

*МИНИСТЕРСТВО ОБЩЕГО И
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ РФ
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ
РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОГО МОДЕЛИРОВАНИЯ*

МЕТОДЫ НЕЙРОИНФОРМАТИКИ

СБОРНИК НАУЧНЫХ ТРУДОВ

Под редакцией доктора
физико-математических наук
А.Н. Горбаня

КРАСНОЯРСК
КГТУ
1998

УДК 519

Методы нейроинформатики / Под. ред. А.Н. Горбаня; отв. за выпуск М.Г. Доррер. КГТУ, Красноярск, 1998. 205 с.

Сборник статей молодых ученых – представителей красноярской школы нейроинформатики. Большинство работ связано с созданием нейросетевых алгоритмов обработки данных и реализацией этих алгоритмов на персональных компьютерах. Представлены новые алгоритмы решения классической проблемы заполнения пробелов в данных, описаны технологии нейросетевого производства явных знаний из данных, продемонстрированы различные приложения нейросетевых методов – от очень популярных сейчас геоинформационных систем (ГИС) до систем психологического тестирования и предсказания отношений в группе.

Другая группа работ посвящена развитию теории. Даны оценки необходимой точности во входных сигналах и элементах сети при заданой требуемой точности на выходе. Без решения такой задачи невозможно создавать надежные нейронные сети “в железе”. Две статьи посвящены проблеме “бесструктурного параллелизма” – устройствам, совершающими вычисления при беспорядочном взаимодействии различных элементов.

Описан также опыт подготовки учебно-исследовательских задач для работы со студентами и школьниками и дан обзор результатов решения этих задач.

Печатается при поддержке
ФЦП "ИНТЕГРАЦИЯ"

Печатается в авторской редакции

ЛР № 020375 от 7.02.1997 г.

© Коллектив авторов, 1998

Редакционно-издательский отдел КГТУ
660074, Красноярск, ул. Киренского, 26

ПРЕДИСЛОВИЕ РЕДАКТОРА

“Что день грядущий мне готовит?” Стремительный прогресс информатики размывает границы между реальностью и фантазией. Системы искусственного интеллекта на основе алгоритмических языков развивались во всемирной программе “Пятое поколение компьютеров”. Сейчас ей на смену пришла другая программа – “Вычисления в реальном мире”. Ее сверхзадача – разработать системы, способные самостоятельно, без помощи переводчика-поводыря человека действовать в “естественном” (а не только в специально приготовленном) внешнем мире. Огромная доля этой программы отводится на создание искусственных нейросетевых систем.

В словах "искусственные нейронные сети" слышатся отзвуки фантазий об андроидах и бунте роботов, о машинах, заменяющих и имитирующих человека. Эти фантазии интенсивно поддерживаются многими разработчиками нейросистем: рисуется не очень отдаленное будущее, в котором роботы осваивают различные виды работ, просто наблюдая за человеком, а в более отдаленной перспективе – человеческое сознание и личность перегружаются в искусственную нейронную сеть – появляются шансы на вечную жизнь.

Если перейти к прозаическому уровню повседневной работы, то искусственные нейронные сети – это всего-навсего сети, состоящие из связанных между собой простых элементов – формальных нейронов. Значительное большинство работ по нейроинформатике посвящено переносу различных алгоритмов решения задач на такие сети.

Ядром используемых представлений является идея о том, что нейроны можно моделировать довольно простыми автоматами, а вся сложность мозга, гибкость его функционирования и другие важнейшие качества определяются связями между нейронами. Каждая связь представляется как совсем простой элемент, служащий для передачи сигнала и его линейного усиления или ослабления. Предельным выражением этой точки зрения может служить лозунг: "структура связей – все, свойства элементов – ничто".

Совокупность идей и научно-техническое направление, определяемое описанным представлением о мозге, называется

коннекционизмом (по-английски connection – связь). Как все это соотносится с реальным мозгом? Так же, как карикатура или шарж со своим прототипом-человеком – весьма условно. Это нормально: важно не буквальное соответствие живому прототипу, а продуктивность технической идеи.

С коннекционизмом тесно связан следующий блок идей:

- 1) однородность системы (элементы одинаковы и чрезвычайно просты, все определяется структурой связей);
- 2) надежные системы из ненадежных элементов и "аналоговый ренессанс" – использование простых аналоговых элементов;
- 3) "голографические" системы – при разрушении случайно выбранной части система сохраняет свои полезные свойства.

Предполагается, что система связей достаточно богата по своим возможностям и достаточно избыточна, чтобы скомпенсировать бедность выбора элементов, их ненадежность, возможные разрушения части связей.

Коннекционизм и связанные с ним идеи однородности, избыточности и голографичности еще ничего не говорят нам о том, как же такую систему научить решать реальные задачи. Хотелось бы, чтобы это обучение обходилось не слишком дорого.

Существует большой класс задач: нейронные системы ассоциативной памяти, статистической обработки, фильтрации и др., для которых связи формируются по явным формулам. Но еще больше (по объему существующих приложений) задач требует неявного процесса. По аналогии с обучением животных или человека этот процесс называют обучением.

Обучение нейронных сетей оказалось продуктивным и при решении задач на обычных персональных компьютерах. Оказалось, что если обучить нейронную сеть решать какую-либо задачу, а затем вырезать из нее все связи, кроме необходимых, без которых эту задачу не решить, то получается очень полезное явное представление знаний о способе решения. Из данных, использованных для обучения, получаем явные знания. Например, обучая сеть предсказывать социально-политическую ситуацию, получаем полезную политологическую теорию, заложим обработку экономических данных – получим экономические знания и т.д.

По-настоящему эффективная работа нейронных сетей требует параллелизма. Можно даже сказать, что искусственные нейронные сети –

одно из мощных средств программирования высокопараллельных компьютеров. Обкатка нейросетевых алгоритмов на персональных компьютерах позволяет, однако, усовершенствовать и сами алгоритмы – дефицит вычислительных ресурсов (отсутствие параллельности) приводит здесь к более жесткому отбору и быстрой эволюции.

В предлагаемом сборнике собраны статьи молодых ученых – представителей красноярской школы нейроинформатики. Большинство статей связано с созданием нейросетевых алгоритмов обработки данных и реализацией этих алгоритмов на персональных компьютерах (А.А. Россиев, В.Г. Царегородцев, Н.А. Погребная, М.Г. Доррер, А.А. Питенко). Здесь есть и новые алгоритмы решения классической проблемы заполнения пробелов в данных, и описание технологии нейросетевого производства явных знаний из данных, а также различные приложения нейросетевых методов: от очень популярных сейчас геоинформационных систем (ГИС) до систем психологического тестирования и предсказания отношений в группе.

Другая группа работ посвящена развитию теории. В статье М.Ю. Сенашовой даны оценки необходимой точности во входных сигналах и элементах сети при заданной требуемой точности на выходе. Без решения такой задачи невозможно создавать надежные нейронные сети “в железе”. Е.О. Горбунова занимается проблемой “бесструктурного параллелизма” – грубо говоря, устройствами, совершающими вычисления при беспорядочном взаимодействии различных элементов.

Особое место занимает статья Л.И. Жукова, который описал свой богатый опыт подготовки учебно-исследовательских задач для работы со студентами и школьниками и дал обзор результатов решения этих задач.

Все работы были выполнены в учебно-научном центре по нейроинформатике, объединяющем академический институт и несколько вузов. Это один из центров, созданных по проекту № 68 Федеральной целевой программы “ИНТЕГРАЦИЯ” (направление 2.1).

Принятые к публикации работы были доложены и прошли конкурсный отбор на семинаре, поддержанном в рамках другого проекта ФЦП “ИНТЕГРАЦИЯ” (№ K0790+K0691, направление 1.6).

Работы получали также поддержку различных фондов и программ, ссылки на которые приведены в конце каждой статьи.

МОДЕЛИРОВАНИЕ ДАННЫХ ПРИ ПОМОЩИ КРИВЫХ ДЛЯ ВОССТАНОВЛЕНИЯ ПРОБЕЛОВ В ТАБЛИЦАХ

А.А.Россиев

Институт вычислительного моделирования СО РАН

660036, Красноярск-36, ИВМ СО РАН,

E-mail: alexross@cc.krascience.rssi.ru

Предложен и реализован метод последовательного моделирования набора данных одномерными многообразиями (кривыми). Метод интерпретируется как построение конвейера нейронов для обработки данных с пробелами. Другая возможная интерпретация – итерационный метод главных компонент и нелинейный факторный анализ для данных с пробелами.

1. Общая схема метода

Для заполнения пробелов в данных в общем случае обычные уравнения регрессии практически неприменимы из-за большого разнообразия возможного расположения пропущенных данных – $2^n - 1$ вариантов для n -мерного вектора данных. Более перспективным представляется моделирование множества данных многообразиями M малой размерности. Вектор данных x с k пробелами представляется как k -мерное линейное многообразие L_x , параллельное k координатным осям, которые соответствуют пропущенным данным. При наличии априорных ограничений на пропущенные значения место L_x занимает прямоугольный параллелепипед $P_x \subset L_x$. Особую роль во всем дальнейшем рассмотрении играют две точки:

$\text{Pr}_M(x)$ – ближайшая к x (соответственно к L_x или P_x) точка M ,

и для некомплектных данных – $\text{Im}_M(x)$ – ближайшая к M точка L_x (или, соответственно, P_x).

С использованием этого многообразия M производятся следующие операции:

заполнение пробелов – замена x на $\text{Im}_M(x)$,

ремонт данных (замещение данных моделью) – замена x на $\text{Pr}_M(x)$.

Регрессионная зависимость одной части данных от другой определяется через заполнение пробелов, соответствующих зависимым переменным, по известным значениям независимых.

Итерационный процесс моделирования данных состоит в том, что для исходных данных строится наилучшая (в определенном точном смысле) модель – многообразие M малой размерности. Далее из данных x (соответственно L_x или P_x) вычитаются проекции $\text{Pr}_M(x)$. Получаем отклонения от первой модели. Для этого множества отклонений снова строится простая модель и т.д., пока все отклонения не станут достаточно близки к нулю. В следующем разделе описаны простейшие линейные модели данных, которые могут интерпретироваться как сингулярные разложения таблиц с пробелами. Далее с использованием линейных моделей строятся простейшие нелинейные.

2. Итерационный метод главных компонент для данных с пропусками

Пусть задана прямоугольная таблица, клетки которой либо заполнены действительными числами или значком @, означающим отсутствие данных. Требуется правдоподобным образом восстановить отсутствующие данные. При более детальном рассмотрении возникают три задачи:

заполнить пропуски в таблице;

отредактировать таблицу – изменить значения известных данных таким образом, чтобы наилучшим образом работали модели, используемые при восстановлении пропущенных данных;

построить по таблице вычислитель, заполняющий пробелы в приходящей для анализа строке данных с пробелами (в предположении, что данные в этой строке связаны теми же соотношениями, что и в строках таблицы).

Для решения этих задач предлагается использовать метод последовательного приближения множества векторов данных (строк таблицы) прямыми.

Основная процедура – поиск наилучшего приближения таблицы с пропусками матрицей вида $x_i y_j + b_j$.

Пусть задана таблица с пропусками $A=(a_{ij})$. Ставится задача поиска наилучшего приближения A матрицей вида $x_i y_j + b_j$ методом наименьших квадратов:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - x_i y_j - b_j)^2 \rightarrow \min. \quad (1)$$

Если фиксированы два из трех векторов x_i , y_j и b_j , то третий легко находится по явным формулам. Задаваясь практически произвольными начальными приближениями для двух из них, ищем значение третьего, далее, объявляем неизвестным другой вектор из трех, находим его значение, наконец, находим третий и т.д. (по кругу) – эти простые итерации, очевидно, сходятся. Более того, по фиксированному x_i , можно сразу по явным формулам посчитать значения y_j и b_j – таким образом расщепление производится не на три, а на две составляющие.

При фиксированных векторах y_j и b_j значения x_i , доставляющие минимум форме (1), определяются из равенств $\partial\Phi/\partial x_i=0$ следующим образом:

$$x_i = \left(\sum_{\substack{j \\ a_{ij} \neq @}} (a_{ij} - b_j) y_j \right) / \left(\sum_{\substack{j \\ a_{ij} \neq @}} (y_j)^2 \right).$$

При фиксированном векторе x_i значения y_j и b_j , доставляющие минимум форме (1), определяются из двух равенств $\partial\Phi/\partial y_j=0$ и $\partial\Phi/\partial b_j=0$ следующим образом:

Для каждого j имеем систему из двух уравнений относительно y_j и b_j :

$$\begin{cases} y_j A_{01}^j + b_j A_{00}^j = B_0^j \\ y_j A_{11}^j + b_j A_{10}^j = B_1^j \end{cases}, \text{ где } A_{kl}^j = \sum_{\substack{i \\ a_{ij} \neq @}} x_i^{k+l}, B_k^j = \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i^k, k=0..1, l=0..1.$$

Выражая из первого уравнения b_j и подставляя полученное значение во второе, получим:

$$y_j = \frac{B_1^j - B_0^j \frac{A_{10}^j}{A_{00}^j}}{A_{11}^j - A_{01}^j \frac{A_{10}^j}{A_{00}^j}}, b_j = \frac{B_0^j - y_j A_{01}^j}{A_{00}^j}.$$

Начальные значения:

y – случайный, нормирован на 1 (т.е. $\sum_j y_j^2 = 1$)

$$b_j = \frac{1}{n_j} \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij}, \text{ где } n_j = \sum_{\substack{i \\ a_{ij} \neq @}} 1 \text{ (число известных данных в } j\text{-ом}$$

столбце), т.е. b_j определяется как среднее значение в столбце.

Критерий остановки – малость относительно улучшения $\Delta\Phi/\Phi$, где $\Delta\Phi$ – полученное за цикл уменьшение значения Φ , а Φ – само текущее значение. Второй критерий – малость самого значения Φ . Окончательно: процедура останавливается, если $\Delta\Phi/\Phi < \varepsilon$ или $\Phi < \delta$ для некоторых $\varepsilon, \delta < 0$.

Последовательное исчерпание матрицы A .

Для данной матрицы A ищем наилучшее приближение матрицей P_1 вида $x_i y_j + b_j$. Далее, для $A - P_1$ ищем наилучшее приближение этого же вида P_2 и т.д. Контроль ведется, например, по остаточной дисперсии столбцов.

Q -факторное заполнение пропусков есть их определение из суммы Q полученных матриц вида $x_i y_j + b_j$,

Q -факторный “ремонт” таблицы – замена ее на сумму Q полученных матриц вида $x_i y_j + b_j$.

Пусть в результате описанного процесса построена последовательность матриц P_q вида $x_i y_j + b_j$ ($P_q = x_i^q y_j^q + b_j^q$),

исчерпывающая исходную матрицу A с заданной точностью. Опишем операцию восстановления данных в поступающей на обработку строке a_j с пробелами (некоторые $a_j=@$). Для каждого q по заданной строке определим число $x^q(a)$ и вектор a_j^q :

$$\begin{aligned}
 a_j^0 &= a_j \quad (a_j \neq @); \\
 x^1(a) &= \left(\sum_{\substack{j \\ a_j \neq @}} (a_j^0 - b_j^1) y_j^1 \right) / \left(\sum_{\substack{j \\ a_j \neq @}} (y_j^1)^2 \right); \\
 a_j^1 &= a_j^0 - b_j^1 - x^1(a) y_j^1 \quad (a_j \neq @); \\
 &\dots\dots\dots \\
 x^q(a) &= \left(\sum_{\substack{j \\ a_j \neq @}} (a_j^{q-1} - b_j^q) y_j^q \right) / \left(\sum_{\substack{j \\ a_j \neq @}} (y_j^q)^2 \right); \\
 a_j^q &= a_j^{q-1} - b_j^q - x^q(a) y_j^q \quad (a_j \neq @); \\
 &\dots\dots\dots
 \end{aligned} \tag{2}$$

Здесь многообразие M – прямая, координаты точек на M задаются параметрическим уравнением $z_j = t y_j + b_j$, а проекция $\text{Pr}_M(a)$ определяется согласно (2):

$$\begin{aligned}
 \text{Pr}(a) &= t(a) y_j + b_j; \\
 t(a) &= \left(\sum_{\substack{j \\ a_j \neq @}} (a_j - b_j) y_j \right) / \left(\sum_{\substack{j \\ a_j \neq @}} (y_j)^2 \right).
 \end{aligned} \tag{3}$$

Для Q -факторного восстановления данных полагаем:

$$\bar{a}_j = \sum_{q=1}^Q x^q(a) y_j^q + b_j^q, \quad (a_j \neq @). \tag{4}$$

Если пробелы отсутствуют, то описанный метод приводит к обычным главным компонентам – сингулярному разложению исходной

таблицы данных. В этом случае, начиная с $q=2$, $P_q = x_i^q y_j^q$ ($b=0$). В общем случае это не так и центрирование к данным с пробелами неприменимо.

Также следует учесть, что при отсутствии пробелов, полученные прямые будут ортогональны, то есть получим ортогональную систему факторов (прямых). Исходя из этого, при неполных данных возможен процесс ортогонализации полученной системы факторов, который заключается в том, что исходная таблица восстанавливается при помощи полученной системы, после чего эта система пересчитывается заново, но уже на полных данных.

3. Квазилинейные факторы и формулы Карлемана

Использование нелинейных одномерных моделей данных (кривых) принципиально не отличается от использования линейных моделей (прямых). Предлагается использовать квазилинейные модели, допускающие простые явные формулы обработки данных и опирающиеся на описанный алгоритм построения линейных моделей.

Пусть, как и в случае линейных моделей, задана таблица с пропусками A . Построение квазилинейных моделей, наилучшим образом приближающих данные, предлагается проводить в несколько этапов.

Построение линейной модели – решение задачи (1). Для определенности полагаем, что $(y,b)=0$, $(y,y)=1$ – этого всегда можно добиться.

Интерполяция (сглаживание)

Строится вектор-функция $f(t)$, минимизирующая функционал:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt, \text{ где } \alpha > 0 - \text{параметр}$$

сглаживания.

При этом решение задачи сглаживания дается либо одним полиномом, либо кубическими сплайнами.

Экстраполяция

Вектор-функция $f(t)$ экстраполируется при помощи касательных.

Сглаженная вектор-функция $f(t)$ экстраполируется с некоторого конечного множества $\{t_k\}$ (которое не обязательно связано с проекциями на прямую $z_j=ty_j+b_j$ исходных строк данных) на всю вещественную прямую с использованием формул Карлемана [1] (с помощью формул Карлемана экстраполируется отклонение кривой $f(t)$ от прямой $ty+b$).

$$f(t) \approx ty + b + \sum_{k=1}^m (f(t_k) - t_k y - b) \frac{2(e^{\lambda t} - e^{\lambda t_k})}{\lambda(e^{\lambda t} + e^{\lambda t_k})(t - t_k)} \prod_{\substack{j=1 \\ j \neq k}}^m \frac{(e^{\lambda t_k} + e^{\lambda t_j})(e^{\lambda t} - e^{\lambda t_j})}{(e^{\lambda t_k} - e^{\lambda t_j})(e^{\lambda t} + e^{\lambda t_j})}, \quad (5)$$

где λ – параметр метода, характеризующий, насколько широка полоса на плоскости комплексных чисел, в которой гарантированно голоморфна экстраполируемая функция (эта ширина равна π/λ).

Интерполяция со сглаживанием

При решении задачи интерполяции со сглаживанием необходимо нормировать значения x_i и a_{ij} таким образом, чтобы они лежали в отрезке $[-1,1]$. В последствии, при вычислении значения функции, полученные значения денормируются.

Интерполяция полиномом степени n

В случае аппроксимация полиномом степени n ставится задача наилучшего приближения матрицы A полиномами вида $f_j(x) = f_n^j x^n + f_{n-1}^j x^{n-1} + \dots + f_1^j x + f_0^j$, т.е.:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq 0}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt, \text{ где } \alpha > 0 - \text{параметр}$$

сглаживания.

Так как значения $x_i=(a_{i,y})$, где a_i – i -ая строка матрицы A , фиксированы (вычислены на предыдущем этапе), то значения коэффициентов полинома f_k^j ($k=0..n$), доставляющие минимум функционалу Φ , определяются из системы равенств $\partial\Phi/\partial f_k^j = 0$ ($k=0..n$) следующим образом:

Вычислим значение интеграла при параметре сглаживания:

$$f_j'(t) = \sum_{k=1}^n k f_k^j x^{k-1}, \quad f_j''(t) = \sum_{k=2}^n k(k-1) f_k^j x^{k-2}, \quad \text{тогда:}$$

$$\left(f_j''(t)\right)^2 = \sum_{k,l=2}^n k(k-1)l(l-1) f_k^j f_l^j x^{k+l-4}. \quad \text{Следовательно, интеграл}$$

будет равен (учитывая, что искомая функция определена на отрезке $[-1, 1]$):

$$\begin{aligned} \int_{-1}^{+1} \left(f_j''(t)\right)^2 dt &= \sum_{k,l=2}^n f_k^j f_l^j k(k-1)l(l-1) \frac{x^{k+l-3}}{k+l-3} \Big|_{-1}^{+1} = \\ &= \sum_{\substack{k,l=2 \\ k+l - \text{четное}}}^n f_k^j f_l^j k(k-1)l(l-1) \frac{2}{k+l-3}. \end{aligned}$$

Тогда:

$$\frac{d}{df_k^j} \int_{-1}^{+1} \left(f_j''(t)\right)^2 dt = \sum_{\substack{l=2 \\ l \neq k \\ k+l - \text{четное}}}^n f_l^j k(k-1)l(l-1) \frac{2}{k+l-3} + 2 f_k^j k^2 (k-1)^2 \frac{2}{2k-3}.$$

Окончательно, для $k=0..n$, имеем:

$$\begin{aligned} \frac{\partial \Phi}{\partial f_k^j} &= -2 \sum_{\substack{i \\ a_{ij} \neq @}} (a_{ij} - \sum_{l=0}^n f_l^j x_i^l) x_i^k + \\ &+ \sum_{\substack{l=2 \\ l \neq k \\ k+l - \text{четное}}}^n f_l^j k(k-1)l(l-1) \frac{2}{k+l-3} + 2 f_k^j k^2 (k-1)^2 \frac{2}{2k-3} = 0. \end{aligned}$$

Группируя коэффициенты при f_k^j ($k=0..n$), получим:

$$\sum_{l=0}^n A_{kl}^j(0) f_l^j = B_k^j(0), \quad \text{для } k=0..n, \quad \text{где:}$$

$$A_{kl}^j(0) = \sum_{\substack{i \\ a_{ij} \neq @}} x_i^{k+l} + \begin{cases} 0 & , \text{если } k < 2 \\ & \text{или } l < 2 \text{ или } k+l - \text{нечетное} \\ 2k^2(k-1)^2 \frac{2}{2k-3} & , \text{если } k = l \\ k(k-1)l(l-1) \frac{2}{k+l-3} & , \text{если } k \neq l \end{cases},$$

$$B_k^j(0) = \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i^k.$$

Возьмем 1-ое уравнение, выразим f_0^j через оставшиеся и подставим полученное выражение во все остальные уравнения системы:

$$\sum_{l=1}^n A_{0l}^j(0) f_l^j + A_{00}^j(0) f_0^j = B_0^j(0) \Rightarrow f_0^j = \frac{B_0^j(0) - \sum_{l=1}^n A_{0l}^j(0) f_l^j}{A_{00}^j(0)}.$$

Отсюда получаем:

$$\begin{aligned} \sum_{l=1}^n A_{kl}^j(0) f_l^j + A_{k0}^j(0) \frac{B_0^j(0) - \sum_{l=1}^n A_{0l}^j(0) f_l^j}{A_{00}^j(0)} &= B_k^j(0) \text{ для } k=1..n, \Rightarrow \\ \sum_{l=1}^n \left(A_{kl}^j(0) - A_{0l}^j(0) \frac{A_{k0}^j(0)}{A_{00}^j(0)} \right) f_l^j &= B_k^j(0) - B_0^j(0) \frac{A_{k0}^j(0)}{A_{00}^j(0)}, \text{ для } k=1..n. \end{aligned}$$

Эту систему можно записать в следующем виде:

$$\sum_{l=1}^n A_{kl}^j(1) f_l^j = B_k^j(1), \text{ для } k=1..n, \text{ где } A_{kl}^j(1) = A_{kl}^j(0) - A_{0l}^j(0) \frac{A_{k0}^j(0)}{A_{00}^j(0)},$$

$$B_k^j(1) = B_k^j(0) - B_0^j(0) \frac{A_{k0}^j(0)}{A_{00}^j(0)}.$$

Теперь аналогичным образом исключаем f_1^j и т.д.

Пусть m таких исключений уже сделано, тогда имеем систему:

$$\sum_{l=m}^n A_{kl}^j(m) f_l^j = B_k^j(m), \text{ для } k=m..n. \text{ Аналогично описанному выше из}$$

первого уравнения системы ($k=m$) выразим f_m^j и подставим в оставшиеся уравнения. В итоге получим:

$$f_m^j = \frac{B_m^j(m) - \sum_{l=m+1}^n A_{ml}^j(m) f_l^j}{A_{mm}^j(m)},$$

$$\sum_{l=m+1}^n \left(A_{kl}^j(m) - A_{ml}^j(m) \frac{A_{km}^j(m)}{A_{mm}^j(m)} \right) f_l^j = B_k^j(m) - B_m^j \frac{A_{km}^j(m)}{A_{mm}^j(m)}, \quad \text{для}$$

$k=m+1..n$. Следовательно:

$$\sum_{l=m+1}^n A_{kl}^j(m+1) f_l^j = B_k^j(m+1), \quad \text{для } k=m+1..n, \text{ где}$$

$$A_{kl}^j(m+1) = A_{kl}^j(m) - A_{ml}^j(m) \frac{A_{km}^j(m)}{A_{mm}^j(m)}, \quad B_k^j(m+1) = B_k^j(m) - B_m^j(m) \frac{A_{km}^j(m)}{A_{mm}^j(m)}.$$

Таким образом, для нахождения f_k^j ($k=0..n$) получили следующие рекуррентные соотношения:

$$A_{kl}^j(0) = \sum_{\substack{i \\ a_{ij} \neq @}} x_i^{k+l}, \quad B_k^j(0) = \sum_{\substack{i \\ a_{ij} \neq @}} a_{ij} x_i^k,$$

$$A_{kl}^j(m+1) = A_{kl}^j(m) - A_{ml}^j(m) \frac{A_{km}^j(m)}{A_{mm}^j(m)}, \quad B_k^j(m+1) = B_k^j(m) - B_m^j(m) \frac{A_{km}^j(m)}{A_{mm}^j(m)},$$

$$f_k^j = \frac{B_k^j(k) - \sum_{l=k+1}^n A_{kl}^j(k) f_l^j}{A_{kk}^j(k)}, \quad \text{для } k=n..0.$$

Интерполяция кубическими сплайнами

Требуется решить с использованием кубических сплайнов следующую задачу сглаживания:

$$\Phi = \sum_{\substack{i,j \\ a_{ij} \neq @}} (a_{ij} - f_j(\sum_k a_{ik} y_k))^2 + \alpha \int_{-\infty}^{+\infty} (f''(t))^2 dt, \quad \text{где } \alpha > 0 - \text{параметр}$$

сглаживания.

При решении этой задачи имеем отрезок $[-1,1]$. Разобьем его на n частей точками \bar{x}_s , $s=0..n$, где: $-1 = \bar{x}_0 < \bar{x}_1 < \dots < \bar{x}_{n-1} < \bar{x}_n = 1$. При этом: $h_s = \bar{x}_s - \bar{x}_{s-1}$.

Пусть $\bar{x} = x - \bar{x}_{s-1}$, где $\bar{x}_{s-1} < x < \bar{x}_s$, $s=1..n$, тогда имеем:

$$f(x) = f_{s3} \bar{x}^3 + f_{s2} \bar{x}^2 + f_{s1} \bar{x} + f_{s0} = \sum_{l=0}^3 f_{sl} \bar{x}^l.$$

Посчитаем производные:

$$f'(x) = \sum_{l=1}^3 l f_{sl} \bar{x}^l, \quad f''(x) = \sum_{l=2}^3 l(l-1) f_{sl} \bar{x}^{l-1}.$$

Значение интеграла при коэффициенте гладкости:

$$\begin{aligned} I &= \int_{-1}^1 (f''(t))^2 dt = \sum_{s=1}^{n-1} \int_{\bar{x}_{s-1}}^{\bar{x}_s} \left(\sum_{l=2}^3 l(l-1) f_{sl} t \right)^2 dt = \\ &= \sum_{s=1}^{n-1} \left(\sum_{k,l=2}^3 f_{sk} f_{sl} k(k-1)l(l-1) \frac{t^{k+l-3}}{k+l-3} \Big|_{\bar{x}_{s-1}}^{\bar{x}_s} \right) = \\ &= \sum_{s=1}^{n-1} \left(\sum_{k,l=2}^3 f_{sk} f_{sl} k(k-1)l(l-1) \frac{h_s^{k+l-3}}{k+l-3} \right) \end{aligned}$$

Таким образом, исходная задача сглаживания запишется в следующем виде:

$\Phi = H + \alpha I \rightarrow \min$, где

$$H = \sum_{i \neq @} \left(a_i - \sum_{l=0}^3 f_{sl} \bar{x}_i^l \right), \quad I = \sum_{s=1}^{n-1} \left(\sum_{k,l=2}^3 f_{sk} f_{sl} k(k-1)l(l-1) \frac{h_s^{k+l-3}}{k+l-3} \right).$$

Полученная сплайн-функция должна быть непрерывная в узлах сетки вместе со своей первой и второй производной.

Поэтому из условий непрерывности получим следующую систему:

$$\begin{cases} f(\bar{x}_s - 0) = f(\bar{x}_s + 0) \\ f'(\bar{x}_s - 0) = f'(\bar{x}_s + 0), \text{ где } s=1..n-1. \\ f''(\bar{x}_s - 0) = f''(\bar{x}_s + 0) \end{cases}$$

Подставив соответствующие значения, получим условия на коэффициенты сплайнов:

$$\begin{cases} f_{s+1,0} = f_{s3} h_s^3 + f_{s2} h_s^2 + f_{s1} h_s + f_{s0} \\ f_{s+1,1} = 3f_{s3} h_s^2 + 2f_{s2} h_s + f_{s1} \\ f_{s+1,2} = 3f_{s3} h_s + f_{s2} \end{cases}, \text{ где } s=1..n-1.$$

Исходная задача сглаживания решается методом множителей Лагранжа:

$$L = \Phi + \sum_{s=1}^{n-1} \lambda_{s0} \varphi_{s0} + \sum_{s=1}^{n-1} \lambda_{s1} \varphi_{s1} + \sum_{s=1}^{n-1} \lambda_{s2} \varphi_{s2}, \text{ где:}$$

$$\begin{cases} \varphi_{s0} = f_{s+1,0} - f_{s3} h_s^3 - f_{s2} h_s^2 - f_{s1} h_s - f_{s0} \\ \varphi_{s1} = f_{s+1,1} - 3f_{s3} h_s^2 - 2f_{s2} h_s - f_{s1} \\ \varphi_{s2} = f_{s+1,2} - 3f_{s3} h_s - f_{s2} \end{cases}, \text{ при } s=1..n, \text{ — линейные}$$

ограничения, полученные из условий непрерывности.

Нужно решить следующую систему:

$$\begin{cases} \frac{\partial L}{\partial f_{sl}} = 0, \text{ для } l = 0..3, s = 1..n \\ \frac{\partial L}{\partial \lambda_{sl}} = 0, \text{ для } l = 0..2, s = 1..n-1 \end{cases}, \text{ т.е. систему из } (7n-3) \text{ уравнений и}$$

с $(7n-3)$ неизвестными.

Первые строчки этой системы можно расписать следующим образом:

$$\frac{\partial L}{\partial f_{sl}} = \frac{\partial H}{\partial f_{sl}} + \alpha \frac{\partial I}{\partial f_{sl}} + L_{sl}, \text{ где:}$$

$$\frac{\partial H}{\partial f_{sl}} = -2 \sum_{\substack{i \\ a_i \neq @}} \left(a_i - \sum_{p=0}^3 f_{sp} \bar{x}_i^p \right) \bar{x}_i^l,$$

$$\frac{\partial I}{\partial f_{sl}} = \sum_{\substack{k=2 \\ k \neq l}}^3 f_{sk} k(k-1)l(l-1) \frac{2}{k+l-3} + 2f_{sl} l^2 (l-1)^2 \frac{2}{2l-3},$$

$$\begin{cases} L_{s0} = \lambda_{s-1,0} - \lambda_{s0} \\ L_{s1} = \lambda_{s-1,1} - \lambda_{s0} h_s - \lambda_{s1} \\ L_{s2} = \lambda_{s-1,2} - \lambda_{s0} h_s^2 - \lambda_{s1} 2h_s - \lambda_{s2} \\ L_{s3} = -\lambda_{s0} h_s^3 - \lambda_{s1} 3h_s^2 - \lambda_{s2} 3h_s \end{cases}, \text{ при этом коэффициенты с}$$

индексом $s-1$ берутся для $s=2..n$, коэффициенты с индексом s берутся для $s=1..n-1$.

Подставив все это в исходную систему, можно заметить, что она имеет практически блочно-диагональный вид, где размер отдельных

блоков достаточно мал, поэтому, несмотря на большой размер $((7n-3) \times (7n-3))$, она с хорошей точностью решается методом Гаусса.

Экстраполяция

Необходимость в экстраполяции вызвана двумя обстоятельствами: во-первых, построенная на втором этапе сглаженная зависимость в принципе интерполяционная и не может быть экстраполирована, во-вторых, в ней фактически содержится в явном виде информация о каждой строке матрицы данных. Сглаживание просто многочленом небольшой степени по методу наименьших квадратов свободно от второго недостатка (информация "сворачивается" в несколько коэффициентов), но также не дает хорошей экстраполяции.

Экстраполяция прямыми

При экстраполяции прямыми полученная функция экстраполируется с отрезка $[-1,1]$ на всю вещественную ось за счет добавления касательных к функции в концах отрезка.

Экстраполяция при помощи формул Карлемана

С помощью формул Карлемана экстраполируется отклонение кривой $f(t)$ от прямой $ty+b$. Формулы Карлемана обеспечивают хорошую экстраполяцию аналитических функций на всю прямую (конечно, строго говоря, в каждом конкретном случае нет гарантий, что именно по формуле (5) будет получена наилучшая экстраполяция, однако существует ряд теорем о том, что формула (5) и родственные ей дают наилучшее приближение в различных классах аналитических функций [1]).

Последовательность квазилинейных факторов

Точка на построенной кривой $f(t)$, соответствующая полному ("комплектному") вектору данных a строится как $f((a,y))$, в этом и заключается квазилинейность метода: сначала ищется проекция вектора данных на прямую $Pr(a)=ty+b$, $t=(a,y)$, затем строится точка на кривой $f(t)$. Также и для неполных векторов данных – сначала ищется точка на прямой по формуле (3), а затем – соответствующая точка на кривой $f(t)$ при $t=t(a)$.

После построения кривой $f(t)$ данные заменяются на уклонения от модели. Далее снова ищется наилучшее приближение вида $x_i y_j + b_j$ для матрицы уклонений, вновь строится сглаживание, потом – экстраполяция по Карлеману и т.д., пока уклонения не приблизятся в достаточной степени к нулю.

Действуя по этой схеме, для таблиц данных с пропусками $A=(a_{ij})$ получаем последовательность уклонений a_{ij}^q , таблиц $P_q = x_i^q y_j^q + b_j^q$ (для определенности полагаем $(y^q, b^q)=0$, $(y^q, y^q)=1$ – этого всегда можно добиться) и кривых $f^q(t)$.

Это последовательности связаны следующим образом:

$a_{ij}^0 = a_{ij}$, $P_q = x_i^q y_j^q + b_j^q$ – наилучшее приближение такого вида (решение задачи (1)) для таблицы a_{ij}^{q-1} ,

$$t_i^q = \left(\sum_{\substack{j \\ a_{ij} \neq @}} (a_{ij}^{q-1} - b_j^q) y_j^q \right) / \left(\sum_{\substack{j \\ a_{ij} \neq @}} (y_j^q)^2 \right), \quad (6)$$

кривая $f^q(t)$ получается в два этапа: сглаживается зависимость $\varphi_j^q(t)$, заданная на конечном множестве $\{t_j^q\}$ следующим образом: $\varphi_j^q(t_i^q) = a_{ij}^{q-1}$, если $a_{ij}^{q-1} \neq @$ и $\varphi_j^q(t_i^q) = x_i^q y_j^q + b_j^q$, если $a_{ij}^{q-1} = @$, – после чего сглаженная зависимость экстраполируется с помощью формул (5). Последовательность уклонений определяется с помощью кривой $f^q(t)$ и значений t_j^q : $a_{ij}^q = a_{ij}^{q-1} - f_j^q(t_i^q)$.

В результате исходная таблица предстает в виде $a_{ij} \cong \sum_q f_j^q(t_i^q)$.

Если $a_{ij} \neq @$, то эта формула аппроксимирует исходные данные, если же $a_{ij} = @$, то она дает способ восстановления данных. Большой интерес вызывает вопрос: сколько слагаемых следует брать для обработки данных? Существует несколько вариантов ответов, но большинство из них

подчиняется эвристической формуле: *число слагаемых должно быть минимальным среди тех, что обеспечивают удовлетворительное (терпимое) тестирование метода на известных данных.*

Алгоритм применим в том случае, когда матрица данных не может быть приведена перестановкой строк и столбцов к следующему блочно-диагональному виду:

$$A = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & A_n \end{bmatrix}, \quad (7)$$

где 0 – прямоугольные матрицы с нулевыми элементами.

Приведенный алгоритм заполнения пробелов не требует их предварительного априорного заполнения – в отличие от многих других алгоритмов, предназначенных для той же цели. Однако же он требует предварительной нормировки ("обезразмеривания") данных – перехода в каждом столбце таблицы к "естественной" единице измерения. Чаще всего производится нормировка на единичное среднеквадратичное отклонение в столбцах или на единичный разброс данных в каждом столбце (если нет каких-либо специфических ограничений, связанных со смыслом задачи).

4. Нейронный конвейер

Построенный алгоритм допускает нейросетевую интерпретацию. С каждой кривой $f_q(t)$ связан один сумматор (его веса – координаты вектора y^q), набор из n свободных слагаемых ("порогов") – координат вектора b^q и n нелинейных преобразователей, каждый из которых вычисляет одну координату точки на кривой по формуле (5). Действует такой "нейрон" на вектор a входных сигналов (содержащий пробелы) так: по формуле (3) вычисляется $t(a)$ (работает сумматор), далее нелинейные элементы вычисляют $f_j^q(t(a))$, затем разность $f_j^q(t(a))$ ($a_j \neq @$) передается следующему нейрону. При прохождении a по этому конвейеру одновременно накапливается сумма величин $f_j^q(t(a))$ ($a_j = @$). Они и образуют вектор выходных сигналов – предлагаемые значения

пропущенных данных. При необходимости провести ремонт данных накапливается сумма величин $f_j^q(t(a))$ для каждой координаты j .

Структура нейрона нестандартна (рис.1) (можно сравнить с [2]) – он имеет один входной сумматор и n нелинейных преобразователей (по размерности вектора данных).

Также не вполне обычен способ работы сумматора (3) – для некомплектных векторов данных вычисляется скалярное произведение с имеющимися данными и производится дополнительная нормировка (с учетом только тех координат, значения которых для входного вектора известны).

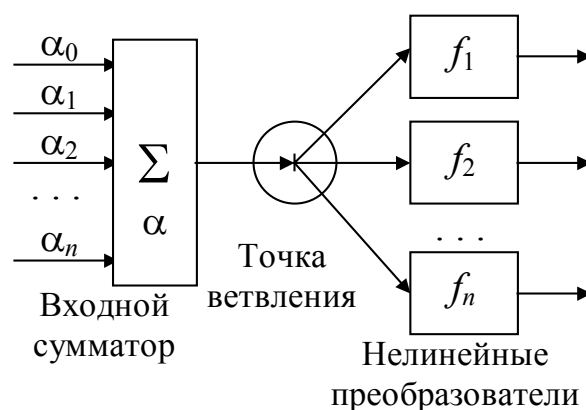


Рис. 1. Нейрон конвейера

Примечателен описанный способ построения этих нейронов. Их характеристики вычисляются по очереди, причем сначала строится сумматор (с помощью решения задачи (1)), затем – нелинейные преобразователи (по формулам Карлемана), далее сумматор следующего нейрона и т.д.

Все построенные нейроны работают поочередно (в обычном смысле здесь столько же слоев, сколько нейронов), однако они образуют конвейер и освободившиеся нейроны могут переходить к новому вектору данных, поэтому при последовательном поступлении данных время обработки пропорционально числу нейронов, но производительность (количество обработанных векторов данных в единицу времени) определяется временем срабатывания одного нейрона и не зависит от их числа.

Автор признателен А.Н. Горбаню и С.В. Макарову за руководство работой и полезные идеи.

Работа выполнена при поддержке Министерства науки и технологий РФ (подпрограмма "Перспективные информационные технологии", проект № 05.04.1291) и ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.).

ЛИТЕРАТУРА

1. Айзенберг Л.А. Формулы Карлемана в комплексном анализе. Первые приложения. Новосибирск: Наука, 1990. 248 с.
2. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 256 с.
3. Горбань А.Н., Макаров С.В., Россиев А.А. Итерационный метод главных компонент для таблиц с пробелами // Третий сибирский Конгресс по прикладной и индустриальной математике (ИНПРИМ-98) (памяти С.Л.Соболева). Новосибирск, июнь 1998. Тезисы докладов. Ч. V. Новосибирск: Издательство Института математики, 1998. С. 74.
4. Горбань А.Н., Макаров С.В., Россиев А.А. Нейронный конвейер для восстановления пробелов в таблицах и построения регрессии по малым выборкам с неполными данными // V Международная конференция "Математика, компьютер, образование", Дубна, январь 1998. Тезисы. С. 53.

ФИНИТНОСТЬ И ДЕТЕРМИНИРОВАННОСТЬ ПРОСТЫХ ПРОГРАММ ДЛЯ КИНЕТИЧЕСКОЙ МАШИНЫ КИРДИНА

Е. О. Горбунова

*Институт вычислительного моделирования СО РАН,
Красноярский государственный технический университет*

660036, Красноярск-36, ИВМ СО РАН,

E-mail: gkat@cc.krascience.rssi.ru

Предлагается новая абстрактная модель параллельных вычислений – кинетическая машина Кирдина. Ожидается, что эта модель сыграет ту же роль для параллельных вычислений, что и нормальные алгоритмы Маркова, машины Колмогорова и Тьюринга или схемы Поста для последовательных вычислений. Неформально кинетическую машину Кирдина можно описать следующим образом. Есть банка, в которой плавают слова. В нее добавляем правила-катализаторы; одни из них, сталкиваясь со словами, способствуют их распаду, другие, встречая пару подходящих слов, способствуют их синтезу, а третьи заменяют в словах некоторые подцепочки. В работе описаны основные способы реализации вычислений и исследованы свойства простейших программ для кинетической машины Кирдина.

1. Введение

В настоящее время параллельные вычисления сводятся к «распараллеливанию» последовательных алгоритмов при наличии однотипных операций над однотипными данными. Но когда число параллельно работающих «процессорных элементов» имеет высокий порядок, то задача программирования приобретает качественно иной характер. Некоторую надежду на получение способа программирования таких систем для более общих задач дают нейронные сети: программу имитации можно создать в идеологии «однотипные задания для разных элементов». Несмотря на то, что большие усилия прилагаются для решения задач параллельной обработки данных, проблема эффективного программирования для вычислительных систем с мелкозернистым параллелизмом до сих пор остается открытой.

Предлагается новая абстрактная модель параллельных вычислений – кинетическая машина Кирдина, предложенная А.Н.Кирдиным в октябре 1997 года на конференции «Нейроинформатика и ее приложения» [1]. Ожидается, что эта модель сыграет ту же роль для параллельных вычислений, что и нормальные алгоритмы Маркова, машины Колмогорова и Тьюринга или схемы Поста для последовательных вычислений.

В работе описаны основные способы реализации вычислений и исследованы свойства простейших программ для кинетической машины Кирдина. Часть результатов была ранее анонсирована на Международном конгрессе по индустриальной и прикладной математике ИНПРИМ-98 [2].

2. Понятие кинетической машины Кирдина

Пусть L – алфавит символов. L^* – множество всех конечных слов или цепочек в алфавите L . Обработываемой единицей является ансамбль слов M из алфавита L , который отождествляется с функцией F_M с конечным носителем на L^* , принимающей неотрицательные целые значения – $F_M: L^* \rightarrow N \cup \{0\}$. Значение $F_M(s)$ интерпретируется как число экземпляров слова s в ансамбле M .

Обработка состоит в совокупности элементарных событий, которые происходят недетерминированно и параллельно. Элементарное событие $S: M \mapsto M'$ состоит в том, что из ансамбля M изымается ансамбль K^- (это возможно, если для всех s $F_{K^-}(s) \leq F_M(s)$) и добавляется ансамбль K^+ , т.е. $F_{M'} = F_M - F_{K^-} + F_{K^+}$. Ансамбли K^- и K^+ однозначно задаются правилами или командами, которые объединяются в программу. Команды могут быть только трех видов:

Пусть u, w, v, f, g, k, q, s – терминальные символы, обозначающие некоторые цепочки символов из L , являющиеся подцепочками слов из L^* .

1. *Распад*. $uvw \rightarrow uf + gw$

где u, w – произвольные, v, f, g – фиксированы. Распад однозначно задается тройкой цепочек (v, f, g) . Обозначим P_1 множество таких троек.

При применении этой команды функция F_M изменит свои значения следующим образом:

$$F_{M'}(uvw) = F_M(uvw) - 1 ,$$

$$F_{M'}(uf) = F_M(uf) + 1 ,$$

$$F_{M'}(gw) = F_M(gw) + 1 .$$

Команда распада применима, если $F_M(uvw) > 0$.

2. *Синтез*. $uk + qw \rightarrow usw$

u, w – произвольные, k, q, s – фиксированы. Синтез однозначно задается тройкой цепочек (k, q, s) . Обозначим P_2 множество таких троек.

При применении этой команды функция F_M изменит свои значения следующим образом:

$$F_{M'}(uk) = F_M(uk) - 1 ,$$

$$F_{M'}(qw) = F_M(qw) - 1 ,$$

$$F_{M'}(usw) = F_M(usw) + 1 .$$

Команда синтеза применима, если $F_M(uk) > 0, F_M(qw) > 0$.

3. *Прямая замена*. $uvw \rightarrow usw$

u, w – произвольные, v, s – фиксированы. Прямая замена однозначно задается парой цепочек (v, s) . Обозначим P_3 множество таких пар.

При применении этой команды функция F_M изменит свои значения следующим образом:

$$F_{M'}(uvw) = F_M(uvw) - 1 ,$$

$$F_{M'}(usw) = F_M(usw) + 1 .$$

Команда прямой замены применима, если $F_M(uvw) > 0$.

Программа ***P*** применима к ансамблю M , если хотя бы одна его команда применима к M . Программа ***P*** однозначно определяется множествами P_1, P_2 и P_3 .

Таким образом, элементарное событие S однозначно определяется правилом p , содержащимся в списке команд программы P , и ансамблем K^- , определяемым этим правилом, таким, что $F_{K^-}(s) \leq F_M(s)$ для любого s . Допустимое элементарное событие S для ансамбля M и программы P – это такое, для которого существует правило p , содержащееся в списке команд программы P , и значения функции F_M для слов, стоящих в левой части этого правила, положительны.

Скажем, что N допустимых событий совместны, если $F_M - \sum_{i=1}^n F_i^- \geq 0$, где F_i^- – изымаемый ансамбль для i -го события.

Ансамбль M является *финальным* для данной программы P , если никакая команда программы к нему не применима.

Программа P называется *финитной* для данного ансамбля M , если, применяя к нему команды программы в любом порядке, пока это возможно, мы обязательно получим некоторый финальный ансамбль.

Программа P является *детерминированной* для ансамбля M , если все финальные ансамбли совпадают.

Ансамбль M будем называть «*райским садом*» для программы P , если он не может быть получен ни из одного ансамбля применением программы P .

Неформально кинетическую машину Кирдина можно описать следующим образом. У нас есть банка, в которой плавают слова. Мы добавляем в эту банку правила-катализаторы, одни из них, сталкиваясь со словами, способствуют их распаду, другие, встречая пару подходящих слов, способствуют их синтезу, а третьи заменяют в словах некоторые подцепочки (рис.1).

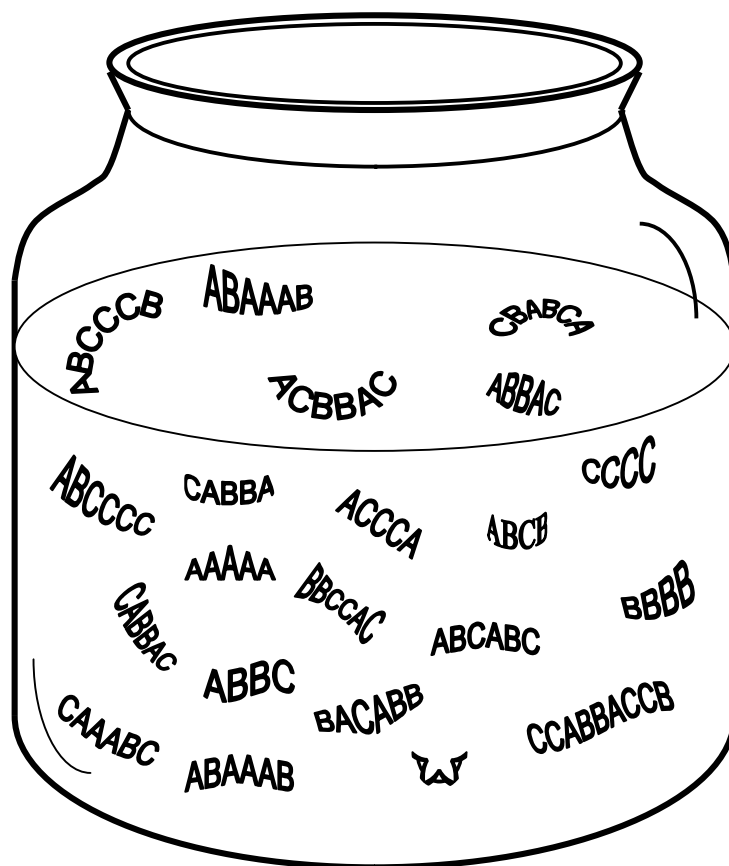


Рис. 1. Банка со словами

3. Модели выполнения программы

Элементарные события в кинетической машине Кирдина происходят недетерминированно и параллельно. Предлагаются некоторые простые варианты модели выполнения программы P .

3.1. Последовательная модель

Пусть в один дискретный момент времени происходит только одно элементарное событие из допустимых $S_1 : M_0 \rightarrow M_1$, в следующий момент происходит одно из допустимых событий для ансамбля M_1 . Выбор этого события недетерминирован. Получаем ряд элементарных событий:

$$M_0 \xrightarrow{S_1} M_1 \xrightarrow{S_2} M_2 \dots \xrightarrow{S_n} M_n$$

Если элементарные события S_1^1 и S_1^2 одновременно допустимы для ансамбля M_0 , то оба они могут служить продолжением выполнения программы:

$M_0 \xrightarrow{S_1^1} M_1^1$ и $M_0 \xrightarrow{S_1^2} M_1^2$, где ансамбли M_1^1 и M_1^2 , вообще говоря, различны. Таким образом, для одной программы возможны различные временные ряды ансамблей. Если программа финитна, то все эти ряды конечны. В противном случае среди всех рядов выполнения программы существует бесконечный ряд. Для детерминированной программы все финальные ансамбли, порожденные различными рядами элементарных событий одинаковы.

3.2. Параллельно-последовательная модель

Обозначим S_i^* некоторое множество совместных событий для данного ансамбля M и программы P . Все события множества S_i^* происходят параллельно и приводят к ансамблю M_i , далее выполняется некоторый набор совместных событий для ансамбля M_i .

$$M_0 \xrightarrow{S_1^*} M_1 \xrightarrow{S_2^*} M_2 \dots \xrightarrow{S_n^*} M_n$$

Разбиение на множества совместных событий S^* для данного ансамбля M неоднозначно. Различные S_i^* для данного ансамбля M приводят к различным ансамблям M_i . Следовательно, такой временной ряд совместных множеств элементарных событий также неоднозначен.

Пример. Пусть программа P для кинетической машины Кирдина состоит из одной команды и имеет следующий вид: $uK + Qw \rightarrow uSw$.

Ансамбль M отождествляем с функцией F_M :

$$F_M(AK)=1, F_M(BK)=1, F_M(QA)=1, F_M(QB)=1.$$

Для таких программ и ансамбля возможны четыре допустимых события, которые определяются изымаемыми ансамблями над следующими словами:

1. Слова AK , QA определяют элементарное событие S_1 ;

2. Слова AK, QB определяют элементарное событие S_2 ;
3. Слова BK, QA определяют элементарное событие S_3 ;
4. Слова BK, QB определяют элементарное событие S_4 .

Нетрудно заметить, что не все из этих допустимых событий совместны. Выпишем полный список множеств совместных элементарных событий:

$$S_1^* = \{S_1\}, S_2^* = \{S_2\}, S_3^* = \{S_3\}, S_4^* = \{S_4\}, S_5^* = \{S_1, S_4\}, S_6^* = \{S_2, S_3\}.$$

Отсюда видно, что в рамках параллельно-последовательной модели для нашего примера существует шесть различных способов перехода к следующему ансамблю.

3.3. Максимальная параллельно-последовательная модель

Множество элементарных событий, к которому нельзя добавить ни одного допустимого события, не теряя совместности, назовем максимальным множеством совместных событий S_{\max}^* . В предыдущем примере множества S_5^* и S_6^* являются максимальными множествами совместных событий. То есть к исходному ансамблю M может быть применено два разных максимальных множества совместных событий, которые порождают различные ансамбли. Мы видим, что максимальная параллельно-последовательная модель также не однозначна. Переход от ансамбля M к некоторому ансамблю M' однозначно определяется только в следующих случаях:

- когда ансамблю M и программе P соответствует ровно одно допустимое событие;
- (для максимальной параллельно-последовательной модели) когда весь набор допустимых событий для ансамбля M и программы P является одновременно совместным.

4. Программы, состоящие из одной команды

4.1. Распад

Рассмотрим программу, состоящую из одной команды вида

$$uvw \rightarrow uf + gw.$$

Применимость

Эта программа применима к ансамблю M , если существуют такие цепочки $u, w \in L^*$, что $F_M(uvw) > 0$.

Финитность

Лемма 1. Программа P , состоящая из одной команды распада $uvw \rightarrow uf + gw$ не будет финитной, если $v \subset f$ или $v \subset g$.

Доказательство. Пусть $v \subset f$ или $v \subset g$, т.е. команда программы имеет вид $uvw \rightarrow uf_1vf_2 + gw$ или $uvw \rightarrow uf + g_1vg_2w$. В каждом из этих случаев применение программы порождает новую цепочку uf_1vf_2 или g_1vg_2w , к которой опять применима программа и т.д. бесконечно. Лемма доказана.

Пример. $uAw \rightarrow uAB + BAu$. Программа применима к любому ансамблю, хотя бы одно слово которого содержит символ A . При каждом распаде такого слова будут порождаться еще два слова, к которым применима эта программа, поэтому такая программа не будет финитной.

Лемма 2. Программа P , состоящая из одной команды распада $uvw \rightarrow uf + gw$ не будет финитной для ансамбля M , если

- 1) существует два, возможно одинаковых разбиения цепочки $v = a_1b_1 = a_2b_2$;
- 2) $f = b_1b_2x$, $g = ua_1a_2$, где x и u – некоторые фиксированные цепочки;
- 3) Существует слово uvw ($F_M(uvw) > 0$) такое, что $u = u_1a_1$ или $w = b_2w_1$.

Доказательство. По условиям леммы:

$$v = a_1b_1 = a_2b_2, uvw \rightarrow ub_1b_2x + ua_1a_2w,$$

$$F_M(u_1a_1vw) > 0 \text{ или } F_M(uvb_2w_1) > 0.$$

Для доказательства не финитности достаточно указать одну бесконечную последовательность элементарных событий.

Разберем распад слова u_1a_1vw :

$$1) \{u_1a_1vw\} \mapsto \{u_1\underline{a_1}\underline{b_1}b_2x, ua_1a_2w\};$$

- 2) к слову $u_1 \underline{a_1} \underline{b_1} b_2 x$ программа применима, т.к. $v = a_1 b_1$:
 $\{u_1 \underline{a_1} \underline{b_1} b_2 x\} \mapsto \{u_1 b_1 b_2 x, y a_1 \underline{a_2} \underline{b_2} x\}$;
- 3) к слову $y a_1 \underline{a_2} \underline{b_2} x$ программа применима, т.к. $v = a_2 b_2$:
 $\{y a_1 \underline{a_2} \underline{b_2} x\} \mapsto \{y \underline{a_1} \underline{b_1} b_2 x, y a_1 a_2 x\}$.
- 4) слово $y \underline{a_1} \underline{b_1} b_2 x$ содержит цепочку $\underline{a_1} \underline{b_1} b_2$, как и слово $u_1 \underline{a_1} \underline{b_1} b_2 x$, распад которого разбирался в пункте 2). Это слово породит слово, содержащее цепочку $a_1 \underline{a_2} \underline{b_2}$, распад которого разбирался в пункте 3). И так бесконечно на каждом шаге распада будут поочередно порождаться слова, содержащие цепочки $\underline{a_1} \underline{b_1} b_2$ или $a_1 \underline{a_2} \underline{b_2}$.

Теперь разберем распад слова $uvb_2 w_1$:

- 1) $\{uvb_2 w_1\} \mapsto \{u b_1 b_2 x, y a_1 a_2 b_2 w_1\}$;
- 2) слово $y a_1 a_2 b_2 w_1$ содержит цепочку $a_1 a_2 b_2$, оно, как и в предыдущем случае, породит слово, содержащее цепочку $\underline{a_1} \underline{b_1} b_2$ и т. д.

Таким образом, мы указали бесконечные последовательности событий при условиях леммы. Лемма доказана.

Критерий финитности распада

Программа P , состоящая из одной команды распада $uvw \rightarrow uf + gw$ финитна для ансамбля M , если

I. $v \not\subset f, v \not\subset g$

II. для любых двух разбиений $v = a_1 b_1 = a_2 b_2$, где возможно $a_1 = a_2$ и $b_1 = b_2$, хотя бы одно из трех условий не выполняется:

1. $f = b_1 b_2 f_1$, где f_1 – произвольная фиксированная цепочка
2. $g = g_1 a_1 a_2$, где g_1 – произвольная фиксированная цепочка
3. Существует слово uvw , $F_M(uvw) > 0$ такое, что $u = u_1 a_1$ или $w = b_2 w_1$, где u_1 и w_1 – произвольные цепочки.

Доказательство. Рассмотрим произвольное слово вида $uvw \in M$. Построим для него в виде двоичного дерева вывод всех возможных слов. Во второй строчке каждого узла дерева будем обозначать условия, при которых к данному слову применима программа распада, в третьей строчке будет вид данного слова при этих условиях. Потомок наследует условия всех своих предков. Стрелкой влево будем показывать слово вида uf , стрелкой вправо – gw . Правую и левую ветви дерева приведем отдельно.

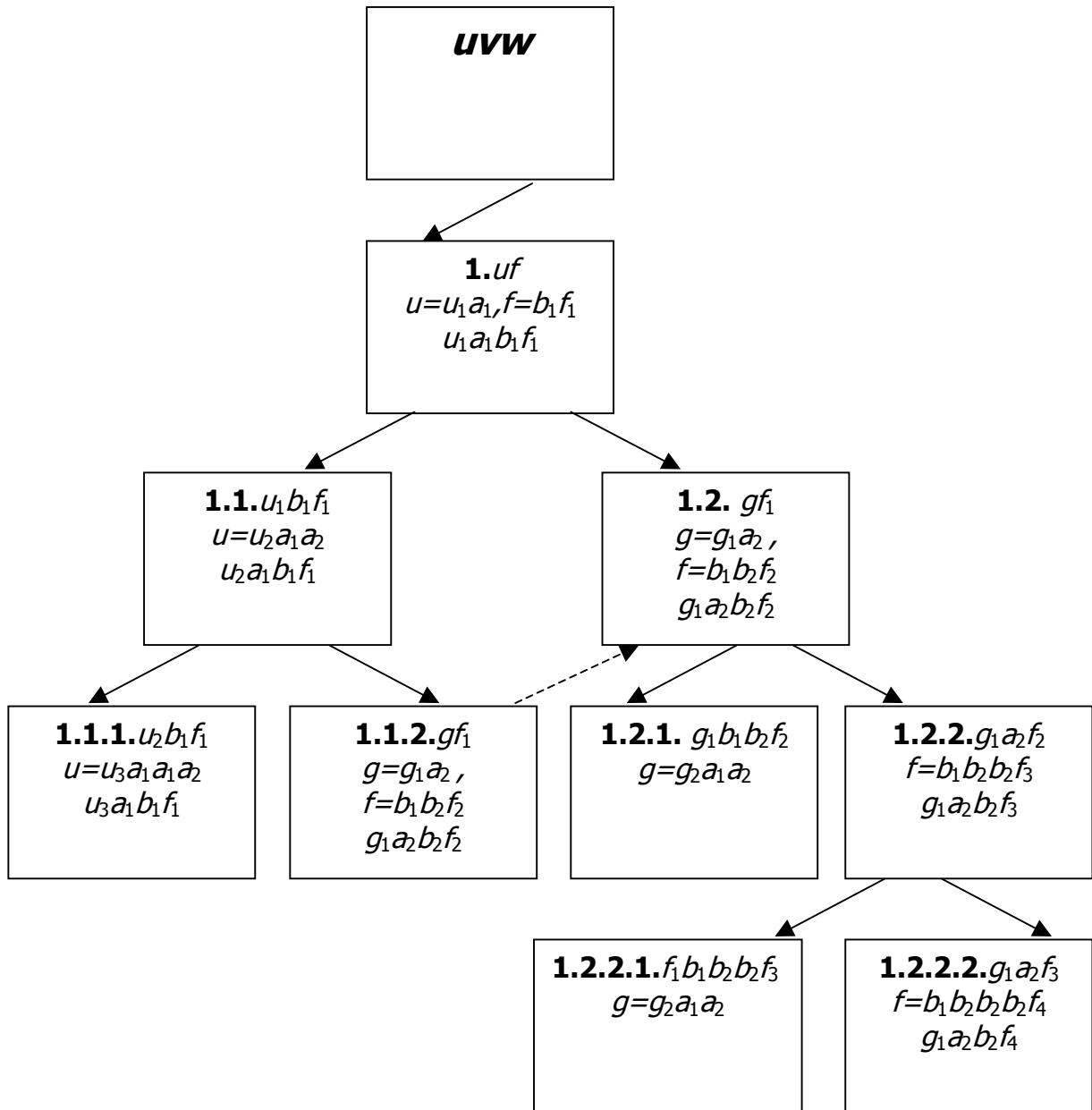


Рис. 2. Левая ветвь дерева распада.

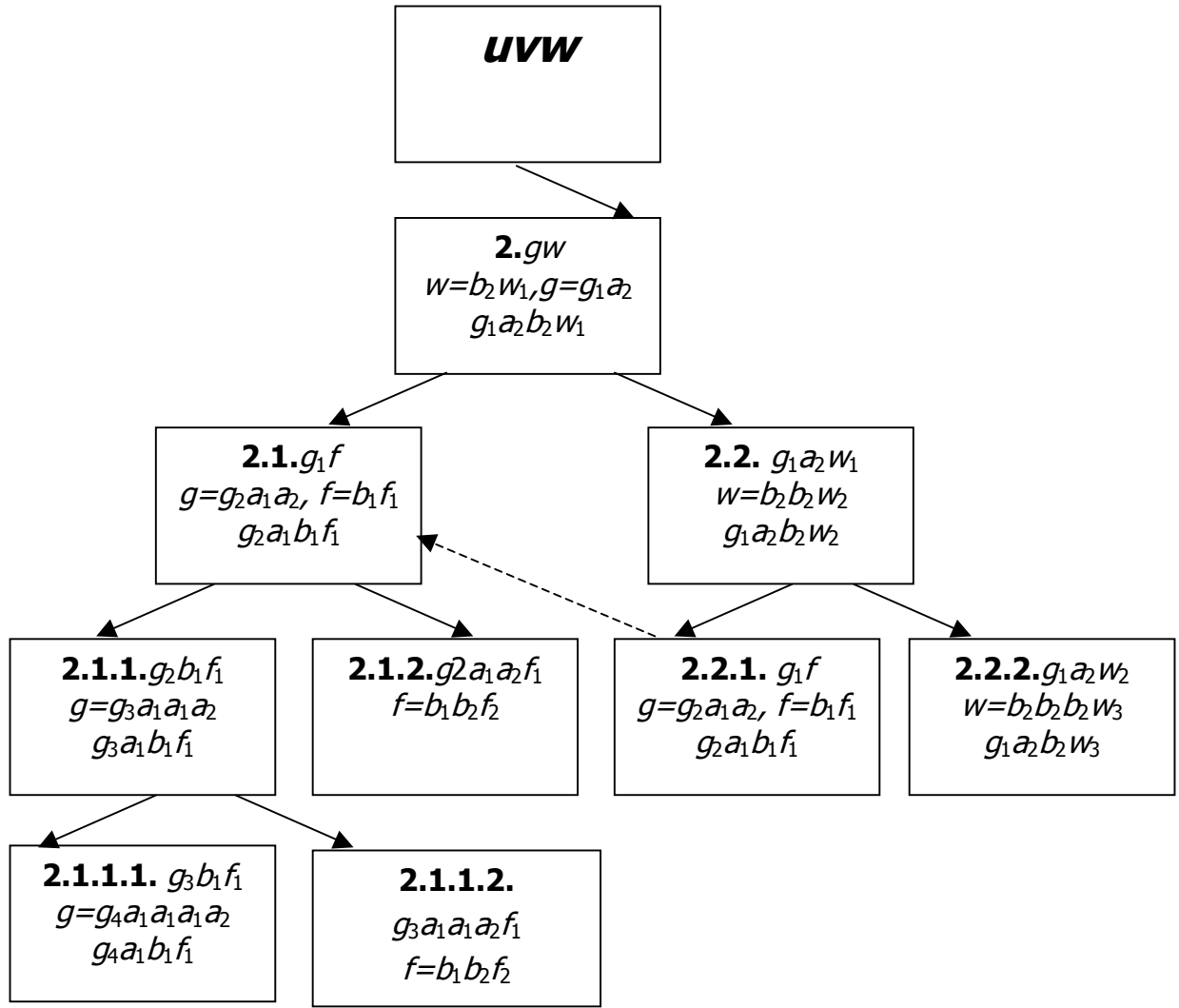


Рис. 3. Правая ветвь дерева распада

Дадим пояснения к схемам. Самая левая ветвь максимально будет иметь длину равную $|u|$ при условии, что подцепочка u имеет вид $a_1...a_1a_2$, в остальных случаях эта ветвь будет короче. Вправо на каждом шаге она будет порождать слово gf_1 , условия распада которого аналогичны ветви 1.2. Ветвь 1.2 вправо максимально будет иметь длину равную $|f|$ при условии, что подцепочка f имеет вид $b_1b_2...b_2$, в остальных случаях эта ветвь будет короче.

Влево на каждом шаге ветвь 1.2 будет порождать цепочки вида $g_1b_1b_2...b_2f_2$, которые получены при условии $f=b_1b_2f_2$, а их распад возможен только в случае если $g=g_2a_1a_2$, а это противоречит условиям критерия.

Правая ветвь двоичного дерева симметрична с точностью до переобозначений.

Таким образом, мы видим, что при выполнении условий критерия все последовательности элементарных событий будут конечны для программы, состоящей из одной команды распада, что и следовало доказать.

Пример. Нефинитная программа: $uabw \rightarrow ubb + aaw$

Пусть слово $aab \in M$.

$aab \rightarrow \{abb, aa\};$

$abb \rightarrow \{bb, aab\}.$

Получили слово aab , с которого начали. К нему данная программа также применима.

Детерминированность.

Программа не будет детерминированной, если в исходном ансамбле M_0 или в одном из ансамблей, полученных одним или несколькими элементарными событиями на ансамбле M_0 , существует цепочка, из которой вычленение подцепочек v происходит неоднозначно. Это может происходить в тех случаях, когда один или несколько символов из начала цепочки v содержится в её конце в том же порядке.

Пусть цепочка v включена в слово h ($F_M(h) > 0$) более одного раза, не пересекаясь. Т.е. слово h можно представить в виде $u_1 v^{(1)} u_2 v^{(2)} \dots u_n v^{(n)} w$, где n – число включений v в h . Элементарные события, происходя в любом порядке, разобьют слово h на $n+1$ слова: $u_1 f$, $gu_2 f$, ..., $gu_n f$, gw .

Критерий детерминированности распада

Программа распада недетерминирована, если существует разбиение цепочки $v = aba$ и выполняется хотя бы одно из условий:

1. $\exists u, w$ такие, что $F_M(uababaw) > 0$;
2. $f = xababau$ или $g = xababau$, где x, u – фикс. цепочки;
3. $f = df_1$, $\exists u_1, w$ такие, что $F_M(u_1 cvw) > 0$, где $cd = ababa$;

4. $g=g_1c$, $\exists u, w_1$ такие, что $F_M(uv dw_1) > 0$, где $cd=ababa$.

В остальных случаях распад детерминирован.

Доказательство. По условию $uabaw \rightarrow uf+gw$.

1. $\exists u, w$ такие, что $F_M(uababaw) > 0$. К слову $uababaw$ распад применим двумя способами:

$uababaw \rightarrow uf+gbaw$ и $uababaw \rightarrow uabf+gw$, что приведет к двум различным ансамблям. Следовательно, в этом случае распад недетерминирован.

2. $f=xababay$ или $g=xababay$, где x, y – фикс. цепочки.

Команда распада имеет один из следующих видов:

$uabaw \rightarrow uxababay + gw$;

$uabaw \rightarrow uf+gxababay$;

$uabaw \rightarrow uxababay+gx_1ababay_1$.

В каждом из этих случаев, любое допустимое элементарное событие порождает ансамбль, содержащий слова с неоднозначным вычленением цепочки aba , к которой применим пункт 1 данного критерия.

3. $f=df_1$, $\exists u_1, w$ такие, что $F_M(u_1cvw) > 0$, где $cd=ababa$.

Здесь cd – это некоторое, возможно, другое разбиение цепочки $ababa$.

Команда распада принимает вид:

$uabaw \rightarrow udf_1+gw$.

Применением программы к слову u_1cvw получаем два слова u_1cdf_1 и gw .

$u_1cdf_1 = u_1ababaf_1$, а применение распада к этому слову недетерминированно (см. пункт 1)

4. $g=g_1c$, существуют u, w_1 такие, что $F_M(uvdw_1) > 0$, где $cd=ababa$.

Команда распада принимает вид:

$uabaw \rightarrow uf+g_1cw$.

Применением программы к слову $uvdw_1$ получаем два слова uf и g_1cdw .

$g_1cdw = g_1ababaw$, а применение распада к этому слову недетерминированно (см. пункт 1)

Если существует разбиение цепочки $v=aba$, но не выполняется ни одно из условий 1)-4), то в исходном ансамбле, а также во всех ансамблях, полученных элементарными событиями из исходного, не появляется слова, содержащего цепочку $ababa$. Следовательно, такой распад также будет детерминирован.

Пусть цепочку v невозможно представить в виде цепочки aba , тогда невозможны пересекающиеся включения v ни в какое слово. Следовательно, невозможна недетерминированность ни для какого ансамбля.

Критерий детерминированности распада доказан.

Пример. $uABABw \rightarrow uf + gw, F_M(ABABAB) > 0$.

Цепочка v представима в виде разбиения aba : $a=AB$, b – пустая цепочка. Слово $ABABAB$ удовлетворяет условию 1) критерия, следовательно, распад этого слова будет недетерминирован. Действительно, из слова $ABABAB$ можно получить два различных ансамбля применением команды распада:

$$F_{M1}(f)=1, F_{M1}(gAB)=1;$$

$$F_{M2}(Abf)=1, F_{M2}(g)=1.$$

Ансамбль «Райский сад» для программы распада не содержит ни одной пары цепочек uf и gw .

4.2. Синтез

Рассмотрим программу, состоящую из одной команды вида

$$uk+qw \rightarrow usw.$$

Применимость. Эта программа применима ко всем ансамблям, содержащим пару слов вида uk и qw . $F_M(uk) > 0, F_M(qw) > 0$

Финитность. Очевидно, что программа всегда финитна. Она будет склеивать все пары соответствующих слов до тех пор, пока они не исчерпаются.

Детерминированность. В общем случае программа будет недетерминирована, так как склеивает произвольные цепочки неоднозначно.

Критерий детерминированности синтеза

Программа синтеза детерминированна тогда и только тогда, когда выполняется хотя бы одно из условий:

1. $\left\{ \left\{ uk \mid F_M(uk) > 0 \right\} = 1 \text{ и } \left\{ qw \mid F_M(qw) > 0 \right\} = 1 \right\}$
2. $\left\{ \left\{ uk \mid F_M(uk) > 0 \right\} = 1 \text{ и } F_M(uk) \geq \sum_{qw \in M} F_M(qw) \right\}$
3. $\left\{ \left\{ qw \mid F_M(qw) > 0 \right\} = 1 \text{ и } F_M(qw) \geq \sum_{uk \in M} F_M(uk) \right\}$

Доказательство.

1. Пусть в исходном ансамбле $F_M(uk)=n$, $F_M(qw)=m$, где u и w – некоторые фиксированные цепочки, и других слов вида uk и qw нет. Финальным ансамблем для данного ансамбля будет всегда следующий:

При $n < m$ $F_{M'}(usw) = F_M(usw) + n$, $F_{M'}(uk) = 0$, $F_{M'}(qw) = m - n$.

При $m < n$ $F_{M'}(usw) = F_M(usw) + m$, $F_{M'}(uk) = n - m$, $F_{M'}(qw) = 0$.

2. Пусть в исходном ансамбле

$F_M(uk)=n$, $F_M(qw_1)=m_1$, $F_M(qw_2)=m_2, \dots, F_M(qw_l)=m_l$,

где u , w_1 , w_2, \dots, w_l – некоторые фиксированные цепочки, причем

$\sum_{i=1}^l m_i \leq n$, и других слов вида uk и qw нет. Финальным ансамблем для

данного ансамбля будет всегда следующий:

$F_{M'}(uk) = n - \sum_{i=1}^l m_i$, $F_{M'}(qw_1)=0$, $F_{M'}(qw_2)=0, \dots, F_{M'}(qw_l)=0$,

$F_{M'}(usw_1)=m_1$, $F_{M'}(usw_2)=m_2, \dots, F_{M'}(usw_l)=m_l$.

3. Пусть в исходном ансамбле

$F_M(qw)=m$, $F_M(u_1k)=n_1$, $F_M(u_2k)=n_2, \dots, F_M(u_lk)=n_l$,

где w , u_1 , u_2, \dots, u_l – некоторые фиксированные цепочки, причем $\sum_{i=1}^l n_i \leq m$,

и других слов вида uk и qw нет. Финальным ансамблем для данного ансамбля будет всегда следующий:

$F_{M'}(qw) = m - \sum_{i=1}^l n_i$, $F_{M'}(u_1k)=0$, $F_{M'}(u_2k)=0, \dots, F_{M'}(u_lk)=0$,

$$F_M(u_1sw)=n_1, F_M(u_2sw)=n_2, \dots, F_M(u_lsw)=n_l.$$

Видно, что условие 1 является частным случаем условия 2 или 3 в зависимости от того, что больше: n или m . Если эти условия не выполняются, то можно указать, по крайней мере, два различных ансамбля, которые будут финальны для данной программы и исходного ансамбля. Действительно, пусть исходным ансамблем будет следующий:

$$F_M(uk)=n, F_M(qw_1)=m_1, F_M(qw_2)=m_2, \dots, F_M(qw_l)=m_l,$$

где u, w_1, w_2, \dots, w_l – некоторые фиксированные цепочки, но $\sum_{i=1}^l m_i \geq n$ и других слов вида uk и qw нет, и пусть для определенности $\forall j=1 \dots l$ $\sum_{i=1}^l m_i - m_j \leq n$. Укажем два возможных финальных ансамбля.

$$1. F_M'(uk)=0, F_M'(qw_1)=\sum_{i=1}^l m_i - n, F_M'(qw_2)=0, \dots, F_M'(qw_l)=0,$$

$$F_M'(usw_1)=m_1 - \sum_{i=1}^l m_i + n, F_M'(usw_2)=m_2, \dots, F_M'(usw_l)=m_l.$$

$$2. F_M'(uk)=0, F_M'(qw_1)=0, F_M'(qw_2)=\sum_{i=1}^l m_i - n, \dots, F_M'(qw_l)=0,$$

$$F_M'(usw_1)=m_1, F_M'(usw_2)=m_2 - \sum_{i=1}^l m_i + n, \dots, F_M'(usw_l)=m_l.$$

Недетерминированность в случае невыполнения условия 3 доказывается аналогично. Таким образом, критерий детерминированности для программы, состоящей из одной команды синтеза, доказан.

Пример недетерминированного синтеза

$$uK+Qw \rightarrow uSw. F_M(AK)=1, F_M(QB)=1, F_M(QA)=1$$

В ансамбле M одно слово вида uK и два различных слова вида Qw , следовательно, этот ансамбль не удовлетворяет ни одному из условий критерия. Значит, синтез будет недетерминирован. Действительно, здесь есть два допустимых несовместных события. Финальными могут являться два различных ансамбля:

$$F_{M1}(ASA)=1, F_{M1}(QB)=1$$

и

$$F_{M2}(ASB) = 1, F_{M2}(QA) = 1.$$

Пример детерминированного синтеза.

$$uK + Qw \rightarrow uSw. F_M(AK) = 3, F_M(QB) = 1, F_M(QA) = 1$$

Данный ансамбль попадает под условие 2 критерия, следовательно, синтез будет детерминирован. Финальным будет ансамбль M_1 такой, что

$$F_M(AK) = 1, F_M(ASB) = 1, F_M(ASA) = 1$$

Ансамблями «Райский сад» для этой программы будут все такие ансамбли M , что для любого слова t такого, что $F_M(t) > 0$ $s \not\subset t$.

4.3. Прямая замена

Рассмотрим программу, состоящую из одной команды вида $uvw \rightarrow usw$.

Применимость. Эта программа применима к ансамблю M , если существуют такие цепочки $u, w \in L^*$, что $F_M(uvw) > 0$.

Финитность. Прямая замена не изменяет количества слов в ансамбле, она будет финитна если $v \not\subset s$.

Детерминированность. Условия детерминированности прямой замены сходны с условиями детерминированности распада. Здесь также проверяется разложимость цепочки v на подцепочки вида aba и отслеживается вхождение цепочки вида $ababa$ в слова исходного ансамбля или ансамбля, полученного применением программы прямой замены к исходному.

Критерий детерминированности прямой замены

Прямая замена не детерминирована, если существует разбиение цепочки $v = aba$ и выполняется хотя бы одно из условий:

1. $\exists u, w$ такие, что $F_M(uababaw) > 0$;
2. $s = xababay$, где x, y – фикс. цепочки;
3. $s = ds_1$, $\exists u_1, w$ такие, что $F_M(u_1cvw) > 0$, где $cd = ababa$;
4. $s = s_1c$, $\exists u, w_1$ такие, что $F_M(uv dw_1) > 0$, где $cd = ababa$;
5. $s = d$, $\exists u_1, w_1$ такие, что $F_M(u_1c v ew_1) > 0$, где $cde = ababa$;

В остальных случаях прямая замена детерминирована.

5. Заключение

В статье представлена идеальная ансамблевая модель параллельных вычислений – кинетическая машина Кирдина, рассмотрены простые программы для нее и доказаны критерии финитности и детерминированности для таких программ. Ожидается, что эта модель сыграет ту же роль для параллельных вычислений, что и нормальные алгоритмы Маркова, машины Колмогорова и Тьюринга или схемы Поста для последовательных вычислений.

Работа выполнена при поддержке ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.) и Красноярского краевого фонда науки (грант 7F0113).

ЛИТЕРАТУРА

1. Кирдин А.Н. Идеальная ансамблевая модель параллельных вычислений // Нейроинформатика и ее приложения. Тезисы докладов V Всеросс. семинара. – Красноярск, КГТУ, 1997. С.101.
2. Горбунова Е.О. Анализ простейших программ для идеальной ансамблевой модели параллельных вычислений // Тез. докл. Третьего сибирского конгресса по прикладной и индустриальной математике (ИНПРИМ – 98). – Новосибирск: изд-во Института математики, 1998. – С.77.

АЛГОРИТМИЧЕСКАЯ УНИВЕРСАЛЬНОСТЬ КИНЕТИЧЕСКОЙ МАШИНЫ КИРДИНА

Е.О.Горбунова

*Институт вычислительного моделирования СО РАН,
Красноярский государственный технический университет
660036, Красноярск-36, ИВМ СО РАН,
E-mail: gkat@cc.krascience.rssi.ru*

Рассматривается новая ансамблевая модель параллельных мелкозернистых вычислений – кинетическая машина Кирдина (КМК). Доказывается, что детерминированная КМК является эффективным вычислителем. Приводится метод реализации частично-детерминированной КМК при помощи специально устроенной системы машин Тьюринга. Дается понятие о статистической реализации КМК.

Кинетическая машина Кирдина является моделью вычислений в условиях идеального мелкозернистого параллелизма. В предыдущей статье было дано понятие кинетической машины Кирдина и рассмотрены критерии финитности и детерминированности. Неформально, сводя полученные результаты, можно сказать, что команды распада и прямой замены недетерминированы, если выполнены условия 1&2 или 1&3 из следующего списка.

1. Заменяемая цепочка v имеет разложение типа aba , т.е. начало и конец ее совпадают.
2. В словах, к которым применимы эти команды, встречаются цепочки вида $ababa$, т.е. цепочку v можно выделить двумя способами.
3. В словах, полученных после применения этих команд, встречаются цепочки вида $ababa$, т.е. цепочку v можно выделить двумя способами.

Для программ, состоящих из произвольного числа команд распада и прямой замены, эти критерии легко обобщаются.

Программа, состоящая из команд синтеза, всегда финитна и в общем случае недетерминирована.

Мы предполагаем, что КМК будет носить универсальный характер. В связи с этим возникает вопрос, как КМК соотносится со стандартными последовательными формализмами.

Существует много формальных способов описания последовательных алгоритмов. Например, следующие:

1. Машины Тьюринга.
2. Грамматики Хомского типа 0.
3. Нормальные алгоритмы Маркова.
4. Системы Поста.
5. Большинство языков программирования.

Алгоритм, записанный в одном из этих формализмов, можно промоделировать алгоритмом, записанным в любом другом из них. В этом смысле все перечисленные формализмы эквивалентны. Далее мы промоделируем работу машины Тьюринга с помощью кинетической машины Кирдина. МТ детерминирована, следовательно соответствующая ей КМК также будет детерминирована и будет состоять только из команд прямой замены. Такую КМК мы будем моделировать с помощью нормальных алгоритмов Маркова (НАМ). Отсюда получим алгоритмическую универсальность КМК, поскольку существует тезис Чёрча-Тьюринга, который говорит, что любой вычислительный процесс, который можно разумно назвать алгоритмом, можно промоделировать на машине Тьюринга.

Сначала дадим кратко понятие о МТ и НАМ.

Машина Тьюринга состоит из:

1) управляющего устройства, которое может находиться в одном из состояний, образующих конечное множество $Q = \{q_1, \dots, q_n\}$. Среди состояний выделены начальное состояние q_1 и заключительное состояние q_z ;

2) бесконечной ленты, разбитой на ячейки, в каждой из которых может быть записан один из символов конечного алфавита $A = \{a_1, \dots, a_m\}$. В начальный момент времени только конечное число ячеек ленты заполнено непустыми символами, остальные ячейки содержат пустой символ λ ;

3) считывающей и пишущей головки, которая в каждый момент времени обозревает ячейку ленты, в зависимости от символа в этой ячейке

и состояния управляющего устройства записывает в ячейку символ, сдвигается на ячейку влево или вправо, или остается на месте; при этом управляющее устройство переходит в новое состояние .

Для любого внутреннего состояния q_i и символа a_j однозначно заданы:

- а) следующее состояние q_i' ;
- б) символ a_j' , который нужно записать вместо a_j в ту же ячейку;
- в) направление сдвига головки d_k , обозначаемое символом в алфавите $D=\{L,R,E\}$ ($L=Left$, $R=Right$, $E=Equal$).

Это задание описывается таблицей или системой команд, имеющих вид $q_i a_j \rightarrow q_i' a_j' d_k$.

Полную конфигурацию машины Тьюринга обозначим тройкой $\alpha_1 q_i \alpha_2$, где q_i — текущее внутреннее состояние, α_1 — слово слева от головки, α_2 — слово, образованное символом, обозреваемым головкой, и символами справа от него, причем слева от α_1 и справа от α_2 нет непустых символов.

В качестве элементарной операции, на базе которой строятся *нормальные алгоритмы Маркова*, выделяется подстановка одного слова вместо другого. Если P и Q —слова в алфавите A , то выражения $P \rightarrow Q$ и $P \rightarrow \cdot Q$ будем называть *простой и заключительной формулами подстановки* в алфавите A соответственно. Конечный список формул подстановки в алфавите A

$$\left\{ \begin{array}{l} P_1 \rightarrow (\cdot) Q_1 \\ P_2 \rightarrow (\cdot) Q_2 \\ \vdots \\ P_r \rightarrow (\cdot) Q_r \end{array} \right.$$

называется *схемой алгоритма* и порождает следующий алгоритм в алфавите A . Условимся предварительно говорить, что слово T *входит* в слово Q , если существуют такие (возможно пустые) слова U, V , что $Q=UTV$.

Работа алгоритма может быть описана следующим образом. Пусть дано слово P в алфавите A . Находим первую в схеме алгоритма формулу

подстановки $P_m \rightarrow (\cdot) Q_m$ такую, что P_m входит в P . Совершаем подстановку слова Q_m вместо самого левого вхождения слова P_m в слово P . Пусть R_1 – результат такой подстановки. Если $P_m \rightarrow (\cdot) Q_m$ – заключительная формула подстановки, то работа алгорифма заканчивается и его значением является R_1 . Если формула подстановки $P_m \rightarrow (\cdot) Q_m$ – простая, то применим к R_1 тот же поиск, который был только что применен к P , и так далее. Если мы в конце концов получим такое слово R_i , что ни одно из слов P_1, \dots, P_r не входит в R_i , то работа алгорифма заканчивается и R_i будет его значением. При этом возможно, что описанный процесс никогда не закончится. В таком случае мы говорим, что алгорифм неприменим к слову P . Алгорифм, определенный таким образом, называется *нормальным алгорифмом* (или *алгорифмом Маркова*) в алфавите A .

Теорема. *Детерминированная кинетическая машина Кирдина эквивалентна любому последовательному стандартному алгоритмическому формализму, такому как машина Тьюринга, нормальные алгоритмы Маркова, порождающие грамматики Хомского типа 0. Следовательно, она является эффективным вычислителем.*

Доказательство. Построим КМК, моделирующую произвольную машину Тьюринга. Поскольку машина Тьюринга детерминирована и последовательна, КМК будет детерминирована и в каждый момент времени будет существовать только одно допустимое событие. Алфавитом для КМК будет $L = A \cup Q$. Начальный ансамбль состоит из одного слова, которое соответствует стандартной начальной конфигурацией машины

Команда МТ	Команда КМК
$q_i a_j \rightarrow q_i' a_j' R$	$u q_i a_j w \rightarrow u a_j' q_i' w$ $u a_j' q_i' \rightarrow u a_j' q_i' \lambda$
$q_i a_j \rightarrow q_i' a_j' L$	$u a_1 q_i a_j w \rightarrow u q_i' a_1 a_j' w$... $u a_m q_i a_j w \rightarrow u q_i' a_m a_j' w$ $q_i a_j w \rightarrow q_i' \lambda a_j' w$
$q_i a_j \rightarrow q_i' a_j' E$	$u q_i a_j w \rightarrow u q_i' a_j' w$

Тьюринга $M_0 = \{ q_0 \alpha \}$, где $\alpha \in A^*$. Моделирование команд машины Тьюринга опишем таблицей, в первом столбце которой находится команда машины Тьюринга, во втором — соответствующая ей система команд КМК.

Программа для полученной детерминированной КМК состоит только из команд прямой замены.

Теперь рассмотрим моделирование детерминированной КМК, состоящей из команд прямой замены, с помощью нормальных алгоритмов Маркова.

Все команды $uvw \rightarrow usw$ КМК заменяем на правила $v \rightarrow s$. В конце добавляем заключительное правило подстановки $\lambda \rightarrow \cdot \lambda$, где λ пустой символ. Теорема доказана.

Проблема остановки для машины Тьюринга в общем случае неразрешима, следовательно *финитность КМК для программ, состоящих из команд прямой замены, также неразрешима.*

В общем случае КМК недетерминированна. Естественно, что она не может быть полностью эквивалентна детерминированному устройству. Тем не менее, мы увидели, что она полностью включает в себя все детерминированные универсальные вычислители. Что же можно сказать о КМК в недетерминированном случае? Выделим еще один класс КМК.

Назовем КМК *частично детерминированной*, если ее программа состоит из детерминированных команд прямой замены и распада и произвольных команд синтеза.

Частично детерминированную КМК можно промоделировать специально устроенной системой алгоритмических формализмов, пусть для определенности это будут машины Тьюринга.

Пусть у нас есть частично детерминированная КМК, $|M|$ — количество слов в ее ансамбле. Изначально иницируется $|M|$ машин Тьюринга, каждая из них обрабатывает собственное слово. Программа для каждого состоит из команд прямой замены КМК и команд распада. Применение команды распада означает инициацию новой МТ, причем конфигурация первой соответствует слову uf , а конфигурация второй — слову gw .

При этом одновременно работает некий надвычислитель, программа которого состоит из всех команд синтеза исходной программы.

Он сравнивает конфигурации устройств со словами uk и qw . Если одновременно существуют обе эти конфигурации, то он отключает одно из этих устройств, а конфигурацию другого устанавливает usw .

Представляет особый интерес “статистический” метод реализации КМК. В пределе больших ансамблей он даст кинетическое поведение, аналогичное поведению сложной системы химических реакций (откуда, собственно, и происходит название “*кинетическая* машина Кирдина”). При статистической реализации каждой (i -й) команде сопоставляется некоторое неотрицательное число r_i - “константа скорости”. Саму реализацию можно представить так: за малое Δt с вероятностью $r_i \Delta t$ выбирается одна из команд-реакций, а с вероятностью $1 - \sum_i r_i \Delta t$ - пустая команда (ничего не происходит); вероятность выбора двух или более команд пренебрежимо мала ($o(\Delta t)$). Если выбрана i -я команда-реакция, то из ансамбля случайно и равновероятно выбираются слова в количестве, необходимом для выполнения команды. Если команда применима к этому набору слов, то она исполняется и переходим к следующему Δt , если же она неприменима, то ансамбль не изменяется и также переходим к следующему Δt . В пределе $\Delta t \rightarrow 0$ получаем случайный процесс - статистическую модель КМК.

В работе рассмотрены различные методы реализации кинетической машины Кирдина: от детерминированной до статистической и показано, что она полностью включает в себя все детерминированные универсальные вычислители.

Работа выполнена при поддержке ФЦП “ИНТЕГРАЦИЯ” (проект № 68, напр. 2.1.) и Красноярского краевого фонда науки (грант 7F0113).

ЛИТЕРАТУРА

1. Кирдин А.Н. Идеальная ансамблевая модель параллельных вычислений // Нейроинформатика и ее приложения. Тезисы докладов V Всеросс. семинара. – Красноярск, КГТУ, 1997. С.101.
2. Горбунова Е.О. Анализ простейших программ для идеальной ансамблевой модели параллельных вычислений // Тез. докл. Третьего сибирского конгресса по прикладной и индустриальной математике

(ИНПРИМ – 98). – Новосибирск: изд-во Института математики, 1998. – С.77.

3. Марков А.А., Нагорный Н.М. Теория алгоритмов.– М.: Наука. Гл. ред. физ.-мат. лит., 1984.
4. Успенский В.А., Семенов А.Л. Теория алгоритмов: основные открытия и приложения.– М.: Наука. Гл. ред. физ.-мат. лит., 1987.– (Б-чка программиста).

ПОГРЕШНОСТИ НЕЙРОННЫХ СЕТЕЙ. ВЫЧИСЛЕНИЕ ПОГРЕШНОСТЕЙ ВЕСОВ СИНАПСОВ

М.Ю.Сенашова

Институт вычислительного моделирования СО РАН

660036, Красноярск-36, ИВМ СО РАН,

E-mail: amse@cc.krascience.rssi.ru

Рассматриваются нейронные сети слоистой структуры, состоящие из слоев стандартных нейронов. Изучаются ошибки, возникающие при технической реализации сетей, а также при шумах и повреждениях.

Определены максимально допустимые погрешности, возможные для весов синапсов и сигналов сети, исходя из условия, что вектор выходных сигналов сети должен вычисляться с заданной точностью. Используются два типа оценок погрешности: гарантированные интервальные оценки и среднеквадратические оценки погрешностей.

Показано, что оценки допустимых погрешностей можно получить в ходе специального процесса “обратного распространения точности”. Он состоит в функционировании сети с той же системой связей, но от выходов к входам и с заменой элементов на двойственные. Эта двойственность принципиально отличается от той, которая используется в классическом методе вычисления градиентов оценки с помощью обратного распространения ошибок (back propagation of errors).

1. Введение

В настоящее время существуют различные технические реализации нейронных сетей, в том числе нейроимитаторы, то есть компьютерные модели нейронных сетей. Нейроимитаторы являются гибкими средствами для изучения сетей и работы с ними. С нейроимитаторами можно выполнять различные операции – обучать, определять наиболее и наименее значимые связи, контрастировать, то есть удалять наименее значимые связи и т. д.

Существует подход, получающий все большее распространение, при котором сначала конструируется и обучается нейроимитатор, а затем

создается техническая реализация полученной нейросети с уже вычисленными весами синапсов.

Нейроимитатор, работающий на универсальных цифровых ЭВМ, позволяет вычислять веса синапсов с большой точностью, которую трудно получить при других технических реализациях сети (в первую очередь – аналоговых) в силу ограниченной точности технических устройств. Поэтому следует ограничиться некоторой точностью, с которой может работать сеть, то есть выбрать интервал, в котором могут изменяться значения вектора выходных сигналов сети. Исходя из выбранной погрешности для выходных сигналов сети, следует вычислить допустимые погрешности для весов синапсов и сигналов сети.

Оценка погрешностей сигналов сети очень полезна при решении задачи упрощения нейронной сети. Зная допустимую погрешность выходного сигнала какого-либо элемента сети, мы можем заменять его более простыми, но менее точными элементами так, чтобы в итоге ошибка не превышала заданную.

2. Структура сети

Предполагаем, что сеть имеет слоистую структуру. Это самоограничение позволит несколько сократить изложение, но не влияет на общность рассмотрения - исследование любой сети может быть формально сведено к изучению слоистых сетей.

Сеть слоистой структуры состоит из слоев стандартных нейронов, связанных между собой синапсами с весами, полученными при обучении. Причем сигналы передаются только в одном направлении, с предыдущего слоя на следующий. Под стандартным нейроном [1,2] понимается набор элементов, состоящий из адаптивного сумматора, нелинейного преобразователя и точки ветвления (рис.1). Точка ветвления – это элемент, посылающий выходной сигнал нелинейного преобразователя на вход нескольких стандартных нейронов следующего слоя.

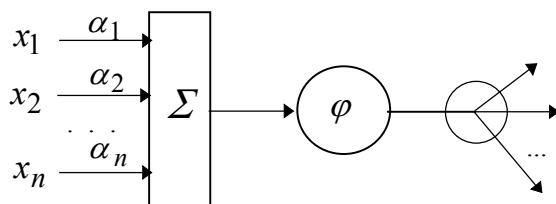


Рис.1.

Так как мы имеем дело с сетями слоистой структуры, состоящими из слоев стандартных нейронов, то выходные сигналы одного слоя являются входными сигналами другого слоя. В свою очередь, внутри самого стандартного нейрона выходной сигнал одного элемента (например, сумматора) является входным сигналом другого элемента (например, нелинейного преобразователя). Таким образом, можно проследить, начиная с выходных сигналов сети, от какого элемента сети пришел сигнал к данному элементу.

Стандартный нейрон является типичным участком любой нейронной сети. Поэтому достаточно выяснить, как вычисляются допустимые погрешности для элементов стандартного нейрона. В результате получим возможность вычислить допустимые погрешности для любого участка сети, двигаясь по сети от нейрона к нейрону в обратном направлении.

3. Два базовых подхода к оценкам погрешности

Рассмотрим два подхода к решению задачи вычисления погрешностей сигналов сети. При первом подходе (*гарантированные интервальные оценки*) вычисляются допустимые интервалы для погрешностей сигналов сети такие, что погрешность вектора выходных сигналов гарантированно (с вероятностью 1) не превышает заданную. При втором подходе (*среднеквадратические оценки погрешностей*) вычисляются среднеквадратические отклонения погрешностей сигналов. При этом часто используется предположение о том, что погрешности различных сигналов являются независимыми случайными величинами.

Существует принципиальное различие между этими двумя типами оценок. Гарантированные интервальные оценки исходят из рассмотрения наихудших возможных случаев, сколь бы малой не была их вероятность.

Поэтому они, как правило, завышают реально имеющую место ошибку и слишком пессимистичны с практической точки зрения. Среднеквадратичные оценки, наоборот, стирают возможные большие отклонения и могут оказаться слишком оптимистичными.

Важное различие между двумя типами оценок демонстрируют следующие *формулы сложения*.

1. *Формула сложения для интервальных оценок.* Пусть для двух величин x, y определены гарантированные интервалы значений $x=x_0\pm\Delta_x$, $y=y_0\pm\Delta_y$. Тогда для их суммы имеем гарантированную оценку: $x+y=x_0+y_0\pm(\Delta_x+\Delta_y)$, то есть $\Delta_{x+y}=\Delta_x+\Delta_y$.

2. *Формула сложения для среднеквадратичных отклонений* Пусть для двух независимых величин x, y определены среднеквадратичные отклонения σ_x, σ_y . Тогда $\sigma_{x+y}=(\sigma_x^2+\sigma_y^2)^{1/2}$.

4. Погрешности весов синапсов

В [6,7] описан метод обратного распространения точности для сигналов нейронных сетей. Он состоит в функционировании сети с той же системой связей, но от выходов к входам и с заменой элементов на двойственные. Эта двойственность принципиально отличается от той, которая используется в классическом методе вычисления градиентов оценки с помощью обратного распространения ошибок (back propagation of errors). В ходе процесса сигнал, проходящий по связи в обратном направлении, является допустимой погрешностью сигнала, проходящего по этой связи в прямом направлении.

В методе обратного распространения точности были получены формулы для вычисления погрешностей входных и внутренних сигналов сети при заданных погрешностях ее выходных сигналов. В работах [6,7] для элементов стандартного нейрона при известной погрешности выходного сигнала гарантированная интервальная оценка погрешности входного сигнала вычисляется следующим образом:

1) *точка ветвления.* Если погрешности выходных сигналов точки ветвления равны $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_k$, то погрешность входного сигнала точки ветвления не должна превышать $\min\{\varepsilon_i\}_{i=1}^k$;

2) *нелинейный преобразователь*. Если погрешность выходного сигнала нелинейного преобразователя равняется ε , то погрешность его входного сигнала не должна превышать $\varepsilon / \max |\varphi'(x)|$, где y – выходной сигнал нелинейного преобразователя, φ – его функция активации и $x \in [\varphi^{-1}(y - \varepsilon), \varphi^{-1}(y + \varepsilon)]$ или в линейном приближении $\varepsilon / |\varphi'(A)|$, где A – входной сигнал нелинейного преобразователя при прямом функционировании сети;

3) *адаптивный сумматор*. Если при обратном распространении погрешность выходного сигнала адаптивного сумматора равняется ε , то погрешность каждого входа сумматора не должна превышать ε_i , где $\varepsilon_i \leq \varepsilon / \sum_{i=1}^n |\alpha_i|$ для равномерного распределения и $\varepsilon_i = \varepsilon / (n \cdot \alpha_i)$ для пропорционального распределения.

Для среднеквадратических оценок погрешностей входных сигналов при известных погрешностях выходных сигналов формулы имеют вид:

1) *точка ветвления*. Если дисперсии погрешностей выходных сигналов точки ветвления равны D_1, D_2, \dots, D_k , то в качестве погрешности входного сигнала точки ветвления выбирается $\min \{D_i\}_{i=1}^k$;

2) *нелинейный преобразователь*. Если среднеквадратическое отклонение погрешности выходного сигнала нелинейного преобразователя равняется σ , то погрешность его входного сигнала не должна превышать $\sigma / |\varphi'(A)|$, где A – входной сигнал нелинейного преобразователя при прямом функционировании и φ – его функция активации;

3) *адаптивный сумматор*. Если среднеквадратическое отклонение погрешности выходного сигнала адаптивного сумматора равняется σ , то погрешность каждого входа сумматора не должна превышать σ_i , где $\sigma_i \leq \sigma / \sum_{i=1}^n \alpha_i^2$ для равномерного распределения и $\sigma_i = \sigma / (n \cdot \alpha_i)$ для пропорционального распределения.

При вычислении погрешностей сигналов предполагалось, что веса синапсов погрешностей не имеют и погрешности выходных сигналов сети образуются за счет погрешностей входных сигналов. Теперь рассмотрим противоположную задачу. Будем предполагать, что веса синапсов имеют некоторые погрешности, а входные сигналы сети погрешностей не имеют. В этом случае погрешности выходных сигналов сети образуются лишь за счет погрешностей, вносимых весами синапсов. Требуется оценить

допустимые погрешности весов синапсов. Их оценка полезна при решении задачи упрощения нейронной сети. В частности, зная погрешности весов синапсов, мы можем заменять сумматоры более простыми, но менее точными элементами так, чтобы в итоге ошибка не превышала заданную.

Пусть нам дана обученная нейронная сеть слоистой структуры, веса синапсов и сигналы которой не имеют погрешностей. Будем называть сигналы и веса синапсов этой сети точными.

Предположим теперь, что веса синапсов данной сети имеют некоторые погрешности, а входные сигналы сети не имеют погрешностей. Нам задана допустимая погрешность выходных сигналов сети. Требуется вычислить величины допустимых погрешностей весов синапсов и сигналов такие, что погрешность выходного сигнала при них не превышает заданную.

Будем вычислять допустимые погрешности весов синапсов, начиная с последнего слоя сети, продвигаясь от слоя к слою в обратном порядке.

Единственный элемент стандартного нейрона, в работе которого принимают участие веса синапсов, это адаптивный сумматор. Для вычисления допустимых погрешностей весов синапсов мы будем рассматривать именно его. При вычислении допустимых погрешностей сигналов остальных элементов стандартного нейрона мы будем пользоваться формулами, полученными в методе обратного распространения точности для сигналов сети.

Мы будем рассматривать два типа оценки погрешностей: интервальные оценки погрешностей весов синапсов и среднеквадратические оценки.

5. Гарантированные интервальные оценки погрешностей весов синапсов

Вычислим сначала допустимые погрешности весов синапсов для одного примера из обучающей выборки. Считаем, что на входы сети с точными весами синапсов были поданы точные входные сигналы и, после необходимых вычислений, получены точные выходные сигналы. Таким образом, нам известны точные сигналы для всей сети.

Нам задана допустимая погрешность δ выходного сигнала сети, который одновременно является выходным сигналом нейрона последнего слоя сети. Пользуясь методом обратного распространения точности, вычислим допустимую погрешность ε входного сигнала этого нейрона по полученной выше формуле: $\varepsilon = \delta / |\varphi'(A)|$, где A – точный выходной сигнал сумматора последнего слоя сети.

Таким образом, мы вычислили допустимую погрешность выходного сигнала сумматора. Теперь нам следует вычислить допустимые погрешности весов синапсов этого сумматора, а также погрешности его входных сигналов, которые образуются при прохождении предыдущих слоев сети. Это можно сделать двумя способами: распределяя допустимую погрешность выходного слоя сумматора по его входам пропорционально или равномерно.

Рассмотрим сначала пропорциональное распределение погрешностей. Обычно принято рассматривать адаптивный сумматор как устройство с входными сигналами x_i , умножающее x_i на веса α_i и затем их складывающее. В данном конкретном случае нам будет удобнее рассматривать адаптивный сумматор в следующем виде (рис.2).

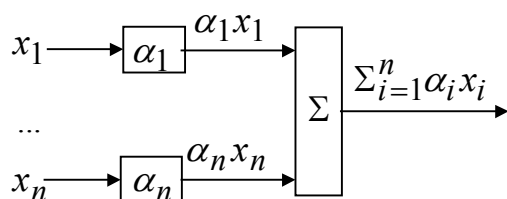


Рис. 2

Будем считать, что входными сигналами сумматора являются не x_i , а уже преобразованные весами синапсов $\alpha_i x_i$, которые затем складываются уже без умножения.

Рассмотрим сумматор с n входами. Допустимая погрешность выходного сигнала сумматора равна ε . Будем распределять эту погрешность поровну по входам сумматора. То есть погрешность каждого входа равняется $\frac{\varepsilon}{n}$. Пусть x_i – точный i -ый входной сигнал сумматора, α_i – точный вес синапса его i -го входа, ε_{I_i} – погрешность входного сигнала i -го входа, ε_{α_i} – погрешность i -го веса синапса. Рассмотрим следующие

интервалы: $[x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}]$ – возможные значения входного сигнала,

$[\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}]$ – возможные значения веса синапса, $\left[\alpha_i x_i - \frac{\varepsilon}{n}; \alpha_i x_i + \frac{\varepsilon}{n}\right]$

– интервал в котором может изменяться значение произведения i -го входного сигнала с погрешностью на i -ый вес синапса с погрешностью. Нам требуется, чтобы

$$[x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}] \leq \left[\alpha_i x_i - \frac{\varepsilon}{n}; \alpha_i x_i + \frac{\varepsilon}{n}\right] \quad (1)$$

Интервалы перемножаются следующим образом:

$$[\underline{a}; \bar{a}] \cdot [\underline{b}; \bar{b}] = [\min\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}; \max\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}].$$

Рассмотрим возможные расположения интервалов $[x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}]$ и $[\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}]$ на числовой оси.

$$1) x_i - \varepsilon_{I_i} \geq 0; \alpha_i - \varepsilon_{\alpha_i} \geq 0.$$

Перемножим интервалы:

$$\begin{aligned} & [x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}] = \\ & [(x_i - \varepsilon_{I_i})(\alpha_i - \varepsilon_{\alpha_i}); (x_i + \varepsilon_{I_i})(\alpha_i + \varepsilon_{\alpha_i})] = \\ & [\alpha_i x_i - \alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}; \alpha_i x_i + \alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}] \end{aligned}$$

Таким образом, мы получили интервал, в котором изменяются погрешность i -го входного сигнала и погрешность i -го веса синапса:

$$[-\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}; \alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}].$$

Этот интервал должен быть меньше, либо равен интервалу изменения погрешности произведения входного сигнала и веса синапса:

$\left[-\frac{\varepsilon}{n}; \frac{\varepsilon}{n}\right]$. Но интервал $\left[-\frac{\varepsilon}{n}; \frac{\varepsilon}{n}\right]$ симметричен относительно 0, а интервал

$[-\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}; \alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}]$ сдвинут вправо. Поэтому для выполнения условия (1) нам достаточно, чтобы выполнялось условие

$$\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}.$$

$$2) x_i - \varepsilon_{I_i} > 0, \alpha_i - \varepsilon_{\alpha_i} < 0, \alpha_i + \varepsilon_{\alpha_i} \geq 0.$$

Пусть $x_i > 0, \alpha_i > 0$. Рассуждая аналогично случаю 1), получаем:

$$[\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} - \varepsilon_{I_i} \varepsilon_{\alpha_i}; \alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i}] \leq \left[-\frac{\varepsilon}{n}; \frac{\varepsilon}{n}\right].$$

Отсюда $\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $x_i > 0, \alpha_i < 0$ получаем неравенство

$$-\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}.$$

$$3) x_i - \varepsilon_{I_i} < 0, x_i + \varepsilon_{I_i} \geq 0, \alpha_i - \varepsilon_{\alpha_i} \leq 0, \alpha_i + \varepsilon_{\alpha_i} > 0.$$

При $x_i > 0, \alpha_i > 0$ получаем $\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $x_i > 0, \alpha_i < 0$ получаем $-\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $x_i < 0, \alpha_i > 0$ получаем $\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $x_i < 0, \alpha_i < 0$ получаем $-\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

$$4) x_i - \varepsilon_{I_i} < 0, x_i + \varepsilon_{I_i} > 0, \alpha_i - \varepsilon_{\alpha_i} > 0.$$

При $x_i \geq 0$ получаем $\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $x_i < 0$ получаем $\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

$$5) x_i + \varepsilon_{I_i} \leq 0, \alpha_i + \varepsilon_{\alpha_i} \leq 0.$$

В этом случае $-\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

$$6) x_i + \varepsilon_{I_i} < 0, \alpha_i - \varepsilon_{\alpha_i} < 0, \alpha_i + \varepsilon_{\alpha_i} > 0.$$

При $\alpha_i \geq 0$ получаем $\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $\alpha_i < 0$ получаем $-\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

$$7) x_i - \varepsilon_{I_i} < 0, x_i + \varepsilon_{I_i} > 0, \alpha_i + \varepsilon_{\alpha_i} < 0.$$

При $x_i \geq 0$ получаем $-\alpha_i \varepsilon_{I_i} + x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

При $x_i < 0$ получаем $-\alpha_i \varepsilon_{I_i} - x_i \varepsilon_{\alpha_i} + \varepsilon_{I_i} \varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n}$.

Как видно из рассмотренных выше случаев, вид неравенства зависит не от расположения интервалов на числовой оси, а от знаков x_i и α_i .

Из рассмотренных случаев выделим виды неравенств, которые получаются при умножении интервалов в зависимости от знаков x_i и α_i :

$$1) x_i \geq 0, \alpha_i \geq 0 : \frac{\varepsilon}{n} \geq x_i \varepsilon_{\alpha_i} + \alpha_i \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i};$$

$$2) x_i < 0, \alpha_i \geq 0 : \frac{\varepsilon}{n} \geq -x_i \varepsilon_{\alpha_i} + \alpha_i \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i};$$

$$3) x_i \geq 0, \alpha_i < 0 : \frac{\varepsilon}{n} \geq x_i \varepsilon_{\alpha_i} - \alpha_i \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i};$$

$$4) x_i < 0, \alpha_i < 0 : \frac{\varepsilon}{n} \geq -x_i \varepsilon_{\alpha_i} - \alpha_i \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i}.$$

Таким образом, общая формула для x_i и α_i любого знака:

$$\frac{\varepsilon}{n} \geq |x_i| \varepsilon_{\alpha_i} + |\alpha_i| \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i}.$$

Отсюда можем получить формулу для вычисления допустимой погрешности i -го входного сигнала:

$$\varepsilon_{I_i} \leq \frac{\frac{\varepsilon}{n} - |x_i| \cdot \varepsilon_{\alpha_i}}{|\alpha_i| + \varepsilon_{\alpha_i}}.$$

В этой формуле две неизвестных величины: ε_{I_i} и ε_{α_i} . Для того, чтобы вычислить ε_{I_i} , необходимо присвоить ε_{α_i} какое-либо значение.

Представим ε_{α_i} в виде $\varepsilon_{\alpha_i} = k \frac{\varepsilon}{n}$, где k – некоторый коэффициент. Тогда предыдущая формула принимает вид:

$$\varepsilon_{I_i} \leq \frac{\frac{\varepsilon}{n} - |x_i| \cdot k \frac{\varepsilon}{n}}{|\alpha_i| + k \frac{\varepsilon}{n}}.$$

Вынося в числителе множитель $\frac{\varepsilon}{n}$, получаем $\frac{\varepsilon}{n} \cdot (1 - k|x_i|)$. Для нас имеет смысл только положительное значение ε_{I_i} . Поэтому нам нужно,

чтобы выполнялось неравенство $1 - k|x_i| > 0$, то есть $0 < k < \frac{1}{|x_i|}$. Выбирая k с учетом полученного неравенства, находим значение ε_{α_i} и затем, используя ε_{α_i} , можем вычислить ε_{I_i} . Получили формулу для пропорционального распределения погрешности по входам сумматора.

Перейдем теперь к другому способу вычисления ε_{α_i} и ε_{I_i} . Рассмотрим равномерное распределение погрешности по входам сумматора. Пусть $A = \sum_{i=1}^n \alpha_i x_i$ – точный выходной сигнал сумматора, ε – допустимая погрешность его выходного сигнала. Выходной сигнал сумматора с погрешностями должен попадать в интервал $[A - \varepsilon; A + \varepsilon]$. С другой стороны, погрешность выходного сигнала сумматора складывается из погрешностей его входных сигналов и погрешностей весов синапсов: $\sum_{i=1}^n [x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}]$. Нам требуется, чтобы

$$\sum_{i=1}^n [x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}] \leq [A - \varepsilon; A + \varepsilon].$$

Как было показано выше, при перемножении интервалов $[x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] \cdot [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}]$ получаются интервалы, несимметричные относительно 0. Максимальные по модулю концы интервалов погрешности вычисляются по следующей формуле: $|x_i| \varepsilon_{\alpha_i} + |\alpha_i| \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i}$. Предполагаем, что ε_{α_i} для каждого конкретного сумматора равны между собой и ε_{I_i} равны между собой. Складывая максимальные по модулю концы перемноженных интервалов, получаем следующую оценку:

$$\varepsilon_{\alpha_i} \sum_{i=1}^n |x_i| + \varepsilon_{I_i} \sum_{i=1}^n |\alpha_i| + n \varepsilon_{\alpha_i} \varepsilon_{I_i} \leq \varepsilon.$$

Отсюда получается оценка для допустимых погрешностей входных сигналов:

$$\varepsilon_{I_i} \leq \frac{\varepsilon - \varepsilon_{\alpha_i} \sum_{i=1}^n |x_i|}{\sum_{i=1}^n |\alpha_i| + n \varepsilon_{\alpha_i}}.$$

Из этой формулы нам требуется найти значения ε_{α_i} и ε_{I_i} . Чтобы иметь возможность вычислить ε_{I_i} , представим погрешность ε_{α_i} как

$\varepsilon_{\alpha_i} = k\varepsilon$, где k – некоторый коэффициент. Тогда предыдущая формула принимает вид:

$$\varepsilon_{I_i} \leq \frac{\varepsilon - k\varepsilon \sum_{i=1}^n |x_i|}{\sum_{i=1}^n |\alpha_i| + nk\varepsilon}.$$

Преобразуем числитель правой части неравенства, вынося множитель ε : $\varepsilon \cdot (1 - k \sum_{i=1}^n |x_i|)$. Так как ε_{I_i} определяет концы интервала $[-\varepsilon_{I_i}; \varepsilon_{I_i}]$, в котором может изменяться погрешность, то для нас имеет смысл только положительное значение ε_{I_i} . Поэтому нам нужно, чтобы

выполнялось неравенство $1 - k \sum_{i=1}^n |x_i| > 0$, то есть $0 < k < \frac{1}{\sum_{i=1}^n |x_i|}$. Выбирая

k с учетом полученного неравенства, находим значение ε_{α_i} и затем, используя ε_{α_i} , вычисляем ε_{I_i} . Получили формулы для равномерного распределения погрешности выходного сигнала сумматора по его входам.

Мы вычислили погрешности для весов синапсов и сигналов одного слоя. Погрешности ε_{I_i} являются допустимыми погрешностями входных сигналов сумматора и, одновременно, погрешностями выходных сигналов нелинейных преобразователей предыдущего слоя. Погрешности предыдущего слоя вычисляются так же, как было описано выше. Сначала вычисляются погрешности входного сигнала нелинейного преобразователя, затем погрешности весов синапсов и погрешности входных сигналов сумматора. После этого переходим к следующему слою и так до тех пор, пока не дойдем до первого слоя сети.

Так как мы предположили, что входные сигналы сети не имеют погрешностей, то формула для вычисления весов синапсов первого слоя сети отличается от формулы, по которой вычисляются все остальные погрешности весов синапсов. В этом случае $\varepsilon_{I_i} = 0$. Для пропорционального распределения допустимых погрешностей весов синапсов получаем следующую формулу вычисления погрешностей весов синапсов: $\varepsilon_{\alpha_i} \leq \frac{\varepsilon}{n|x_i|}$, где ε – допустимая погрешность выходного сигнала

сумматора первого слоя. Для равномерного распределения допустимых погрешностей формула имеет вид: $\varepsilon_{\alpha_i} \leq \frac{\varepsilon}{\sum_{i=1}^n |x_i|}$.

Если считать, что входные сигналы сети имеют некоторые погрешности, то можно пользоваться формулами для вычисления допустимых погрешностей внутренних слоев сети. Эти формулы позволяют вычислить допустимые погрешности и входных сигналов сети.

Так как для величины ε_{α_i} на каждом шаге возможен выбор из некоторого интервала значений, то, вообще говоря, для каждого примера из обучающей выборки можно вычислять различные варианты допустимых погрешностей весов синапсов и сигналов. Если на каком-то слое допустимые погрешности нас не устраивают, мы можем вернуться к предыдущему слою, выбрать новые значения ε_{α_i} , удовлетворяющие условию, и пересчитать допустимые погрешности с учетом новых значений.

6. Среднеквадратические оценки погрешностей весов синапсов

Перейдем теперь к оценке среднеквадратических отклонений погрешностей весов синапсов. Предполагаем, что погрешности весов синапсов являются независимыми случайными величинами. При этом можно налагать менее жесткие ограничения при вычислении погрешностей весов синапсов и сигналов сети.

Нам задана δ – допустимая погрешность выходного сигнала сети. Будем рассматривать ее как величину среднеквадратического отклонения погрешности выходного сигнала сети σ_{out} .

Выходной сигнал сети одновременно является выходным сигналом нелинейного преобразователя. Поэтому, зная величину среднеквадратического отклонения погрешности выходного сигнала σ_{out} , мы можем вычислить среднеквадратическое отклонение погрешности входного сигнала нелинейного преобразователя методом обратного распространения точности. Среднеквадратическое отклонение погрешности входного сигнала вычисляется по формуле: $\sigma = \frac{\sigma_{out}}{|\varphi'(A)|}$, где

φ – функция активации нелинейного преобразователя, A – его точный входной сигнал.

Среднеквадратическое отклонение погрешности входного сигнала нелинейного преобразователя σ , в свою очередь, является одновременно среднеквадратическим отклонением погрешности выходного сигнала сумматора. Дисперсия выходного сумматора равна σ^2 .

Зная дисперсию выходного сигнала сумматора, мы можем вычислять среднеквадратические отклонения погрешностей весов синапсов и входных сигналов сумматора.

Как и для интервальных оценок, получим формулы для пропорционального и равномерного распределения среднеквадратических отклонений погрешностей по входам сумматора.

Предварительно заметим, что математические ожидания случайных величин ε_{I_i} и ε_{α_i} равны нулю. Случайные величины ε_{I_i} и ε_{α_i} являются независимыми, поэтому

$$\begin{aligned} D(\varepsilon_{I_i} \cdot \varepsilon_{\alpha_i}) &= D\varepsilon_{I_i} \cdot D\varepsilon_{\alpha_i} + (M\varepsilon_{I_i})^2 \cdot D\varepsilon_{\alpha_i} + (M\varepsilon_{\alpha_i})^2 \cdot D\varepsilon_{I_i} = \\ &= D\varepsilon_{I_i} \cdot D\varepsilon_{\alpha_i}. \end{aligned}$$

Выясним сначала, как вычисляются среднеквадратические отклонения погрешностей для пропорционального распределения.

Как было показано для интервальной оценки погрешностей, при пропорциональном распределении получили формулу $\frac{\varepsilon}{n} \geq \left| x_i \right| \varepsilon_{\alpha_i} + \left| \alpha_i \right| \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i}$. Рассмотрим дисперсию обеих частей этого неравенства:

$$\frac{\sigma^2}{n} = D\left(\left| x_i \right| \varepsilon_{\alpha_i} + \left| \alpha_i \right| \varepsilon_{I_i} + \varepsilon_{\alpha_i} \varepsilon_{I_i} \right) = x_i^2 \cdot \sigma_{\alpha_i}^2 + \alpha_i^2 \sigma_{I_i}^2 + \sigma_{\alpha_i}^2 \cdot \sigma_{I_i}^2.$$

$$\text{Отсюда получаем } \sigma_{I_i}^2 = \sqrt{\frac{\sigma^2 / n - x_i^2 \cdot \sigma_{\alpha_i}^2}{\alpha_i^2 + \sigma_{\alpha_i}^2}}.$$

В этой формуле две неизвестных величины: σ_{I_i} и σ_{α_i} . Чтобы вычислить σ_{I_i} , требуется присвоить некоторое значение величине σ_{α_i} .

Представим $\sigma_{\alpha_i}^2 = k\sigma^2$. Тогда числитель подкоренного выражения можно представить в виде $\sigma^2 \cdot (1/n - k \cdot x_i^2)$. Нам требуется, чтобы числитель был положительной величиной. Поэтому следует выбрать $0 < k < 1/(nx_i^2)$. Выбирая такое k , мы определяем $\sigma_{\alpha_i}^2$ и можем вычислить σ_{I_i} . Мы получили формулу для пропорционального распределения среднеквадратических отклонений погрешностей.

Дисперсия выходного сигнала сумматора равняется σ^2 . Как было показано выше, погрешность выходного сигнала сумматора складывается из погрешностей сигналов и весов синапсов: $\sum_{i=1}^n [x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}]$. Рассмотрим дисперсию этой суммы:

$$\begin{aligned} D\left(\sum_{i=1}^n [x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}]\right) &= \\ &= \sum_{i=1}^n D[x_i - \varepsilon_{I_i}; x_i + \varepsilon_{I_i}] [\alpha_i - \varepsilon_{\alpha_i}; \alpha_i + \varepsilon_{\alpha_i}] = \\ &= \sum_{i=1}^n D(x_i \cdot \alpha_i) + \sum_{i=1}^n D(x_i \cdot \varepsilon_{\alpha_i}) + \sum_{i=1}^n D(\alpha_i \cdot \varepsilon_{I_i}) + \sum_{i=1}^n D(\varepsilon_{\alpha_i} \cdot \varepsilon_{I_i}) = \\ &= \sum_{i=1}^n x_i^2 \cdot D\varepsilon_{\alpha_i} + \sum_{i=1}^n \alpha_i^2 D\varepsilon_{I_i} + \sum_{i=1}^n D\varepsilon_{\alpha_i} \cdot D\varepsilon_{I_i} = \\ &= \sum_{i=1}^n x_i^2 \cdot \sigma_{\alpha_i}^2 + \sum_{i=1}^n \alpha_i^2 \sigma_{I_i}^2 + \sum_{i=1}^n \sigma_{\alpha_i}^2 \cdot \sigma_{I_i}^2 = \sigma^2. \end{aligned}$$

Предполагаем, что дисперсии погрешностей входных сигналов сумматора $\sigma_{I_i}^2$ равны между собой и дисперсии погрешностей его весов синапсов $\sigma_{\alpha_i}^2$ равны между собой.

Вычислим среднеквадратическое отклонение погрешностей входных сигналов сумматора:

$$\sigma_{I_i} = \sqrt{\frac{\sigma^2 - \sigma_{\alpha_i}^2 \sum_{i=1}^n x_i^2}{\sum_{i=1}^n \alpha_i^2 + n \cdot \sigma_{\alpha_i}^2}}$$

В этой формуле две неизвестных величины: σ_{I_i} и σ_{α_i} . Чтобы вычислить σ_{I_i} , требуется задать некоторое значение величине σ_{α_i} так, чтобы числитель был положительной величиной. То есть следует выбрать

$0 < k < 1 / \sum_{i=1}^n x_i^2$. Выбирая такое k , мы определяем $\sigma_{a_i}^2$ и можем вычислить σ_{I_i} . Мы получили формулы для равномерного распределения среднеквадратических отклонений погрешностей.

Таким образом, мы можем вычислить среднеквадратические отклонения погрешностей весов синапсов и входных сигналов сумматора. Вычислив эти отклонения для одного слоя, можно переходить к предыдущему слою и т.д.

Как и в случае интервальных оценок, выбор σ_{a_i} на каждом шаге достаточно произволен. Если вычисленные среднеквадратические отклонения погрешностей для данного слоя по каким-то причинам нас не устраивают, можно вернуться на предыдущий слой, выбрать другое k , удовлетворяющее накладываемому на него условию, и вычислить новые величины для погрешностей.

Вообще говоря, вычисленные допустимые погрешности весов синапсов могут достаточно сильно отличаться как в разных слоях, так и внутри одного слоя. Если требуется, чтобы погрешности весов синапсов имели достаточно близкие значения, можно попытаться выровнять их за счет итерационного вычисления погрешностей весов синапсов. То есть мы уменьшаем допустимые погрешности весов синапсов последних слоев сети. За счет этого должны увеличиться допустимые погрешности первых слоев сети. Если при этом допустимые погрешности первых слоев сети получились больше, чем допустимые погрешности последних слоев, то увеличиваем допустимые погрешности последних слоев сети. Итерации повторяются до тех пор, пока полученные результаты нас не устроят.

7. Заключение

В работе получены формулы для вычисления допустимых погрешностей весов синапсов. Эти формулы полезны при упрощении нейронных сетей (т.е. при замене вещественных весов синапсов весами синапсов с некоторым конкретным набором значений), а также при создании технической реализации на основе уже обученного нейроимитатора.

Работа выполнена при поддержке ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.) и Красноярского краевого фонда науки (грант 7F0113).

ЛИТЕРАТУРА

1. Горбань А.Н. Обучение нейронных сетей. –М.: СП “ПараГраф”, 1990. – 159 с.
2. Kimura T., Shima T. Synapse weight accuracy of analog neuro chip // Proceedings of International Joint Conference on Neural Networks. – Japan, Nagoya, October 25-29, 1993. – Vol.1. – P. 891-894.
3. Anguita D., Ridella S., Rovetta S. Limiting the effects of weight errors in feed forward networks using interval arithmetic // Proceedings of International Conference on Neural Networks (ICNN’96). – USA, Washington, June 3-6, 1996. – Vol.1. – P. 414-417.
4. Edwards P., Murray A. Modelling weight- and input-noise in MLP learning // Proceedings Of International Conference on Neural Networks (ICNN’96). – USA, Washington, June 3-6, 1996. – Vol.1. – P. 78-83.
5. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.
6. Senashova Masha Yu., Gorban Alexander N., and Wunsch Donald, “Back-Propagation of Accuracy”// Proc. IEEE/INNS International Coonference of Neural Networks, Houston, IEEE, 1997, pp.1998-2001
7. Горбань А.Н., Сенашова М.Ю. Погрешности в нейронных сетях// Вычислительный центр СО РАН в г.Красноярске. Красноярск, 1997. 38 с., библиогр. 8 назв. (Рукопись деп. в ВИНТИ, 25.07.97, №2509-В97)
8. Сенашова М.Ю. Глава 6. Погрешности в нейронных сетях // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.

НЕЙРОСЕТЕВЫЕ МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ В ЗАДАЧАХ ПРОГНОЗА КЛИМАТИЧЕСКИХ ХАРАКТЕРИСТИК И ЛЕСОРАСТИТЕЛЬНЫХ СВОЙСТВ ЛАНДШАФТНЫХ ЗОН

В.Г. Царегородцев, Н.А. Погребная

Институт вычислительного моделирования СО РАН,

Институт леса СО РАН

660036, Красноярск-36, ИВМ СО РАН,

E-mail: tsar@cc.krascience.rssi.ru

В настоящей работе рассматриваются проблемы, возникающие при обработке таблиц экспериментальных данных. Описываются нейросетевые методы обработки информации для задач классификации и построения регрессионных зависимостей. Работа может служить первым введением в нейроинформатику для специалистов-прикладников. Подробно описан опыт применения нейросетевых технологий для решения задач прогноза, классификации и глобального моделирования изменений климата. С помощью нейросетевой транспонированной линейной регрессии, нейросетевой нелинейной регрессии и нейросетевых классификаторов решены задачи прогнозирования неизвестных значений климатических параметров, классификации ландшафтных зон и глобального моделирования последствий изменения климата на основе таблицы данных, собранных с метеостанций Сибири и Дальнего Востока.

Введение

В настоящее время в связи с широким распространением вычислительной техники и повышением ее мощности актуальным становится вопрос эффективного ее применения для решения различных задач моделирования, прогнозирования, классификации и идентификации в экологии, климатологии и метеорологии. Особенность данных проблемных областей заключается в малом числе теоретически обоснованных и хорошо согласующихся с реальными данными вычислительных моделей. Поэтому прикладные задачи часто решаются на основе моделей, построенных по таблицам экспериментальных данных.

При этом проблему представляет как сложность учета всех факторов, влияющих на ситуацию в конкретных территориях, так и сложность сбора территориально распределенной информации. В связи с этим часто приходится обрабатывать неполную информацию при наличии дублирующих друг друга либо малоинформативных признаков.

В настоящей работе рассматриваются проблемы, возникающие при обработке таблиц экспериментальных данных. Описываются нейросетевые методы обработки информации для задач классификации и построения регрессионных зависимостей. Работа может служить первым введением в нейроинформатику для специалистов-прикладников.

Подробно описан опыт применения нейросетевых технологий для решения задач прогноза, классификации и глобального моделирования изменений климата. С помощью нейросетевой транспонированной линейной регрессии, нейросетевой нелинейной регрессии и нейросетевых классификаторов решены задачи прогнозирования неизвестных значений климатических параметров, классификации ландшафтных зон и глобального моделирования последствий изменения климата на основе таблицы данных, собранных с метеостанций Сибири и Дальнего Востока.

1. Проблемы обработки таблиц экспериментальных данных

Имеется таблица экспериментальных данных, где каждому объекту (строке таблицы) соответствует набор значений (столбцов) его независимых и зависимых признаков. Под независимыми (базовыми) признаками понимаются те признаки, на основе которых необходимо предсказать значения зависимых – искомых или целевых признаков, которые могут быть не известны. Все задачи классификации и прогнозирования для такой таблицы можно свести к четырем классическим постановкам [14]:

- 1) Распознавание образов (предсказание для объекта значения некоторого его целевого признака, выраженного в шкале наименований).
- 2) Предсказание значения числового (порядкового или количественного) признака для объекта.
- 3) Динамическое прогнозирование значения числового признака объекта, использующее временные измерения значений этого же признака (анализ временных рядов).

4) Автоматическая группировка объектов.

Каждая из перечисленных постановок сводится к единой задаче заполнения пропусков в таблице данных. При автоматической группировке объектов в таблицу добавляется новый столбец, содержащий информацию о разбиении всего множества объектов на группы похожих. Для иных постановок прогнозируются неизвестные значения признаков у тех объектов, где имеется пропущенная информация. Для этого требуется нахождение зависимостей в таблице экспериментальных данных [2,3,10,14]. При этом появляются следующие специфические особенности:

Таблица данных априорно является неполной, поскольку невозможно в общем случае описать все независимые и зависимые признаки, существенные для моделирования объекта или процесса. Это связано и с нашим ограниченным представлением о моделируемом объекте, и с ограничениями на возможность проведения тех или иных измерений.

Задачи приходится решать при высокой априорной неопределенности, когда практически ничего неизвестно о виде функций распределения вероятностей в пространстве признаков. Всякое “сильное” предположение (о нормальности или унимодальности распределения, некоррелированности признаков и т.д.) ставит вопрос об адекватности предлагаемого вида действительному.

При изучении сложных объектов возникают большие трудности при задании исходной системы признаков для их описания. Поэтому в признаковом пространстве может быть много “дублирующих” и “шумящих” признаков. В результате проблема выбора наиболее информативной подсистемы признаков приобретает важное значение, поскольку уменьшение числа признаков часто улучшает качество решения (и сокращает экономические и временные затраты на измерения или сбор информации). Желательно иметь возможность определения значимости каждого признака для принятия решения и выделения минимально необходимого набора базовых признаков для прогнозирования целевого признака с заданной точностью.

Для описания объектов используются признаки, измеренные в разных шкалах и, возможно, разнотипные – количественные (выраженные в шкалах интервалов, отношений и абсолютных значений), порядковые

(шкалы порядка, частичного порядка, рангов, баллов) или номинальные (шкалы наименований).

В связи со сложностью проведения измерений, отказом датчиков, историческими причинами в таблице могут отсутствовать некоторые значения исходных и целевых признаков у отдельных объектов. В данных всегда присутствуют ошибки разной природы, шум, а также имеются противоречия отдельных измерений друг другу. За исключением простых случаев, искажения в данных не могут быть устранены полностью.

Классификация объектов проблемной области, вводимая человеком, может не совсем точно отражать существующую в проблемной области естественную кластерную структуру объектов, что создает дополнительные трудности.

Эти особенности не связаны только с перечисленными выше проблемными областями, но могут встретиться везде, где используется построение зависимостей по таблице экспериментальных данных. Также не зависит от проблемной области возможность сведения любой задачи прогнозирования и классификации к задаче "правдоподобного" заполнения пропусков в таблице.

Для обработки эмпирических данных традиционно используются классические методы математической статистики [11,12,17]. Можно получить для отсутствующих значений их условные математические ожидания (условия – значения других величин, описывающих конкретную ситуацию) и характеристики разброса – доверительные интервалы. Для решения задачи классификации методы математической статистики строят разделяющие поверхности между классами в признаковом пространстве. Однако достоверное статистическое оценивание требует либо очень большого объема известных данных, либо очень сильных предположений о виде функций распределения, и работает обычно только при нормальных или близких к нормальным функциях распределения. Поэтому при вычислении условного математического ожидания требуется проверять гипотезу о распределении эмпирических данных по нормальному закону или использовать аппарат непараметрической статистики, восстанавливающей оценки плотностей распределения вероятностей.

Поэтому часто для решения задач классификации и прогнозирования используются различные эмпирические (статистически не достоверные) методы обработки информации. Естественным

требованием является желание иметь единую парадигму построения различных эмпирических моделей, решающих задачу предсказания и классификации, не привязанную к конкретной проблемной области и предлагающую процедуры решения типовых проблем, встающих при обработке таблиц экспериментальных данных. В связи с этим можно сформировать список требований, которым должен удовлетворять эмпирический метод обработки информации:

1. Работа при наличии пропусков в таблице.
2. Работа в случае, если число измеренных признаков превышает число объектов, и число объектов достаточно мало.
3. Наложение достаточно слабых ограничения на вид распределений.
4. Обеспечение возможности обработки разнотипных экспериментальных данных (без сведения всех признаков к одной шкале) и инвариантность к допустимым преобразованиям шкал признаков.
5. Достаточно высокая вычислительная эффективность.
6. Простота получения результата и отсутствие привязки к конкретной проблемной области.

Эти требования является очень важными. Обычно при использовании эмпирических методов предполагается, что пользователя удовлетворит просто получение прогноза. Далее на основе своих профессиональных знаний он сделает вывод о применимости полученного результата, сопоставив машинный прогноз со своими экспертными оценками. Пользователь может протестировать на имеющихся у него данных несколько эмпирических методов, выбрать наилучший метод по точности достигаемого прогноза и в дальнейшем использовать именно его. При этом, поскольку пользователь обычно смутно представляет, как следует задавать параметры метода для достижения наилучшей точности прогноза, и обычно не может заложить в метод сформировавшуюся у него эмпирическую модель проблемной области, то возникают проблемы недостаточной эффективности использования методов и неадекватности построенных моделей имеющимся экспертным соображениям.

Можно ввести еще одно требование, основанное на желании применить разработанный метод прогноза или классификации либо построенную эмпирическую модель для решения этих же задач, например, в другом регионе либо по истечению некоторого времени без

необходимости их коррекции. Хотя это всегда практически недостижимо, требуется хотя бы обеспечить возможность адаптации модели к новому региону путем простого переобучения на новых данных.

2. Искусственные нейронные сети

Для решения задач классификации и прогноза используются различные математические методы: математической статистики, кластерного, регрессионного, факторного анализа и т.д. По сравнению с этими методами подход, основанный на применении искусственных нейронных сетей, обладает рядом преимуществ и удовлетворяет введенным ранее требованиям. Во-первых, создается единая для всех задач вычислительная парадигма. Используя нейронные сети со сравнительно небольшим числом нейронов, можно решать достаточно сложные задачи классификации и прогноза. Нейронные сети представляют собой обучающиеся модели, что позволяет просто "доучивать" их при поступлении новых данных либо "переучивать" для обработки данных из другого региона. Возможно использование малых обучающих выборок, не обеспечивающих получение статистически достоверных результатов классическими методами. В нейронных сетях можно использовать любое количество независимых и зависимых признаков, число примеров для разных классов (при решении задачи классификации) может быть различным. Также возможно решение одной сетью одновременно нескольких задач классификации или прогноза. В нейронной сети есть процедура подсчета значимости независимых признаков и возможность минимизации их числа.

После такого многообещающего анонса перейдем непосредственно к описанию того, что такое искусственные нейронные сети и как они решают задачи.

Искусственные нейронные сети, а в более общем случае – и вся нейроинформатика, появились при попытке моделирования мозга не на основе нейробиологических идей, а на основе кибернетических. Так, в основе всего нейросетевого подхода лежит идея построения вычислительного устройства из большого числа параллельно работающих простых процессирующих элементов – нейронов. Эти нейроны функционируют независимо друг от друга и связаны между собой

однонаправленными каналами передачи информации – синапсами (как и при именовании нейрона, здесь взяты нейробиологические термин и абстракция). Получаемая при этом искусственная нейронная сеть обладает по аналогии с ее биологическими прототипами следующими преимуществами:

Высокоэффективная параллельно - последовательная обработка информации и, вообще, максимальный потенциальный параллелизм и наиболее эффективное использование любой параллельной вычислительной архитектуры по сравнению с другими вычислительными технологиями.

Высокая надежность и устойчивость к отказам отдельных элементов, составляющих нейронную сеть.

Место программирования занимает процесс обучения (или настройки) нейронной сети. Под обучением понимается процесс адаптации нейронной сети для достижения минимума некоторого оценивающего функционала – например, качества решения сетью поставленной задачи.

В нейронной сети выделена группа входов и группа выходов. По входным рецепторам нейронная сеть принимает информацию и затем, пропуская эту информацию через себя и преобразуя ее с помощью процессорных элементов, генерирует выходные сигналы.

2.1. Элементы нейронных сетей

В нейроинформатике выработана специальная "схемотехника", в которой элементарные устройства – сумматоры, синапсы, нейроны и т.п. объединяются в сети, предназначенные для решения задач [6,9].

Самый заслуженный и, вероятно, наиболее важный элемент нейросистем – это *адаптивный сумматор*. Адаптивный сумматор вычисляет скалярное произведение вектора входного сигнала x на вектор параметров α . На схемах будем обозначать его так, как показано на рис.1. Адаптивным называем его из-за наличия вектора настраиваемых параметров α . Для многих задач полезно иметь линейную неоднородную функцию выходных сигналов. Ее вычисление также можно представить с помощью адаптивного сумматора, имеющего $n+1$ вход и получающего на 0-й вход постоянный единичный сигнал (рис.2).

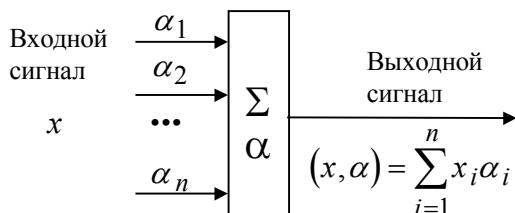


Рис. 1. Адаптивный сумматор

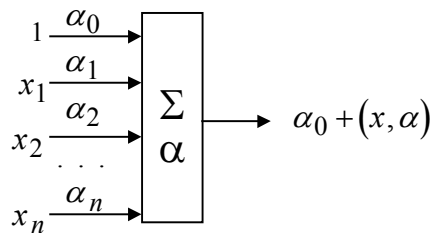


Рис. 2. Неоднородный сумматор

Нелинейный преобразователь сигнала изображен на рис.3. Он получает скалярный входной сигнал x и переводит его в $\varphi(x)$. *Точка ветвления* служит для рассылки одного сигнала по нескольким адресам (рис.4). Она получает скалярный входной сигнал x и передает его всем своим выходам. *Стандартный формальный нейрон* составлен из входного сумматора, нелинейного преобразователя и точки ветвления на выходе (рис.5). *Линейная связь - синапс* – отдельно от сумматоров не встречается, однако для некоторых рассуждений бывает удобно выделить этот элемент (рис.6). Он умножает входной сигнал x на "вес синапса" α .

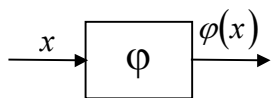


Рис.3. Нелинейный преобразователь сигнала

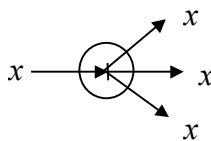


Рис.4. Точка ветвления

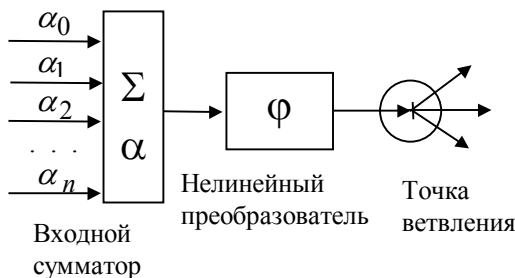


Рис. 5. Формальный нейрон

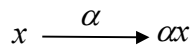


Рис. 6. Линейная связь (синапс)

2.2. Архитектуры нейронных сетей

В "зоопарке" искусственных нейронных сетей можно выделить две базовые архитектуры – *слоистые* и *полносвязные сети*.

Слоистые сети: нейроны расположены в несколько слоев (рис.7). Нейроны первого слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до k -го слоя, который выдает выходные сигналы. Если

не оговорено противное, то каждый выходной сигнал i -го слоя подается на вход всех нейронов $i+1$ -го. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: все нейроны первого слоя получают каждый входной сигнал.

Полносвязные сети: каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам.

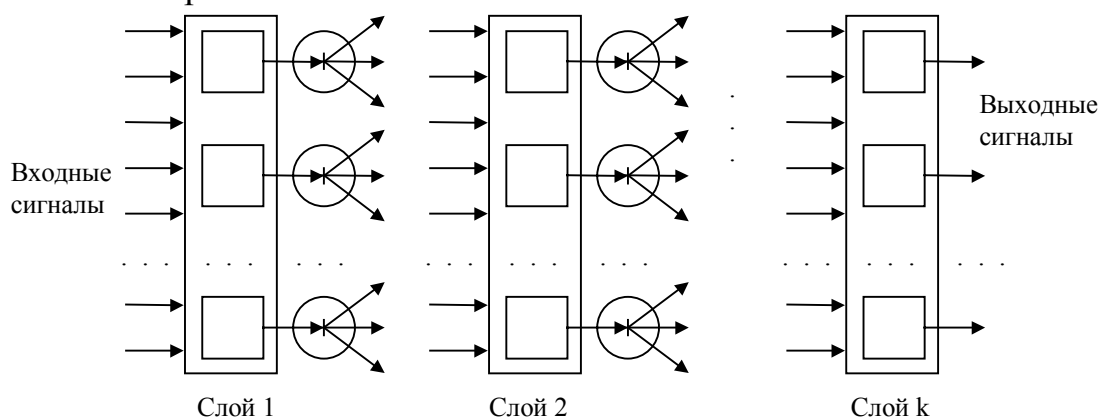


Рис. 7. Слоистая нейронная сеть

Элементы слоистых и полносвязных сетей могут выбираться по-разному. Существует, впрочем, стандартный выбор – нейрон с неоднородным адаптивным линейным сумматором на входе (рис.5).

2.3. Решение задач нейронными сетями

Можно выделить два класса задач, решаемых обучаемыми нейронными сетями. Это задачи предсказания и классификации.

Задачи предсказания или прогнозирования являются, по-существу, задачами построения регрессионной зависимости выходных данных от входных. Нейронные сети могут эффективно строить сильно нелинейные регрессионные зависимости. При решении задач классификации нейронная сеть строит разделяющую поверхность в признаковом пространстве, а решение о принадлежности ситуации тому или иному классу принимается самостоятельным, не зависящим от сети устройством – интерпретатором ответа сети [7,9]. Наиболее простой интерпретатор

возникает в задаче бинарной классификации (классификации на два класса). В этом случае достаточно одного выходного сигнала сети, а интерпретатор относит, например, ситуацию к первому классу, если выходной сигнал меньше нуля, и ко второму, если он больше или равен нулю.

Классификация на несколько классов требует усложнения интерпретатора. Широко используется интерпретатор "победитель забирает все", где число выходных сигналов сети равно числу классов, а номер класса будет соответствовать номеру нейрона, выдавшего максимальный выходной сигнал.

Одна нейронная сеть может одновременно предсказывать несколько чисел, либо одновременно решать задачи и прогнозирования, и классификации.

2.4. Подача входных сигналов и снятие выходных сигналов сети

На веса синапсов сети обычно наложены требования принадлежности некоторому диапазону значений. Наиболее часто используемые нелинейные функции нейронов также обычно выдают значения из некоторого диапазона. Это приводит к тому, что обычно нельзя подавать сети входные сигналы в их истинном диапазоне величин и получать от сети выходные сигналы в требуемом диапазоне.

Поэтому перед подачей сети входных сигналов их необходимо нормировать, например, в диапазон значений $[-1,1]$ или $[0,1]$, либо делать так, чтобы входные сигналы не слишком сильно выходили за пределы этих отрезков. Наиболее просто нормировку можно выполнить так. Каждая компонента входного вектора данных x_i заменяется величиной

$$x_i = \frac{x_i - (\max x_i + \min x_i) / 2}{(\max x_i - \min x_i) / 2},$$

где $\max x_i$ и $\min x_i$ – соответственно максимальное и минимальное значения для данной компоненты, вычисленные по всей обучающей выборке. По этой же формуле пересчитываются и компоненты векторов ответов.

Можно нормировать и по-другому, например, пересчитывать выборку так, чтобы разброс данным был единичным.

Здесь имеется одна сложность. Любое изменение обучающей выборки должно соответственно менять и правило нормирования данных. Поэтому поступают обычно так: в качестве *min* и *max*-величин берут данные из существующего на настоящий момент задачника и в дальнейшем нормировку не меняют, а если предполагается, что в дальнейшем поступят сильно отличающиеся данные, то *min* и *max*-величины задаются пользователем по его оценкам. Иными словами, эти величины должны вводиться в момент создания сети и в дальнейшем не зависеть от обучающей выборки.

Выходные сигналы сети должны нормироваться в диапазон истинных значений по обращенным формулам.

Для сетей-классификаторов нормировка выходных сигналов не нужна, поскольку пользователь получает не собственно выходные сигналы сети, а результат отнесения ситуации интерпретатором ответа к тому или иному классу. Здесь каждый выходной вектор задачника преобразуется так, чтобы правильно обрабатываться интерпретатором ответа. Например, при классификации на три класса и интерпретаторе "победитель забирает все" номер класса будет кодироваться вектором из трех чисел, где компонента вектора, соответствующая номеру класса, имеет значение 1, а остальные две – -1.

Другой проблемой является ситуация, когда в таблице данных имеются пробелы. Вместо таких отсутствующих компонент данных можно подавать нуль, можно исключать некомплектные вектора из обучающей выборки, можно перед обучением сети решать задачу заполнения пробелов в данных некоторыми правдоподобными значениями.

2.5. Обучение нейронных сетей

Обучение нейронной сети на некотором "задачнике" означает минимизацию функционала невязки между выходными сигналами сети и сигналами, которые требуется получить. Минимизация функционала невязки (оценки в дальнейшем) производится путем такой подстройки обучаемых параметров a_i нейронов сети, чтобы сеть на некоторый входной вектор сигналов X выдавала ответ Y' , который был бы по возможности ближе к требуемому ответу Y . Иными словами, чтобы для задачника с

числом примеров N (т.е. для N пар $\{X_i, Y_i\}, i=1..N$) достигался минимум суммарной функции оценки:

$$H = \sum_{i=1}^N \|Y_i - Y'_i\| \longrightarrow \min$$

В качестве функции нормы выступает покомпонентная сумма квадратов элементов вектора $Y - Y'$ (оценка МНК), либо более специализированная. Применение более специализированных оценок ускоряет процесс обучения сети. Так, можно использовать оценки, позволяющие задавать требования к точности решения задачи. Это позволяет прекращать процесс обучения, когда достигнута удовлетворяющая пользователя точность. Для решения задач классификации можно строить специализированные оценки, основанные на используемом при решении задачи интерпретаторе ответа. Подробно разнообразные интерпретаторы ответа и построенные по ним оценки описаны в [7,9].

Минимизация функции оценки выполняется с привлечением градиентных методов оптимизации. Преобразовав по некоторым правилам структуру сети, подавая на выход сети частные производные функции оценки по выходным сигналам и используя так называемое двойственное функционирование, мы можем получить для каждого подстроечного параметра сети и для каждого входного сигнала частные производные функции оценки по значению этого параметра или сигнала. Для вектора параметров сети вектор соответствующих частных производных будет градиентом функции оценки, поэтому возможна градиентная оптимизация функции оценки, в ходе которой нейронная сеть "обучается" давать требуемые ответы на подаваемые ей входные сигналы.

Сеть лучше обучать по суммарному градиенту (градиенту по всем примерам задачника), что ускоряет процесс обучения, и применять специализированные алгоритмы оптимизации, надстраиваемые над простейшим градиентным спуском. Для вычисления суммарного градиента необходимо просуммировать вектора градиентов, вычисляемые для каждого примера задачника (всего N векторов). Естественно, что нет необходимости одновременного хранения в памяти ЭВМ градиентов для всех примеров, процесс вычисления суммарного градиента реализуется как процесс последовательного накопления.

2.6. Вычисление градиента функции оценки по подстроечным параметрам сети

Теперь о том, как же собственно вычисляется градиент функции оценки по подстроечным параметрам сети. Вычисление градиента производится в ходе функционирования нейронной сети, двойственной к исходной. У сети двойственности сигналы передаются в обратном направлении – от выходов сети к ее входам, а элементы, из которых строится двойственная сеть, являются двойственными к исходным. Рассмотрим сначала некоторый достаточно абстрактный элемент сети. Ему передается n значений переменных x_i , на основе которых вычисляется значение функции $f(x_1, \dots, x_n)$ и это значение рассылается по m выходным связям. В двойственной сети этот элемент получает на выходе набор из m двойственных сигналов μ_i , вычисляет их сумму μ и передает на каждый из n входов произведение μ и частной производной функции $f(x_1, \dots, x_n)$ по значению подаваемого на соответствующий вход сигнала. Этот процесс представлен на рис. 8. Здесь видно, что при построении двойственной сети все точки ветвления переходят в двойственные к ним элементы – простые сумматоры.

Теперь рассмотрим процесс построения нейрона, двойственного к формальному. Для этого представим формальный нейрон (рис. 5) более подробно (рис. 9), представив входной адаптивный сумматор нейрона в виде набора элементов, вычисляющих попарные произведения (веса синапса и приходящего на синапс сигнала), и простого сумматора. По описанному выше правилу построения двойственного элемента преобразуем все элементы, составляющие формальный нейрон. Нейрон, двойственный к формальному, и правила его функционирования показаны на рис. 10. Здесь видно, что при построении двойственной сети все простые сумматоры переходят в двойственные к ним элементы – точки ветвления. Для более понятной иллюстрации этого факта и было использовано представление адаптивного сумматора с помощью простого сумматора и набора синапсов на рис. 9. Из нейронов, двойственных к формальным, и строится двойственная сеть. Разумеется, двойственные сети могут быть построены и для сетей, состоящих из более сложных по конструкции нейронов.

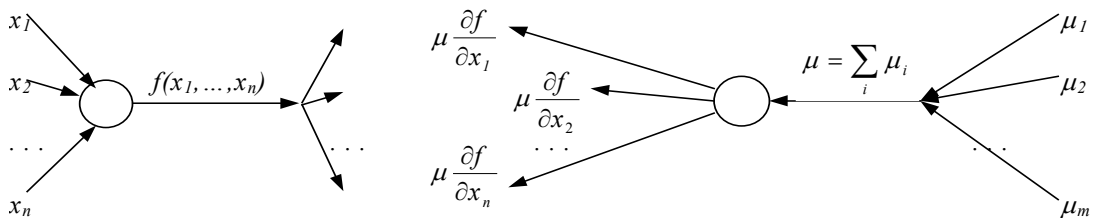


Рис.8. Прямое и двойственное функционирование абстрактного элемента сети

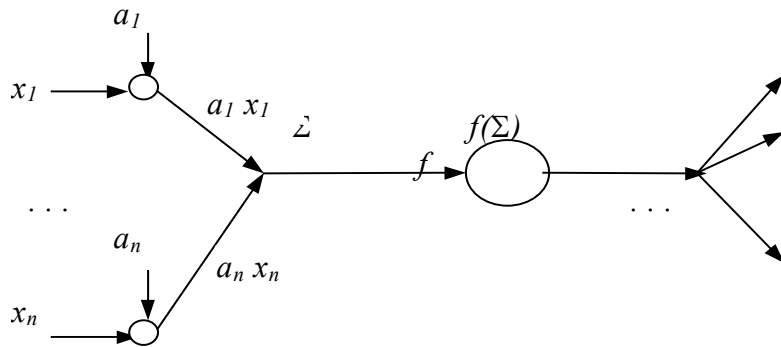


Рис. 9. Функционирование формального нейрона

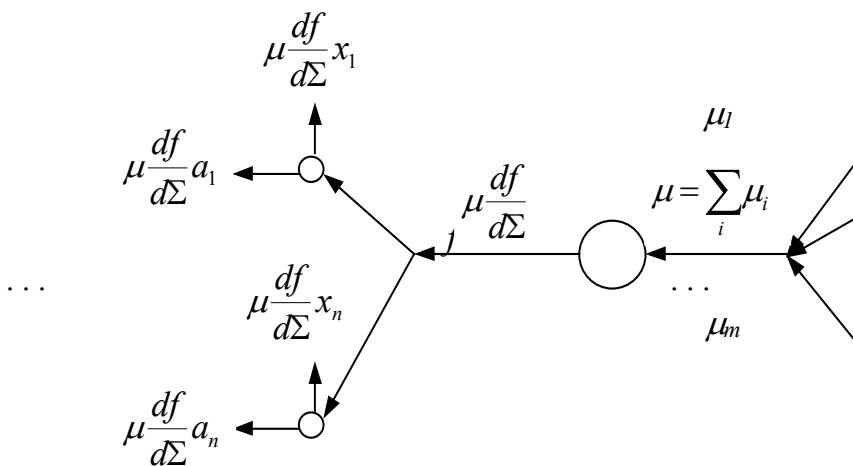


Рис. 10. Двойственное функционирование формального нейрона

Итак, построена сеть, двойственная к исходной. Теперь определимся с двойственными переменными μ , подаваемыми на выходы нейронов последнего слоя. Чтобы решить задачу вычисления с помощью двойственной сети градиента функции оценки по подстроечным параметрам исходной сети, в качестве двойственных переменных необходимо брать частные производные функции оценки H по выходным сигналам сети. Тогда для всех элементов и входных сигналов сети будут вычислены частные производные функции оценки по значениям этих параметров и сигналов.

Использование двойственной сети для вычисления градиента позволяет вычислять весь вектор градиента за время, всего в 2-3 раза большее, чем время вычисления выходных сигналов в ходе обычного функционирования исходной сети. Это происходит потому, что все действия организованы рационально, без дублирования вычислений, и градиент вычисляется в максимально параллельном режиме.

Единственным требованием, налагаемым на нейронные сети и функции оценки для того, чтобы можно было использовать двойственное функционирование для вычисления градиента, является существование непрерывных первых производных функций оценки и всех элементов сети по их входным сигналам.

Подробно принцип двойственности для обучения нейронных сетей рассмотрен в [6,9].

2.7. Факторы, влияющие на обучение нейронной сети

Рассмотрим факторы, от которых зависит успешность обучения нейронной сети правильному решению задачи. В первую очередь, сеть должна быть достаточно гибкой, чтобы научиться правильно решать все примеры обучающей выборки. Поэтому в нейронной сети должно быть достаточное количество нейронов и связей.

На основании обучающей выборки достаточно сложно определить, сколько слоев и нейронов сети необходимо. Поэтому поступают обычно так. Обучают сеть со структурой, предлагаемой программой-нейроимитатором по умолчанию, а в дальнейшем, если сеть не может обучиться, пробуют обучить сеть большего размера. На практике при решении разнообразных задач практически не встречается ситуации, когда

требуется нейросеть с более чем сотней нейронов – обычно хватает нескольких десятков нейронов и даже меньшего числа.

Однако даже увеличение размера нейронной сети не поможет, если обучающая выборка противоречива. Иными словами, в обучающей выборке присутствуют задачи с одинаковыми условиями, но разными ответами (одинаковыми входными векторами данных, но разными выходными). Таким задачам нейронная сеть обучиться не может. Здесь возникает проблема разрешения такой противоречивой ситуации. Появление таких конфликтных примеров может, допустим, означать недостаточность набора входных признаков, поскольку при расширении признакового пространства конфликтным примерам могут соответствовать разные значения добавляемого признака и критическая ситуация будет исчерпана. В любом случае пользователь должен решить эту проблему, хотя бы даже простым исключением конфликтных примеров из задачника.

После обучения нейронной сети необходимо провести ее тестирование на тестовой выборке для определения точности решения не входивших в обучающую выборку задач. Точность решения очень сильно зависит от репрезентативности обучающей выборки. Обычно при решении различных неформализованных задач в разных проблемных областях точность в 70-90% правильных ответов на тестовой выборке соответствует проценту правильных ответов при решении этих же задач специалистом-экспертом.

Может оказаться так, что нейронная сеть обучилась, но дает много ошибок на тестовой выборке. Природу этого явления нужно изучать в каждом конкретном случае. Одна возможная причина – нерепрезентативность обучающей выборки, когда обучающая выборка не охватывает всего множества ситуаций (выборка мала или просто узкоспециализирована). Иной причиной большого количества ошибок, на этот раз, только при решении задачи классификации, может быть неодинаковое число примеров разных классов. При этом при тестировании нейросеть будет достаточно хорошо распознавать примеры класса, для которого в обучающей выборке было большинство примеров, и относить к этому же классу много примеров другого класса. Поэтому желательно, чтобы в обучающей выборке было примерно одинаковое число примеров для каждого класса, или, по крайней мере, не было отличия на порядок и более. Если же неравномерность распределения примеров по классам есть

особенность проблемной области, то нужно использовать при обучении сети оценки "с весами" – модификации стандартных оценок, позволяющие уравнивать вклад примеров разных классов в суммарную функцию оценки [9].

2.8. Упрощение нейронных сетей

Сокращение множества параметров сети и входных сигналов может преследовать несколько целей:

Упрощение последующей аппаратной реализации нейронной сети;

Сокращение объема используемой памяти и повышение быстродействия программ–нейроимитаторов;

Удешевление процесса сбора данных за счет сокращения их объема; возможность решения задачи на основе меньшего объема входных данных;

Упрощение (контрастирование) нейронной сети строится как последовательный процесс исключения из сети наименее значимого элемента и дальнейшего доучивания сети. Если после шага упрощения невозможно доучивание сети до требуемой точности, то возвращаемся к сети, полученной на предыдущем шаге, и завершаем процесс упрощения.

В следующем разделе опишем процедуру вычисления показателей значимости [4,5,9].

2.9 Вычисление показателей значимости параметров и входных сигналов сети

У нейронной сети имеется набор n -мерных векторов данных x^i , $i=1,...,N$ (задачник) с координатами x_j^i , $j=1,...,n$, M -мерный вектор параметров a с координатами a_k , $k=1,...,M$, и некоторая функция оценки $H(x,a)$, оценивающая работу сети с параметрами a на векторе данных x . Требуется выделить у сети наименее значимые параметры a_k и сигналы x_j и модифицировать систему, отбрасывая эти параметры и компоненты данных.

Для каждого a_k определено некоторое значение a_k^0 и сокращение a_k означает приравнивание $a_k=a_k^0$. Аналогично для каждой компоненты

данных x_j и любого примера x^i определено x_j^{i0} и отбрасывание j -ой компоненты данных для i -го примера означает приравнивание $x_j = x_j^{i0}$.

Такая процедура допускает очень большую свободу в доопределении процедуры сокращения описания. В качестве простейшего базового варианта будем $x_j^{i0} = \frac{1}{n} \sum_{p=1}^n x_j^p$ иметь в виду $a_k^0 = 0$ и для всех i (параметры обращаются в нуль, данные заменяются средними по выборке). В случае данных, отнормированных, например, в диапазон $[-1, 1]$, в качестве x_j^{i0} , естественно, принимается нуль.

Показатели значимости вычисляются в два этапа: сначала они вычисляются для одного вектора данных (примера), а потом – по всей выборке. Для данного x^p значимости a_k и x_j оцениваются так:

$$\chi(a_k | x^p) = \left| \frac{\partial H(x^p, a)}{\partial a_k} (a_k - a_k^0) \right|;$$

$$\chi(x_j | x^p) = \left| \frac{\partial H(x^p, a)}{\partial x_j} (x_j^p - x_j^{p0}) \right|.$$

Здесь χ – просто вычисленные в линейном приближении абсолютные величины изменения H при сокращении описания, частные производные вычисляются при двойственном функционировании. Естественно, значение функции оценки должно быть ненулевым. Поэтому, если используются специализированные функции оценки, на этапе вычисления значимости необходимо возвращаться к оценке МНК.

Оценка по всему задачку производится с использованием некоторой нормы, например, с использованием нормы в виде суммы модулей

$$\chi(a_k) = \sum_p \chi(a_k | x^p); \quad \chi(x_j) = \sum_p \chi(x_j | x^p)$$

или максимума модулей.

$$\chi(a_k) = \max \chi(a_k | x^p); \quad \chi(x_j) = \max \chi(x_j | x^p).$$

Часто приходится иметь дело с системой, которая меняет свои параметры, например, в ходе обучения. Тогда к моменту принятия решения о значимости может быть накоплена информация о частных

производных H в разных точках $a=a^1, \dots, a^q$. Тогда показатели значимости для одного вектора данных вычисляются так:

$$\chi(a_k | x^p) = \frac{1}{q} \sum_{s=1}^q \left| \frac{\partial H(x^p, a^s)}{\partial a_k} \right| * |a_k - a_k^0|;$$

$$\chi(x_j | x^p) = \frac{1}{q} \sum_{s=1}^q \left| \frac{\partial H(x^p, a^s)}{\partial x_j} \right| * |x_j^p - x_j^{p0}|.$$

Усредняются абсолютные значения производных, а приращения берутся в той точке, в которой будет проводиться процедура сокращения. Далее для всей выборки показатели значимости усредняются в одной из норм.

Использование вычисленных таким образом показателей значимости часто позволяет очень сильно сократить как число входных сигналов сети (остается только минимально необходимый для правильного решения задачи набор признаков), так и число элементов сети.

Если необходимо не исключать из сети адаптивные элементы, а приводить их значения к конечному набору выделенных значений, то для каждого a_k в качестве a_k^0 принимается ближайшее к a_k выделенное значение. Такая задача возникает при бинаризации весов синапсов сети – приведении весов синапсов к величинам -1 или 1.

3. Транспонированная задача регрессии

Представляют интерес не только задачи регрессии. Рассмотрим наиболее общую постановку задачи прогнозирования неизвестных значений. Пусть имеется таблица эмпирических данных, содержащая пропуски. Требуется заполнить эти пропуски и, при добавлении в таблицу информации о новых объектах, для которых отсутствуют значения некоторых признаков, прогнозировать отсутствующие значения. Классическая задача регрессии разбивает все множество признаков объекта на группы зависимых и независимых признаков и для каждого зависимого признака строит регрессионную зависимость от независимых свойств. Однако здесь имеются две трудности. Предположение о том, что регрессионные зависимости одинаковы для всех объектов выборки, выполняется далеко не всегда. Также может потребоваться большое число

регрессионных зависимостей – по одной для любой возможной комбинации пропусков в данных.

Если хорошей точности приближения известных данных добиться не удалось (невозможно построение регрессионной зависимости, общей для всех объектов), то можно предположить, что существуют такие признаки, которые не включены в рассматриваемую таблицу данных, но которые, тем не менее, являются значимыми и несут более существенную информацию и именно в них и заключены скрытые различия, не дающие построить искомые регрессионные зависимости. Эти неучтенные и неизмеренные свойства, от которых зависят наблюдаемые параметры, называют "скрытыми параметрами", а предположение о том, что все дело в них – *гипотезой о скрытых параметрах*.

Одна из простейших форм предположения о скрытых параметрах - *гипотеза о качественной неоднородности выборки*. Она означает, что скрытые параметры принимают сравнительно небольшое конечное число значений и всю выборку можно разбить на классы, внутри которых скрытые параметры, существенные для решения интересующей нас задачи регрессии, постоянны. Каждой такой выборке будет соответствовать "хорошая" регрессионная зависимость.

Построить классификацию (без учителя), соответствующую данной гипотезе, можно только на основе предположений о форме искомой регрессионной зависимости наблюдаемых параметров от наблюдаемых же параметров внутри классов (задача мозаичной регрессии). Если предполагается линейная зависимость, то эта задача классификации решается методом динамических ядер, только место точек – центров тяжести классов – занимают линейные многообразия, каждое из которых соответствует линейному регрессионному закону своего класса [9].

Регрессионные зависимости, которые строятся с помощью нейронных сетей, также образуют вполне определенный класс и для них тоже возможна соответствующая классификация без учителя. Достаточно большая нейронная сеть может освоить любую непротиворечивую обучающую выборку, однако, как показывает опыт, если малая нейронная сеть не может обучиться, то из этого можно извлечь полезную информацию. Если не удастся построить удовлетворительную регрессионную зависимость при заданном (небольшом) числе нейронов, то из обучающей выборки исключаются наиболее сложные примеры до тех

пор, пока сеть не обучится. Так получается класс, который предположительно соответствует одному значению скрытых параметров. Далее обучение можно продолжить, сформировав из отброшенных примеров отдельный класс (скорректировав классификационную модель).

Транспонируем таблицу данных (поменяем местами слова "объект" и "признак"). Рассмотрим гипотезу о том, что значения признака одного объекта могут быть функциями значений того же признака других объектов и эти функции одни и те же для всех признаков (или классов признаков). Получаем формально те же задачи регрессии (*транспонированные задачи регрессии*). Есть, однако, два содержательных отличия транспонированных задач от исходных:

Инвариантность к смене шкал измерения – кажется маловероятным, чтобы существенные связи между признаками различных объектов зависели от шкалы измерения, поэтому необходимо, чтобы уравнения транспонированной регрессии были инвариантны относительно смены шкалы измерения любого признака x (обычно – линейного неоднородного преобразования $x' = ax + b$, однородная часть которого описывает смену единицы измерения, а свободный член – сдвиг начала отсчета);

В традиционных задачах регрессии предполагается, что объектов достаточно много (N), по сравнению с числом признаков n , иначе (при $N < n$) точные линейные соотношения возникнут просто из-за малого числа объектов, так как через N точек всегда можно провести линейное многообразие размерности $N-1$. В противовес этому "транспонированное" предположение о достаточно большом числе признаков ($n > N$) кажется нереалистичным.

Требование инвариантности к смене шкал приводит к специальным ограничениям на вид функций регрессии, а недостаточность количества признаков (в сравнении с числом объектов) для построения транспонированной регрессии вынуждает нас для каждого объекта искать небольшую группу объектов, по свойствам которых можно восстановить свойства исходного. Иными словами, задача ставится так: найти для каждого объекта наилучшую формулу, выражающую его вектор признаков через векторы признаков других объектов (которых должно быть по возможности меньше) и инвариантную относительно смены шкал измерения признаков.

Итак, требуется построить отношение, связывающее объекты с группами объектов, по которым строятся интерполяционные формулы. Предположим, что некоторый большой набор свойств – внешних, эмпирических данных об объекте (явление) является сюръекцией небольшого набора внутренних, теоретических переменных (сущности). Эта идея позволяет сделать предположение о том, что размер опорной группы объектов, по которой наилучшим образом восстанавливаются свойства данного объекта, не только не должен превосходить размер набора свойств (иначе заведомо возникнут точные линейные соотношения), но и быть малым настолько, насколько это позволяет заданная точность [6,8,9].

Если предположить, что для некоторого множества объектов зависимость между теоретическим и эмпирическим линейна, и векторы теоретических параметров объектов данного множества лежат в линейном многообразии размерности q , то размер опорной группы не будет превосходить $q+1$.

Другое условие, налагаемое на искомую формулу, требует инвариантности к смене шкал измерений:

$$f(ay^1+b, \dots, ay^q+b) = a f(y^1, \dots, y^q) + b$$

Способ реализации этого ограничения — представление f в виде [6,17]

$$f(y, \dots, y^q) = M_y + \sum_{i=1}^q \varphi_i (y^1 - M_y, \dots, y^q - M_y) (y^i - M_y),$$

где $M_y = \frac{1}{q} \sum_{i=1}^q y^i$ — барицентр опорной группы, а φ_i — однородные функции нулевой степени : для любого a : $\varphi_i(az_1, \dots, az_q) = \varphi_i(z_1, \dots, z_q)$.

Простейшие примеры таких φ :

$$1) \varphi = \text{const}; \quad 2) \varphi = \frac{a_1 z_1 + \dots + a_q z_q}{b_1 z_1 + \dots + b_q z_q}; \quad 3) \varphi = \frac{a_1 \varphi_1 + \dots + a_q \varphi_q}{b_1 \varphi_1 + \dots + b_q \varphi_q}$$

где все φ_j в 3) имеют вид 2) с различными наборами коэффициентов a_i, b_i .

Если в качестве искомой формулы рассматривать линейную комбинацию векторов опорной группы (линейную транспонированную регрессию), то требуемой инвариантности можно достичь, наложив некоторое условие на коэффициенты разложения. Таковым условием является равенство суммы коэффициентов единице:

$$\tilde{\mathbf{y}} = \sum_i \alpha_i \mathbf{y}^i, \sum_i \alpha_i = 1.$$

Рассматривались два варианта нахождения коэффициентов разложения, соответствующих некоторой опорной группе объектов. Первый:

$$\tilde{\mathbf{y}} = \mathbf{m}_y + \sum_{i=1}^q \beta_i (\mathbf{y}^i - \mathbf{m}_y), \quad \alpha_i = \beta_i + \frac{1}{q} - \frac{1}{q} \sum_{k=1}^q \beta_k \quad (1)$$

где $\tilde{\mathbf{y}}$ – восстановленный вектор свойств, \mathbf{y}^i – вектор свойств i -го объекта опорной группы, q – мощность опорной группы, $\mathbf{m}_y = \frac{1}{q} \sum_{i=1}^q \mathbf{y}^i$, – среднее значение (барицентр опорной группы).

Во втором случае в качестве \mathbf{m}_y выбирался один из векторов опорной группы.

$$\tilde{\mathbf{y}} = \mathbf{y}^t + \sum_{i=1}^q \beta_i (\mathbf{y}^i - \mathbf{y}^t), \quad \alpha_i = \beta_i, \quad \alpha_t = 1 - \sum_{\substack{k=1 \\ k \neq t}}^q \beta_k \quad (2)$$

Заметим, что легко построить нейронную сеть, вычисляющую такие формулы [9,19].

Коэффициенты разложения определяются в ходе минимизации одного из следующих критериев:

$$\text{а) } \|\mathbf{y} - \tilde{\mathbf{y}}\|^2 + \varepsilon^2 \|\alpha\|^2 \rightarrow \min,$$

если в качестве нормы используется евклидова норма, или

$$\text{б) } \|\mathbf{y} - \tilde{\mathbf{y}}\| + \varepsilon \|\alpha\| \rightarrow \min$$

при использовании нормы в виде суммы модулей компонент вектора. Второе слагаемое в обоих критериях вводится для получения устойчивого решения – при уменьшении модулей коэффициентов разложения возможные искажения в значениях компонент векторов

опорной группы приведут к меньшим ошибкам при восстановлении неизвестных компонент искомого вектора.

В случае а) точное решение находится из системы линейных уравнений. Введем обозначения: Y – матрица векторов опорной группы, n строк, q столбцов. n – число известных компонент восстанавливаемого вектора y ; $\mathbf{Y} = (y^i - m_y)$ – матрица Y , в которой из каждого столбца вычтен вектор m_y (y^t в случае 2); M – матрица, все элементы которой равны 1; m – вектор, все компоненты которого равны 1; E – единичная матрица; α, β – вектора размерностью q .

Для выражения (1)

$$\tilde{y} = m_y + Y\alpha, \quad \alpha = \frac{1}{q} m + \left(E - \frac{1}{q} M \right) \beta.$$

Дифференцируя выражение а) и приравнявая нулю, получаем:

$$\left(\mathbf{Y}^T \mathbf{Y} + \varepsilon \left(E - \frac{1}{q} M \right) \right) \beta = \mathbf{Y}^T (y - m_y).$$

Для выражения (2): e^t – вектор, t -ая компонента которого равна 1, остальные 0; $L_t = (e^t)$ – матрица, столбцы которой равны вектору e^t . $L_t^T L_t = M, L_t^T e^t = m$. Имеем

$$\begin{aligned} \tilde{y} &= y^t + Y\beta, \quad \alpha = e^t + (E - L_t)\beta, \\ \left(Y^T Y + \varepsilon (E + M) \right) \beta &= Y^T (y - y^t) + \varepsilon m \end{aligned}$$

Система уравнений решается для известных значений компонент вектора y , полученное решение используется для предсказания неизвестных значений.

В случае критерия б) в качестве начального приближения для каждого испытуемого набора рассматриваются β , минимизирующие невязку $\Delta = \|y - \tilde{y}\|$. Минимум всего критерия находится градиентным методом оптимизации; для вычисления градиента можно сделать переход в нейросетевой базис.

Для построения для объекта его оптимальной опорной группы возможны два варианта. Первый задает фиксированный размер (мощность) опорной группы и ищет оптимальную опорную группу в ходе полного перебора всех опорных групп заданной мощности. Оптимальной опорной

группе будет соответствовать наименьшее значение критерия а) или б). Другой вариант (оптимизационный) предполагает первоначальное задание избыточного числа объектов в опорной группе и последующее сокращение ее размера в результате отбрасывания наименее значимых объектов (входящих в транспонированную регрессионную формулу с наименьшим по модулю весом α_i). Этот вариант разумнее реализовать в нейросетевом базисе.

После построения для объекта оптимальной опорной группы возможно прогнозирование значений отсутствующих у него признаков на основе единого регрессионного уравнения. При этом единственным требованием для возможности прогноза является необходимость наличия значений соответствующих признаков у объектов его опорной группы.

4. Применение нейросетевых технологий для обработки таблицы климатических данных

Для обработки была взята база данных по климату и растительности Института леса СО РАН в г.Красноярске, построенная по средним многолетним значениям характеристик климата, измеренных на 170 метеостанциях.

Каждая из 170 станций охарактеризована 20 полями, из которых от 5 до 9 полей занимают характеристики климата, остальные – характеристики растительного и почвенного покрова. По этим качественным характеристикам определена зональная принадлежность каждой станции к той или иной ландшафтной зоне и сделано предположение, что данные зоны имеют достаточно устойчивую корреляцию с некоторыми из климатических характеристик – теплом, относительным увлажнением, степенью континентальности климата и др. Зоны самого крупного порядка – тундра, лес и степь; между ними есть полосы перехода – лесотундра и редколесье на севере и лесостепь на юге. Этого перечня должно было хватить для того, чтобы прогнозировать (точнее идентифицировать) по климату зональный тип растительности, дающий название ландшафтной зоне.

База данных была сверена со справочниками по климату СССР и дополнена новыми данными. В настоящее время база данных имеет 250 записей (метеостанций) и 25 полей.

На основе описанной таблицы данных выполнялись работы по прогнозированию, классификации и глобальному моделированию с применением искусственных нейронных сетей.

4.1. Заполнение пропусков в таблице климатических данных

Решалась задача заполнения пропусков в значениях 6 интегральных климатических параметров:

- годовая сумма температур;
- годовая сумма осадков;
- коэффициент увлажнения;
- дефицит влаги;
- индекс сухости;
- индекс континентальности.

Для 121 метеостанции имелись значения всех шести климатических параметров, у остальных в различных комбинациях отсутствовали значения одного-трех параметров. Целью работы являлось прогнозирование отсутствующих значений.

Первым этапом работы являлось оценивание точности прогнозирования известных данных при помощи нейросетевых методов обработки информации. Использовалась нейросетевая нелинейная регрессия и транспонированная линейная регрессия. Определение точности прогноза проводилось в режиме скользящего экзамена на основе данных по 121 метеостанции, где имелись значения всех шести параметров. Оценивалась точность прогнозирования одного из климатических параметров на основе информации о пяти других.

При использовании транспонированной линейной регрессии для каждой из метеостанций по пяти климатическим параметрам отбиралась наилучшая опорная группа из нескольких метеостанций и по построенному транспонированному регрессионному уравнению на основе значений искомого признака у метеостанций наилучшей опорной группы прогнозировалось значение искомого признака целевой метеостанции и прогнозное значение затем сравнивалось с известным значением этого признака у данной метеостанции. Затем для метеостанции

прогнозировался другой климатический параметр после построения новой (возможно, включающей другие метеостанции) опорной группы по набору из оставшихся пяти параметров. Далее для каждого из параметров по всему набору из 121 метеостанции находились средние и максимальные ошибки прогноза известных значений. Такая работа по нахождению проводилась несколько раз для оценки влияния различных параметров метода транспонированной линейной регрессии (например, числа метеостанций в опорной группе) на достигаемую точность прогноза.

При использовании нейросетевой нелинейной регрессии процесс оценивания точности прогноза являлся модификацией скользящего экзамена, примененного для транспонированной линейной регрессии. По-прежнему прогнозировался один из параметров метеостанции по значениям пяти других параметров. При этом весь набор из 121 станции разбивался на 5 частей и 4 части (80% от числа метеостанций) использовались для обучения нейронной сети, а оставшаяся часть – для определения точности прогноза искомого климатического параметра. Для прогнозирования каждого из параметров обучались 5 нейронных сетей (каждая по своему набору из 80% метеостанций) и определялись средняя и максимальная ошибки прогноза этого параметра. Таким образом, было обучено 30 нейронных сетей.

В Табл. 1 представлены результаты по полученной точности прогноза. Для каждого климатического параметра представлены максимальная и средняя ошибки, выраженные в процентах от диапазона изменения соответствующего параметра. Для транспонированной линейной регрессии представлены точности прогноза при разном числе метеостанций в опорной группе (две, три и четыре станции). Чтобы сравнить полученную точность прогноза с точностью, достигаемой классическими методами построения зависимостей, в таблице приведена точность прогноза классической линейной регрессии.

Можно сделать следующие выводы. Нейросетевая нелинейная регрессия и транспонированная линейная регрессия (при четырех метеостанциях в опорной группе) показали лучшую точность прогноза по сравнению с классической линейной регрессией. Поскольку при решении задачи регрессии при переходе от линейных регрессионных зависимостей (классическая линейная регрессия) к нелинейным (нейросетевая регрессия) наблюдается повышение точности прогноза, то по аналогии можно

предположить потенциальное повышение точности прогноза при переходе от транспонированных линейных регрессионных зависимостей к нелинейным транспонированным регрессионным зависимостям (даже к простейшим дробно-рациональным, представленным в гл.3).

Таблица 1.

Достигнутая точность прогноза климатических параметров

Использованный метод построения регрессионной зависимости		Ошибка (максимальная (max), средняя (avg)) прогноза известных данных, в % от диапазона изменения параметра, по климатическим параметрам											
		Коэфф-т увлажнения		Годовая сумма температур		Дефицит влаги		Индекс континентальности		Индекс сухости		Годовая сумма осадков	
		max	avg	max	avg	max	avg	max	avg	max	avg	max	avg
Линейная регрессия		20,3	2,0	53,4	7,7	29,9	9,9	33,1	9,3	38,3	9,0	22,3	2,9
Нейросетевая нелин. Регрессия		4,4	1,1	10,1	3,0	14,8	5,0	11,8	3,2	13,9	4,1	5,5	1,2
Трансп. лин. регрессия	2 метеостанции	19,9	1,4	49,3	8,9	43,9	10,4	64,0	10,3	39,5	8,2	34,2	1,9
	3 метеостанции	21,8	1,3	34,6	5,7	39,	11,2	35,5	8,7	42,3	7,9	27,4	1,3
	4 метеостанции	18,7	1,2	32,5	5,7	29,2	7,5	27,6	7,7	37,6	6,1	12,9	1,5

Тем не менее, не надо на основании только этой работы делать вывод, что транспонированная линейная регрессия уступает регрессии нейросетевой. Имеются опыты по весьма эффективному применению транспонированной регрессии, даже простейшей линейной, как в данной работе, для обработки иных таблиц данных [18]. У транспонированной регрессии как у метода анализа эмпирических данных имеется одно преимущество, которое для некоторых задач может быть значимым. Поскольку для каждого объекта строится оптимальная опорная группа, то в результате получается некоторая группировка объектов – пары "искомый объект"–"малая группа объектов, наилучшим образом описывающая

искомый". Теперь можно решать некоторые содержательные задачи на основе такой группировки объектов.

В приведенном примере каждой метеостанции соответствует точка на карте (если рассматривать географические координаты станции – ее широту и долготу). Если для каждой станции проанализировать территориальное размещение станций, входящих в ее оптимальную опорную группу, то оказывается, что станции опорной группы расположены достаточно компактно относительно исходной. Иными словами, не происходит интерполяции климатических параметров, например, восточносибирских метеостанций по значениям параметров станций Западной Сибири и Дальнего Востока, или станций района Красноярск по станциям Таймыра и Алтая. Такая глобальная интерполяция, конечно, возможна, но по точности проигрывает локальным моделям климатической ситуации. Можно сказать, что достаточно существенный вклад в климатическую ситуацию района вносят особенности самого района, как-то: удаленность от океана, рельеф местности и другие параметры, влияние которых можно учитывать при прогнозировании климатических параметров метеостанции по близлежащим метеостанциям.

Вернемся к исходной задаче заполнения пропусков в таблице данных. Тестирование точности прогноза проводилось в условиях, достаточно сильно отличающихся от реальных – у метеостанций в действительности могли отсутствовать значения более чем одного параметра. Реально из 111 метеостанций, для которых восстанавливались отсутствующие значения, 84 станции имели пропущенные значения одного параметра, 19 – двух и 8 метеостанций – трех параметров. Определение точности прогноза одновременно двух и более параметров в данной работе не рассматривалось, но, естественно, эта точность оказалась бы ниже точности, достигнутой при прогнозе одного параметра (из-за сокращения набора признаков, по которым делался бы прогноз).

Точность прогноза, достигнутая нейросетевыми методами, оказалась достаточной и позволила провести дальнейшую работу по прогнозированию неизвестных значений климатических параметров у отдельных метеостанций. При этом в большинстве случаев прогнозные значения, получаемые разными методами, были достаточно близки друг к

другу и к экспертным заключениям, выполнявшимся с привлечением дополнительной информации.

Здесь может возникнуть справедливое замечание: можно ли доверять прогнозу, сделанному на основании обработки такой достаточно малой таблицы данных. Излагаемые далее результаты будут получены даже на еще меньших объемах данных. Однако повторим, что в случае отсутствия теоретически обоснованных моделей единственным способом проверки адекватности построенной модели является рассмотрение точности прогноза модели на тестовой выборке, а точность получилась достаточной. Также специфика данных проблемных областей такова, что прогноз модели непосредственно не "пойдет в дело" – исследователь будет сравнивать прогноз модели с собственными экспертными заключениями и использовать наиболее правдоподобное значение.

Результаты данного исследования позволяют предположить, что нейросетевые методы восстановления зависимостей с успехом могут быть использованы при решении различных задач прогнозирования при обработке климатической, метеорологической и экологической информации.

4.2. Построение классификационной модели ландшафтных зон и секторов континентальности

В этом разделе описаны исследования, выполнявшиеся для решения различных задач классификации на основе описанных выше климатических данных. В разделах 4.2.1 и 4.2.2 описана постановка задач классификации (разделения) метеостанций, находящихся в различных ландшафтных зонах и секторах континентальности, и приведены результаты по достигнутой на тестовых выборках точности прогноза. В качестве независимых признаков бралось несколько интегральных климатических параметров. В разделе 4.3 описан опыт моделирования глобального изменения климата, когда для ряда метеостанций по оценкам экспертов на 2030г. были изменены значения рассматриваемых климатических параметров и прогнозировалась потенциально возможная смена климатических зон и секторов континентальности у этих станций, что привело бы к изменению ландшафтных зон и смене лесообразователей в соответствующих районах.

4.2.1. Классификация ландшафтных зон Сибири

Для решения задачи классификации ландшафтных зон выбрано четыре крупных зональных класса :

1. Тундра и северное редколесье (лесотундра);
2. Лес;
3. Лесостепь;
4. Степь.

Для полученные этой обобщенной зональной классификации были переработаны данные по более точной зональной классификации, включающей 20 мелких климатических зон Сибири.

Ландшафтные зоны: тундра, лес, степь – это основные природные зоны Сибири. Лесотундра и лесостепь являются экотонами. Экотон – это промежуточная полоса между двумя соседними зонами. Экотон лесостепи представляет особый интерес с точки зрения классификации и прогноза: куда отнести лесостепь, к лесу или степи, или рассматривать ее как самостоятельную ландшафтную зону? Вопрос этот дискуссионен и решается разными специалистами по-разному [13,15].

Деление на зоны не является достаточным, так как в разных экорегионах зоны имеют свои самобытные черты состава, структуры, и функционирования. Поэтому широтно-зональное деление необходимо сочетать с секторным, проводимым с учетом континентальности. Континентальность характеризует суровость климата в регионах и имеет очень большое значение для классификации растительных зон Сибири [16].

Для решения задачи классификации секторов континентальности все лесные зоны (лес, лесостепь и лесотундра) на втором шаге построения иерархической системы разделены на три сектора континентальности:

1. континентальный,
2. резко континентальный,
3. крайне континентальный.

Таким образом, система включает 4 зоны, 3 сектора и 12 производных единиц – секторно-зональных классов.

Задача классификации состоит в следующем: определить, к какому зональному классу и сектору континентальности относится тот или иной набор значений климатических параметров метеостанций.

Для решения задачи классификации были выбраны следующие климатические параметры:

1. Дефицит влажности.
2. Сумма температур за период с температурами выше 5°C.
3. Сумма температур за период с температурами выше 10°C (*Gdd10*).
4. Средняя температура января (*Mjan*).
5. Средняя температура июля (*Mjul*).
6. Сумма осадков за холодный период года (*OsdCold*).

Мнемонические имена некоторых климатических параметров здесь приводятся для последующего удобства записи формул. Дополнительно в данную выборку были включены три параметра:

Сумма осадков за год (*Osd*), вычисленная по формуле $Osd = OsdCold + OsdWarm$, где *OsdWarm* – сумма осадков теплого периода.

Коэффициент увлажнения (*Kv*), вычисленный по формуле Мезенцева
$$Kv = \frac{Osd}{0.2 * Gdd10 + 306}.$$

Индекс континентальности (*IndK*), вычисленный по формуле Конрада,
$$IndK = \frac{1.7 * Amplituda}{\sin(shirota + 10)} - 14,$$
 где $Amplituda = Mjul + |Mjan|$, а *shirota* – географическая широта метеостанции.

Обучающая и тестирующая выборки для нейронных сетей содержат эти девять входных параметров (климатических признаков), и один выходной параметр – код ландшафтной зоны.

В работе строилось несколько классификационных моделей путем последовательного уточнения классификации.

На первом этапе исследований проводилось деление на 3 зональных класса – тундру-лесотундру, лес и степь. Лесостепь как зональный класс не была включена в выборку чтобы изучить поведение лесостепи как экотона, промежуточного между лесом и степью. В обучающую выборку включены данные для 99 метеостанций, а в тестирующую – для 64 станций. В обучающей и тестирующей выборках примеры делятся по классам следующим образом (здесь и далее первым числом является число примеров в обучающей, а вторым – в тестовой выборке):

- класс 1 (тундра и лесотундра) – 18 и 6 примеров,
- класс 2 (лес) – 65 и 54 примеров,

- класс 3 (степь) – 16 и 4 примера.

На этой обучающей выборке было обучено 15 нейросетей с различными параметрами. Процент правильных ответов на тестовой выборке колебался в диапазоне от 81 до 86%. Для сети с наилучшей точностью прогноза ошибка распределяется так: к лесотундровой зоне ошибочно отнесены 2 из 54 метеостанций лесной зоны; к лесной зоне – 2 из 6 метеостанций лесотундровой зоны; к степной зоне – 4 метеостанции из 54 метеостанций лесной зоны.

Самыми информативными климатическими параметрами из девяти использовавшихся для этой сети оказались сумма осадков за год, средняя температура июля и осадки холодного периода. В работе только определялась информативность входных сигналов, а работа по сокращению их числа не проводилась.

При тестировании 32-х станций лесостепи этой нейросетью получились следующие результаты:

- 14 метеостанций попали в лесную зону,
- 18 – в степь.

В общем случае лесостепная зона распределялась между лесом и степью следующим образом при тестировании всех 15-ти обученных нейронных сетей:

- от 11 до 20 метеостанций попадали в лесную зону,
- от 12 до 21 – в степь.

Отсюда следует, что лесостепные станции не только образуют пограничный класс (экотон), но и занимают определенное место в климатическом пространстве. Поэтому правомерно выделение лесостепи в качестве самостоятельной биоклиматической зоны. Применение обучаемых нейросетевых классификаторов позволило внести ясность в классификационное положение лесостепи, которое оставалось дискуссионным на протяжении долгого времени. Подтверждена точка зрения ландшафтоведов-географов, выделяющих лесостепь в качестве самостоятельной природной зоны наряду с лесом и степью.

На втором этапе исследования выполнено отделение лесостепи от леса и степи. В обучающую выборку включены данные для 108 метеостанций, а в тестирующую – для 60 станций. В обучающей и тестирующей выборках примеры делятся по классам следующим образом:

- класс 1 (лес) – 66 и 46 примеров,

- класс 2 (лесостепь) – 24 и 8 примеров,
- класс 3 (степь) – 18 и 6 примеров.

Обучено 15 нейросетей с различными параметрами и точностью прогноза на тестовой выборке от 73 до 82%. Для наилучшей нейросети получились следующие результаты: к лесной зоне ошибочно отнесены 2 из 8 метеостанций лесостепной зоны и 1 из 6 метеостанций степной зоны; к лесостепной зоне – 3 из 46 метеостанций лесной зоны и 2 из 6 метеостанций степной зоны; к степной зоне – 1 из 8 метеостанций лесостепной зоны и 2 из 46 станций лесной зоны.

Самыми информативными параметрами для лучшей нейросети оказались осадки холодного периода, индекс континентальности и температура января. Весьма знаменательно, что и в работах географов осадки холодного периода выделяются как главный, наиболее информативный признак для отделения леса от лесостепи [1].

Сравнивая результаты классификаций ландшафтных зон с лесостепью и без нее, можно сделать вывод, что лесостепь, как пограничный класс, выделяется несколько хуже других. Интервал колебания процента правильных ответов увеличился от 5 до 9%. В дополнение к тому, что экотоны трудно разграничить, лесостепь – это самая освоенная человеком ландшафтная зона. Вмешательства человека настолько велики, что изменились естественные границы и покров лесостепной зоны. Но это не означает, что лесостепь нельзя идентифицировать по климатическим параметрам. Класс "лесостепь", как и три других класса, дает вполне допустимые погрешности при тестировании: скачков через соседний класс нет, а ошибочные определения зоны связаны с тем, что в соседний класс попадают метеостанции, близкие к границе между соседними зонами.

Для более точного выделения зональных классов и наиболее информативных признаков целесообразно вводить бинарные классификации – разделения двух соседних зон.

На следующем этапе исследований сделаны 3 бинарные классификации:

1. Лесотундра и лес.
2. Лес и лесостепь.
3. Лесостепь и степь.

Рассмотрим первую бинарную классификацию. В обучающую выборку включены данные для 83 метеостанций, а в тестирующую – для 53 станций. В обучающей и тестирующей выборках примеры делятся по классам следующим образом:

- класс 1 (лесотундра) – 20 и 4 примера,
- класс 2 (лес) – 63 и 49 примеров.

Обучено 10 нейросетей с различными параметрами. При тестировании процент правильных ответов колебался от 87 до 93%. Наилучшей нейросетью к лесотундровой зоне ошибочно отнесено 5 из 49 метеостанций лесной зоны. Самыми информативными параметрами для данной нейронной сети оказались годовая сумма осадков, осадки холодного периода и средняя температура января.

Данное ранжирование параметров по информативности не поддается пока объяснению. Согласно общепринятому взгляду, северная граница леса определяется температурой июля или другими показателями из группы "теплообеспеченность". Все же результат косвенно говорит о суровости зимы, и в этом есть своя логика.

Следующая бинарная классификация – разделение леса и лесостепи. В обучающую выборку включены данные для 89 метеостанций, а в тестирующую – для 55 станций. В обучающей и тестирующей выборках примеры делятся по классам следующим образом:

- класс 1 (лес) – 63 и 49 примеров,
- класс 2 (лесостепь) – 26 и 6 примеров.

Обучено 15 нейросетей с различными параметрами. При тестировании процент правильных ответов колебался от 81 до 96%, наилучшая нейросеть ошибалась так: к лесной зоне отнесена 1 из 6 метеостанций лесостепной зоны; к лесостепной зоне отнесены 2 из 49 метеостанций лесной зоны. Самыми информативными параметрами для этой сети оказались средняя температура января, индекс континентальности и дефицит влаги.

Экологическое значение этого ранжирования параметров трудно интерпретировать, поскольку экспертная оценка не совпадает. Дефицит влаги должен быть на первом месте, а осадки зимнего периода – на следующем. Значимость температуры января и индекса континентальности оценить сложно. Этот результат мог возникнуть потому, что лесные районы расположены в Западной и Средней Сибири, а лесостепь – в

Восточной. Иными словами, температурный класс зон один, а секторы континентальности – разные.

И последняя бинарная классификация – разделение лесостепи и степи. В обучающую выборку включены данные для 43 метеостанций, а в тестирующую – для 17 метеостанций. В обучающей и тестирующей выборках примеры делятся по классам следующим образом:

- класс 1 (лесостепь) – 21 и 11 примеров,
- класс 2 (степь) – 18 и 6 примеров.

Обучено 20 нейросетей с различными параметрами. При тестировании наилучшей нейросети ошибочно отнесена к лесостепной зоне 1 из 6 метеостанций степной зоны. Самыми информативными параметрами для лучшей сети оказались коэффициент увлажнения, годовая сумма осадков и средняя температура июля. По оценкам экспертов, это отличное выделение наиболее значимых признаков. Показатели увлажнения и должны играть роль дифференцирующих признаков при разделении степи и лесостепи.

Обсуждая результаты, можно сделать вывод, что бинарная классификация ландшафтных зон оказалась более точной (увеличился процент правильных ответов). Для каждой из классификаций имеется собственное ранжирование климатических параметров по степени важности для принятия решения – отнесения метеостанции к тому или иному классу.

4.2.2. Идентификация лесных зон по континентальности

В данной работе все лесные зоны (лесотундра, лес и лесостепь) разделены на три сектора континентальности:

1. Континентальный.
2. Резко континентальный.
3. Крайне континентальный.

Первая классификация сделана отдельно для равнин. В обучающую выборку включены данные для 76 метеостанций, а в тестирующую – для 62 метеостанций. Взяты те же входные параметры, что и для рассмотренной ранее классификации ландшафтных зон, выходным параметром (классом) является сектор континентальности. В обучающей и тестирующей выборках примеры делятся по классам следующим образом:

- класс 1 (континентальный сектор) – 19 и 15 примеров,

- класс 2 (резко континентальный) – 24 и 20 примеров,
- класс 3 (крайне континентальный) – 33 и 27 примеров.

Обучено 12 нейросетей. При тестировании процент правильных ответов колебался от 72 до 82%. Наилучшая сеть ошибается так: к континентальному сектору отнесены 3 из 20 метеостанций резко континентального сектора и 1 из 27 метеостанций крайне континентального сектора; к резко континентальному сектору отнесены 3 из 15 станций континентального сектора и 2 из 27 станций крайне континентального сектора; к крайне континентальному сектору отнесены 3 из 20 станций резко континентального сектора. Самыми информативными параметрами для данной нейронной сети оказались коэффициент увлажнения, средняя температура июля и годовая сумма осадков.

Для классификации секторов в горах в обучающую выборку включены данные для 44 метеостанций, а в тестирующую – для 18 метеостанций. В обучающей и тестирующей выборках примеры делятся по классам следующим образом:

- класс 1 (континентальный) – 15 и 5 примеров,
- класс 2 (резко континентальный) – 12 и 4 примера,
- класс 3 (крайне континентальный) – 17 и 9 примеров.

Для разделения секторов в горах обучено 16 нейросетей. При тестировании процент правильных ответов колебался от 72 до 89%. При тестировании нейросети с лучшей точностью прогноза получились следующие результаты: к резко континентальному сектору отнесена 1 из 9 метеостанций крайне континентального сектора, к крайне континентальному сектору отнесена 1 из 4 метеостанций резко континентального сектора. Самыми информативными параметрами для данной нейронной сети оказались осадки холодного периода, индекс континентальности и дефицит влаги. Это хороший результат. Все три параметра действительно значимы и имеют экологический смысл применительно к ситуации.

Информативность климатических параметров для классификации секторов континентальности на равнине и в горах сильно отличается, поэтому эти секторы нежелательно объединять в одну классификацию. Нужно отметить, что для классификации равнинных секторов индекс континентальности не играет большой роли (по сравнению с его значимостью для классификации секторов в горах). Хотя этот параметр, по

мнению ландшафтоведов, имеет важное значение для классификации секторов.

По итогам работы по разделению секторов можно сделать следующие выводы: данная классификационная модель, как и классификация ландшафтных зон, пригодна для дальнейших исследований, т.к. по небольшому количеству основных параметров климата она дает неплохие результаты.

4.3. Прогнозирование возможного изменения ландшафтных зон и секторов континентальности

Для прогноза крупных зональных классов – ландшафтных зон или зонобиомов – разрабатываются модели глобального уровня [19,20,22]. Считается, что изменения температуры января и июля наиболее доступны для долгосрочного прогноза, а поскольку по ним можно вычислить индекс континентальности, то можно предложить последний как один из наиболее доступных производных параметров. Известно, что с континентальностью связана повторяемость воздушных масс океанического и континентального происхождения и режимы тепла и влаги. Эксперты предсказывают, что в наших широтах средняя температура января увеличится на 3-4°C, а июля – на 1-2°C, и эти данные согласуются у различных авторов. Это означает, что континентальность снизится, и режимы тепла будут менее суровы в году.

Труднее предсказать режим увлажнения. Поэтому в данной работе взято некоторое приближение сценариев. Так, предполагается, что количество осадков холодного периода может увеличиться на 10-30 мм, а количество годовых осадков – на 30-50 мм. И эти все значения меняются по долготе (см. Таблица 2).

Еще хуже дело обстоит с учетом взаимодействия систем в силу не только прямых, но и обратных связей климат–растительность, и множества косвенных взаимодействий (лес – почва, лес – насекомые, пожары и др. экстремальные события). Разработка общей концепции прогноза наземных экосистем еще далека от завершения.

Задача данного исследования – сделать первый шаг к прогнозу изменения ландшафтного покрова на основе базы данных по климату и растительности, опираясь на фундаментальную концепцию природной зональности Докучаева–Берга–Григорьева–Будыко. Учитывая характер

базы данных, следует ограничиться точечным прогнозом, т.е. исследованием возможности смены секторно-зональной принадлежности конкретных точек (метеостанций) согласно одному из сценариев климата на 2030 год. Весьма актуальной представляется возможность выявления тех территорий, которые могут оказаться особенно подвергнутыми риску сокращения лесных площадей.

Таблица 2.

Прогноз изменения климатических параметров по секторам
континентальности на 2030 год

Климатические Параметры	Континентальный и резко континентальный секторы	Крайне Континентальный сектор
Температура января	+ 3°	+ 4°
Температура июля	+ 1°	+ 2°
Сумма осадков за холодный период года	+ 30 мм	+ 10 мм
Сумма осадков за год	+ 50 мм	+ 30 мм

Для решения задачи прогноза смены ландшафтных зон из обучающей и тестирующей выборки для нейронных сетей исключены четыре параметра, которые пока недоступны для прогноза:

1. Коэффициент увлажнения.
2. Дефицит влажности.
3. Сумма температур за период выше 5°C.
4. Сумма температур за период выше 10°C.

Итак, в обучающей и тестируемой выборках осталось пять параметров:

1. Индекс континентальности.
2. Средняя температура января.
3. Средняя температура июля.
4. Сумма осадков за холодный период года.
5. Сумма осадков за год.

Для прогноза смены ландшафтных зон и секторов континентальности на 2030г. выбрана описанная ранее последовательно

уточняемая классификационная модель ландшафтных зон (без бинарной классификации зон) и деление на три сектора континентальности. На выборке из оставшихся пяти параметров для каждой классификационной модели были обучены наборы нейросетей, и для каждой классификационной модели было выбрано по одной нейросети с наибольшим процентом правильных ответов:

Для классификации ландшафтных зон без лесостепи – 83% правильных ответов. При тестировании к лесотундровой зоне ошибочно отнесены 5 из 54 станций лесной зоны, к лесной зоне – 1 из 6 станций лесотундровой зоны и 1 из 4 станций степной зоны, к степной зоне – 3 из 54 станций лесной зоны.

Для классификации леса, лесостепи и степи – 80%. При тестировании к лесной зоне ошибочно отнесены 2 из 8 метеостанций лесостепной зоны и 1 из 6 станций степной зоны, к лесостепной зоне – 4 из 46 станций лесной зоны и 2 из 6 станций степной зоны, к степной зоне – 2 из 8 станций лесостепной зоны и 1 из 6 станций степной зоны.

Для классификации секторов континентальности на равнине – 82%. При тестировании к континентальному сектору ошибочно отнесены 3 из 20 станций резко континентального сектора, к резко континентальному сектору – 2 из 15 станций континентального сектора и 3 из 27 станций крайне континентального сектора, к крайне континентальному сектору – 2 из 20 станций резко континентального сектора.

Для классификации секторов в горах – 94%, к резко континентальному сектору ошибочно отнесена 1 из 4 метеостанций крайне континентального сектора.

Самыми информативными параметрами при прогнозе оказались следующие:

- Для ландшафтных зон без лесостепи – средняя температура января, осадки холодного периода и сумма осадков за год.
- Для ландшафтных зон с лесостепью – осадки холодного периода и сумма осадков за год.
- Для секторов континентальности на равнине – осадки холодного периода, сумма осадков за год и индекс континентальности.
- Для секторов континентальности в горах – средняя температура июля, индекс континентальности и средняя температура января.

При обучении нейросетей для прогноза на основании пяти параметров распознаваемость классов, в общем, не ухудшилась: понижение процента правильных ответов произошло на 1-2%. Распознаваемость секторов континентальности в горах улучшилась с 89 до 94%. Стоит отметить, что обучение нейронных сетей на меньшем количестве входных параметров происходило значительно дольше. Интервал колебания процента правильных ответов стал больше. В целом данную систему классификации можно использовать для прогноза, предварительно убедившись в ее объективности.

При дальнейшем исследовании возможной смены ландшафтных зон нужно дополнительно включить в обучающую и тестирующую выборки новые климатические параметры, доступные для прогноза.

В качестве примера на полученных нейросетях по данным эксперта выполнен точечный прогноз ситуации на 2030г. для 40 метеостанций, расположенных в разных ландшафтных зонах и секторах континентальности. Определялась возможность изменения лесорастительного потенциала метеостанций, т.е. их зональной и секторной принадлежности при заданном сценарии климата.

По таблице "Изменение климатических параметров по секторам на 2030 год" (Таблица 2) для 40 метеостанций были изменены следующие параметры: средняя температура января, средняя температура июля, сумма осадков за холодный период года, сумма осадков за год. На основании измененных параметров был вычислен индекс континентальности по формуле Конрада. Далее использовались обученные нейронные сети для определения зон и секторов, которые должны соответствовать измененным климатическим параметрам метеостанций.

При прогнозе разные станции ведут себя по-разному. 26 метеостанций остались в тех же зонах и секторах (например, Богучаны, Енисейск, Игарка, Красноярск, Томск). А те, которые были ближе к климатическим границам, переходят в другой класс. У 8 метеостанций изменилась зона (например: Братск, Кежма, Якутск, Янск), эти зоны имеют потенциал, соответствующий соседней, более южной зоне. И для 7 станций изменился сектор континентальности (например: Канск, Минусинск, Абакан), эти секторы имеют потенциал, соответствующий соседнему, менее континентальному сектору. Соответственно прогнозу в данных зонах и секторах должна измениться и растительность.

Отсюда следует, что к 2030г. все современные ландшафтные зоны сохранятся на территории Сибири, хотя некоторые смещения границ, возможно, произойдут. Общий вывод данного точечного прогноза сводится к тому, что тенденции к сменам на более "южную" зону могут проявиться во всех звеньях ландшафтов: в растительности, почвах, гидрологическом режиме и т.д. Тенденция смены более северных зон более южными и менее континентальными, отмечаемая другими авторами [21], в целом подтверждается.

Оценивая первые опыты моделирования, не следует особенно доверять полученным результатам, поскольку авторы не учитывали обратное влияние лесных экосистем на климат и многие косвенные взаимодействия в ландшафтах. В данном случае важен не столько результат прогноза, сколько методика прогноза. Фактически мы прогнозируем, какое новое равновесное состояние будет устойчивым, если температура июля вырастет на 1-2 градуса, января – на 3-4 градуса, осадки за год вырастут на 30-50 мм., осадки холодного периода – на 10-30 мм. Если какая-либо точка переходит в другой, соседний класс, это означает для нее потенциальное состояние растительного покрова, тогда как реальный покров не сразу достигнет этого состояния. Нужно, чтобы прошло, по меньшей мере, несколько сотен лет (время релаксации лесной растительности) для достижения нового равновесного состояния в ландшафте при данном климатическом сценарии.

Определим теоретическое и прикладное значение прогноза. Очевидно, что лесное хозяйство должно принимать во внимание долгосрочные прогнозы климата для того, чтобы планировать лесовосстановление, поддержание лесистости и устойчивости лесного покрова с учетом региональных особенностей.

Моделирование с помощью нейронных сетей позволяет выделить те объекты (регионы), которые могут подвергнуться максимальному риску. Так, например, не следует проводить массовые посадки леса в тех районах, где наиболее высок риск засухи, пожара. Здесь важнее провести первоочередную противопожарную профилактику лесных массивов и ограничить посадки леса наиболее благоприятными условиями топографии и почв.

5. Заключение

В данной работе описаны результаты применения нейросетевых методов обработки информации для решения задач прогнозирования и классификации при обработке климатических данных. Задачи решаются достаточно эффективно и результаты их решения могут быть использованы в дальнейшем. Очень просто выглядит возможность моделирования глобального изменения климата и оценки последствий этого изменения.

К настоящему времени накоплен очень большой опыт по применению нейронных сетей для решения задач в различных проблемных областях – не только в климатологии (здесь это наш первый опыт), но и в медицине, психологии и политологии, диагностике и оптимальном управлении в технических системах и т.д. [9]. Нейронные сети позволяют решать различные неформализованные задачи (задачи, где алгоритм решения неизвестен). Исследователь при этом получает очень эффективную модель проблемной области и может очень просто моделировать различные ситуации, предъявляя сети различные данные и оценивая ответ, выдаваемый сетью. Нейронные сети могут применяться и в ситуациях, когда известен сравнимый по точности прогноза метод решения, но критично, например, время получения результата, поскольку обученная нейронная сеть решает предъявляемую ей задачу очень быстро.

Большинство пользователей (от простых пользователей нейросетевых экспертных систем до специалистов-исследователей в области нейроинформатики), применяющих нейронные сети, отмечают единственный существенный недостаток нейронных сетей. А именно, человеку не понятно, какие "рассуждения" используются внутри нейронной сети при решении задачи. Иными словами, очень трудно понять процесс получения сетью результата. Первый шаг в этом направлении уже сделан. Сеть может количественно оценить влияние входных признаков на точность решения задачи. Ранжируя входные признаки по этим показателям значимости и последовательно исключая наименее значимые (дообучая при этом сеть извлекать нужную информацию из меньшего объема данных), можно получить минимально необходимый для правильного решения задачи набор входных параметров. Это уже дает информацию для содержательного анализа.

В настоящее время ведутся работы по созданию новой технологии, позволяющих не только решать задачи с помощью обучаемых нейронных сетей, но и генерировать вербализованное (на естественном языке) описание процесса решения сетью задачи. Фактически, по таблице экспериментальных данных, описывающих проблемную область, можно будет получить новое знание – явный алгоритм решения поставленной неформализованной задачи.

Авторы благодарны А.Н. Горбаню, Д.И. Назимовой, Л.Ф. Ноженковой за руководство работой.

Работа выполнена при финансовой поддержке Красноярского краевого фонда науки (гранты 6F0181, 7F0113), Министерства науки и технологий РФ (подпрограмма "Перспективные информационные технологии", проект № 05.04.1291) и ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.).

ЛИТЕРАТУРА

1. Бессолицина Е.П., Какарека С.В., Крауклис А.А., Кремер Л.К. Геосистемы контакта тайги и степи: юг Центральной Сибири. – Новосибирск: Наука, 1991. – 217 с.
2. Браверман Э.М., Мучник И.Б. Структурные методы обработки эмпирических данных. – М., Наука. Гл. ред. физ.-мат. лит., 1983. – 464с.
3. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. – М.: Наука, 1979. – 448с.
4. Гилев С.Е., Коченов Д.А., Миркес Е.М., Россиев Д.А. Контрастирование, оценка значимости параметров, оптимизация их значений и их интерпретация в нейронных сетях // Доклады III Всероссийского семинара “Нейроинформатика и ее приложения”.- Красноярск, 1995. - С.66-78.
5. Горбань А.Н. Обучение нейронных сетей. М.": изд. СССР-США СП "ParaGraph", 1990. - 160 с.
6. Горбань А.Н. Проблема скрытых параметров и задачи транспонированной регрессии // Нейроинформатика и ее приложения. Тезисы докладов V Всероссийского семинара. Красноярск: изд. КГТУ, 1997. – с.57-58.
7. Горбань А.Н., Миркес Е.М. Оценки и интерпретаторы ответа для сетей двойственного функционирования. Вычислительный центр СО РАН в г.

- Красноярске. Красноярск, 1997. - 24 с. (Рукопись деп. в ВИНТИ 25.07.97, № 2511-B97)
8. Горбань А.Н., Новоходько А.Ю., Царегородцев В.Г. Нейросетевая реализация транспонированной задачи линейной регрессии // Нейроинформатика и ее приложения. Тезисы докладов IV Всероссийского семинара, 5-7 октября 1996 г. Красноярск: изд. КГТУ, 1996. – с.37-39.
 9. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. –Новосибирск: Наука (Сиб. отделение), 1996. – 276с.
 10. Загоруйко Н.Г., Елкина В.Н., Лбов Г.С. Алгоритмы обнаружения эмпирических закономерностей. – Новосибирск: Наука, 1985. – 110с.
 11. Кендалл М., Стьюарт А. Статистические выводы и связи. – М.: Наука, 1973. – 900с.
 12. Кендалл М., Стьюарт А. Многомерный статистический анализ и временные ряды. – М.: Наука, 1976. – 736 с.
 13. Ландшафты юга Восточной Сибири. Карта м-ба 1: 1 500 000 / Ред. В. Б. Сочава. – М.:ГУГК, 1977.
 14. Лбов Г.С. Методы обработки разнотипных экспериментальных данных. – Новосибирск: Наука, 1981. – 157с.
 15. Мильков Ф. Н. Лесостепной ландшафт и его зональное подразделение / Изв. АН СССР. Сер. геогр., № 5, 1951. С. 3-14.
 16. Назимова Д.И., Молокова Н.И., Джансеитов К.К. Высотная поясность и климат в горах южной Сибири / География и природные ресурсы., № 2, 1981. С. 68-78.
 17. Рао С.Р. Линейные статистические методы. – М.: Наука, 1968. – 548 с.
 18. Царегородцев В.Г. Транспонированная регрессия в задаче интерполяции свойств химических элементов. // Вестник КГТУ. Информатика, вычислительная техника, управление. Сб. научных трудов / Красноярск: КГТУ. 1997. 139с. – с.31-36.
 19. Gorban A.N., Novokhodko A.Yu. Neural Networks In Transposed Regression Problem, Proc. of the World Congress on Neural Networks, Sept. 15-18, 1996, San Diego, CA, Lawrence Erlbaum Associates, 1996, pp. 515-522.
 20. Solomon A. M. and Cramer W. Biospheric Implications of Global Environmental Change / Repr. From Solomon et al. Vegetation Dynamics and Global Change. Chapman and hall, London, 1976. p. 25-51.

21. Tchebakova N.M., Mousured R.A., Leemans R. and Nazimova D.I. Possible Vegetation Shifts in Siberia under Climatic Change. / Impacts of Climate Change on Ecosystems, 1995. p. 67-83.
22. Tchebakova N.M., Mousured R.A. and Nazimova D.I. A Siberian Vegetation Model Based on Climatic Parameters / Can. J. For. Res. 24, 1994. p. 1597-1607.
23. Walter, Hand Box. Global Classification of Natural Terrestrial Ecosystem / Vegetatio. 32 (2), 1976. p. 75-81.
24. Кирдин А.Н., Новоходько А.Ю., Царегородцев В.Г. Глава 7. Скрытые параметры и транспонированная регрессия // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.
25. Миркес Е.М. Глава 9. Логически прозрачные нейронные сети и производство явных знаний из данных // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.
26. Горбань А.Н. Глава 1. Возможности нейронных сетей // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.
27. Горбань А.Н. Глава 2. Решение задач нейронными сетями // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.
28. Горбань А.Н. Глава 3. Быстрое дифференцирование, двойственность и обратное распространение ошибки // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.

ИНТУИТИВНОЕ ПРЕДСКАЗАНИЕ НЕЙРОСЕТЯМИ ВЗАИМООТНОШЕНИЙ В ГРУППЕ

М.Г. Доррер

Сибирский государственный технологический университет,

660049, Красноярск, пр. Мира 82

E-mail: dorrer@sibstu.kts.ru

В работе решается задача моделирования и прогнозирования системы взаимоотношений в группе на основе состояния и поведения исследуемых. Задача решается при помощи нейросетевых технологий с использованием нейроимитатора *MultiNeuron* на баз психологических и социологических методик – социометрии и минимизированного психологического опросника. Показано, что искусственная нейронная сеть позволяет с удовлетворительной точностью предсказывать отношения типа «человек-человек» и «человек-группа», а также может служить вспомогательным механизмом при разработке психологических методик.

1. Проблема оценки взаимоотношений

В работе практических психологов, имеющих дело с подбором персонала или исследующих взаимоотношения внутри уже сложившихся групп (примером первого может служить психолог-консультант по подбору персонала, примером второго - офицер по работе с личным составом в частях, классный руководитель в школе) постоянно возникает задача установления и прогноза межличностных отношений в группе.

Под отношением в данной работе понимается психологический феномен, сутью которого является возникновение у человека психического образования, аккумулирующего в себе результаты познания конкретного объекта действительности (в общении это другой человек или группа людей), интеграции всех состоявшихся эмоциональных откликов на этот объект, а также поведенческих ответов на него [2]. Кроме того, общение обыкновенно происходит в условиях определенной ситуации: в присутствии других людей, которые для общающихся в разной степени субъективно значимы, на фоне какой-то конкретной деятельности, при действии каких-либо экспериментальных факторов.

2. Общая задача экспериментов

В данной работе была поставлена задача смоделировать и, по возможности, спрогнозировать систему взаимоотношений в группе на основе состояния и поведения исследуемых, оставляя в стороне такие аспекты формирования отношений между людьми, как внешний облик, приписываемые человеку цели и мотивы [2]. Оценке и прогнозу подвергались межличностные «статусно-ролевые» [11] отношения в группах. Оценка совместимости «человек-человек» и «группа-человек» велась по оценке статуса исследуемых - индивидуальной (от каждого к каждому) и групповой (от группы к человеку).

Задача моделирования и прогнозирования взаимоотношений людей в группе (коллективе) неоднородна - она может быть условно подразделена на следующие подзадачи:

- прогноз вхождения исследуемого в сложившийся коллектив;
- прогноз совместимости между собой двух исследуемых.

Кроме того, при проведении экспериментов предполагалось апробировать к задаче прогноза межличностных отношений методику интуитивной выдачи предсказания минуя создание описанной (дескриптивной) [3] реальности.

3. Применяемые в экспериментах психологические методики

Для определения фактических отношений в исследуемых группах применялась социометрическая методика. Данная методика позволяет определить положение исследуемого в системе межличностных отношений той группы, к которой он принадлежит. Социометрическое исследование группы обычно проводится тогда, когда группа включает в себя не менее 10 человек и существует не менее одного года. Всем членам исследуемой группы предлагается оценить каждого из товарищей (включая и самого себя - появляется возможность изучения самооценки исследуемых). В стандартном варианте методики оценка ведется по трехступенчатой шкале предпочтений - «приемлю - безразличен - отвергаю». Однако для получения большей разрешающей способности методики шкала была модифицирована до десятибалльной. В используемом варианте социометрического исследования применялось

следующее задание: «Оцените своих товарищей, задав себе вопрос: «Насколько я бы хотел работать с этим человеком в одной группе?». Поставьте в соответствующей графе оценку от 1 до 10 баллов по следующему принципу: 1 - не хочу иметь с ним ничего общего, 10 - с этим человеком я бы хотел работать сильнее всего».

Результатом исследования для каждого из испытуемых в группе являлась стеновая оценка статуса и экспансивности. Стен [1] представляет собой усредненную оценку, нормированную в предположении, что оценки распределены по закону нормального распределения и, следовательно, выполняется правило «трех сигм». Статусом именуется стеновый балл всех оценок, сделанных данному члену группы, экспансивностью - стеновый балл всех оценок, сделанных данным испытуемым всем остальным представителям группы.

В процессе экспериментов предполагалось подтвердить (или отвергнуть) гипотезу о том, что нейросеть позволяет на основе психологических особенностей людей (представителей группы) моделировать взаимоотношения в группе и выдавать прогноз по вхождению в группу нового члена и по взаимоотношениям двух индивидуумов. Предполагалось также оценить качество прогноза - возможные значения ошибок и их распределение.

Описание личностных качеств испытуемых предполагалось получить на основе опросника, составленного А.Г. Копытовым (ППФ КГУ). Опросник включает в себя три субтеста, каждый из которых составлен из вопросов, предназначенных для определения константных свойств человеческой личности - темперамента, эмоциональности, контактности и т. п. Общее число вопросов - 90, в первом субтесте - 29, во втором - 25 и в третьем - 36.

Сбор данных производился путем проведения одновременного анкетирования в студенческих учебных группах по опроснику А.Г. Копытова и социометрического исследования. Затем результаты социометрии обрабатывались на специально разработанной программе, рассчитывающей стеновые оценки статуса и экспансивности.

Эксперименты по обучению нейронных сетей производились на нейросетевом имитаторе MultiNeuron v2.0 в режиме предиктора, то есть нейросети, имеющей на выходе вещественное число (подробное описание MultiNeuron v2.0 - см. [13], [15]).

4. Эксперименты по предсказанию группового статуса

В этой серии экспериментов предполагалось проверить, насколько нейронные сети способны моделировать вхождение в группу отдельного человека.

По вышеописанной методике были обследованы три студенческие группы - третьего, четвертого и пятого курсов, общее число собранных анкет - 48 (19, 17 и 12 по группам соответственно). Результаты анкетирования каждой из групп был составлен задачник, представляющий собой реляционную таблицу, включающую следующие поля:

$№$ - автоиндексируемый номер записи, ID - номер испытуемого по списку группы, $w1_1$ - $w1_29$ - ответы на вопросы первого субтеста, $w2_1$ - $w2_25$ - ответы на вопросы второго субтеста, $w3_1$ - $w3_36$ - ответы на вопросы третьего субтеста, $to1$ - $to30$ - оценки, выставленные данным испытуемым членам группы (строка социометрической матрицы), St - значение стеновой оценки статуса данного испытуемого, Ex - экспансивности.

Для первого и третьего субтестов, у которых вопрос имел два варианта ответа («Да»/«Нет»), ответ кодировался по принципу 1 - «Да», 2 - «Нет». Второй субтест, имеющий три варианта ответов («а», «б», «в») - 1 - вариант «а», 2 - «б», 3 - «в».

При формировании структуры задачника поля $w1_1$ - $w3_36$ были обозначены как входные, поле Ex - как выходное. Нейросеть в процессе обучения должна была приобрести умение предсказывать статус члена группы по его ответам на опросник А.Г. Копытова.

На первом этапе были проведены эксперименты для выяснения оптимальных параметров нейросети, предназначенной для решения задачи предсказания статуса члена группы. Из-за малочисленности выборок эксперименты велись в режиме «скользящего тестирования», когда для решения задачи обучается столько же сетей, сколько задач в задачнике. При обучении каждой из сетей одна задача исключалась, и потом сеть тестировалась по ней. Для оценки качества предсказания $H_{выб}$ применялся

средний модуль ошибки $H_{выб} = \frac{\sum |H_i|}{n_{выб}}$, чем ниже значение - тем,

соответственно лучше предсказание. Результаты этого этапа экспериментов сведены в таблицу 1.

Значения чисел нейронов - N_{neu} - были взяты из следующих соображений: нейросети с числом нейронов менее 16 обучались решению

Таблица 1

Результаты экспериментов по подбору оптимальных параметров нейросети, решающей задачу предсказания статуса исследуемых.

№№	Характеристики сети		$H_{выб}$
	N_{neu}	σ	
1	16	0.1	2,475
2	16	0,4	2,791
3	16	0,7	2,488
4	32	0,1	2,569
5	32	0,4	3,006
6	32	0,7	3,384
7	64	0,1	2,891
8	64	0,4	2,703
9	64	0,7	2,676

задачи неустойчиво, процесс оптимизации постоянно заходил в тупик, а $H_{выб}$ во всех таких экспериментах превышало 3 (30% относительной погрешности). 64 является максимально допустимым значением числа нейронов для программы *MultiNeuron v.2.0*. Значения характеристических чисел нейронов были распределены в интервале от 0.1 до 0.7, поскольку данный интервал является, по опыту, накопленному в группе «НейроКомп» [4], [5], [6], [7], [8], [9], [11], [13], [15], интервалом, в котором как правило лежат оптимальные характеристические числа нейронов.

Таким образом, по результатам данной серии экспериментов оптимальным было признано количество нейронов, равное 16, и характеристический параметр нейрона равный 0.1, поскольку данные значения обеспечивают наилучшую выборочную оценку качества прогноза $H_{выб}$.

Следующим этапом работы была серия экспериментов, позволяющих оценить точность предсказания статуса исследуемых внутри групп. Для каждой из групп было выполнено обучение сетей для

проведения скользящего контроля. Затем результаты скользящего контроля фиксировались и сводились в табл. 2.

Таблица 2

Результаты экспериментов по установлению точности предсказания статуса исследуемых внутри групп

№	Количество испытуемых	N _{выб}
1	19	2,587
2	17	2,854
3	12	2,475

Однако, по опыту применения нейроимитаторов, известно, что на одних и тех же обучающих выборках предсказание выдаваемое сетью может существенно разниться.

Причина этого в том, что начальная карта синаптических весов генерируется случайным образом. Для преодоления данной проблемы в практике создания нейросетей (см. например [7]) используется предсказание ответов группой сетей, обученных на одних и тех же данных - консилиумом.

Таблица 3

Оценки погрешностей предсказания статуса исследуемых в группах консилиумами сетей.

N _{иссл}	H ₁	H ₂	H ₃	H ₄	H ₅	H ₆	H ₇	H ₈	H ₉	H ₁₀	H _{ср}	N _{выб}
19	3,02	3,68	3,88	4,13	3,14	3,38	4,09	3,46	2,82	3,32	3,49	2,83
17	3,32	4,80	4,33	4,50	4,46	3,15	3,72	4,31	3,20	4,51	4,03	3,84
12	2,20	2,68	3,23	2,59	3,86	2,96	2,82	3,28	3,52	2,58	2,97	2,41

Решено было применить этот метод и для данной задачи. При проведении скользящего контроля по выборке для каждого из случаев обучалась не одна нейросеть, а десять.

Фиксировались средние выборочные значения ошибки предсказания статуса каждым из экспертов, а затем, оценивалась погрешность предсказания всем консилиумом.

Для этого в качестве ответа на каждую из задач скользящего контроля подавалось среднее значение ответов десяти нейросетей - экспертов. Результаты этого эксперимента представлены в табл. 3.

Здесь $N_{иссл}$ - число исследуемых в данной группе, $H_1 - H_{10}$ - средние ошибки предсказания статуса для каждой из сетей консилиума, $H_{ср}$ - среднее значение ошибки по всем сетям консилиума, $H_{выб}$ - ошибка предсказания всем консилиумом.

Таким образом, по трем группам средний модуль ошибки составляет 3,08 (или, в относительных цифрах, средняя погрешность составляет 30,8%).

Такая погрешность является удовлетворительной для задачи предсказания статуса членов группы, поскольку как правило не выводит испытуемого из групп классификации - «лидер»-«середняк»-«аутсайдер», то есть отражает тенденцию вхождения в группу нового человека.

Кроме того, при статистическом исследовании экспериментальных выборок было вычислено среднее расстояние d между случайными оценками x_i и x_j

$$d = \frac{1}{N(N-1)} \sum_{i,j} |x_i - x_j| : i \neq j,$$

где N - количество элементов выборки.

Можно считать, что d характеризует математическое ожидание расстояния между двумя случайными примерами выборки.

Для экспериментов установления статуса тестируемых в группе $d = 4.033$, или 40.33%. Таким образом можно утверждать, что полученная сетью погрешность (30,8%) значительно отличается от случайной.

Следующая серия экспериментов производилась с целью уяснения, насколько можно предсказывать результаты вхождения в одну группу на базе опыта, накопленного сетью по другой группе.

В ходе экспериментов для каждой из групп был обучен консилиум из десяти нейросетей (их характеристики, как и в предыдущих экспериментах, $N_{neu}=16$, $\sigma=0,1$). Здесь задачник подавался для обучения полностью, то есть сеть обучалась предсказанию статуса по всем представителям группы. Затем на сетях этого консилиума тестировались две другие группы.

Для сглаживания фактора случайности при генерации сетей в качестве вычисленных значений при расчете ошибки определения статуса брались по каждой оценке средние значения из вычисленных десятью

сетями консилиума. Результаты этой серии экспериментов представлены в табл. 4.

Здесь **Об.** - порядковый номер группы, по которой обучались нейронные сети консилиума, **Тст.** - порядковый номер группы, по которой сети тестировались.

Таблица 4

Результаты перекрестного тестирования

Об.	Тст.	H ₁	H ₂	H ₃	H ₄	H ₅	H ₆	H ₇	H ₈	H ₉	H ₁₀	H _{ср}	H _{выб}
1	2	1,87	3,96	2,85	3,65	4,62	1,82	2,82	1,97	1,77	4,32	2,97	2,48
1	3	2,26	3,98	3,58	3,61	2,36	2,46	3,64	2,16	2,55	3,11	2,97	1,79
2	1	4,31	4,03	3,92	3,48	4,17	3,66	3,83	4,33	4,03	3,78	3,95	3,5
2	3	3,82	1,81	2,91	3,43	2,75	3,13	3,08	2,53	2,57	3,06	2,91	2,05
3	1	3,4	4,09	3,21	2,91	2,76	3,65	3,03	2,56	2,89	3,51	3,20	2,79
3	2	3,60	3,28	3,72	2,94	4,24	4,30	3,91	4,35	3,60	4,13	3,81	3,77

При анализе данной серии экспериментов заметны следующие закономерности:

- предсказание социального статуса испытуемых нейронными сетями, обученными по другим группам (не по тем, в которой производилось определение статуса при социометрическом опросе) по своему качеству несколько хуже, чем такое же предсказание, сделанное нейросетями, обученными на этой же группе;

- однако, в большинстве случаев (в двух третях из проведенных экспериментов) оценка качества (средний модуль ошибки $H_{\text{выб}}$) является приемлемой (менее 3 баллов или, в относительных значениях - менее 30%);

- хорошо видно, как при предсказании статуса испытуемых в группах реализуется принцип создания надежных систем из ненадежных элементов, заложенный в концепцию нейронных сетей: ошибка предсказания одной нейросетью может составлять неприемлемо большую величину, однако консилиум из нескольких нейросетей решает задачу

существенно лучше - ошибка предсказания консилиумом сетей меньше большинства из ошибок отдельных сетей, она также всегда меньше чем среднее значение ошибок отдельных сетей консилиума.

После оценки качества предсказания между группами решено было проверить гипотезу о том, что нейросеть может накапливать опыт не только по отдельной группе, но и аккумулировать его по любой заданной последовательности испытуемых. Для проверки гипотезы была предпринята следующая серия экспериментов: данные по всем группам были объединены в один задачник, по которому проводилось скользящее тестирование консилиума из десяти сетей - экспертов. Результаты данной серии из 480 экспериментов представлены в табл. 5.

Таблица 5
Результаты тестирования консилиумов сетей, обученных по полной выборке.

№	$H_{\text{выб}}$
1	3,02
2	2,56
3	2,88
4	3,04
5	2,94
6	2,88
7	2,74
8	2,46
9	2,59
10	3,12
Весь консилиум	2,32

Видно, что, как и в предыдущей серии экспериментов, погрешность каждого из экспертов (и, как минимум, математическое ожидание погрешности) выше, чем погрешность консилиума, то есть математическое ожидание оценок по консилиуму сетей всегда (или, вернее, в большинстве случаев) ближе к верному ответу, чем оценки отдельных экспертов.

Кроме того, легко заметить, что предсказание статуса исследуемых в группе улучшается с накоплением выборки - оценка погрешности предсказания, сделанного нейросетями, обученными по объединенной выборке лучше, чем в любых других экспериментах.

Иначе говоря, нейросети обладают возможностью аккумулировать опыт предсказания социометрического статуса исследуемых в группе, причем этот опыт не локален - навык, полученный на исследуемых одной группы значим и для оценки исследуемых, принадлежащих к другим группам.

Этот результат подтверждает тезис, приведенный в [18], о том, что оценки равных в группе устойчивы и, видимо, на них не влияет изменение состава группы.

Причина этого феномена, предположительно, в том, что при предсказании статуса испытуемых информация о них существенно ограничена - отсутствуют данные анамнестического плана, данные об их социальном положении.

Этим практически исключается из состава используемых в прогнозе данных информация о внешнем облике, принадлежности к социокультурной или национальной группе - то есть вся социальная история личности и коллектива в целом, хотя известно, что эти факторы могут вызвать существенное различие в поведении людей со схожим типом личности.

Информация же о константных психологических качествах испытуемых относительно однородна от группы к группе, что позволяет нейросети накапливать опыт, основанный на ней.

5. Нейросетевое исследование структуры опросника

Следующим этапом работ по прогнозу статуса испытуемых в группах было определение значимости вопросов опросника и исключение из него наименее значимых вопросов.

Согласно результатам главы 2 это может привести к улучшению качества прогноза, выдаваемого нейросетью. Для решения данной задачи была использована возможность вычисления значимости параметров, заложенная в MultiNeuron.

Были обучены пять нейронных сетей по задачику, включающему все три группы исследуемых, затем, средствами MultiNeuron, определены числовые значения значимости сигналов, соответствующих вопросам опросника.

После этого список вопросов был отсортирован по среднему значению величины значимости. В результате была получена следующая картина (вопросы размещены по убыванию значимости):

1_6. Вы обычно говорите без запинок?

1_23. Вы обычно предпочитаете делать несложные дела, не требующие от Вас большой энергии?

1_7. Легко ли Вы можете найти другие варианты решения известной задачи?

3_24. Самое трудное для Вас - это справиться с собой.

3_28. Вы склонны принимать все слишком близко к сердцу.

3_22. Вам нетрудно внести оживление в довольно скучную компанию.

1_2. Легко ли Вам выполнять работу, требующую длительного внимания и большой сосредоточенности?

1_1. Легко ли Вы генерируете идеи, связанные с работой?

3_10. Вы не раз замечали, что незнакомые люди смотрят на Вас критически.

3_8. Иногда у Вас пропадает или изменяется голос, даже если Вы не простужены.

2_3. Окружающим известно, что у меня много разных идей, и я почти всегда могу предложить какое-то решение проблемы.

1_19. Обычно Вы предпочитаете легкую работу?

1_27. Дрожат ли у Вас иногда руки во время ссоры?

3_20. Некоторые так любят командовать, что Вам все хочется делать наперекор, хотя Вы знаете, что они правы.

2_25. Бывает, что я говорю незнакомым людям о вещах, которые кажутся мне важными, независимо оттого, спрашивают меня, или нет.

2_19. Если начальство или члены семьи меня в чем-то упрекают, то, как правило, только за дело

3_3. Дурные предчувствия всегда оправдываются

2_24. Обычно я спокойно переношу самодовольных людей, даже когда они хвастаются или другим образом показывают, что они высокого мнения о себе.

2_11. Устаревший закон должен быть изменен

3_29. Вы любите готовить (пищу)

3_35. Вы вели дневник.

1_8. Вы когда-нибудь опаздываете на свидание или работу?

2_5. Ко дню рождения, к праздникам (я люблю делать подарки / затрудняюсь ответить / считаю, что покупка подарков несколько неприятная обязанность)

1_9. Часто ли Вам не спится из-за того, что вы поспорили с друзьями?

2_21. При равной продолжительности дня мне было бы интереснее работать(столяром или поваром / не знаю, что выбрать / официантом в хорошем ресторане)

1_3. Испытываете ли Вы чувство беспокойства, что Вас неправильно поняли в разговоре?

1_5. Быстры ли у Вас движения рук?

3_4. Вы очень часто не в курсе дел и интересов тех людей, которые Вас окружают.

1_28. Испытываете ли Вы тягу к напряженной ответственной деятельности?

3_7. Нравятся ли Вам “первоапрельские” шутки?

1_17. Трудно ли Вам говорить очень быстро?

1_15. Всегда ли Вы платили бы за провоз багажа на транспорте, если бы не опасались проверки?

3_25. Временами Вам так нравится ловкость какого-нибудь преступника, что Вы надеетесь, что его не поймают.

1_10. Нравится ли Вам быстро бегать?

3_33. Ваши родители и другие члены семьи часто придираются к Вам

2_2. У меня бывают такие волнующие сны, что я просыпаюсь

3_18. Вы совершаете много поступков, о которых потом жалеете (больше и чаще чем другие)

2_10. Думаю, что обо мне правильнее сказать, что я (вежливый и спокойный / верно нечто среднее / энергичный и напористый)

3_34. Временами, когда Вы плохо себя чувствуете, Вы бываете раздражительными.

3_12. Держитесь ли Вы обычно “в тени” на вечеринках или в компаниях?

1_20. Медленны ли Ваши движения, когда Вы что-то мастерите?

3_2. Иногда Вам очень хотелось навсегда уйти из дома

3_31. Вы стараетесь избегать конфликтов и затруднительных положений.

3_16. Иногда по несколько дней Вы не можете отделаться от какой-нибудь пустяковой мысли.

3_11. Вы знаете, кто виноват в большинстве Ваших неприятностей.

1_21. Вы обычно предпочитаете выполнять только одну операцию?

1_18. Дрожат ли у Вас иногда руки во время ссоры?

1_14. Все ли Ваши привычки хороши и желательны?

3_14. Не все Ваши знакомые Вам нравятся.

3_15. Предпочитаете ли Вы иметь поменьше приятелей, но зато особенно близких Вам.

3_13. Иногда Вы не уступаете людям не потому, что дело действительно важное, а просто из принципа.

3_26. Если Вам не грозит штраф, то Вы переходите улицу там, где Вам удобно, а не там, где положено.

2_7. Мне нравится работа разнообразная, связанная с частыми переменами и поездками, даже если она немного опасна

1_29. Нравится ли Вам быстро говорить?

3_9. Вам неловко входить в комнату, где уже собрались и разговаривают люди

2_20. Бывает, что я говорю незнакомым людям о вещах, которые кажутся мне важными, независимо оттого, спрашивают меня, или нет.

3_21. Вы предпочитаете не заговаривать с людьми, пока они сами к Вам не обратятся.

3_23. Когда Вы узнаете об успехах близкого знакомого, у Вас появляется чувство, что Вы неудачник.

1_24. Сосет ли у Вас под ложечкой перед ответственным разговором?

2_14. Мне доставляет удовольствие совершать рискованные поступки только ради забавы

3_6. Временами в голову приходят такие мысли, что лучше о них никому не рассказывать

2_13. Иногда какая-нибудь навязчивая мысль не дает мне заснуть

2_8. Я предпочел бы иметь дачу (в оживленном дачном поселке / предпочел бы нечто среднее / уединенную, в лесу)

2_1. Я предпочитаю несложную классическую музыку современным популярным мелодиям?

2_22. Когда мною пытаются командовать, я нарочно делаю все наоборот

3_17. Вы часто беспокоитесь о чем-нибудь.

1_22. Бывает ли так, что Вы говорите о вещах, в которых не разбираетесь?

1_16. . Обычно Вам трудно переключать внимание с одного дела на другое?

2_4. У меня бывают такие волнующие сны, что я просыпаюсь

1_11. Испытываете ли Вы постоянную жажду деятельности?

3_19. В гостях Вы держитесь за столом лучше, чем дома.

3_36. Вы легко смущаетесь.

3_30. Вы не осуждаете того, кто стремится взять от жизни все, что может.

2_16. Если бы я работал в хозяйственной сфере, мне было бы интереснее

1_25. Считаете ли Вы свои движения медленными и неторопливыми?

3_32. Справляетесь ли Вы с делом лучше, обдумывая его самостоятельно, а не обсуждая с другими.

2_12. Если кто-то разозлился на меня (Я постарался бы его успокоить / я не знаю, что бы я предпринял / это вызвало бы у меня раздражение)

1_12. Быстро ли Вы читаете вслух?

3_5. Иногда Вы так настаиваете на чем-нибудь, что люди начинают терять терпение

2_18. Обычно я могу сосредоточенно работать, не обращая внимания на то, что люди вокруг меня очень шумят

1_26. Ваша речь обычно медленна и нетороплива?

2_17. Вечер, проведенный за любимым занятием, привлекает меня больше, чем оживленная вечеринка

2_15. Я делаю людям резкие критические замечания, если мне кажется, что они того заслуживают

1_4. Любите ли Вы игры в быстром темпе?

1_13. Если Вы обещали что-то сделать, всегда ли Вы выполняете свое обещание независимо от того, удобно это Вам или нет?

2_9. Я провожу много свободного времени, беседуя с друзьями о тех прежних событиях, которые мы вместе пережили когда-то.

2_6. Иногда у меня бывали огорчения из-за того, что люди говорили обо мне дурно за глаза без всяких на то оснований.

3_27. Вы часто испытываете тягу к новым впечатлениям, к тому, чтобы встряхнуться, испытать возбуждение.

2_23. Люди относятся ко мне менее благожелательно, чем я того заслуживаю своим добрым к ним отношением.

3_1. Часто ли Вы переходите на другую сторону улицы, чтобы не встречаться с кем-нибудь из знакомых?

Для определения значимости субтестов теста было произведено вычисление средней значимости по вопросам каждого из них. Субтесты распределились в следующем порядке: наиболее значимый - 1-й, далее - 3-й и наименее значимый - 2-й. Данное распределение можно проиллюстрировать гистограммой (рис. 1). Для построения этой гистограммы все вопросы, отсортированные в порядке убывания значимости, были разбиты на девять десятков, а затем для каждой из них было подсчитано число вхождений вопросов, принадлежащих первому, второму и третьему субтесту.

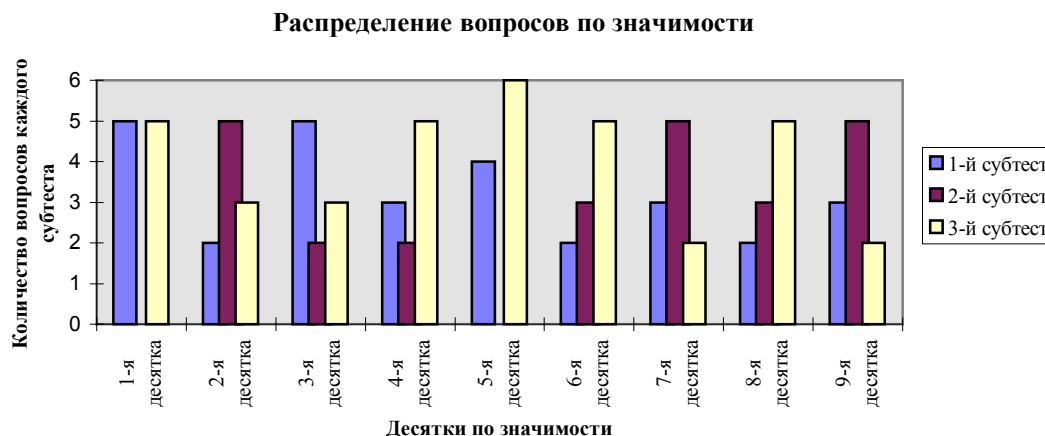


Рис. 1. Диаграмма распределения вопросов теста по их значимости для предсказания статуса испытуемых.

Для вопросов первого субтеста виден эксцесс распределения в сторону большей значимости, второго - в сторону меньшей, а вопросы третьего - относительно равномерно распределены по всему интервалу.

Была произведена серия экспериментов с целью выяснить достаточный для нейросети объем опросника. На каждом этапе исключалась половина из имеющихся вопросов опросника.

При исключении половины вопросов скользящий контроль консилиума сетей, обученных на выборке по всем группам, дал среднюю погрешность в 24%, при исключении трех четвертей вопросов - в 28% и, наконец, при исключении семи восьмых нейросети обучиться не смогли.

Таким образом, примерно половина вопросов и без того изначально минимизированного теста оказалась для нейросети избыточной, даже приводящей к ухудшению оценки качества предсказания. Оптимальным можно признать опросник из половины вопросов, максимальных по своей значимости для нейронной сети, поскольку результаты тестирования для него лучше, чем для всех остальных вариантов, включая и полный набор вопросов.

6. Оценка оптимизации задачника нейросетью с позиций теории информации

Разницу между первоначальным (заданным психологом) и требуемым нейросети для успешного решения задачи объемом опросника можно оценить с позиций теории информации [17].

Начальное количество информации, содержащейся в тесте можно оценить исходя из того, что вопросы первого и третьего тестов бинарны (варианты ответов «Да» и «Нет», вероятность наступления каждого из них - 0.5), а ответы на вопросы второго - могут с равной вероятностью соответствовать наступлению одного из трех событий, которые будем считать равновероятными (варианты ответов «А», «Б» и «В», $p=0.333$). Тогда, исходя из формулы Шеннона

$$I(\xi, \xi) = - \sum_{i=1}^n p_i \log_2 p_i$$

и учитывая, что количество вопросов в первом субтесте - 29, во втором - 25 и в третьем - 36 можем вычислить суммарное количество информации, содержащееся в ответах на вопрос теста:

$$I_{\text{теста}} = - \sum_{i=1}^{29} \frac{1}{2^{29}} \log_2 \frac{1}{2^{29}} + - \sum_{i=1}^{35} \frac{1}{3^{35}} \log_2 \frac{1}{3^{35}} + - \sum_{i=1}^{2^{36}} \frac{1}{2^{36}} \log_2 \frac{1}{2^{36}} = 104.62.$$

После исключения половины вопросов из-за их малой значимости для нейронной сети в оптимизированном опроснике осталось 16 вопросов первого субтеста, 9 - второго и 20 - третьего. Количество информации, оставшееся после оптимизации:

$$I_{\text{теста}} = - \sum_{i=1}^{2^{16}} \frac{1}{2^{16}} \log_2 \frac{1}{2^{16}} + - \sum_{i=1}^9 \frac{1}{3^9} \log_2 \frac{1}{3^9} + - \sum_{i=1}^{2^{20}} \frac{1}{2^{20}} \log_2 \frac{1}{2^{20}} = 50.26,$$

то есть количество информации при оптимизации сократилось несколько более чем вдвое.

7 Эксперименты по предсказанию парных взаимоотношений

В этой серии экспериментов предполагалось установить, способны ли нейросети воспроизвести взаимоотношения пары испытуемых.

Обучающие выборки имели следующую структуру: № - номер примера, ID_From - номер оценивающего, ID_To - имя оценивающего, ID_To - номер оцениваемого, Name_To - имя оценивающего, w1_1_From - w3_36_From - ответы на вопросы опросника А.Г. Копытова, данные оценивающим, w1_1_To - w3_36_To - ответы на вопросы опросника А.Г. Копытова, данные оцениваемым, Osen - данная оценка.

В задачник включались строки, соответствующие всем клеткам социометрической матрицы кроме диагональных, отвечающих за самооценку испытуемых.

Был сформирован задачник по группе 5-го курса. В него вошли 132 примера, по которым было произведено обучение соответствующего числа сетей по методике скользящего контроля.

В силу большой трудоемкости задачи обучения по выборкам такого объема и размерности (обучение одной сети занимает около 40 мин.) обучения консилиумов не проводилось.

Результат скользящего контроля следующий: средняя относительная ошибка предсказания парных взаимоотношений в группе составила 33,1%.

Затем было вычислено среднее расстояние d между случайными оценками x_i и x_j , вычисляемое, как и в п.3.4, по формуле

$$d = \frac{1}{N(N-1)} \sum_{i,j}^N |x_i - x_j| : i \neq j,$$

где N - количество примеров обучающей выборки.

Данная величина составила 6.612 (или, относительно шкалы измерения признака, 66.12%), то есть отличие предсказания сети от случайного почти двукратное.

Таким образом, можно говорить, что нейронные сети могут предсказывать не только усредненный статус члена группы, но и взаимоотношения между двумя произвольно взятыми личностями.

Работа выполнена при финансовой поддержке ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.).

ЛИТЕРАТУРА

1. Анастаси А. Психологическое тестирование. - М. Педагогика, 1982 - кн.1 - с.320, кн.2 - с.360.
2. Бодалев А.А. О взаимосвязи общения и отношения // Вопросы психологии - 1994 - №1 - с.122-126.
3. Бурлачук Л.Ф., Коржова Е.Ю. К построению теории измеренной индивидуальности в психодиагностике. // Вопросы психологии - 1994 - №5 - с.5-12.
4. Гилев С.Е. Сравнение характеристических функций нейронов. // Тезисы докладов III всероссийского семинара «Нейроинформатика и ее приложения». - Красноярск: изд. КГТУ, 1995 - с.82.
5. Гилев С.Е., Горбань А.Н., Миркес Е.М. и др. Определение значимости обучающих параметров для принятия нейронной сетью решения об ответе. // Нейроинформатика и нейрокомпьютеры: Тезисы докладов рабочего семинара, 8-11 октября 1993 г. - Красноярск: Институт биофизики СО РАН, 1993. - с.8.
6. Гилев С.Е., Коченов Д.А., Миркес Е.М., Россиев Д.А. Контрастирование, оценка значимости параметров, оптимизация их значений и их интерпретация в нейронных сетях // Тезисы докладов III всероссийского семинара «Нейроинформатика и ее приложения». - Красноярск: изд. КГТУ, 1995 - с.66-78.
7. Горбань А.Н. Обучение нейронных сетей.- М. СП ПараГраф - 1990.
8. Горбань А.Н., Миркес Е.М. Контрастирование нейронных сетей. // Тезисы докладов III всероссийского семинара «Нейроинформатика и ее приложения». - Красноярск: изд. КГТУ, 1995 - с.78-79.
9. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. - Новосибирск: Наука, 1996 - с.144.
10. Зеличенко А.И. Интеллектуальные системы и психологическое знание. // В книге: Компьютеры и познание. - М.: Наука, 1990 - с.69-86.

11. Коченов Д.А., Россиев Д.А. Аппроксимация функций класса $C[a,b]$ нейросетевыми предикторами // Тезисы докладов рабочего семинара «Нейроинформатика и нейрокомпьютеры», Красноярск, 8-11 октября 1993 г. - Красноярск, 1993 - с.13.
12. Практическая психология для преподавателей. / под ред. М.К.Тутушкиной // М.: Филин, 1997 - с. 167.
13. Россиев Д.А., Винник Н.Г. Предсказание «удачности» предстоящего брака нейросетевыми экспертами. // Нейроинформатика и ее приложения: Тезисы докладов II всероссийского рабочего семинара, 7-10 октября 1994 г. - Красноярск, 1994. - с.45.
14. Россиев Д.А., Гилев С.Е., Коченов Д.А. MultiNeuron, версии 2.0 и 3.0 // Тезисы докладов III всероссийского семинара «Нейроинформатика и ее приложения». - Красноярск: изд. КГТУ, 1995 - с.14.
15. Россиев Д.А., Головенкин С.Е., Назаров Б.В. и др. Определение информативности медицинских параметров с помощью нейронной сети // Диагностика, информатика и метрология - 94: Тезисы научно-технической конференции, Санкт-Петербург, 28-30 июня 1994 г. - СПб., 1994. - с.348.
16. Россиев Д.А., Коченов Д.А. Пакет программ «MultiNeuron» - «Configurator» - «Tester» для конструирования нейросетевых приложений. // Нейроинформатика и ее приложения: Тезисы докладов II всероссийского рабочего семинара, 7-10 октября 1994 г. - Красноярск, 1994. - с.30.
17. Шеннон К. Работы по теории информации в кибернетике, пер. с англ., М., 1963, с. 243-332.
18. Шнейдерман Б. Психология программирования. М.: Радио и связь, 1984 - с.139.
19. Россиев Д.А. Глава 5. Медицинская нейроинформатика // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.

АППРОКСИМАЦИЯ МНОГОМЕРНЫХ ФУНКЦИЙ ПОЛУТОРАСЛОЙНЫМ ПРЕДИКТОРОМ С ПРОИЗВОЛЬНЫМИ ПРЕОБРАЗОВАТЕЛЯМИ

М.Г. Доррер

Сибирский государственный технологический университет,

660049, Красноярск, пр. Мира 82

E-mail: dorrer@sibstu.kts.ru

В работе описаны теоретические предпосылки и программная реализация искусственной нейронной сети, обладающей важным новым свойством: генерацией структуры «от простого к сложному». Описан итерационный процесс наращивания объема сети. Используются два условия на ограничение требуемого объема - по достижению требуемой точности или по критерию сравнения константы Липшица для сети и выборочной константы Липшица.

1. Постановка проблемы

Функция F на R задана набором своих значений в случайных точках пространства $(x_1, y_1), \dots, (x_N, y_N)$. Построим ее аппроксимацию при помощи комбинаций $f_i(\bar{X}, \bar{a})$ - функций из набора Φ , гладких и непрерывно дифференцируемых. Тогда

$F_1 = F - f(x, a)$ - ошибка аппроксимации F функцией $f_1(\bar{X}, \bar{a})$;

$F_2 = F_1 - f_1(x, a)$ - ошибка предыдущего шага аппроксимации

$F_3 = F_2 - f_2(x, a)$

Аппроксимация может вестись не только подбором коэффициентов, но и выбором на каждом шаге функций $f_i(x, a)$ из Φ . Таким образом, может быть получено разложение функции F в сходящийся ряд вида:

$$F = \sum f_i(a, x)$$

Решение задачи аппроксимации может быть получено путем минимизации функционала качества, соответствующего квадрату отклонения:

$$H = \sum (F(\bar{a}^i, \bar{x}^i) - y_i)^2 \rightarrow \min,$$

Задача состоит в приближении функции F , заданной исходной выборкой точек, при помощи нейросети-предиктора с неизвестным заранее количеством нейронов и видом функции, используемой в преобразователе каждого из нейронов.

Решение может быть представлено как итерационный процесс, состоящий из следующих шагов:

- Подключение нового нейрона;
- Оптимизация ошибки предсказания значений в заданных точках для текущего нейрона путем подбора функции преобразователя, ее параметров и весов синапсов;

Если заданная точность достигнута, то процесс можно остановить, в противном случае - процесс повторяется сначала, причем параметры уже обученных нейронов фиксируются, так что каждый новый нейрон обучается вычислять погрешность, оставшуюся от предыдущих.

Количество итераций процесса исчерпания ошибки может быть также ограничено из условия превышения нижней оценки константы Липшица для конструируемой нейронной сети над верхней оценкой выборочной константы Липшица.

2. Аналитическое решение

Пусть Y_i - приближаемое очередным слоем значение. Тогда Y_0 - само значение приближаемой функции в точках экспериментальной выборки, а Y_1 и последующие - погрешности вычисления на соответствующем шаге.

Обучение ведется оптимизацией параметров сети каким либо из градиентных методов по всему задачнику.

Тогда при обучении k -го нейрона

$$Y_k = w_{k1}^{(2)} F_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i \right),$$

соответственно H (функция ошибки) для всего задачника будет иметь вид

$$H = \sum_{j=1}^{N \cdot p} \left[w_{k1}^{(2)} F_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) - Y_k^j \right]^2,$$

то есть в качестве критерия близости аппроксимируемой и аппроксимирующей функций выбрана сумма квадрата ошибки по всей обучающей выборке.

Для обучения каждого очередного нейрона используются частные производные функции F_k по весам синапсов первого слоя $w_{ik}^{(1)}$:

$$\frac{\partial H_k}{\partial w_{ik}^{(1)}} = \sum_{j=1}^{N \cdot p} 2 \left[w_{k1}^{(2)} F_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) - Y_k^j \right] \left[w_{k1}^{(2)} \alpha_k x_i^j F'_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) \right],$$

параметру нейрона α_k

$$\frac{\partial H_k}{\partial \alpha_k} = \sum_{j=1}^{N_{np}} 2 \left[w_{k1}^{(2)} F_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) - Y_k^j \right] \left[w_{k1}^{(2)} \left(\sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) F'_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) \right]$$

и весу синапса второго (выходного) слоя $w_{k1}^{(2)}$ соответствующему данному нейрону

$$\frac{\partial Y_k}{\partial w_{k1}^{(2)}} = \sum_{j=1}^{N_{np}} 2 \left[w_{k1}^{(2)} F_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) - Y_k^j \right] \left[F_k \left(\alpha_k \sum_{i=1}^{N_{ex}} w_{ik}^{(1)} x_i^j \right) \right],$$

где N_{np} - число примеров обучающей выборки.

Однако, если вычисление функции H связано с затратами процессорного времени порядка T_H , то вычисление ее градиента традиционным способом потребует времени порядка

$$T_{gradH} = n T_H,$$

где n - число переменных функции H . Учитывая, что в задачах, для которых традиционно применяются нейросети, величина n может достигать нескольких тысяч, аналитическое решение для вычисления градиента функции ошибки следует признать неприемлемым.

Однако при описании решающей функции F в виде сети автоматов вычисление градиента функции ошибки H может быть представлено как функционирование системы, двойственной исходной. При таком подходе

$$T_{gradH} \approx C T_H,$$

где C - константа, не зависящая от размерности n и в большинстве случаев примерно равная 3.

Таким образом, мы приходим к записи решения исходной задачи в идеологии нейронных сетей.

3. Запись решения в идеологии нейросетей

Как уже было сказано, выше, ряд вида $F = \sum f_i(a, x)$ может быть представлен как нейронная сеть, имеющая структуру, показанную на рис.1.

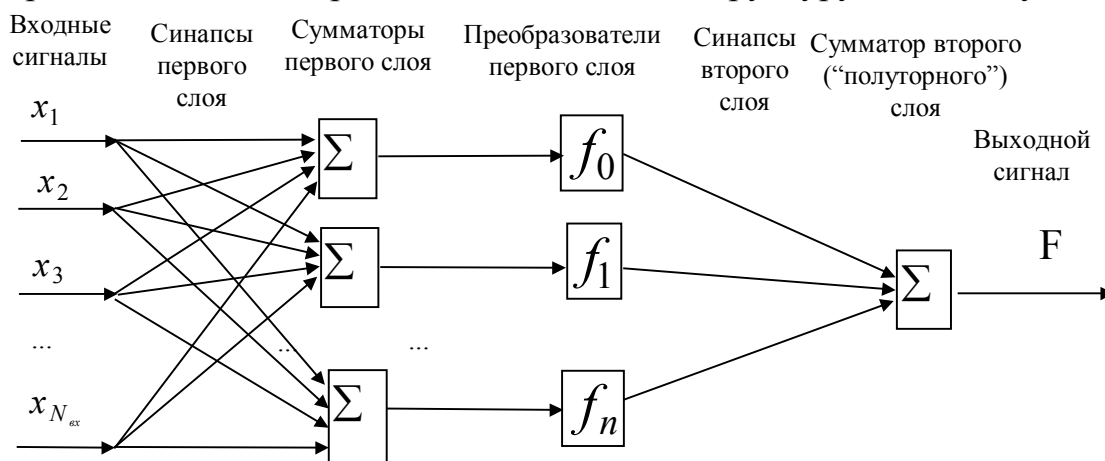


Рис.1. Структура «поточной» нейросети

Тогда вычисление градиента функции ошибки H можно производить, используя схему, двойственную данной. При этом решение задачи может быть существенно упрощено применением следующего приема. Поскольку обучение следующего слоя начинается тогда, когда предыдущие уже обучены, а их связи зафиксированы, то, фактически, каждый нейрон обучается отдельно от других, а в качестве значения, вычисляемого k -м нейроном берется погрешность вычисления функции предыдущими $k-1$ - нейронами, или F_i .

Процесс обучения всей сети сводится этим ее разбиением к ряду последовательных процессов обучения структур вида, представленного на рис. 2.

Здесь $x_1 - x_N$ - входные сигналы сети, $W_{ij}^{(1)}$ - синапсы первого слоя от i -го входа к j -му сумматору первого слоя, $W_{j1}^{(2)}$ - синапсы второго слоя

от j -го преобразователя к выходному сумматору, f_j - j -й преобразователь, символом Σ обозначаются тривиальные сумматоры.

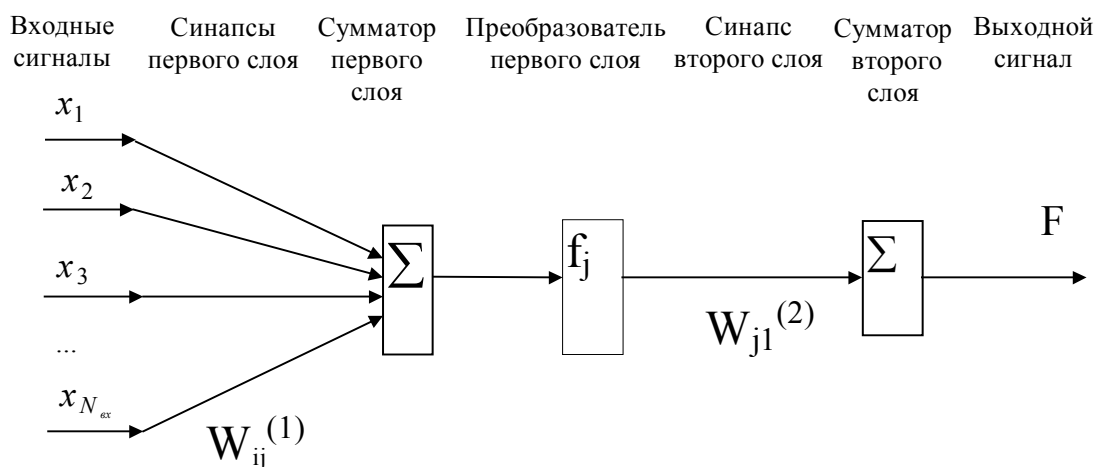


Рис. 2. Структура одного «потока» сети

Далее в тексте такая структура будет именоваться «потоком сети» или просто «потоком».

Вычисление производных H (функции ошибки) для сети, представляющего собой один «поток», можно вести на основе аппарата двойственных функций и алгоритма обратного распространения ошибки. Схема прохождения сигналов при обратном функционировании показана на рис. 3.

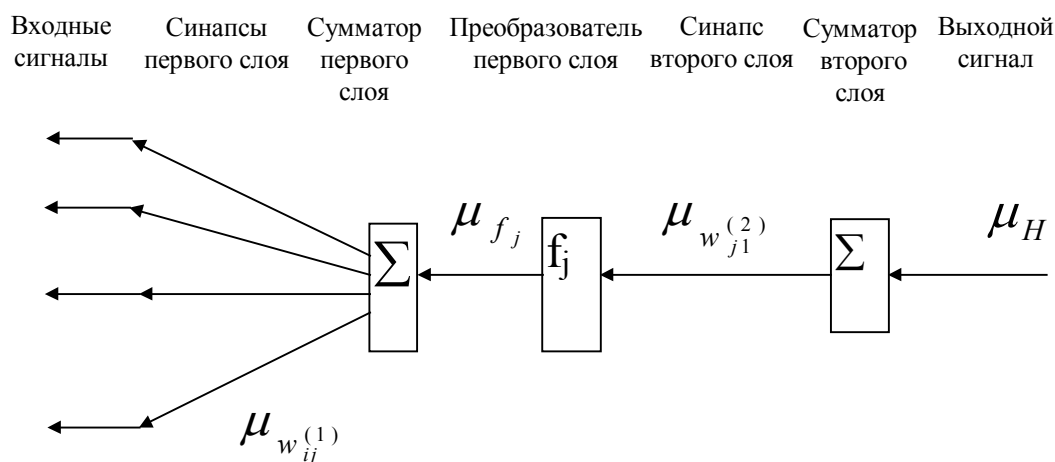


Рис. 3. Распространение двойственных сигналов в одном «потоке»

Здесь μ_H - двойственный сигнал от функции оценки, передаваемый без изменения через тривиальный сумматор второго слоя,

$\mu_{w_{j1}}^{(2)}$ - двойственный сигнал от соответствующего синапса второго слоя, μ_{f_j} - двойственный сигнал от преобразователя j -го «потока», передающийся сумматору для раздачи на синапсы, $\mu_{w_{ij}}^{(1)}$ - двойственные сигналы соответствующих синапсов первого слоя.

4. Алгоритмическая часть

Обучение сети, состоящей из «потоков» производится в соответствии с алгоритмом, представленным на рис. 4.

Здесь H - значение оценки сети, накопленное по всему задачнику, ε - константа малости ошибки.

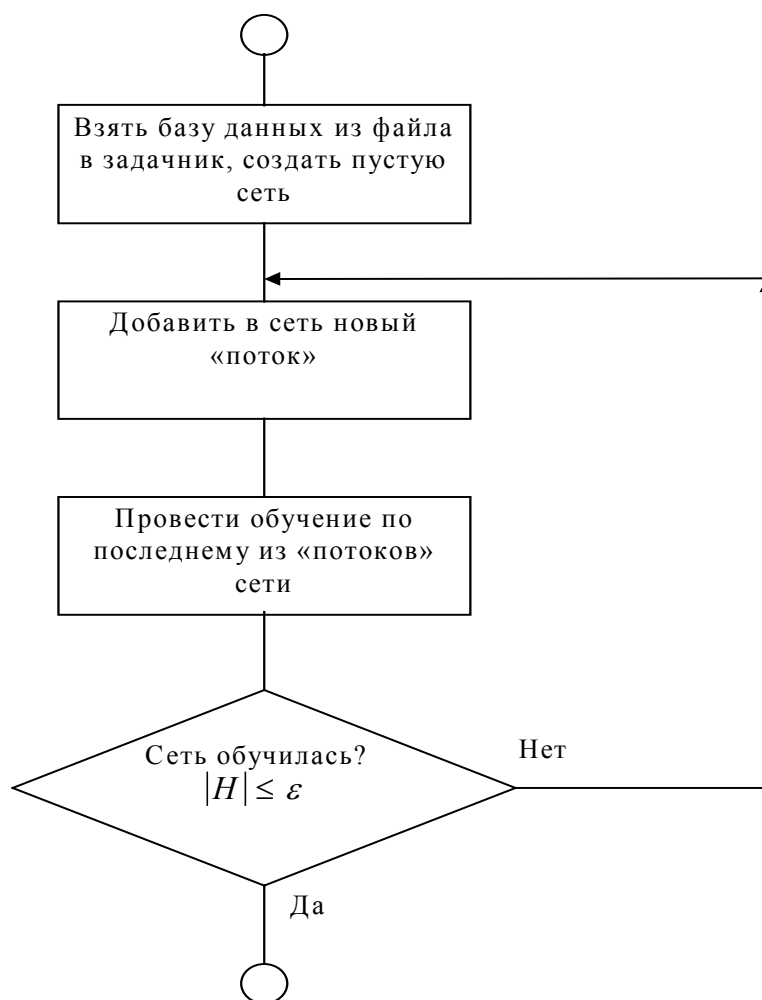


Рис. 4. Блок-схема обучения «поточной» сети

Для обучения каждого из потоков используется алгоритм, показанный на рис.5. Здесь $N_{\text{ф}}$ - общее число функций, по которому идет обучение, α_0 - карта параметров сети, h_0 - шаг оптимизации в начале цикла, s - градиент функции оценки H по обучаемым параметрам.

Используемое в алгоритме условие остановки формируется из двух подусловий, скомбинированных через «или»:

- Число шагов оптимизации, во время которых $\frac{|H_p - H_s|}{|H_p|} \leq \varepsilon$ превысило заранее заданную величину (при обучении использовалось значение $N_{sh}=15$), то есть сеть в данной конфигурации уже не может улучшить оценку;
- Достигнуто заданное значение функции оценки $|H_s| \leq \varepsilon$, то есть сеть уже обучена.

При обучении одного потока используются процедуры подбора шага оптимизации - *Increase* (поиск шага в сторону возрастания, блок-схему см. на рис. 6), *Decrease* (поиск шага в сторону убывания, блок-схему см. на рис. 7) и *Parabola* (поиск оптимального шага по формулам параболического поиска блок-схему см. на рис. 8).

В процедурах используются следующие обозначения:

$H(\dots)$ - функция оценки сети, накопленная по всему задатнику;

h_1, h_2, h_3 - различные значения величины шага оптимизации, используемые при ее подборе;

W - величина шага в вершину параболы, проходящей через точки $(h_1, H_1), (h_2, H_2), (h_3, H_3)$. Вычисляется по формуле:

$$W = \frac{h_3 + h_1}{2} - \frac{H_3 - H_1}{\frac{H_3 - H_2}{h_3 - h_2} - \frac{H_2 - H_1}{h_2 - h_1}}$$

H_1, H_2, H_3 - значения функции ошибки, соответствующие смещению обучаемых параметров по направлению градиента на величину шага h_1, h_2 и h_3 .

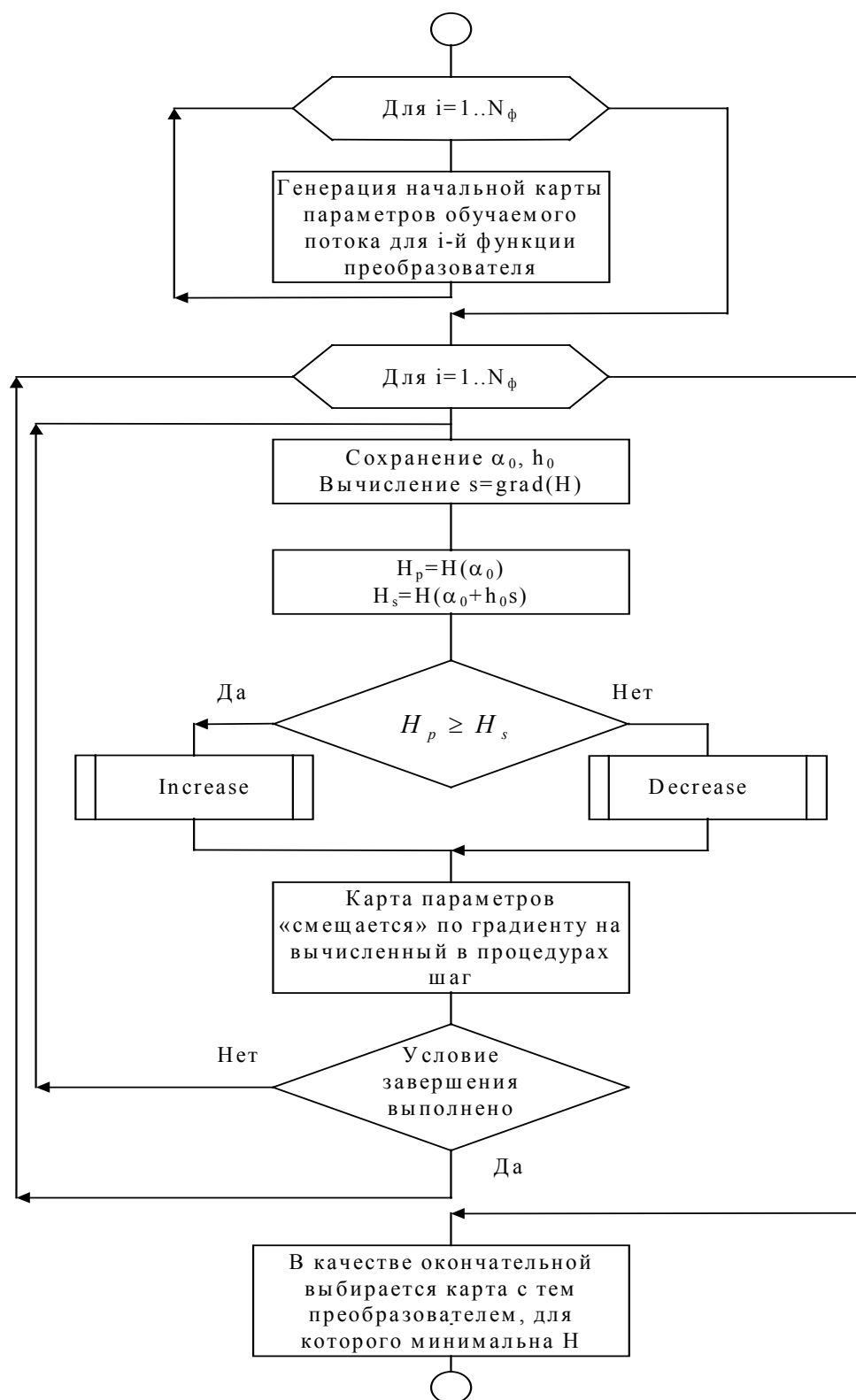


Рис. 5. Алгоритм обучения одного «потока»

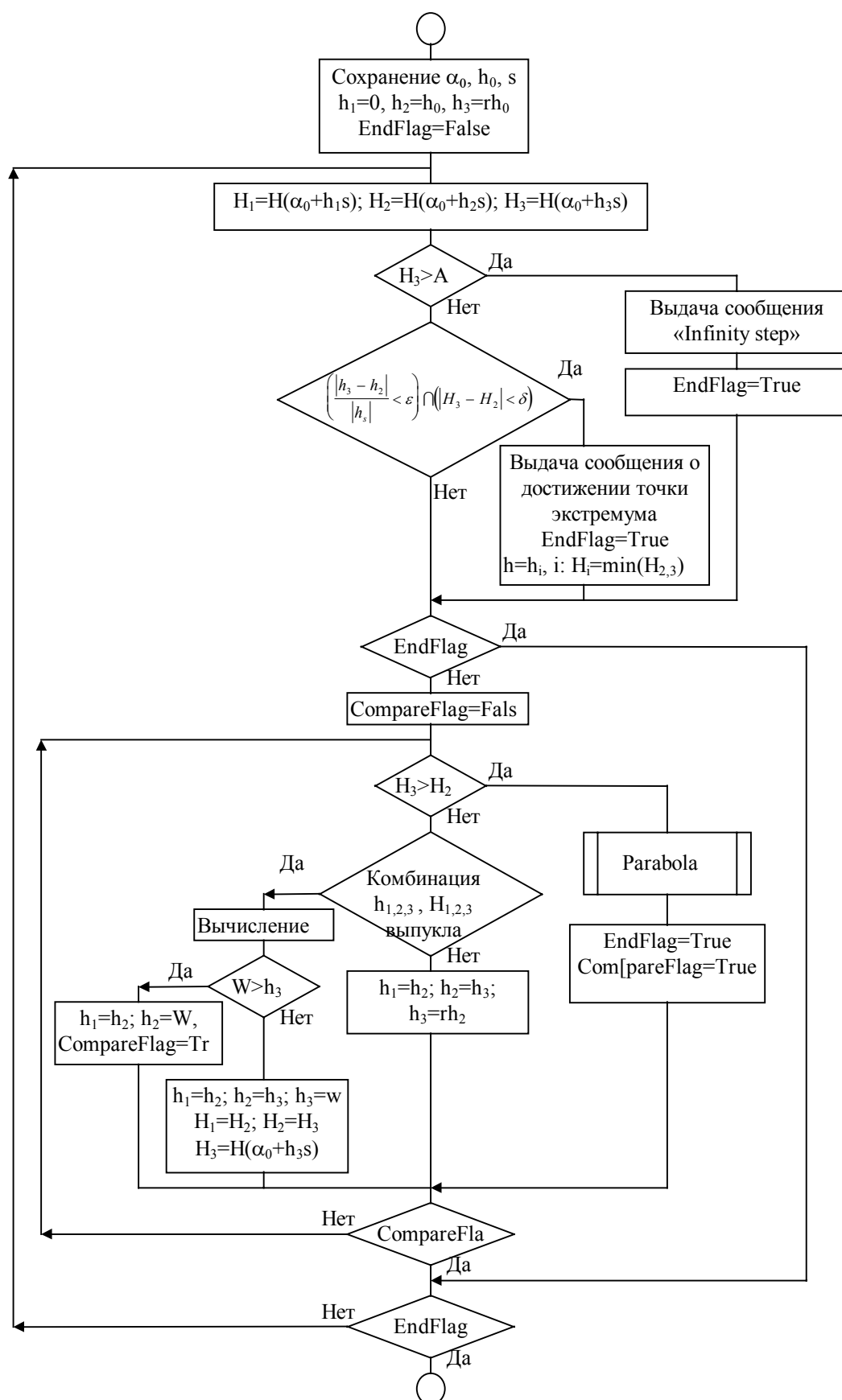


Рис. 6. Блок-схема процедуры Increase

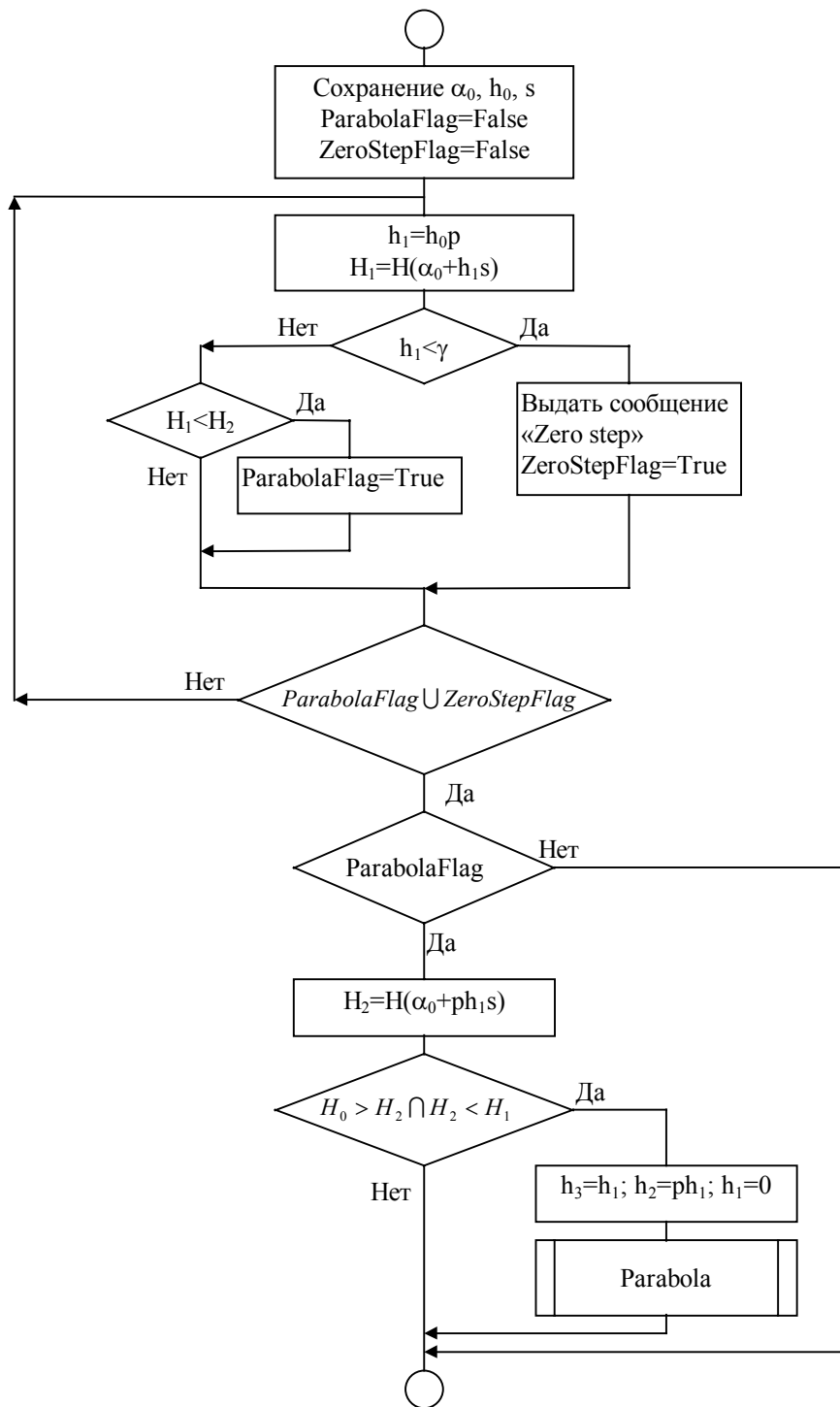


Рис. 7. Блок-схема процедуры Decrease

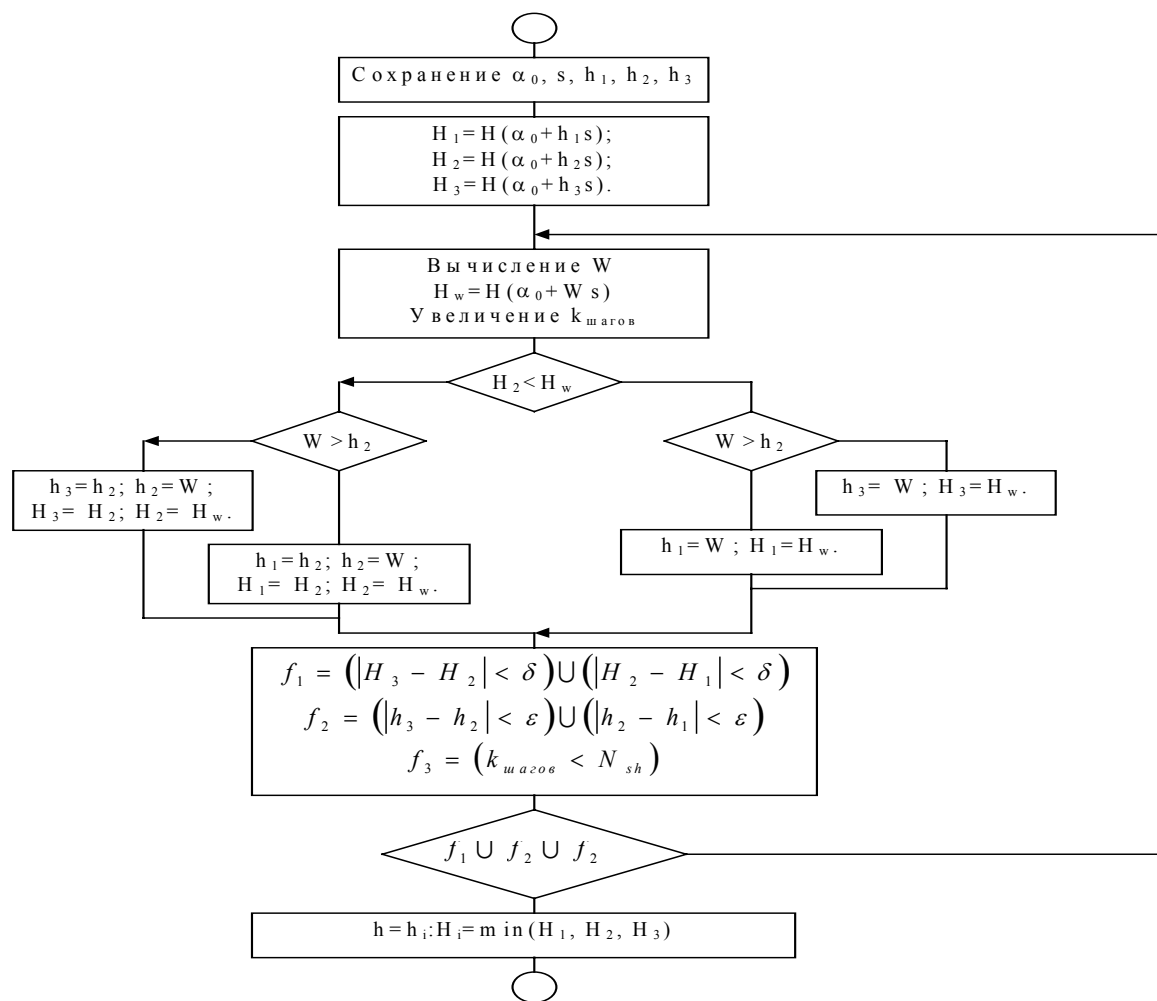


Рис. 8. Процедура Parabola

Условие выпуклости комбинации $h_{1,2,3}$, $H_{1,2,3}$ определяется формулой

$$H_2 < \frac{h_3 - h_2}{h_3 - h_1} H_1 + \frac{h_2 - h_1}{h_3 - h_1} H_1$$

Если выражение истинно, то условие выпуклости выполнено.

Теперь, рассмотрев алгоритмы обучения сети, перейдем к описанию компонентов, структуры и функционирования сети.

5. Оценка информационной емкости нейронной сети при помощи выборочной константы Липшица

Условие остановки процесса пошагового исчерпания ошибки может основываться также на оценке полноты функции, заданной нейронной сетью. В случае, если число элементов сети задано (для каждого шага наращивания «поточной» это так) и значения ее параметров ограничены на определенном интервале (это условие выполняется наложением ограничений на параметры сети), данное условие можно сформулировать с использованием константы Липшица. Константа Липшица вектор-функции

$f(x)$ в области D определяется как $L_f = \sup_{x, y \in D, x \neq y} \frac{\|f(x) - f(y)\|}{\|x - y\|}$. Верхняя

грань может быть вычислена по области определения D . В качестве оценки расстояния используется евклидова норма.

Для суперпозиции вектор-функций $(\varphi \circ f)(x) = \varphi(f(x))$
 $L_{\varphi \circ f} = L_{\varphi} L_f$.

Для линейной комбинации функций $f = \sum_i \alpha_i f_i$ оценка константы Липшица $L_f \leq \sum_i |\alpha_i| L_{f_i}$.

Константа Липшица для адаптивного сумматора, работающего по формуле $\alpha_0 + (x, \alpha)$ имеет вид $L_{\Sigma} = \|\alpha\|$.

Тогда для стандартной комбинации, состоящей из матрицы входных синапсов, сумматора и преобразователя - нейрона с гладкой функцией активации $\varphi(y)$ $L_N \leq \|\alpha\| \max_y |\varphi'(y)|$.

Для прямой суммы вектор-функций $f(x) = f_1(x) \oplus f_2(x) \oplus \dots \oplus f_N(x)$ константа Липшица может быть оценена

как $L_f \leq \left[\sum_i L_{f_i}^2 \right]^{\frac{1}{2}}$.

Таким образом, для слоя нейронов с подбираемыми преобразователями $L_s \leq \left[\sum_i \|\alpha^{(i)}\|^2 \max_y |\varphi_i'(y)|^2 \right]^{\frac{1}{2}}$, где $\alpha^{(i)}$ - вектор весов

синапсов, приходящих на входной сумматор i -го нейрона, а $\varphi_i(y)$ - функция i -го преобразователя.

Если заменить всю область определения функций D на конечное множество (задачник), то условие, определяющее требуемый объем нейронной сети можно получить, сравнивая L_s с оценкой константы

$$L_{\text{выб}} \geq \sup_{x, y \in \{x^P\}, x \neq y} \frac{\|f(x) - f(y)\|}{\|x - y\|}.$$

$L_{\text{выб}}$ является нижней оценкой константы Липшица аппроксимируемой функции. Нейросеть может реализовать данную функцию только в том случае, когда $L_s > L_{\text{выб}}$.

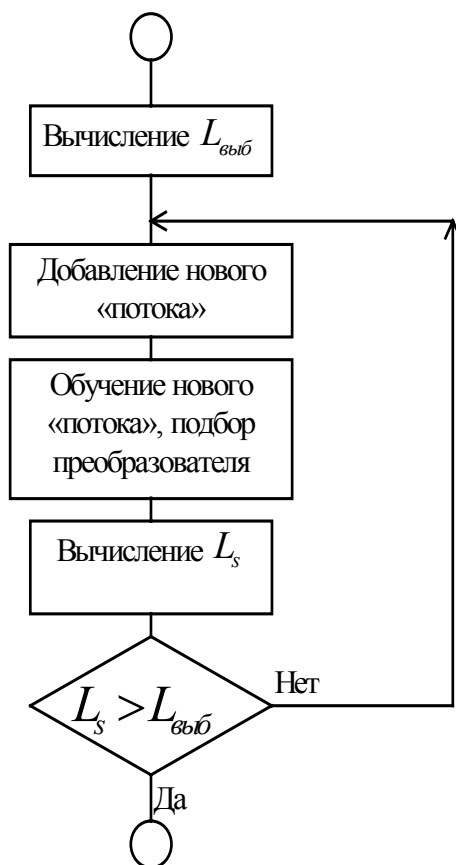


Рис. 9. Алгоритм поэтапного исчерпания ошибки до выполнения условия полноты аппроксимирующей функции.

Исходя из данного условия, можно сформулировать алгоритм обучения сети с поэтапным исчерпанием ошибки так, как это показано в блок-схеме на рис. 9

6. Соглашение о терминологии

Изложение вопросов, связанных со структурой и функционированием сети, тесно связано с программной реализацией задачи. Поэтому по тексту будут встречаться ссылки на реальные структуры программы.

Для описания компонентов сети был использован аппарат объектно-ориентированного программирования [1], реализованный в среде разработки программ *Borland Delphi Developer v.3.0.* (см. [2], [3], [4], [5], [6], [7])

Базовым понятием в языке программирования *Object Pascal*, встроенном в *Delphi*, является **класс** - особый вид записи, включающий поля и методы.

Экземпляр класса называется **объектом**.

Понятие **поле** для объекта аналогично понятию поля для записи.

Метод - процедура или функция, описанная внутри класса и предназначенная для операции над его полями.

7. Компоненты сети

Традиционный состав элементов сети включает в себя следующие элементы: нейрон, синапс, сумматор. Кроме того, в число типовых включены входной и выходной элементы.

На рис. 9 показаны схематические изображения элементов сети, которые далее будут использованы в схемах, описывающих структуру и функционирование программной модели.

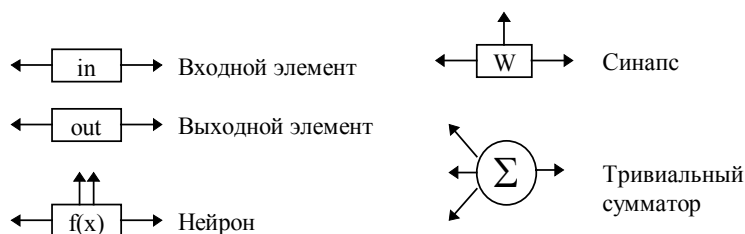


Рис. 9. Типовые элементы нейронной сети

8. Общий элемент сети

Базовым типом элементов, используемых для описания нейронной сети, является *общий элемент сети* - класс, инкапсулирующий основные свойства и методы, характерные для всех компонентов сети. Название этого класса в программной реализации - *TNetPiece*.

Объекты данного класса включают в себя следующие поля:

NextPiece - указатель на следующий элемент сети;

PriorPiece - указатель на предыдущий элемент сети;

ForwOut - значение сигнала, передающегося элементом вперед при прямом функционировании;

BackOut - значение сигнала, передающегося элементом назад при обратном функционировании.

Набор методов включает в себя:

Create - описание создания объекта;

Destroy - действия при разрушении (удалении) объекта;

ForwardTact - действия элемента во время такта прямого функционирования;

BackwardTact - действия элемента во время такта обратного функционирования;

При описании методов ***ForwardTact*** и ***BackwardTact*** они были оставлены пустыми, так как функционирование конкретных элементов сети существенно различно.

Однако введение этих методов имеет достаточно глубокий смысл, поскольку класс ***TNetPiece*** является предком всех прочих классов, описывающих элементы сети, и наличие типовых процедур прямого и обратного функционирования позволяет использовать такие свойства модели объектно-ориентированного программирования как наследование свойств и методов и полиморфизм. Подробнее этот тезис будет раскрыт ниже.

9. Вход сети

Для связи сети с задачиком и передачи используются объекты класса ***TNetInput*** - входной элемент сети.

Данный класс является потомком *TNetPiece*, и поэтому наследует его набор полей и методов этого класса, а кроме того добавлено поле *SourceSignal*, которое содержит номер поля задачника, с которого данный вход сети забирает значение.

Методы *ForwardTact* и *BackwardTact* перекрыты, то есть их код заменен на тот, который соответствует назначению входного элемента.

Метод *ForwardTact* выполняет передачу значения из соответствующего данному элементу поля задачника на выходной сигнал элемента, поле *ForwOut*.

Метод *BackwardTact* передает двойственный сигнал следующего элемента на свой двойственный сигнал (поле *BackOut*).

10. Выход сети

Выходной элемент сети описывает класс *TNetOutput*, также являющийся потомком *TNetPiece*.

В методах *ForwardTact* и *BackwardTact* заложены действия элемента при прямом и обратном тактах функционирования.

Метод *ForwardTact* выполняет передачу сигнала от выхода предыдущего на выход данного элемента, кроме того в поле **H** заносится значение ошибки сети при вычислении функции **Y**.

Метод *BackwardTact* передает на обратный выход элемента (поле *BackOut*) значение двойственного сигнала. Двойственный сигнал μ_H представляет собой производную функции ошибки по выходному сигналу сети:

$$\mu_H = \frac{\partial H}{Y_{сети}} = 2(Y_{сети} - F_{np}),$$

где $Y_{сети}$ - аппроксимированное значение функции, выдаваемое сетью, F_{np} - значение аппроксимируемой функции в данном примере.

11. Синапс сети

Для описания синапсов сети используются объекты класса *TNetSynapse*. Как наследник класса *TNetPiece* он наследует все его поля и методы. Помимо этого в список полей включены *Alpha* - параметр,

представляющий собой вес синапса, и **MuAlpha** - сигнал, двойственный весу синапса.

На такте прямого функционирования метод **ForwardTact** снимает выходной сигнал предыдущего элемента, умножает его на вес синапса и передает на выходной сигнал данного объекта (поле **ForwOut**).

На такте обратного функционирования метод **BackwardTact** передает в поле **BackOut** двойственный сигнал синапса, который может быть вычислен по следующей формуле:

$$\mu_w = Y_{back} \frac{dW}{dx_w} = Y_{back} \alpha_w,$$

где μ_w - двойственный сигнал, передаваемый синапсом, W - функция преобразования в синапсе, x_w - сигнал, поступающий в синапс от предыдущего элемента на такте прямого функционирования, Y_{back} - входной двойственный сигнал, поступающий в синапс от следующего элемента на такте обратного функционирования, α_w - вес синапса.

Кроме того, на обратном такте вычисляется сигнал, двойственный α_w и представляющий собой значение частной производной функции ошибки по этому параметру:

$$\mu_\alpha = Y_{back} \frac{dW}{d\alpha_w} = Y_{back} x_w,$$

где μ_α - сигнал, двойственный α_w .

Для значений α_w , μ_α в классе **TNetSynapse** предусмотрены поля **Alpha** и **MuAlpha**.

12. Тривиальный сумматор

Программной моделью тривиального сумматора является класс **TSummator**.

Помимо полей, унаследованных от класса - предка **TNetPiece**, **TSummator** имеет в своей структуре **PriorPieces**, которое, в отличие от стандартного поля **PriorPiece** является не указателем на предыдущий элемент, а списком указателей на набор таких элементов.

Метод *ForwardTact* осуществляет суммирование выходных сигналов элементов из списка *PriorPieces* и помещает полученный результат в поле *ForwOut*.

На такте обратного функционирования происходит передача двойственного сигнала следующего элемента на двойственный сигнал сумматора *BackOut*.

13. Нейрон

В данной работе под термином «нейрон» подразумевается нелинейный преобразователь вида

$$y = f(x_{вх}, \alpha_n, \alpha_c),$$

где y - выходной сигнал преобразователя, $x_{вх}$ - входной сигнал, α_n - параметр преобразователя, α_c - так называемый «параметр спонтанной активности».

Нейрон описывается в программной модели классом *TNeuron*, выходной сигнал на такте прямого функционирования заносится в поле *ForwOut*.

Обучаемыми в нейроне являются оба параметра - α_n и α_c , поэтому в классе *TNeuron* помимо полей *Alpha* и *AlphaS*, в которых хранятся значения соответствующих параметров, предусмотрены *MuAlpha* и *MuAlphaS*, в которых помещаются значения двойственных им сигналов.

Помимо этого в поле *BackOut* заносится сигнал, двойственный входному. Кроме того, объект класса *TNeuron* характеризуется еще и полем *FunctionType*, представляющим собой номер используемой функции преобразователя в списке функций, используемых при оптимизации.

Вычисление двойственных сигналов в нейронах производится в общем случае по формулам:

$$\mu_\alpha = Y_{back} \frac{df}{d\alpha},$$

где α - параметр, для которого вычисляется двойственный сигнал, μ_α - сам двойственный сигнал.

Список выражений для применяемого в данной работе набора стандартных функций с их производными по основным параметрам приведен в таблице 1.

Таблица 1.

Набор функций нелинейного преобразователя

Ф-ция	$f(x_{ex}\alpha, \alpha_c)$	$\frac{df}{dx_{ex}}$	$\frac{df}{d\alpha_n}$	$\frac{df}{d\alpha_c}$
f_1	$\frac{x_{ex} + \alpha_c}{\alpha_n + x_{ex} + \alpha_c }$	$\frac{\alpha_n}{(\alpha_n + x_{ex} + \alpha_c)^2}$	$-\frac{x_{ex} + \alpha_c}{(\alpha_n + x_{ex} + \alpha_c)^2}$	$\frac{x_{ex} + \alpha_c}{(\alpha_n + x_{ex} + \alpha_c)^2}$
f_2	$arctg(\alpha_n x_{ex} + \alpha_c)$	$\frac{\alpha_n}{1 + (\alpha_n x_{ex} + \alpha_c)^2}$	$\frac{x_{ex}}{1 + (\alpha_n x_{ex} + \alpha_c)^2}$	$\frac{1}{1 + (\alpha_n x_{ex} + \alpha_c)^2}$
f_3	$\sin(\alpha_n x_{ex} + \alpha_c)$	$\alpha_n \cos(\alpha_n x_{ex} + \alpha_c)$	$x_{ex} \cos(\alpha_n x_{ex} + \alpha_c)$	$\cos(\alpha_n x_{ex} + \alpha_c)$

14. Поток сети

Фрагмент сети, состоящий из слоя входных синапсов, сумматора, нейрона и выходного синапса и названный «поток» представлен в программной модели классом **TNetStream**.

Помимо обычных для потомка класса **TNetPiece** полей **NextPiece**, **PriorPiece**, **ForwOut** и **BackOut** данный класс включает в себя **FirstLayer** - список синапсов первого слоя, **Summator** - объект класса **TSummator**, реализующего тривиальный сумматор, **Neuron** - объект класса **TNeuron**, реализующего нелинейный преобразователь - нейрон, **SecondLayer** - объект класса **TSynapse**, описывающий выходной синапс потока.

Прямой такт потока, описанный методом **ForwardTact**, происходит следующим образом:

Перебираются элементы списка **FirstLayer**, для каждого из которых вызывается его собственный метод **ForwardTact**, затем происходит «срабатывание» (вызов этого же метода) для объектов **Summator**, **Neuron** и **SecondLayer**. Затем выходной сигнал объекта заносится в поле **ForwOut** - выходной сигнал всего потока.

Такт обратного функционирования потока, который описан в методе **BackwardTact**, включает в себя следующие действия:

Последовательный вызов собственного метода *BackwardTact* для объектов *SecondLayer*, *Neuron* и *Summator*, затем перебор элементов списка *FirstLayer* и вызов метода *BackwardTact* для каждого синапса, входящего в него.

Структура связей между элементами потока представлена на рис 10.

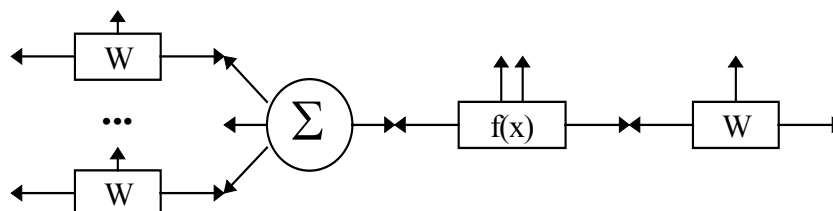


Рис. 10. Структура связей типовых элементов в потоке сети

15. Скомпонованная полутораслойная поточная сеть

Целая сеть представлена классом *TNet*. Данный класс также является потомком класса *TNetPiece* - общего элемента сети, что позволяет, используя свойство полиморфизма объектов, достаточно легко компоновать из отдельных фрагментов нейросети более сложной структуры.

Кроме полей, унаследованных от *TNetPiece*, сеть характеризуется также полями *Inputs* - список входных элементов, *Streams* - список потоков сети, *SecondLayer* - сумматор входного слоя, *Answer* - выходной элемент сети.

Для описания такта прямого функционирования сети используется, так же как в других элементах сети, метод *ForwardTact*. Методом выполняются следующие действия:

Перебор списка входных элементов и вызов метода *ForwardTact* для каждого из них, перебор и прямое функционирование каждого из элементов списка потоков, и затем - для объектов *SecondLayer* и *Answer*.

Метод *BackwardTact*, описывающий обратное функционирование сети, задается следующей последовательностью действий:

Обратное функционирование элементов *Answer*, *SecondLayer*, затем перебор и выполнение методов *BackwardTact* для элементов списков *Streams* и *Inputs*.

Структура связей между стандартными элементами полностью скомпонованной поточной нейросети показана на рис. 11.

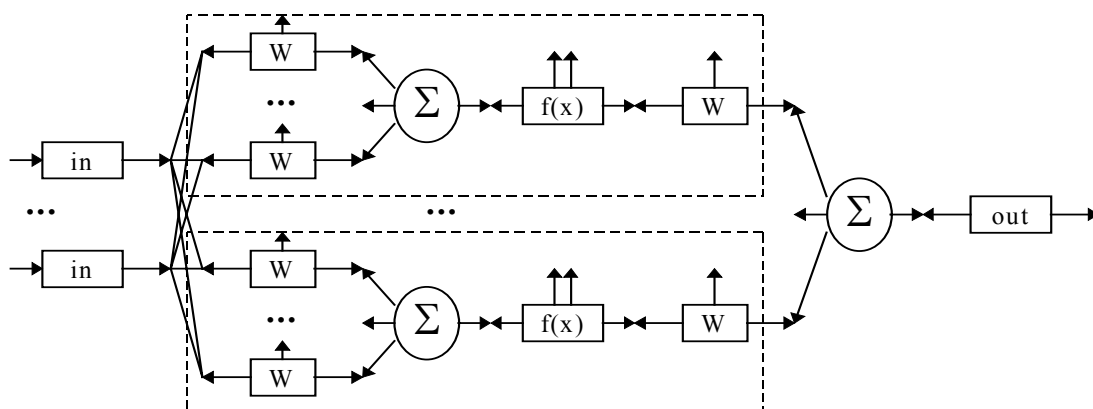


Рис. 11. Структура связи типовых элементов в поточной сети

Схема, приведенная на рис. 11, хорошо иллюстрирует преимущества применения объектно-ориентированного подхода к моделированию нейронных сетей.

Свойство полиморфизма объектов позволяет составлять список предыдущих элементов, например, сумматора, как из простых синапсов, так и из более сложных структур - потоков. Для этого требуется лишь аккуратное выполнение иерархии объектов, описывающих элементы сети.

В перспективе, с появлением параллельных трансляторов, объектно-ориентированный подход за счет свойства инкапсуляции объектов позволит легко перейти к моделированию нейросетей в параллельных системах.

Работа выполнена при финансовой поддержке ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.).

ЛИТЕРАТУРА

1. Буч Г. Объектно-ориентированное программирование с примерами применения. - М.: Конкорд, 1992. - с. 36.
2. Дантенманн Дж., Мишел Дж., Тейлор Д. Программирование в среде Delphi. - Киев: DiaSoft, 1995.
3. Дарахвелидзе П., Марков Е. Delphi - среда визуального программирования. - Санкт-Петербург: BHV, 1996.
4. Килверт Ч. Энциклопедия Delphi 2. Киев: DiaSoft, 1996.

5. Миллер Т., Пауэл Д. И др. Использование Delphi 3. - Киев: Диалектика, 1997 - 768 с.
6. Орлик С. Секреты Delphi на примерах. - М.: Бином, 1996 - 316 с.
7. Оузер Д. Освой самостоятельно Delphi 2. - М.: Бином, 1997 - 624 с.
8. Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики / РАН. Сибирское отделение. - Новосибирск, 1998. - Т.1, №1. - С.11-24.

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ПРИ РЕШЕНИИ АНАЛИТИЧЕСКИХ ЗАДАЧ В ГИС

А.А. Питенко

*Институт вычислительного моделирования СО РАН,
660036, Красноярск-36, ИВМ СО РАН,
E-mail: pitty@cc.krascience.rssi.ru*

Описан подход к использованию нейросетевых технологий для решения аналитических задач в ГИС. Дано изложение некоторых из них и показаны области применения. Представлен макет программной системы.

Нарастающие информационные потоки в современном обществе, разнообразие информационных технологий, повышение сложности решаемых на компьютере задач увеличивают нагрузку на пользователя этих технологий и ставят задачу переноса проблемы выбора и принятия решений с человека на ЭВМ. Одним из путей решения этой задачи является применение аналитических систем, которые могут быть составной частью ГИС [4,5,6].

Можно выделить несколько групп задач, требующих применения таких систем в ГИС:

- обработка видеоизображений;
- преобразование растровых изображений в векторные графические модели;
- обработка картографической информации;
- обработка разнородной информации;
- построение моделей объектов или местности;
- анализ моделей ГИС;
- получение новых знаний;
- получение решений на основе геоинформации.

ГИС являются хорошей средой для внедрения методов искусственного интеллекта и экспертных систем [6]. Это вызвано, с одной

стороны, разнообразием и сложностью данных в ГИС, с другой — наличием большого числа аналитических задач при использовании ГИС.

Электронную карту в ГИС можно рассматривать как многокомпонентную модель реальности. Основными целями ее создания являются:

- графическая коммуникация пространственных отношений и распределений;
- улучшение возможности анализа, обработки и отображения геоинформационных данных;
- визуальное отображение цифровых моделей явлений, невидимых для человеческого глаза;
- автоматизация отображения и картографического анализа в системах управления; исследование объектов, явлений и процессов с учетом динамики их развития и возможного использования;
- получение аналитических решений в графическом виде в режимах реального и разделенного времени.

Основой визуального представления данных при помощи ГИС-технологий служит так называемая графическая среда. Основу графической среды и соответственно визуализации базы данных ГИС составляют векторные и растровые модели.

В общем случае модели пространственных (координатных) данных могут иметь векторное или растровое (ячеистое) представление, содержать или не содержать топологические характеристики. Этот подход позволяет классифицировать модели по трем типам:

- растровая модель;
- векторная нетопологическая модель;
- векторная топологическая модель.

Все эти модели взаимно преобразуемы. Тем не менее, при получении каждой из них необходимо учитывать их особенности. В ГИС форме представления координатных данных соответствуют два основных подкласса моделей – векторные и растровые (ячеистые или мозаичные). Возможен класс моделей, которые содержат характеристики как векторов, так и мозаик. Они называются гибридными моделями.

В дальнейшем под терминами решетка, мозаика, элемент растра будем понимать одно и то же. Основу такой классификации составляет атомарная единица (пространства), содержащая представления площадей, линий и точек.

Между векторными и растровыми изображениями имеется различие, характерное именно для ГИС. Растровые изображения отображают поля данных, т.е. носят полевой характер. Векторные изображения в ГИС, как правило, отображают геоинформационные объекты, т.е. носят объектный характер.

Рассмотрим подробнее растровые модели данных, которые ближе касаются нашей основной задачи, нежели чем векторные. Напомним, что модель данных представляет собой отображение непрерывных последовательностей реального мира в набор дискретных объектов.

В растровых моделях дискретизация осуществляется наиболее простым способом - весь объект (исследуемая территория) отображается в пространственные ячейки, образующие регулярную сеть. При этом каждой ячейке растровой модели соответствует одинаковый по размерам, но разный по характеристикам (цвет, плотность) участок поверхности объекта. В ячейке модели содержится одно значение, усредняющее характеристику участка поверхности объекта. В теории обработки изображений эта процедура известна под названием пикселизация или растеризация.

Если векторная модель дает информацию о том, где расположен тот или иной объект, то растровая - информацию о том, что расположено в той или иной точке территории. Это определяет основное назначение растровых моделей - непрерывное отображение поверхности.

В растровых моделях в качестве атомарной модели используют двумерный элемент – пиксель (ячейка). Упорядоченная совокупность атомарных моделей образует растр, который, в свою очередь, является моделью карты или геообъекта.

Растровые модели имеют следующие достоинства:

- растр не требует предварительного знакомства с явлениями, данные собираются с равномерно расположенной сети точек, что позволяет в дальнейшем на основе статистических методов обработки получать объективные характеристики исследуемых объектов. Благодаря этому растровые модели могут

использоваться для изучения новых явлений, о которых не накоплен материал. В силу простоты этот способ получил наибольшее распространение;

- растровые данные проще для обработки по параллельным алгоритмам и этим обеспечивают более высокое быстроедействие по сравнению с векторными;
- некоторые задачи, например создание буферной зоны, много проще решать в растровом виде;
- многие растровые модели позволяют вводить векторные данные, в то время как обратная процедура весьма затруднительна для векторных моделей;
- процессы растеризации много проще алгоритмически, чем процессы векторизации, которые зачастую требуют экспертных решений.

Данные для анализа могут быть получены из векторных слоев, отражающих поля тематических или/и временных характеристик, растеризацией и записаны в таблицу или напрямую занесены туда из отчетов. Таблица, содержащая атрибуты объектов, называется таблицей атрибутов. В таблице каждому объекту соответствует строка таблицы, каждому тематическому признаку – столбец таблицы. Каждая клетка таблицы отражает значение определенного признака для определенного объекта.

В общем случае ввод информации для задач ГИС осуществляется комплексно: по данным дистанционного зондирования, со снимков спутников, аэроснимков, по материалам дешифрирования снимков, полевым измерениям, по информации с карт.

Далеко не все ГИС снабжены возможностями специализированного анализа, например, геологического. Связано это с тем, что четкой схемы проведения таких работ, не существует и организации, занимающиеся ими, предпочитают производить анализ по собственным методикам и правилам. Работа со специфическими данными специфическим образом является характерной чертой этого типа анализа. Кроме того, взгляды на приемы его проведения могут меняться с течением времени. Поэтому такие возможности в ГИС представляются средствами создания приложений самими пользователями. Сложность состоит в том, для каждой специализированной области возникает необходимость создавать

отдельное приложение к ГИС и часто даже свою методику обработки. Это не всегда возможно и всегда дорого.

Нейронные сети претендуют на то, чтобы стать универсальным аппаратом, решающим разные специфические задачи из разных проблемных областей в ГИС. Такая универсальность обуславливается тем, что нейросети дают стандартный способ решения многих нестандартных задач [3]. И неважно, что специализированная программа решит лучше один класс задач. Важнее, что один нейроимитатор решит и эту задачу и другую и третью и не надо каждый раз создавать специализированные приложения для каждой специфической задачи.

Как правило, модули, реализующие специализированный анализ для разных проблемных областей, решают одинаковые качественные задачи. Перейдя от специфических частных аналитических задач к общему видению проблемы в целом, можно увидеть одно важное обстоятельство. А именно, большинство аналитических задач сводится к одной проблеме, которая легко формулируется, но сложно решается: к проблеме заполнения пропусков в таблице.

Учитывая то, что часто методика обработки неизвестна, с этой задачей справляются лучше всего нейронные сети, которые позволяют строить эмпирические зависимости без привлечения дополнительной информации.

Опишем некоторый круг задач, требующих решения в ГИС, для которых могут быть использованы нейросетевые технологии.

1. Основная задача, к которой, так или иначе, относятся остальные, описанные ниже, это построение слоя. Она означает заполнение его недостающих частей (или построение слоя полностью) по информации, имеющейся в других слоях, на основе нахождения некоторой функциональной зависимости между параметрами, полученными эмпирическим путем, и скрытыми теоретическими параметрами, определяющими сущностные характеристики каждой конкретной точки.

Даны слои качественных характеристик одной и той же территории. Слой, который необходимо восстановить, известен частично. При обучении нейросети используется только та информация из слоев, которая покрывает известные участки слоя с пробелами. Обучившись, нейросеть может распространить знания о зависимости между слоями на отсутствующие области карты. Получившиеся знания обладают

переносимостью за рамки данной территории. Все описанные ниже задачи можно рассматривать как частный случай данной.

Классификационные задачи. Поскольку при сборе информации для БД приходится иметь дело с результатами измерений, определим по этому показателю три типа задач классификации.

К задачам классификации первого типа относятся те, в которых исходные измерения требуется разделить на устойчивые группы. Их называют задачами классификации без учителя, кластеризации, таксономии, типизации. Этот тип классификации основан преимущественно на опытных данных.

Задачи классификации второго типа характеризуются тем, что исходные данные уже сгруппированы и требуется оценить их информативность (значимость) относительно совокупности известных эталонов. Такого рода задачи встречаются при распознавании образов, дешифрировании снимков и т.д.

Задачи классификации третьего типа – задачи разбиения. В них исходные измерения или их функции требуется разбить на устойчивые группы в зависимости от их величины (типичный пример - зонирование).

В ГИС задачи классификации первого типа возникают и решаются при разработке классификаторов, т.е. при организации информационной основы, задачи второго типа - при сборе первичных данных и при использовании ГИС для экспертных решений или оценок. Задачи классификации третьего типа возникают в приложениях ГИС для решения проблем в области экологии, землепользования, статистики и т.п.

2. Вторая решаемая задача – восстановление легенды. Классификация с учителем – генерация объектов слоя по заданным классификационным правилам. Правила задаются во время обучения нейросети и остаются скрытыми от пользователя. Пользователь имеет возможность задавать, по его мнению, полезные для классификации признаки, выбрав слои, участвующие в обучении. Типичная задача поиск полезных ископаемых по косвенным признакам. Эта задача решается на основе информации об уже разведанных месторождениях и полевых съемках косвенных признаков. Знания, полученные при обучении, переносимы на другую территорию с известными косвенными признаками.

3. Зонирование. Основное назначение функций этой группы состоит в построении новых объектов – зон до того на карте не существовавших,

т.е. участков территорий, однородных в смысле некоторого критерия или группы критериев. Границы зон могут либо совпадать с границами ранее существовавших объектов (задача определения "нарезки" избирательных округов по сетке квартального деления), либо строиться в результате различных видов моделирования (зоны экологического риска). Типичные задачи этого типа: выделение зон градостроительной ценности территорий, зон экологического риска, зонирование урбанизированных территорий по транспортной доступности, построение зон обслуживания поликлиник и т.д. Работа может производиться как с растровыми, так и с векторными изображениями.

В сущности, зонирование это – классификация без учителя. Задан набор объектов, каждому объекту сопоставлен вектор значений признаков (строка таблицы). Требуется разбить эти объекты на классы эквивалентности.

Отнесение объекта к классу проводится путем его сравнения с типичными элементами разных классов и выбора ближайшего. Простейшая мера близости объектов – квадрат евклидова расстояния между векторами значений их признаков (чем меньше расстояние, тем ближе объекты). Соответствующее определение признаков типичного объекта – среднее арифметическое значение признаков по выборке, представляющей класс. Другая мера близости, естественно возникающая при обработке сигналов, изображений и т.п. – квадрат коэффициента корреляции (чем он больше, тем ближе объекты). Возможны и иные варианты – все зависит от задачи. Для каждого нового объекта нужно выполнить два действия:

- 1) найти класс, к которому он принадлежит;
- 2) использовать новую информацию, полученную об этом объекте, для исправления (коррекции) правил классификации.

В результате классификации как бы появляются новые имена и правила их присвоения.

4. Создание моделей поверхностей. Это и модели построения изолинейных изображений по регулярным и нерегулярным точкам и модели трехмерной визуализации, например, построение панорамы города в аксонометрической или иной проекции. Расчет производится по содержащимся в базах данных численным характеристикам. Моделироваться могут, как изображения действительного рельефа или

непрерывного поля, современного или с учетом динамических изменений, так и воображаемые поверхности, построенные по одному или нескольким показателям, например, поверхность цен на землю, плотность дорожной сети или населения и т.п.

5. Временной анализ растровых изображений. В качестве таких изображений в ГИС обычно выступают снимки или растеризованные векторные изображения. Преимущество снимков – в их современности и достоверности, поэтому часто встречающийся вид анализа в этой группе – временной. Сравниваются и ищутся различия между снимками различной давности, таким образом, оценивается динамика произошедших изменений. Не менее часто анализируются пространственные взаимосвязи двух или нескольких явлений.

Анализ временных рядов содержит комплекс задач, которые сводятся к построению функций по конечным наборам значений и заполнению пробелов в таблицах. Временные ряды представляют собой специальный вид таблиц и заслуживают отдельного рассмотрения. Для каждого типа объектов выделяется набор постоянных признаков (констант) и множество свойств, меняющихся со временем (переменных признаков). Предполагается, что в любой момент времени для каждого объекта существуют свои значения переменных признаков. Вот, например, три задачи, специфичные для обработки временных рядов:

а) определение констант (всех или части) по известным значениям переменных в разные моменты времени;

б) предсказание значений части переменных в некоторый момент времени по известным значениям констант, переменных в нескольких предшествовавших моментах времени и части переменных в текущий момент;

в) определение объема данных о прошлом, достаточных для предсказания будущего на конкретное время и с заданной точностью.

Обычная задача при временном анализе – получение прогноза. Легко заметить, что решение такой задачи мало чем отличается от решения задачи по восполнению пробелов в слое на основе информации, заключенной в других слоях. Единственное концептуальное отличие состоит в том, что слои вместо качественных признаков содержат количественные, т.е. изменение во времени одного и того же слоя. Взяв за основу известные слои одной и той же территории в количестве $N-1$ (без

последнего), а в качестве восстанавливаемого слой с номером N производим обучение нейросети. Для прогноза подаем на вход нейросети слои в количестве $N-1$ (без первого), тем самым на выходе получаем прогнозируемый слой $N+1$.

6. Анализ значимости. Как уже было сказано, основной задачей является восполнение пробелов в данных и решается она применительно к данной области построением слоя по слою (или нескольким слоям). При этом исследуется вопрос, какие из входных сигналов являются доминирующими, (значимыми) при принятии нейросетью решения, а какие нет. Другими словами, насколько каждый слой участвующий в построении влияет на восполнение пробелов. Такая информация дает знание, например, о том какие признаки можно убрать из рассмотрения, а какие оставить.

Значимость по слою складывается из значимости его точек. Благодаря такой информации можно видеть, какие области из слоев участвующих в качестве входов были значимы при построении, таким образом, мы получаем представление о территориальном распределении значимости. Вербальную оценку этой информации трудно переоценить.

Для обучения нейронной сети создается задачник (обучающее множество), состоящий из примеров вида "вход–правильный выход" [2]. Вход – вектор сигналов, предъявляемый нейросети, выход – вектор сигналов, который нейросеть должна выдать после функционирования. По паре "требуемый выход – полученный выход" можно построить функцию оценки. Чаще всего используется метод наименьших квадратов (МНК). Для ускорения обучения применяют специализированные оценки, например, МНК с люфтом. Это позволяет прекращать процесс обучения, когда достигнута удовлетворяющая пользователя точность.

Обучением называется процесс минимизации функции оценки. Он выполняется с привлечением градиентных методов оптимизации и состоит в поиске таких значений параметров, при которых нейросеть выдает правильный вектор выходных сигналов.

Нейросеть состоит из нелинейных преобразователей сигнала, связанных между собой посредством адаптивных связей, линейно преобразующих проходящий через них сигнал. Такие элементы называются нейронами. В нашем случае нейросеть состоит из Паде-нейронов.

Паде–нейрон состоит из двух адаптивных сумматоров и нелинейного элемента с двумя входами, вычисляющего частное от своих входных сигналов. Получая на входе n -мерный вектор x , Паде–нейрон выдает на выходе отношение $(x, \alpha)/(x, \beta)$, имеет $2n$ настраиваемых параметров – координат векторов α, β . Название происходит от аппроксимации Паде – метода рациональной аппроксимации функций.

С выбранным количеством слоев и нейронов в слое создается слоистая нейросеть. После выбора метода обучения нейросеть запускается на обучение. Нейросеть считается обученной после достижения заданного (малого) значения функции оценки.

Рассмотрим факторы, от которых зависит успешность обучения нейронной сети правильному решению задачи. В общем случае сеть должна быть достаточно гибкой, чтобы научиться правильно решать все примеры обучающей выборки. Поэтому в нейронной сети должно быть достаточное количество нейронов и связей.

На основании обучающей выборки сложно определить количество слоев и нейронов, необходимых сети для правильного решения задачи. Поэтому обычно обучают сеть со структурой, предлагаемой программой-нейроимитатором по умолчанию, если сеть не может обучиться (обучается с недостаточной точностью или совсем не обучается), ее размер увеличивают. На практике при решении разнообразных задач практически не встречается ситуации, когда требуется нейросеть с более чем сотней нейронов – обычно хватает нескольких десятков нейронов и даже меньшего числа.

Встречаются ситуации, когда нейросеть плохо обучается из-за противоречивой обучающей выборки и увеличение размера нейронной сети мало помогает. Это бывает в тех случаях, когда в обучающей выборке присутствуют задачи с одинаковыми условиями, но разными ответами (одинаковыми входными векторами данных, но разными выходными). Появление таких конфликтных примеров может, например, означать недостаточность набора входных признаков, поскольку при расширении признакового пространства конфликтным примерам могут соответствовать разные значения добавляемого признака и критическая ситуация будет исчерпана. В любом случае пользователь должен решить эту проблему, например, исключением конфликтных примеров из задачника.

После обучения нейронной сети необходимо провести ее тестирование на тестовой выборке для определения точности решения не входивших в обучающую выборку задач. Точность правильного решения очень сильно зависит от репрезентативности обучающей выборки. Обычно при решении различных неформализованных задач в разных проблемных областях точность в 70-90% правильных ответов на тестовой выборке соответствует проценту правильных ответов при решении этих же задач специалистом-экспертом.

Разработано программное обеспечение для анализа географических данных на основе обучаемых нейронных сетей. Программа работает со слоями и картами созданными в системе ГеоГраф/*GeoDraw*[1].

Для аналитической обработки векторные слои растрятся. Результат записывается в таблицу. Запись таблицы соответствует точке, поле соответствует слою. Проблема визуализации результатов обработки, решается созданием и раскрашиванием полигональной сетки, где каждой точке после растривания соответствует квадрат сетки. Проблема противоречивости обучающей выборки решается линейной фильтрацией и/или исключением конфликтных примеров.

По данным, полученным после растривания и, возможно, фильтрации, для обучения нейронной сети создается задачник. С выбранным количеством слоев и нейронов в слое создается слоистая нейросеть. После выбора метода обучения нейросеть запускается на обучение. Нейросеть считается обученной после достижения заданного (малого) значения функции оценки.

Обученная нейросеть может распространять знания о зависимости между слоями на отсутствующие области карты. При тестировании нейросети по задачнику и примерам, не входящим в задачник, результат записывается в таблицу, после чего, полигональный слой ("сетка") может быть раскрашен в соответствии с полученными при тестировании значениями.

Также может быть исследован вопрос, какие из входных сигналов являются доминирующими (значимыми) при принятии нейросетью решения, а какие нет. Полученные показатели значимости можно использовать, например, при анализе вектора входных сигналов на избыточность. Значимость параметра зависит от точки карты, поэтому возможна раскраска "сетки" ("слоя значимости") для каждого параметра.

Благодаря такой информации можно видеть, какие области из слоев, участвующих в качестве входов, были значимы при построении слоя, таким образом, получается представление о территориальном распределении значимости.

Программа позволяет выполнять как элементарные операции с картой, так и традиционные операции над нейронными сетями (создание сети, ее обучение, тестирование), а также предобработку (растрирование, фильтрация) и визуализацию результатов.

ЛИТЕРАТУРА

1. ГеоГраф, GeoDraw – профессиональные отечественные ГИС для широкого круга пользователей // ГИС – обозрение. Зима, 1994. С.24 – 25.
2. Гилев С.Е., Коченов Д.А., Миркес Е.М., Россиев Д.А. Контрастирование, оценка значимости параметров, оптимизация их значений и их интерпретация в нейронных сетях. // Нейроинформатика и ее приложения: Материалы III Всероссийского семинара; КГТУ. Красноярск, 1995. – 229 с.
3. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996. – 276с.
4. Коновалова Н.В., Капралов Е.Г. Введение в ГИС. – М., 1997. – 160 с.
5. Хаксхолд В. Введение в городские географические информационные системы. New York: Oxford, 1991. – 321 с.
6. Цветков В.Я. Геоинформационные системы и технологии. – М.: Финансы и статистика, 1998. – 288 с.

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ДЛЯ ПРОВЕДЕНИЯ УЧЕБНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ

Л.А. Жуков

*Сибирский государственный технологический университет
(660049, Красноярск, пр. Мира, 82, СибГТУ, кафедра ИТ)*

Описан опыт работы автора по организации и проведению занятий по нейроинформатике в Сибирском государственном технологическом университете (СибГТУ). В работе приводится анализ творческих возможностей школьников старших классов и студентов для проведения исследований с использованием нейроинформационных технологий. Приведены результаты выполнения некоторых учебно-исследовательских работ. Показаны относительная простота и доступность нейроинформатики и нейроинформационных технологий для студентов и школьников старших классов.

1. Введение

Наличие у школьников и студентов знаний и умений по информатике стало одним из важнейших условий подготовки к профессиональной деятельности в любой области. В настоящее время происходит массовое распространение различных книг, пособий и учебников, работает большое количество различных курсов, предмет «Информатика» введен в государственный образовательный стандарт и в учебные планы большинства учебных заведений. Однако до сих пор существует большое количество различных проблем с изучением и обучением информатике - нехватка современных компьютеров, недостаток хороших учебных пособий, отсутствие лицензионного программного обеспечения и квалифицированных школьных учителей. Одна из причин указанного заключается в задержке начала развития информатики в СССР и последующем нашем технологическом отставании.

С конца 80-х годов во всем мире активно развивается новый раздел знания на стыке информационной науки и математики — нейроинформатика. Главные результаты, получаемые обществом от

нейроинформатики - решение различных трудноформализуемых прикладных задач. Для решения практических задач разрабатываются специальные математические методы, программное обеспечение. Многие доступны уже сейчас. Сразу же встает вопрос - кто в этом разбирается и где взять программное обеспечение?

Нужно сделать оговорку, что в данной работе не обсуждаются источники, достижения и перспективы самой науки, для этого достаточно сослаться на большой список литературы, изданной только в Красноярской школе нейроинформатики [1-14]. Этот список намеренно неполон, чтобы привести полный список работ, опубликованных на русском языке, требуется написать еще одну такую статью, а на английском языке публикуются многотомные сборники тезисов конференций и множество книг. Тем не менее, надо подчеркнуть некоторое отставание в уровне развития нейроинформатики, по крайней мере, по уровню известности и количеству прикладных работ.

В Красноярске с конца 80-х и особенно активно в 90-е годы создана Красноярская научная школа нейроинформатики, кузница научных и педагогических кадров. Начиная с 1993 года в Красноярске проходят ежегодные всероссийские семинары (конференции) по Нейроинформатике. В работе этих конференций участвуют не только ученые и специалисты, но и студенты и даже школьники, если им есть что сказать. Ведущий ученый в Красноярской научной школе нейроинформатики - д.ф.-м.н. А.Н. Горбань. В этой школе также работают д.м.н. Д.А. Россиев, к.ф.-м.н. В.А. Охонин, к.ф.-м.н. Е.М. Миркес, к.ф.-м.н. С.Е. Гилев и другие.

Для существования и развития Красноярской научной школы нейроинформатики, как и для любой другой школы, нужен постоянный приток школьников и студентов. Это способствует развитию и распространению новых идей и способов обработки информации, развитию информационных технологий. Нужны молодые энтузиасты - разработчики и пользователи нейропрограмм. Первым шагом стала организация и проведение в 1991 году 1-й Всесоюзной Олимпиады по нейроинформатике и нейрокомпьютерам. После этого несколько раз проводились Школы Нейроинформатики, параллельно с всероссийскими семинарами по Нейроинформатике, но участвовали в них студенты, аспиранты и молодые ученые.

2. Зимняя Политехническая Школа по Нейроинформатике

Появилась идея организовать такую школу для начинающих - школьников старших классов. Первый раз такая школа была организована в январе 1997 года в ходе проведения краевой Зимней Политехнической Школы (ЗПШ) в Сибирском государственном технологическом университете. После проведения этой школы на всероссийском семинаре-конференции по Нейроинформатике в октябре 1997 года было представлено два доклада школьников о решении прикладных задач [14].

В этом году такая школа была проведена снова. На этот раз в проведении ЗПШ по Нейроинформатике участвовал Горбань А.Н. - прочитал часть лекций. В работе ЗПШ приняли участие школьники из семи разных школ города Красноярска и одной школы города Канска, всего участвовало 27 школьников. Интерес желающих вырос неимоверно.

Для проведения занятий было установлено специальное программное обеспечение - нейроимитатор MultiNeuron [7], разработанный группой НейроКомп в составе Горбань А.Н., Гилев С.Е., Коченов Д.А., Миркес Е.М., Россиев Д.А. Были использованы учебно-методические пособия, разработанные на ВЦ КНЦ СО РАН (Институт вычислительного моделирования РАН) и в СибГТУ [10-12].

Разработана специальная методика решения различных задач с использованием нейроимитаторов [12]. Методика была разработана специально для занятий со школьниками и студентами и ранее уже неоднократно использовалась.

Был заранее подготовлен список заданий. Для того, чтобы заинтересовать школьников, предлагались различные интересные, в том числе - веселые и смешные темы. Уровень сложности заданий - повышенный, однако они были подобраны так, чтобы можно было получить результаты за время работы Школы. В таблице 1 приведены решаемые школьниками задачи. Большая часть задач была приготовлена заранее, некоторые были сформулированы в ходе обсуждения различных идей.

Все задачи так или иначе связаны с изучением различных предметных областей, следовательно с приобретением новых знаний, систематизацией уже имеющихся. Все эти работы, в отличие от “взрослых” работ прикладного характера, пока имеют небольшую

практическую значимость. Тем не менее, эти задачи способствуют подготовке и развитию мышления школьников для дальнейшего применения нейроиформатики на практике. (А может быть какая-либо из задач окажется нужна уже в недалеком будущем?)

После выполнения работ был проведен семинар с докладами участников. Каждый доклад был оценен с учетом мнения всех слушателей. По результатам конкурса работ были определены призовые места и вручены ценные подарки за призовые места и поощрительные призы за старание.

Результаты работы Школы были отмечены в Координационном центре по Красноярскому краю Российской научно-социальной программы для молодежи и школьников “Шаг в будущее”. Одна работа заняла 1-е место в краевом смотре-конкурсе работ по программе “Шаг в будущее”, было получено приглашение на конференцию по программе “Шаг в будущее” в Москве, сделан доклад на краевой конференции НОУ.

Таблица 1

Темы практических работ участников краевой Зимней Политехнической Школы по Нейроиформатике 1998 года (СибГТУ)

N	Тема
1	Разделение на Живое и Неживое (от песчинки и вируса до слона и Солнца) с помощью нейронных сетей
2	Нейросетевое определение уровня развития государства по косвенным географическим факторам
3	Определение принадлежности языка к группе славянских языков с помощью нейронных сетей
4	Нейросетевая классификация компьютерных игр
5	Нейросетевое определение пола человека по косвенным факторам
6	Различение новых и старых русских фильмов (до и после 1990) с помощью нейронных сетей
7	Нейросетевое различение семейств собачьих и волчьих
8	Различение типов цветов (роза и хризантема) с помощью нейронных сетей
9	Разделение классов вертолетов на боевые и транспортно-пассажирские с помощью нейросетей

В апреле 1998 года школьниками из ЗПШ-98 было сделано два доклада на конференции в Сибирском государственном технологическом университете [15, 16]. Доклады вызвали бурный интерес аудитории из преподавателей и студентов. В мае этого же года сделано еще два доклада на конференции “Информатика” в Красноярском государственном техническом университете [17].

3. Задачи

Предлагается список возможных тем для «детских» учебно-исследовательских задач. Здесь и ниже «детские» не надо понимать как легкие или легко решаемые. Некоторые из этих задач трудны из-за сложностей в постановке задачи, другие - из-за недостатка данных. Возможно, что в списке есть неразрешимые сегодня задачи, но узнать это можно только попытавшись решить их. «Детские» здесь указывает на повышенный уровень интереса со стороны школьников и студентов младших курсов. Некоторые из этих задач были ими самостоятельно придуманы.

Искусствоведческие:

1. Определить популярность художественной книги по различным параметрам - жанр, ситуация, год, автор, количество и типы героев, образность...
2. Определить степень привлекательности и/или популярности художественного фильма (кассовость) по различным параметрам - жанр, ситуация, год, киностудия, автор сценария, режиссер, количество и типы героев, наличие спец.эффектов...
3. Определить автора произведения искусства (живопись, литература и т.д.)

Психологические:

1. Определить тип характера и/или темперамента человека.
2. Определить взаимосвязь характера человека и его профессии
3. Определить взаимосвязь характера человека и его любимой компьютерной игры.

Маркетинговые:

1. Определить привлекательность газетной рекламы.
2. Определить привлекательность телевизионной рекламы.

Программистские:

1. Определить качество (учебной) программы по ее параметрам.
2. Разделить языки программирования на группы.

Астрономические:

1. Определить светимость переменной звезды в период максимума блеска по ее небесным координатам, периоду изменения, минимуму блеска.
2. Определить продолжительность существования искусственного космического объекта по параметрам орбиты и прочим данным.

Социологические:

1. Обучить нейросеть различать, «новый русский» или нет.
2. Обучить нейросеть различать национальности по поведению, например, “русский” или нет.
3. Обучить нейросеть различать возраст респондента по косвенным данным. Задача может быть сформулирована для предиктора или классификатора.
4. Определить тип ученого по набору важных личностных параметров социально-биографически-географически-экономического плана. (Вопрос - ученый или нет).

Биологические:

1. Обучить нейросеть различать кошек и собак (семейство кошачьих и семейство псовых) независимо от вида (рысь, оцелот, пума и другие) и породы.
2. Обучить нейросеть различать собак, волков и шакалов - вид и породу.
3. Обучить нейросеть различать породы кошек.
4. Обучить нейросеть различать породы собак.

И кроме этого - многие другие задачи.

4. Результаты

Многочисленные примеры решения подобных задач с помощью нейроинформатики опубликованы в различных работах, сборниках конференций и методических пособиях [6, 10, 13, 14 и другие]. Среди всех примеров решения “взрослых” задач выделяется направление - прикладная медицинская нейроинформатика. По этому направлению выполнено большое количество исследований разными группами авторов для разных областей медицины. Защищены несколько кандидатских и одна докторская

диссертация. Результаты исследований используются в Красноярской Государственной Медицинской Академии, 20-й больнице, Краевом онкологическом диспансере. В последнее время активно развивается приложение нейроинформатики для решения прикладных задач педагогики [14 и другие источники].

Для “детских” задач пока не выделились приоритетные области или области достижения наибольших успехов, да это и понятно. Для своих работ школьники выбирают темы по своим интересам, близкие и понятные. У каждого возраста - свои игрушки, «взрослые» задачи тоже выбираются на свои, близкие и понятные темы. Тем не менее, можно сказать, что у школьников и студентов младших курсов вызывает больший интерес решение задач гуманитарного типа, задач, связанных с человеком и природой. Это соответствует одному из современных направлений развития общества, не реализованному, но планируемому - идее построить гуманное общество.

Можно привести примеры тем некоторых студенческих работ:

1. Нейросетевое разделение семейств псовых и кошачьих по косвенным параметрам;
 2. нейросетевая классификация минералов;
 3. нейросетевое различение групп пород собак;
 4. нейросетевое определение валентности химического элемента;
 5. нейросетевое прогнозирование плотности углеводородов;
 6. нейросетевое различение людей, занятых физической и умственной работой;
 7. нейросетевое определение зависимости цветения комнатных растений от морфологических признаков;
 8. определение автора музыкального произведения с помощью нейросетей;
 9. нейросетевое определение авторства детектива;
 10. стратификация общества с помощью нейросетей;
 11. прогнозирование гидрологической обстановки на реках;
 12. прогнозирование температуры воды в Енисее в районе города Красноярска;
 13. психологическая оценка больных раком молочной железы, перенесших радикальную мастэктомию и расширенную секторальную резекцию.
- и другие.

Работы на указанные темы были представлены школьниками и студентами младших курсов на конференциях Нейроинформатика и других.

Несколько работ по определению авторства произведений искусства и литературы, позволяют сделать вывод о применимости использования нейросетей в качестве консультанта в случае неизвестного или спорного автора. В выполненных экспериментах авторство картин правильно определялось в 83%, музыкальных произведений - более чем в 70% и литературы - в 75%, в среднем [14].

Решалась также искусствоведческая задача определения типа культуры и страны создания художественного фильма по некоторым его характеристикам. Для выполнения работы были выбраны примеры художественных фильмов киностудий Советского Союза и США. По 20 параметрам обученные нейросети определяют правильно более 90% тестовых примеров [14].

Решалась задача определения взглядов философов и ученых на основной вопрос философии (материалистические или идеалистические) по некоторым из основных характеристик их жизни и творчества: материалист или идеалист (ответ), эпоха деятельности, происхождение (страна), материальное положение, реформатор или нет и другим параметрам, которые явно с материализмом не связаны. Показано, что нейросетевой подход позволяет решить поставленную так задачу, а самый значимый параметр - Реформаторство философа [14].

Решалась задача нейросетевой классификации государств по уровню экономического развития по 27 важнейших параметров - Тип воспроизводства населения, Импорт/экспорт нефти, Средняя продолжительность жизни, Плотность населения, Площадь территории и другим. Обученные нейросети показывают правильное определение уровня экономического развития государств для 60-70 % примеров. Были получены нейросети с правильным распознаванием для 90 % примеров. Наиболее значимые входные параметры - Доход на душу населения, Уровни развития автомобильного и трубопроводного транспортов, наименее значимые - Наличие выхода к морю, Население, Территория [15].

Выполнен ряд исследований человека: определение связи возраста с другими характеристиками, определение темперамента человека и определение связи работы человека и его личностных качеств [14].

Проведена работа по стратификации нового общества на слои по новым признакам. В работе показано, что наличие в собственности автомобиля не связано с самоопределением в один из слоев общества. Наличие места жительства и его тип (собственная квартира, аренда, общежитие), а также слагаемые бюджета влияют на определение слоя [14].

Работа по прогнозированию замужества девушек в наиболее благоприятный детородный период (до 25 лет) позволяет определить набор факторов, наиболее способствующих замужеству, и в очередной раз открывает забытые истины, например, что непредсказуемость поведения девушки намного более сильное препятствие для брака, чем недостатки внешности [19].

Выполнено несколько работ по гидро- и метеорологии. Например, в работах по гидрологии показано, что по известным данным о состоянии рек бассейна можно с большой точностью определять состояние реки в выбранном месте. Одна из работ позволяет моделировать температуру Енисея в окрестностях Красноярска с точностью менее 1 градуса [18-19].

Интересно, что при решении последней из списка задач удалось разработать очень простую нейросетевую модель для прогноза температуры в Енисее. После этого на реальных данных гидрологической службы было обучено несколько нейросетей для прогнозирования температуры. Было достигнуто максимальное отклонение при обучении не более $1,2^{\circ}\text{C}$. Средняя правильность прогноза - 80 % на тестовой выборке из 180 примеров. По результатам работы сделан доклад на экологической конференции [18].

Студентами и школьниками выполняются и проектные задачи. В одном из исследований проведен сравнительный анализ методов распознавания образов, выбран математический аппарат, разработан набор инструментальных средств и программа для распознавания бинарных изображений. Цель работы - создание инструментария и системы распознавания символов по их начертанию, т.е. по их растровому изображению, на базе полученного инструментария реализовать программу распознавания арабских чисел заданных монохромным изображением размером 10×15 единиц. Получена программа для распознавания образов [17].

Студент 2 курса СибГТУ написал свой нейроимитатор, хорошо решающий “стандартные” задачи, например, задачу выбора президента [19]. Работа в этом направлении продолжается.

5. Перспективы

В ближайшем будущем предполагается продолжать проведение учебно-исследовательских работ по нейроинформатике. Не обязательно всем студентам стать в дальнейшем учеными или преподавателями, но очень хорошо, что все познакомятся с относительно новым и бурно развивающимся направлением исследований, примут участие в небольшой учебно-исследовательской работе.

Для проведения таких работ в СибГТУ имеется комплект необходимых программных средств, в том числе лицензионные (нейроимитаторы MultiNeuron, NeuroPro, утилита-предобработчик Predmake и другие), методическое обеспечение и опыт таких работ.

Существует возможность для изучения Информатики и Нейроинформатики не только для детей, но и для взрослых, преподавателей или специалистов, на Факультете Повышения Квалификации СибГТУ.

Вывод

Показана относительная простота и доступность нейроинформатики и нейроинформационных технологий для студентов и школьников старших классов, возможность проведения их силами учебно-исследовательских работ и получения заслуживающих внимания результатов.

Работа выполнена при финансовой поддержке ФЦП "ИНТЕГРАЦИЯ" (проект № 68, напр. 2.1.).

ЛИТЕРАТУРА

1. Барцев С.И., Охонин В.А. Адаптивные сети обработки информации // Препринт ИФ СО АН СССР, Красноярск, 1986, №59Б. 20 с.
2. Охонин В.А. Вариационный принцип в теории адаптивных сетей. Препринт ИФ СО АН СССР, Красноярск, 1987, №61Б. 18 с.

3. Барцев С.И., Гилев С.Е., Охонин В.А. Принцип двойственности в организации адаптивных систем обработки информации // Динамика химических и биологических систем. Новосибирск, Наука, 1989. С.6-55.
4. Gorban A.N. Neurocomputing in Siberia // Adv.Model.& Analysis, B, 1992, V. 24, No 2, pp. 21-28.
5. Горбань А.Н. Обучение нейронных сетей.- М.: изд. USSR-USA JV "ParaGraph", 1990.- 160 с. (англ. перевод - Gorban A.N. Training Neural Networks // AMSE Transaction, Scientific Siberian, A, 1993, V. 6. Neurocomputing, pp.1-134).
6. Нейроинформатика и ее приложения. Материалы 3 Всероссийского семинара 6-8 октября 1995. Красноярск: изд. КГТУ, 1995. 229с.
7. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.
8. Горбань А.Н., Фриденберг В.И. Новая игрушка человечества // Мир ПК, 1993, № 9.
9. Горбань А.Н. Нейрокомпьютер или аналоговый ренессанс // МИР ПК, 1994, № 10.
10. Нейропрограммы. Учебное пособие: В двух частях / Под. ред. А.Н.Горбаня. Красноярск: изд. КГТУ, 1994. Часть 1.- 137 с. Часть 2.- 123 с.
11. Горбань А.Н., Россиев Д.А., Коченов Д.А. Применение самообучающихся нейросетевых программ. Раздел 1. Введение в нейропрограммы. Учебно-методическое пособие. Красноярск: СТИ, 1994. - 24с.
12. Горбань А.Н., Гилев С.Е., Коченов Д.А., Россиев Д.А., Миркес Е.М., Жуков Л.А. Применение самообучающихся нейросетевых программ. Раздел 2. MultiNeuron. Красноярск: СибГТУ, 1997.
13. Нейроинформатика и ее приложения: Тезисы докладов 4 Всероссийского семинара, 5-7 октября 1996г./Под ред. А.Н.Горбаня. Красноярск: КГТУ, 1996.- 122с.
14. Нейроинформатика и ее приложения: Тезисы докладов 5 Всероссийского семинара, 3-5 октября 1997г./Под ред. А.Н.Горбаня. Красноярск: КГТУ, 1997.- 139с.
15. Жижаева В.Н., Лукьянов А.А., Григорьев Р.В. и др. Нейросетевое определение уровня экономического развития государства по косвенным географическим факторам // Секция Информационные технологии конференции СибГТУ, 1998 (в печати).

16. Онисько А.В., Мелюков П.А., Замиралов А.А., Гомонов Д.О. и др. Нейросетевая классификация компьютерных игр // Секция Информационные технологии конференции СибГТУ, 1998 (в печати).
17. Черепанов П.В. и др. Создание системы распознавания изображений // Информатика и информационные технологии: Тез. докл. межвузовской научной конференции студентов, аспирантов и молодых ученых. Красноярск: КГТУ, 1998 (в печати)
18. Щукина А.В., Забусова Н.В., Кудрявцева И.П., Раймер Т.П. и др. Использование нейроимитатора MultiNeuron для прогнозирования температуры воды в Енисее в районе Красноярска // 3-я Межвузовская конференция по экологическому образованию. СибГТУ, 1998 (в печати)
19. Нейроинформатика и ее приложения: Тезисы докладов 6 Всероссийского семинара, 2-4 октября 1998г./Под ред. А.Н.Горбаня. Красноярск: КГТУ, 1998 (в печати)
20. Дыхно Ю.А., Мусаева Н.Э., Оводова А.А. и др. Психологическая оценка больных раком молочной железы, перенесших радикальную мастэктомию и расширенную секторальную резекцию // Сборник тезисов 62 студ.научно-практ. конф. Красноярск: КГМА, 1998
21. Дунин-Барковский В.Л. Введение. Нейрокибернетика, нейроинформатика, нейрокомпьютеры // Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин, Е.М. Миркес и др. Новосибирск: Наука, 1998.
22. Россиев Д.А. Глава 5. Медицинская нейроинформатика // Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес и др. Новосибирск: Наука, 1998.

ПРОИЗВОДСТВО ПОЛУЭМПИРИЧЕСКИХ ЗНАНИЙ ИЗ ТАБЛИЦ ДАННЫХ С ПОМОЩЬЮ ОБУЧАЕМЫХ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

*В.Г. Царегородцев*¹

Институт вычислительного моделирования СО РАН

660036, Красноярск-36, ИВМ СО РАН,

E-mail: tsar@cc.krascience.rssi.ru

Рассматриваются возможности извлечения полуэмпирических теорий из таблиц данных при помощи нейронных сетей. Вводится понятие логически прозрачной нейронной сети, формируется набор критериев логической прозрачности, строится набор правил по приведению нейронной сети к логически прозрачному виду. Предлагается технология вербализации логически прозрачной сети - перевода на естественный язык алгоритма решения сетью задачи. Приводится пример решения классической задачи предсказания результатов выборов президента США.

1. Введение

В настоящее время широкое распространение получили искусственные обучаемые нейронные сети. С их помощью эффективно решаются неформализованные задачи классификации и прогнозирования в различных проблемных областях. Обучаясь на некоторой таблице данных, нейросеть формирует навык предсказания или классификации и в дальнейшем может решать задачи с высокой точностью. Однако, обучившись решать задачу, нейронная сеть сформировала навык, логическая структура которого не понятна пользователю.

К настоящему моменту разработано большое число высокоэффективных процедур для упрощения (контрастирования) нейронной сети – снижения избыточности последней [1-3]. В [2,5]

¹ Автор выражает благодарность А.Н.Горбаню, С.Е.Гилеву и Е.М.Миркесу за полезные обсуждения и возможность использования в данной работе большого числа принадлежащих им оригинальных идей.

предложена гипотеза о возможности использования контрастирования для такого упрощения нейронной сети, после которого структура сети станет логически прозрачной. Логически прозрачной назовем сеть, по структуре которой легко можно сформировать понятный пользователю алгоритм решения задачи. Очевидно, что простое упрощение сети не приведет последнюю к логически прозрачному виду – надо строить специальные правила исключения из сети сигналов и элементов, чтобы получить сеть со структурой, близкой к нашему пониманию логической прозрачности.

В настоящей статье описывается процесс порождения явных и понятных алгоритмов решения неформализованных задач (получение полуэмпирических теорий) с помощью обучаемых нейронных сетей. Формализуются критерии логической прозрачности нейронной сети и строится методология приведения нейросети к логически прозрачному виду. Приведены основные идеи, на основе которых можно строить алгоритмы извлечения знаний и нейросетевые программы средства качественно нового уровня, ориентированные на специалистов-когнитологов.

Для изучения нейротехнологий можно порекомендовать работы [1-3,6]. Краткое изложение основных аспектов перечисленных работ приведено в статье [8], включенной в настоящий сборник. Термины и обозначения настоящей статьи соответствуют этим работам.

2. Логически прозрачные нейронные сети

После того, как сеть обучилась решению некоторой задачи, обычно не ясно, какие же правила используются сетью для получения правильных ответов. Сеть произвела из задачника (таблицы данных) скрытое знание – сформировала некоторый навык предсказания или классификации, но логическая структура этого навыка осталась непонятной пользователю. Такое обстоятельство очень сильно разочаровывает: поскольку на основе нейронных сетей часто решаются неформализованные задачи (алгоритм решения которых неизвестен), то даже успешное обучение сети дает нам просто некоторое вычислительное устройство, не приближая к пониманию процесса решения задачи. Действительно, можно описать функционирование сети как пересылку, суммирование и преобразование большого числа сигналов, но понятийная интерпретация крайне

затруднена. Этот факт называется логической непрозрачностью нейронной сети.

Введем фундаментальные критерии логической прозрачности нейронной сети. Эти критерии естественно и просто вводятся по аналогии с критериями простоты и понятности рассуждений, выполняемых человеком.

2.1. Архитектура логически прозрачных сетей

Рассмотрим архитектуры нейронных сетей и выделим из них наиболее логически прозрачные. Поскольку обычно архитектура выбирается в процессе создания нейронной сети и контрастирование в подавляющем большинстве случаев не приводит к изменению архитектуры, то появляется возможность зафиксировать класс изначально логически прозрачных нейросетевых архитектур и в дальнейшем оперировать именно с ними начиная с момента создания сети и заканчивая приведением сети к логически прозрачному виду.

Критерии логической прозрачности нейросетевых архитектур не должны обязательно совпадать с критериями наиболее простой технической или программной реализуемости сети – это все-же разные задачи. Поэтому представляется возможным использование в качестве логически прозрачных сетей со слоистой и только слоистой архитектурой. Слоистые сети наиболее соответствуют нашему представлению о естественном пути решения задачи. А именно, в слоистых сетях происходит некоторое распределение набора входных признаков по нейронам первого слоя, нейроны первого слоя на основании поступивших сигналов генерируют признаки следующего уровня иерархии и т.д. Наличие каких-либо циклов в архитектуре сети затрудняет процесс отслеживания движения признаков по графу вычислений.

Нейросети любой архитектуры могут быть представлены в виде слоистых сетей. Единственное ограничение – в сети не должно быть бесконечных циклов. Иными словами, в сетях с циклами (полносвязных и родственных с ними) процесс функционирования сети должен обрываться после некоторого фиксированного числа тактов. Действительно, имеются сети полносвязной архитектуры, для которых наперед неизвестно число циклов функционирования. Функционирование таких сетей обрывается после установления, например, некоторого равновесного состояния на

выходе, когда новый цикл существенно не меняет значений выходных сигналов сети. При этом для каждого примера, подаваемого сети, число циклов функционирования является различным. Такие сети мало пригодны для приведения к логически прозрачному виду.

Итак, нейросети слоистой архитектуры зафиксированы в качестве логически прозрачных. Возможно приведение слоистой архитектуры к более логически прозрачному виду, а именно, преобразование слоистой сети в монотонную. Монотонность не обязательно применима только к слоистым архитектурам. Под монотонными сетями понимаются такие сети, для каждого из настраиваемых параметров и сигналов (входных и внутренних) которых можно сказать, как изменится (увеличится либо уменьшится) выходной сигнал сети при уменьшении или увеличении значения этого параметра или сигнала. В монотонной сети выделены группы возбуждающих и тормозящих параметров, увеличение значений которых соответственно увеличивает (для возбуждающих параметров) и уменьшает (для тормозящих) выходные сигналы сети, а уменьшение значения этих параметров действует наоборот.

Каждый слой монотонной сети, кроме выходного, разбит на два блока: возбуждающий (В) и тормозящий (Т). Связи между блоками разделяются на два типа: возбуждающие (+) и тормозящие (-). На нелинейные преобразователи нейронов накладывается требование монотонной зависимости выходного сигнала от параметров и входных сигналов. Взаимодействие возбуждающих и тормозящих блоков через возбуждающие и тормозящие связи представлено на рис. 1. От возбуждающего блока к возбуждающему и от тормозящего блока к тормозящему ведут возбуждающие связи, а от возбуждающего к тормозящему и от тормозящего к возбуждающему ведут тормозящие связи. На выходной блок сигналы от возбуждающего блока приходят через возбуждающие связи, а от тормозящего – через тормозящие. При упрощении монотонной сети накладывается дополнительное требование минимизации числа общих входных сигналов возбуждающих и тормозящих слоев. В идеале такое требование приводит к разбиению множества входных сигналов на два непересекающихся подмножества возбуждающих и тормозящих сигналов [1,2].

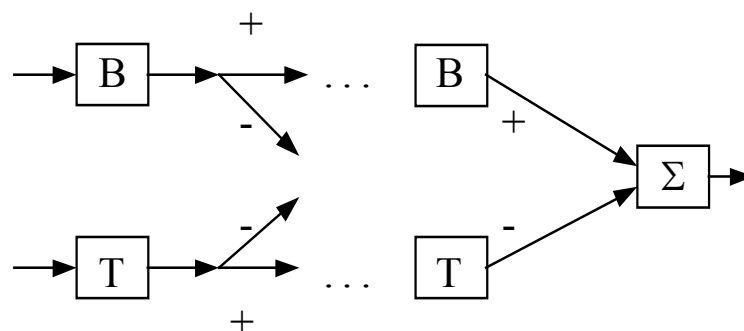


Рис. 1. Монотонная сеть

2.2. Критерии логической прозрачности нейронной сети

Зафиксировав требования к архитектуре, определим набор критериев, следование которым делает сеть более логически прозрачной. Некоторые из критериев были приведены в [2,5]. На основе этих критериев можно строить оценки приближения к идеальному логически прозрачному состоянию сети. Эти критерии могут ранжироваться пользователями по степени важности (в зависимости от субъективного понимания логической прозрачности) и пользователи могут добавлять новые критерии логической прозрачности в эту группу критериев. Перечислим эти критерии:

1. Чем меньше слоев нейронов в сети, тем сеть более логически прозрачна. Однако число слоев зафиксировано при создании сети. Этот критерий важен в тех ситуациях, когда число слоев сети избыточно и после контрастирования возникают отдельные нейроны (или слои нейронов), являющиеся просто передатчиками информации с предыдущего слоя на следующий. Такие нейроны имеют единственный вход (и могут иметь также и единственный выход). Эти нейроны должны по возможности заменяться линиями передачи информации (подробнее об этом – в разделе 3.3). Однако часто не удастся полностью избавиться от излишних слоев и поэтому можно ввести окончательную формулировку первого критерия логической прозрачности – *чем меньше нейронов в каждом из имеющихся путей прохождения сигналов в сети от входа к выходу, тем сеть логически прозрачнее*. Такое утверждение основывается на том факте, что рассуждение понять тем проще, чем меньше длины цепочек рассуждений.
2. Уменьшение числа входных сигналов сети. Этот критерий отсеивает признаковое “шумовое поле”, оставляя минимально

необходимый для правильного решения задачи набор наиболее значимых входных признаков. Надо строить такие рассуждения, которые опираются на минимальное число посылок.

3. *Уменьшение числа нейронов в каждом слое сети.* Этот факт не требует пояснения – мы минимизируем число сигналов, генерируемых каждым слоем сети, что позволяет оставить только действительно значимые промежуточные сигналы (признаки). Надо не только опираться на минимальное число исходных посылок (п.2), но и генерировать минимальное число промежуточных сущностей. Методика контрастирования нейронов описана в главе 2 настоящей работы.
4. *Уменьшение числа приходящих на нейрон сигналов.* Это утверждение опирается на тот факт, что человек может одновременно оперировать с достаточно малым числом сущностей. Минимизируя число приходящих на нейрон сигналов, мы облегчаем пользователю задачу содержательного осмысления признака, генерируемого нейроном, и, может быть, помогаем в именовании этого признака. Однако желательна модификация этого критерия и введение критерия равномерной простоты сети – *на каждый нейрон сети должно приходить не более n сигналов, где n достаточно мало (2-3, но может задаваться и пользователем).* Ведь если рассуждение понять легче, когда оно опирается на малое число посылок, то сделаем так, чтобы все рассуждения были примерно одной степени сложности.
5. *Уменьшение общего числа синапсов в сети.* Критерий ликвидирует все излишние синапсы нейронной сети, которые могут оставаться после достижения предыдущих требований.
6. *Синапс, по которому передается сигнал, логически непрозрачнее неоднородного входа нейрона.* Действительно, неоднородный (пороговый) вход нейрона – это просто константа, в отличие от синапса, выполняющего умножение своего веса на величину поступающего на синапс сигнала. Поэтому в первую очередь из сети должны исключаться синапсы, а только потом – неоднородные входы адаптивных сумматоров. Таким образом, желательно при упрощении сети по предыдущим критериям

исключать только синапсы, а избыточные неоднородные входы удалять на отдельном этапе упрощения сети.

7. *Необходимо приведение значений настраиваемых параметров сети к конечному набору выделенных значений.* На диапазон изменения адаптивных параметров сети обычно наложены некоторые ограничения. Допустим, веса синапсов меняются при обучении в диапазоне $[-1, 1]$. Желательна бинаризация параметров сети – приведение весов синапсов к значениям -1 и 1 . Если такое жесткое требование не выполняется, то можно вводить более широкие наборы дискретных значений. Например, множество $\{-1, -0.5, 0.5, 1\}$ как самое первое расширение исходного набора бинаризованных весов.

2.3. Требования к нелинейности элементов

Вообще то, требование использования более простых нелинейных функций нейронов можно было включить отдельным пунктом (критерием) предыдущего раздела. Однако достаточно большой объем изложения потребовал формирования отдельного раздела.

Имеется теорема, показывающая, что с помощью нейронных сетей можно сколь угодно точно приблизить любую непрерывную функцию, используя *любую* нелинейность нейронов [4]. Хватает всего-навсего одной любой нелинейной функции. Однако такие широкие возможности для решения задачи приведения сети к логически прозрачному виду нам не нужны – нужно ограничиться теми нелинейными функциями, которые допускают простую интерпретацию. Наиболее проста из нелинейных единичная пороговая функция h (функция Хевисайда). Если сумма сигналов, приходящих на нейрон с такой единичной нелинейной функцией, меньше или равна 0, то далее можно не обращать внимания на выходной сигнал нейрона – он будет нулевым и не повлияет на выдаваемые нейронами следующего слоя значения. Если эта сумма входных сигналов больше нуля, то нейрон генерирует единичный выходной сигнал и здесь нам опять не важна величина сигнала, а важно то, что воздействие на нейрон превысило некоторый критический уровень и нейрон возбудился. Такие пороговые нейроны допускают наиболее простую интерпретацию.

С другой стороны, просто линейная функция еще проще... Поэтому имеются предельные случаи:

- “прозрачность” – линейная функция;
- простой порог – функция Хевисайда.

Однако для обучения нейронной сети по методу двойственности необходимо наличие непрерывной первой производной у нелинейной функции нейрона, что для порогового нейрона не достигается. Поэтому можно предложить путь, использующий некоторые непрерывно дифференцируемые функции и переходящий к пороговым тогда, когда дальнейшее обучение сети не нужно.

Рассмотрим одну из наиболее часто используемых нелинейных функций нейрона – сигмоидную функцию $f(A) = A/(c + |A|)$. Здесь c – настраиваемый параметр крутизны сигмоиды. Функция имеет непрерывную первую производную и принимает значения из диапазона $[-1, 1]$. Устремление c к ∞ дает более пологую сигмоиду, а устремление к 0 – сигмоиду, близкую к пороговой функции $\text{sgn}(A)$, однако имеющую третье стабильное состояние, нуль, при нулевом входном сигнале. Однако значение c – параметр крутизны сигмоидной функции – обычно устанавливается одинаковым для всех нейронов в момент создания сети и не включается в число обучаемых параметров, поскольку в большинстве случаев это приводит к ухудшению интер- и экстраполяционных способностей нейронной сети. Действительно, поскольку нейронная сеть первоначально избыточна, то ей обычно проще научиться решению задачи, приближая задачник набором функций, близких к дельта-функциям (а точнее, набором “колоколообразных” функций), чем строя регрессионные зависимости. При этом параметры крутизны при обучении быстро устремляются к нулю, ускоряя обучение сети, но снижая качество решения тестовой выборки.

Исключение из сети малозначимых входных сигналов и элементов снижает избыточность и гибкость нейронной сети и поэтому появляется возможность начинать обучать и параметры крутизны сигмоиды для еще большего снижения избыточности сети.

Итак, надо пытаться заменить сигмоидный элемент пороговым элементом. При этом выход нейрона будет принимать дискретные значения, что упростит пользователю понимание сети. В качестве пороговой функции лучше брать трехзначную пороговую функцию вида

$$f(A) = \begin{cases} -1, & \text{если } A < 0, \\ 0, & \text{если } A = 0, \\ 1 & \text{если } A > 0 \end{cases}$$

С другой стороны, не надо забывать о том, что сигмоиду можно приводить не к более крутой, а к более пологой функции. Тожественная функция $f(A)=A$ хоть и не является предельным случаем сигмоиды, но в окрестности нуля достаточно хорошо аппроксимирует сигмоиду при определенной крутизне последней. Поэтому можно пытаться заменять сигмоиду не только на порог, но и на линейную функцию. Подробно технология такой замены будет изложена в разделе 3.3.

3. Контрастирование нейронов

Описанные [1-3] процедуры вычисления показателей значимости входных сигналов сети и ее настраиваемых параметров позволяют удалять только наименее значимые входы и настраиваемые параметры (синапсы) нейронной сети. Однако нам может потребоваться и упрощение сети путем удаления у нее целого нейрона. Контрастирование отдельных нейронов сети выполняется при сокращении числа нейронов в слоях сети – ранее был введен такой критерий логической прозрачности.

Удаление у сети малозначимых входов и синапсов может привести к удалению у сети нейрона. Действительно, если у нейрона нет ни одного входного сигнала и удален неоднородный вход, то такой нейрон можно исключить из сети (поскольку он не влияет на выходной сигнал сети), удалив как этот нейрон, так и синапсы нейронов следующего слоя, принимающих сигнал от данного. И наоборот, если отконтрастированы все синапсы, по которым нейрон рассылал свой выходной сигнал, то можно удалить этот нейрон со всеми его входными синапсами и неоднородным входом.

Этот пример иллюстрирует важность наличия не только процедуры упрощения, но и процедуры логического анализа структуры сети для упрощения сети. Удаляя элементы, не оказывающие влияния на выходной сигнал сети, процедура логического анализа структуры сети ускоряет процесс упрощения. Однако такое контрастирование нейронов является

некоторым дополнительным результатом при контрастировании синапсов, но не самостоятельной процедурой.

Поэтому не будем уклоняться от поставленного в начале главы вопроса и введем процедуры определения значимости нейронов сети. Дальнейшее изложение предполагает знакомство читателя с алгоритмом двойственного функционирования и процедурами вычисления показателей значимости входных сигналов и синапсов сети [1-3,8]

Вспомним, что при двойственном функционировании сети мы можем получить градиент функции оценки не только по входным сигналам сети и ее настраиваемым параметрам, но и по всем промежуточным сигналам сети, т.е. сигналам, поступающим с синапса на сумматор, с сумматора – на нейрон, с нейрона – на синапсы нейронов другого слоя. Последнее, а именно, возможность получения градиента функции оценки по выходному сигналу нейрона и решает поставленную задачу. Градиент функции оценки используется при вычислении показателей значимости элементов и сигналов сети, поэтому будем использовать стандартные методы и не вводить специальных правил вычисления показателей значимости нейронов.

Представим, что у сети исчез первый слой нейронов, а нейроны второго слоя стали, соответственно, входными нейронами и принимают входные сигналы, являющиеся результатом обработки исходных входных сигналов нейронами бывшего первого слоя. Показатели значимости входных сигналов, вычисленные по известным формулам для новой сети, будут, соответственно, показателями значимости нейронов бывшего первого слоя. Конечно, не надо действительно менять архитектуру сети, надо просто по известным формулам (также, как вычисляется значимость входных сигналов) вычислить еще один набор показателей значимости. Аналогично вычисляются показатели значимости нейронов второго, третьего и последующих слоев. Такие показатели значимости оценивают степень влияния выходных сигналов нейронов на выходные сигналы сети.

Можно предложить альтернативный вариант, не связанный с проведением дополнительных вычислений. Примем в качестве показателя значимости нейрона сумму показателей значимости всех его синапсов (и неоднородного в том числе) и всех синапсов, по которым нейрон рассылает ответ. Этот путь оценивает суммарное влияние всех синапсов нейрона на изменение функции оценки. Чем сумма показателей

значимости синапсов меньше, тем нейрон менее значим для принятия решения.

4. Приведение нейронных сетей к логически прозрачному виду

Итак, имеется набор критериев, показывающих, как нужно модифицировать нейронную сеть для того, чтобы было проще прочесть сформированную сетью навик.

Нельзя сравнивать по степени логической прозрачности различные нейронные сети и нейросетевые архитектуры. Критерии логической прозрачности – это критерии простоты нейронной сети для простоты построения понятного пользователю описания процесса решения сетью задачи. Однако, может встретиться такая задача, где нейронная сеть, не имеющая логически прозрачной архитектуры, будет предпочтительнее (будет давать более понятное пользователю объяснение), чем сети, логически прозрачные по введенным критериям. Поэтому критерии логической прозрачности – это критерии получения *потенциально* более простого и понятного пользователю описания процесса решения сетью задачи. В связи с этим можно сравнивать по степени логической прозрачности только исходную нейронную сеть и все сети, порождаемые из исходной одношаговым упрощением. Критерии логической прозрачности дают нам просто наиболее оптимальные направления одношагового упрощения сети. Действительно, простое последовательное сокращение синапсов сети может дать в итоге сокращение и числа входных сигналов, и числа нейронов, но не гарантирует достижения максимально простого прочтения заложенного в сеть навыка.

Опишем приведение нейронной сети к логически прозрачному виду более формально.

4.1. Наложение ограничений на архитектуру нейросети

Имеются априорно задаваемые ограничения (слоистость, монотонность архитектуры, минимальность числа слоев), накладываемые на нейронную сеть. Эти ограничения нужно учесть на этапе создания нейронной сети.

Может быть, на этапе создания сети надо задавать большую избыточность сети, чем обычно (увеличивать число нейронов в слоях

сети). Хотя это и удлиннит процесс упрощения сети, но все лишние элементы все равно в конце концов будут удалены из сети. Однако, здесь выше вероятность достижения логической прозрачности – ведь из малоизбыточной нейросети сложно сделать логически прозрачную и тут может потребоваться повторное обучение и упрощение новой, более избыточной сети.

Возможно, оптимальным вариантом станет введение некоторой процедуры “роста” сети – при невозможности достижения некоторого критерия (например, равномерной простоты сети) в нужный слой сети вводятся один-два нейрона, сеть дообучается и снова упрощается. Такой вариант не будет требовать большой начальной избыточности сети.

4.2. Упрощение нейросети

После обучения нейронной сети начинается процесс пошагового упрощения с подучиванием. На этом этапе возникает набор критериев – линейных шкал типа "лучше-хуже", задающих направления одношагового изменения архитектуры сети в сторону логической прозрачности. Эти критерии ранжированы пользователем по степени значимости. Следуя этим критериям, нейронная сеть приводится к максимально логически прозрачному виду.

Базовые критерии перечислены в разделе 1.2.

Возможно, по некоторым критериям не нужно упрощать сеть “до упора”, лучше оставить некоторый запас сигналов и элементов. Если же оставить минимальную сеть, то может увеличиться процент неправильных ответов на тестовой выборке. Если в дальнейшем предполагается доучивание логически прозрачной сети новым примерам, то такое упрощение сети “до упора” также снизит способность сети к дальнейшему освоению новых данных.

Таким образом, надо находить некоторый компромисс. Если нужно получить некоторый понятный полуэмпирический алгоритм решения задачи, то сеть нужно упрощать до максимально возможной простоты. А если необходима сеть с возможностями дальнейшей адаптации, то придется оставить в сети некоторое количество избыточных элементов и сигналов.

4.3. Приведение настраиваемых параметров сети к предельным значениям и модификация нелинейных преобразователей нейронов

Получена сеть минимальной архитектуры. Выполняем приведение настраиваемых параметров сети к предельным значениям. Для синапсов сети приведение к предельным значениям означает их бинаризацию. После бинаризации можно попробовать упростить нелинейные функции нейронов.

Изменение нелинейной функции нейрона начинается после того, как у сети отконтрастированы все подлежащие контрастированию синапсы. Если у данного нейрона остался один входной синапс, то такой нейрон является просто передатчиком сигнала с предыдущего слоя на следующий. Можно попытаться избавиться от данного нейрона, заменив его на неадаптивный синапс с единичным весом (тождественную функцию), и дообучить сеть, подстраивая только синапсы, по которым нейрон рассылал свой сигналы (либо даже все синапсы сети). Если такое дообучение невозможно, значит, нелинейность этого нейрона действительно требуется для правильного решения задачи и нейрон придется оставить. Поскольку исходный нелинейный сигмоидный преобразователь не может быть преобразован в линейную функцию путем изменения значения параметра крутизны, то в любом случае работу по замене нелинейного преобразователя линейной связью придется выполнять, заменяя у нейрона нелинейную функцию.

Если у нейрона один выходной сигнал, то от такого нейрона можно попытаться избавиться, передав его набор синапсов нейрону, следующему за ним. Но здесь возникает одна сложность – такое перераспределение весов может нарушить достижимость критерия равномерной простоты сети – у нейрона следующего слоя в итоге может оказаться синапсов больше, чем требуется, и сократить их число (чтобы снова достигнуть критерия равномерной простоты) не удастся. Поэтому такой нейрон придется оставить, хотя он и исключаем из нейросети. Однако, в этом случае нужно попытаться заменить нелинейный преобразователь тождественной функцией.

Ну а если у нейрона несколько входных сигналов либо не выполняется замена нелинейного преобразователя нейрона на тождественную функцию, то можно попробовать привести нелинейный преобразователь к пороговому виду. Здесь используется последовательное

уменьшение значения настраиваемого параметра нелинейной функции и дообучение сети. Если после некоторого момента правильность работы сети сохраняется после замены нелинейной функции на пороговую, то такая замена производится. Иначе продолжаем дальнейшее уменьшение значения настраиваемого параметра до некоторого минимального значения, после которого отказываемся от попытки привести нелинейность к пороговому виду.

Естественно, что при обучении сети по методу двойственности необходимо одновременно пытаться приводить нелинейные функции всех нейронов сети к пороговому виду.

Если в сети после проведения упрощения нелинейных преобразователей не оказалось пороговых элементов, то можно ввести в обучение параметры крутизны сигмоид и попытаться снова удалить из сети лишние синапсы. Поскольку после введения в обучение новых адаптивных параметров гибкость сети повысилась, то от некоторых параметров (синапсов), возможно, удастся при этом избавиться. Ну а если это удалось, то снова вернемся к рассмотрению нелинейных элементов и повторим попытку упрощения сети на уровне нелинейностей нейронов, и т.д. до предела.

4.4. Проведение эквивалентных преобразований структуры нейросети

Поскольку нелинейные функции нейрона обычно являются нечетными функциями (симметричными относительно нуля), как например описанные выше сигмоидная и трехзначная пороговая функции, то для каждого нейрона возможна одновременная смена знака весов всех его синапсов и всех синапсов, по которым рассылается выходной сигнал нейрона, с сохранением правильности решения сетью задачи [7]. Поэтому появляется возможность, допустим, приведения общего числа отрицательных весов синапсов сети к минимуму. Есть гипотеза, что это упростит пользователю процесс вербализации сети.

5. Вербализация нейронных сетей

После того, как сеть приведена к логически прозрачному виду, необходимо представить навык, заложенный в нейронную сеть, в

понятном пользователю виде (вербализовать сеть). Это делается на основе графического визуального либо текстового представления структуры нейронной сети.

Вербализация проводится как процесс восстановления симптом-синдромной структуры понятий предметной области и требует непосредственного участия пользователя.

Входные сигналы сети являются входными симптомами, выходные сигналы нейронов первого слоя сети – синдромами первого уровня и одновременно входными симптомами для нейронов второго слоя, генерирующих синдромы второго уровня, и т.д. Пользователь, последовательно проходя слой за слоем, может на основе описания входных симптомов слоя и их группировки по нейронам давать некоторые осмысленные именованья и интерпретации синдромов, генерируемым этим слоем.

В результате получаем осмысленный текст, описывающий дерево рассуждений в терминах некоторой проблемной области. Программно-нейроимитатор способна дать только некоторый полуфабрикат эмпирической теории – вербальное описание нейронной сети на естественном языке, где имена симптомов и синдромов будут сгенерированы программой автоматически и не привязаны к проблемной области. Пользователь, выполняя когнитивный анализ, должен заменить абстрактные имена на осмысленные.

6. Автоматическая генерация полуэмпирических теорий

Достаточно сложно предложить алгоритм, который автоматически бы получал наиболее логически прозрачную сеть из заданной. При разработке программного продукта, позволяющего приводить нейронные сети к логически прозрачному виду, надо либо давать в руки пользователю весь набор элементарных процедур по упрощению сети, либо строить автоматическую процедуру, производящую упрощение сети несколько раз с варьированием очередности применения методов упрощения и оценкой получаемых результатов для того, чтобы получить оптимальное решение.

Рассмотрим, какие имеются возможности для автоматизации. Во-первых, возможно автоматическое построение одной или нескольких наиболее логически прозрачных сетей, варьируя процедуры упрощения и

оценивая альтернативные варианты. Во-вторых, автоматизируется генерация вербального описания нейронных сетей (полуэмпирических теорий), после чего пользователю остается только задача осмысления полученного текста и привязка к проблемной области. Ситуация, когда пользователь садится за компьютер, задает файл данных, нажимает одну единственную кнопку (или пункт меню) и получает текст, пригодный для вставки в дипломную работу/диссертацию/статью, пока что не возможна. Однако, такое пожелание обрисовывает идеал, к которому нужно стремиться, и показывает пути дальнейшего развития представленной технологии.

Рассмотрим ситуацию, когда можно провести автоматически еще некоторую работу по вербализации. Часто в число входных сигналов сети входят признаки, принимающие конечный набор дискретных значений. Если какой-либо нейрон входного слоя принимает только такие дискретнозначные признаки, то его выходной сигнал будет также принимать конечное множество дискретных значений, т.е. получается промежуточный дискретнозначный синдром. Автоматически можно сгенерировать набор правил, вычисляющих то или иное дискретное значение этого синдрома.

Если же нелинейный преобразователь входного нейрона приведен к пороговому виду, то даже в случае подачи на этот нейрон непрерывных входных сигналов также можно сгенерировать требования к значениям входных сигналов, приводящие к появлению нужного нам значения на выходе нейрона.

Идеи, изложенные в двух предыдущих абзацах, конечно же, распространяются не только на нейроны входного слоя, но и на нейроны следующих слоев. Иными словами, может появиться возможность генерации для некоторого синдрома набора ограничений, накладываемых на значения порождающих этот синдром симптомов, для того, чтобы синдром принял определенное значение.

7. Когнитологические аспекты

Принципиально, что для одной и той же таблицы данных и различных сетей (либо даже одной сети, но с разной начальной случайной генерацией исходных значений настраиваемых параметров) после

обучения, упрощения и вербализации может получиться несколько различных логически прозрачных сетей и, соответственно, несколько полуэмпирических теорий решения задачи. Это не представляется недостатком. При вербализации некоторые синдромы достаточно осмысленны и естественны, другие, напротив, непонятны. Из набора логически прозрачных нейросетей можно отсеять несколько наиболее осмысленных синдромов, объединить их в новую нейронную сеть, при этом введя, если необходимо, некоторые дополнительные нейроны или синапсы для такого объединения. Полученная нейросеть после адаптации и упрощения может быть более понятна, чем любой из ее предков. Таким образом, неединственность полуэмпирических теорий может стать ценным инструментом в руках исследователей-когнитологов.

Можно рассмотреть еще одну возможность применения таких “наиболее осмысленных фрагментов” логически прозрачной нейронной сети. Выделим из сети фрагмент, генерирующий некоторый осмысленный синдром. Теперь для каждого примера в исходной таблице данных можно вычислить значение этого синдрома и добавить в таблицу новый столбец – осмысленный интегральный признак. После расширения признакового пространства можно попытаться сформировать новую нейронную сеть на расширенном наборе входных сигналов, обучить ее и привести к логически прозрачному виду. Использование интегрального синдрома в качестве входного сигнала должно привести к получению более простой нейросети. Из последней также можно выделить некоторый фрагмент и добавить в таблицу еще один (или несколько) интегральных синдромов.

Таким образом, можно получить набор нейронных сетей-экспертов, образующих некоторое дерево решения. Эта макросеть концептуально отличается от исходной сети. В исходной сети базовые признаки использовались только для генерации синдромов первого уровня, те, в свою очередь, – только для генерации синдромов второго уровня. В сети малых экспертов для построения синдрома любого уровня могут быть использованы все ранее сгенерированные синдромы и все базовые признаки.

Подобное использование всех предшествующих сигналов может быть реализовано в исходной слоистой сети только в том случае, если в нее ввести связи не только между соседними слоями. Иными словами, входные сигналы должны поступать не только на нейроны первого слоя,

но и на нейроны всех остальных слоев. Выходы нейронов первого слоя должны передаваться как нейронам второго слоя, так и нейронам последующих слоев. Однако такое усложнение нейросети приводит к очень большому росту числа синапсов в ней, а это удлиняет время приведения такой сети к логически прозрачному виду. При этом не будет гарантироваться получение синдромов, которые будут более осмысленными по сравнению с синдромами, отобранными в ходе “ручной селекции” из нескольких сетей.

8. Влияние функции оценки на логическую прозрачность сети.

Исключение примеров

Точность решения задачи нейронной сетью регулируется функцией оценки [2,6], вычисляющей невязку между выходными сигналами сети и сигналами, которые требуется получить. Чем ниже требуемая точность, тем более простая (и более логически прозрачная сеть) сможет решить задачу. Поэтому для задачи приведения сети к логически прозрачному виду надо устанавливать как можно более низкие требования к точности и использовать специализированные функции оценки.

Не все примеры сеть может решить с одинаковой точностью – в таблице данных могут присутствовать примеры, которые с трудом решаются сетью, в то время, как остальные примеры сеть решает хорошо. Причиной этого может быть некорректность поставленной задачи. Например, в действительности в таблицу данных входят примеры трех классов, а мы пытаемся обучить сеть классификации на два класса. Другой причиной может быть, например, ошибка измерений либо опечатка при вводе данных в таблицу.

Для того, чтобы обнаружить некорректность в данных (либо в постановке задачи), предлагается исключать из таблицы данных наиболее “трудные” примеры (примеры с наибольшим значением функции оценки). Если сеть обучается правильно решать задачу и упрощается до довольно простой структуры, то исключение примеров производить не надо – задача корректна. В противном случае можно предложить следующий подход.

Зададимся требованием к числу правильно решенных примеров. Допустим, что нас устроит правильность решения 95% от общего числа примеров, присутствующих в таблице данных. Тогда построим процесс

обучения и упрощения сети так, чтобы сеть, правильно решающая 95% примеров, считалась правильно обученной решать задачу и, соответственно, упрощалась с сохранением навыка решения 95% примеров. При упрощении сети примеры, входящие в состав 5% наиболее трудных, могут меняться. После завершения процесса упрощения, если в итоге получилась сеть, гораздо более простая по сравнению с логически прозрачной сетью для всего набора задач, необходимо проанализировать наиболее трудные примеры – там могут встретиться ошибки в данных (см., например, [2], стр. 14) либо эти примеры будут “нетипичны” по сравнению с остальными.

Если же сеть с самого начала не может обучиться правильно решать задачу, то будем исключать из таблицы данных наиболее трудные примеры до тех пор, пока сеть не сможет обучиться. Далее надо исследовать статистические различия между набором оставшихся и исключенных данных – может обнаружиться, что отброшенные примеры образуют отдельный класс. Так было при решении задачи нейросетевой постановки диагноза вторичного иммунодефицита по иммунологическим и метаболическим параметрам лимфоцитов. Только коррекция классификационной модели (из отброшенных данных сформировали третий класс в дополнение к двум изначально имевшимся) позволила обучить сеть правильно решать теперь уже измененную задачу ([2] , стр. 15-16).

9. Как выбирают американских президентов

Проиллюстрируем процесс производства знаний из таблиц данных на основе базы данных по результатам выборов президентов США [2].

Имеется таблица данных с результатами 31-ой предвыборной ситуации (с 1860 по 1980 г.). Для каждого выбора в таблице содержатся данные по 12-ти бинарным признакам:

1. Правящая партия была у власти более одного срока?
2. Правящая партия получила более 50% голосов на прошлых выборах?
3. В год выборов была активна третья партия?
4. Была серьезная конкуренция при выдвижении кандидата от правящей партии?

5. Кандидат от правящей партии был президентом в год выборов?
6. Был ли год выборов временем спада или депрессии?
7. Был ли рост среднего национального валового продукта на душу населения более 2,1%?
8. Произвел ли правящий президент существенные изменения в политике?
9. Во время правления были существенные социальные волнения?
10. Администрация правящей партии виновна в серьезной ошибке или скандале?
11. Кандидат от правящей партии – национальный герой?
12. Кандидат от оппозиционной партии – национальный герой?

Также в таблице содержится информация о результатах выборов (победе правящей или оппозиционной партии). Значения бинарных признаков равны -1 (ответ "нет" для входного признака или победа правящей партии) и 1 (ответ "да" для входного признака или победа оппозиции).

Нейронные сети, обученные на этой таблице данных, уверенно предсказывали результаты вторых выборов Рейгана, победу Буша над Дукакисом, обе победы Клинтона.

Рассмотрим результаты построения по этой таблице логически прозрачных сетей и их вербализации. Оказывается, правильно предсказать результаты выборов можно на основании всего 5-ти признаков из 12-ти исходных.

Минимальная нейронная сеть содержит всего один сигмоидный нейрон, у которого на сумматор подаются значения признаков 4,6,8,9,12 (нумерация соответствует номеру признака в вышеприведенном списке) и постоянный единичный сигнал. Однако не все веса синапсов этой сети бинаризованы.

Можно построить однейронные сети с бинаризованными весами синапсов, но для этого приходится оставлять 6 наиболее значимых признаков. Две такие сети приведены на рис. 2. Эти сети, как и ранее упомянутая однейронная сеть с пятью входами, были получены из однослойной сети с 10-ю сигмоидными нейронами и дополнительным адаптивным сумматором, принимающим сигналы всех нейронов, с которого и снимался выходной сигнал сети.

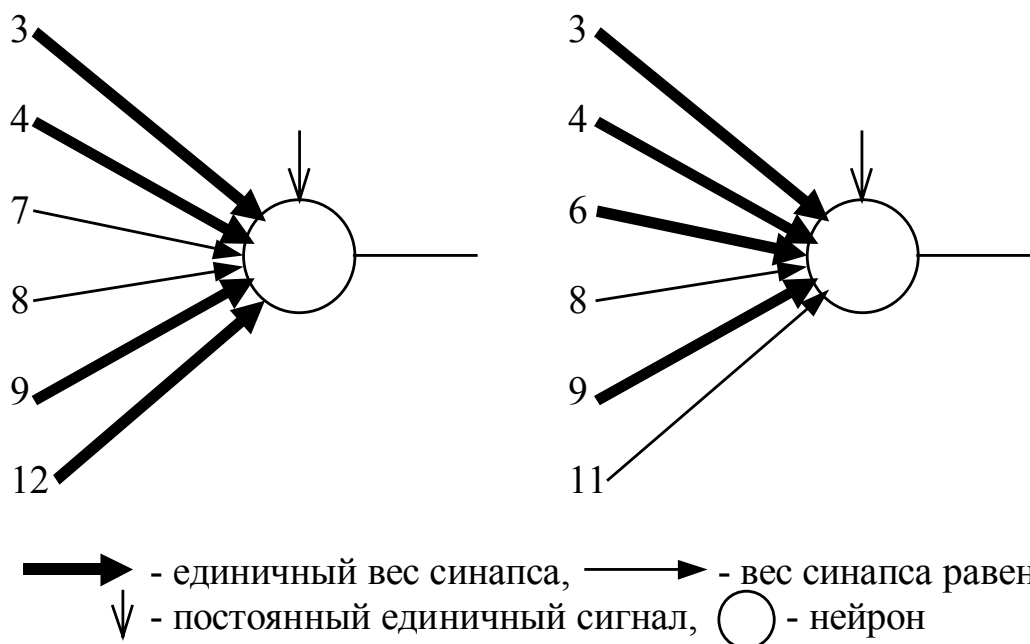


Рис. 2. Однонейронные сети для предсказания результатов выборов президентов США

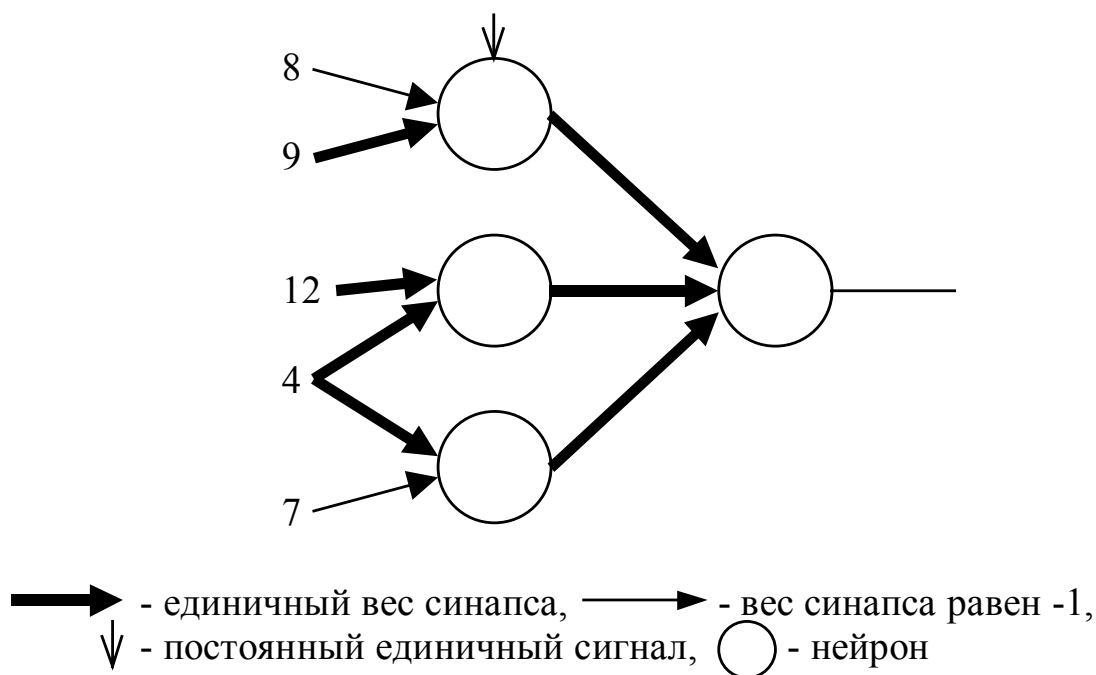


Рис. 3. Логически прозрачная нейронная сеть для предсказания результатов выборов президентов США

Эти сети не являются логически прозрачными по одному из введенных ранее критериев – на нейрон приходит семь сигналов (6 значений признаков и одна константа). Однако уже сейчас видно, какие

признаки способствуют победе правящей, а какие – победе оппозиционной партии.

Можно построить четырехнейронную логически прозрачную сеть, на каждый нейрон которой приходит три или два сигнала. Веса синапсов бинаризованы, используется пять входных сигналов. Она приведена на рис. 3. Эта сеть была получена из двухслойной сети с 10-ю нейронами в каждом слое и дополнительным выходным адаптивным сумматором.

На основе этих нейронных сетей можно строить наборы правил, помогающие как прогнозировать результаты выборов, так и менять предвыборную ситуацию таким образом, чтобы либо остаться у власти (для правящей партии), либо прийти к власти (для оппозиции). Так, для правящей партии плохо, если внутри ее существует конкуренция, если были социальные волнения, если активна третья партия или если отсутствует экономический рост. Для правящей партии хорошо, если президент зарекомендовал себя новатором (произвел существенные изменения в политике). Соответственно, для оппозиционной партии выигрышна прямо противоположная ситуация.

10. Заключение

Применение описанной технологии приведения сети к логически прозрачному виду и последующей вербализации для получения явных знаний из таблиц данных возможно во всех проблемных областях, где нейронные сети давно и с успехом применяются – в медицине, психологии, политологии, экономике, экологии, диагностике и управлении в технических системах и т.д. После приведения к логически прозрачному виду получаем нейронную сеть, не только допускающую простейшую техническую реализацию и использующую минимум данных, но и формирующую явный алгоритм решения задачи. Поскольку нейронные сети часто применяются для решения неформализованных задач, то возможность формирования явных алгоритмов решения поднимает нейронные сети на новый потребительский уровень.

Работа выполнена при поддержке Министерства науки и технологий РФ (подпрограмма "Перспективные информационные технологии", проект № 05.04.1291) и Красноярского краевого фонда науки (грант 7F0113).

ЛИТЕРАТУРА

1. Горбань А.Н. Обучение нейронных сетей. М.: изд. СССР-США СП "ParaGraph", 1990. 160 с. (English Translation: AMSE Transaction, Scientific Siberian, A, 1993, Vol. 6. Neurocomputing, pp.1-134).
2. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.
3. Гилев С.Е., Коченов Д.А., Миркес Е.М., Россиев Д.А. Контрастирование, оценка значимости параметров, оптимизация их значений и их интерпретация в нейронных сетях // Доклады III Всероссийского семинара "Нейроинформатика и ее приложения". – Красноярск: Изд-во КГТУ, 1995.- С.66-78.
4. Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сиб. журн. вычисл. математики / РАН. Сиб. отд-ние. - Новосибирск, 1998. - Т.1, №1. - с.11-24.
5. Горбань А.Н., Миркес Е.М. Логически прозрачные нейронные сети для производства знаний из данных. Вычислительный центр СО РАН в г. Красноярске. Красноярск, 1997. 12 с. (Рукопись деп. в ВИНТИ 17.07.97, № 2434-B97)
6. Горбань А.Н., Миркес Е.М. Оценки и интерпретаторы ответа для сетей двойственного функционирования. Вычислительный центр СО РАН в г. Красноярске. Красноярск, 1997. 24 с. (Рукопись деп. в ВИНТИ 25.07.97, № 2511-B97)
7. A. M. Chen, H.-m. Lu, R. Hecht-Nielsen. On the geometry of feedforward neural network error surfaces // Neural computations, 1993. - 5. pp. 910-927.
8. Царегородцев В.Г., Погребная Н.А. Нейросетевые методы обработки информации в задачах прогноза климатических характеристик и лесорастительных свойств ландшафтных зон. // Настоящий сборник. - С. 65-110.

СОДЕРЖАНИЕ

Предисловие редактора	3
 <i>Россиев А.А. Моделирование данных при помощи кривых для восстановления пробелов в таблицах.....</i>	 6
1. ОБЩАЯ СХЕМА МЕТОДА	6
2. ИТЕРАЦИОННЫЙ МЕТОД ГЛАВНЫХ КОМПОНЕНТ ДЛЯ ДАННЫХ С ПРОПУСКАМИ	7
3. КВАЗИЛИНЕЙНЫЕ ФАКТОРЫ И ФОРМУЛЫ КАРЛЕМАНА	11
4. НЕЙРОННЫЙ КОНВЕЙЕР	20
 <i>Горбунова Е.О. Финитность и детерминированность простых программ для кинетической машины кирдина.....</i>	 23
1. ВВЕДЕНИЕ	23
2. ПОНЯТИЕ КИНЕТИЧЕСКОЙ МАШИНЫ КИРДИНА	24
3. МОДЕЛИ ВЫПОЛНЕНИЯ ПРОГРАММЫ	27
4. ПРОГРАММЫ, СОСТОЯЩИЕ ИЗ ОДНОЙ КОМАНДЫ.....	29
5. ЗАКЛЮЧЕНИЕ	40
 <i>Горбунова Е.О. Алгоритмическая универсальность кинетической машины кирдина.....</i>	 41
 <i>Сенашова М.Ю. Погрешности нейронных сетей. Вычисление погрешностей весов синапсов</i>	 48
1. ВВЕДЕНИЕ	48
2. СТРУКТУРА СЕТИ	49
3. ДВА БАЗОВЫХ ПОДХОДА К ОЦЕНКАМ ПОГРЕШНОСТИ	50
4. ПОГРЕШНОСТИ ВЕСОВ СИНАПСОВ.....	51
5. ГАРАНТИРОВАННЫЕ ИНТЕРВАЛЬНЫЕ ОЦЕНКИ ПОГРЕШНОСТЕЙ ВЕСОВ СИНАПСОВ..	53
6. СРЕДНЕКВАДРАТИЧЕСКИЕ ОЦЕНКИ ПОГРЕШНОСТЕЙ ВЕСОВ СИНАПСОВ.....	60
7. ЗАКЛЮЧЕНИЕ	63
 <i>Царегородцев В.Г., Погребная Н.А. Нейросетевые методы обработки информации в задачах прогноза климатических характеристик и лесорастительных свойств ландшафтных зон.....</i>	 65
ВВЕДЕНИЕ.....	65
1. ПРОБЛЕМЫ ОБРАБОТКИ ТАБЛИЦ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ	66
2. ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ	70
3. ТРАНСПОНИРОВАННАЯ ЗАДАЧА РЕГРЕССИИ.....	83
4. ПРИМЕНЕНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ДЛЯ ОБРАБОТКИ ТАБЛИЦЫ КЛИМАТИЧЕСКИХ ДАННЫХ	89

5. ЗАКЛЮЧЕНИЕ	107
Доррер М.Г. ИНТУИТИВНОЕ ПРЕДСКАЗАНИЕ НЕЙРОСЕТЯМИ ВЗАИМООТНОШЕНИЙ В ГРУППЕ.....	111
1. ПРОБЛЕМА ОЦЕНКИ ВЗАИМООТНОШЕНИЙ.....	111
2. ОБЩАЯ ЗАДАЧА ЭКСПЕРИМЕНТОВ	112
3. ПРИМЕНЯЕМЫЕ В ЭКСПЕРИМЕНТАХ ПСИХОЛОГИЧЕСКИЕ МЕТОДИКИ	112
4. ЭКСПЕРИМЕНТЫ ПО ПРЕДСКАЗАНИЮ ГРУППОВОГО СТАТУСА	114
5. НЕЙРОСЕТЕВОЕ ИССЛЕДОВАНИЕ СТРУКТУРЫ ОПРОСНИКА	120
6. ОЦЕНКА ОПТИМИЗАЦИИ ЗАДАЧНИКА НЕЙРОСЕТЬЮ С ПОЗИЦИЙ ТЕОРИИ ИНФОРМАЦИИ	126
7 ЭКСПЕРИМЕНТЫ ПО ПРЕДСКАЗАНИЮ ПАРНЫХ ВЗАИМООТНОШЕНИЙ	127
Доррер М.Г. АППРОКСИМАЦИЯ МНОГОМЕРНЫХ ФУНКЦИЙ ПОЛУТОРАСЛОЙНЫМ ПРЕДИКТОРОМ С ПРОИЗВОЛЬНЫМИ ПРЕОБРАЗОВАТЕЛЯМИ.....	130
1. ПОСТАНОВКА ПРОБЛЕМЫ.....	130
2. АНАЛИТИЧЕСКОЕ РЕШЕНИЕ	131
3. ЗАПИСЬ РЕШЕНИЯ В ИДЕОЛОГИИ НЕЙРОСЕТЕЙ	133
4. АЛГОРИТМИЧЕСКАЯ ЧАСТЬ	135
5. ОЦЕНКА ИНФОРМАЦИОННОЙ ЕМКОСТИ НЕЙРОННОЙ СЕТИ ПРИ ПОМОЩИ ВЫБОРОЧНОЙ КОНСТАНТЫ ЛИПШИЦА	141
6. СОГЛАШЕНИЕ О ТЕРМИНОЛОГИИ	143
7. КОМПОНЕНТЫ СЕТИ	143
8. ОБЩИЙ ЭЛЕМЕНТ СЕТИ	144
9. ВХОД СЕТИ	144
10. ВЫХОД СЕТИ.....	145
11. СИНАПС СЕТИ.....	145
12. ТРИВИАЛЬНЫЙ СУММАТОР	146
13. НЕЙРОН	147
14. ПОТОК СЕТИ	148
15. СКОМПОНОВАННАЯ ПОЛУТОРАСЛОЙНАЯ ПОТОЧНАЯ СЕТЬ	149
Питенко А.А. ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ПРИ РЕШЕНИИ АНАЛИТИЧЕСКИХ ЗАДАЧ В ГИС	152
Жуков Л.А. ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ДЛЯ ПРОВЕДЕНИЯ УЧЕБНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ.....	164
1. ВВЕДЕНИЕ	164
2. ЗИМНЯЯ ПОЛИТЕХНИЧЕСКАЯ ШКОЛА ПО НЕЙРОИНФОРМАТИКЕ.....	166
3. ЗАДАЧИ	168
4. РЕЗУЛЬТАТЫ.....	169
5. ПЕРСПЕКТИВЫ	173

<i>Царегородцев В.Г.</i> Производство полуэмпирических знаний из таблиц данных с помощью обучаемых искусственных нейронных сетей.....	176
1. ВВЕДЕНИЕ.....	176
2. ЛОГИЧЕСКИ ПРОЗРАЧНЫЕ НЕЙРОННЫЕ СЕТИ.....	177
3. КОНТРАСТИРОВАНИЕ НЕЙРОНОВ	184
4. ПРИВЕДЕНИЕ НЕЙРОННЫХ СЕТЕЙ К ЛОГИЧЕСКИ ПРОЗРАЧНОМУ ВИДУ.....	186
5. ВЕРБАЛИЗАЦИЯ НЕЙРОННЫХ СЕТЕЙ	189
6. АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ПОЛУЭМПИРИЧЕСКИХ ТЕОРИЙ.....	190
7. КОГНИТОЛОГИЧЕСКИЕ АСПЕКТЫ	191
8. ВЛИЯНИЕ ФУНКЦИИ ОЦЕНКИ НА ЛОГИЧЕСКУЮ ПРОЗРАЧНОСТЬ СЕТИ. ИСКЛЮЧЕНИЕ ПРИМЕРОВ.....	193
9. КАК ВЫБИРАЮТ АМЕРИКАНСКИХ ПРЕЗИДЕНТОВ	194
10. ЗАКЛЮЧЕНИЕ.....	197

УДК 519

МОДЕЛИРОВАНИЕ ДАННЫХ ПРИ ПОМОЩИ КРИВЫХ
ДЛЯ ВОССТАНОВЛЕНИЯ ПРОБЕЛОВ В ТАБЛИЦАХ /
Россиев А.А. // Методы нейроинформатики. – Красноярск:
Издательство КГТУ, 1998.

Предложен и реализован метод последовательного моделирования набора данных одномерными многообразиями (кривыми). Метод интерпретируется как построение конвейера нейронов для обработки данных с пробелами. Другая возможная интерпретация – итерационный метод главных компонент и нелинейный факторный анализ для данных с пробелами.

УДК 519

ФИНИТНОСТЬ И ДЕТЕРМИНИРОВАННОСТЬ ПРОСТЫХ
ПРОГРАММ ДЛЯ КИНЕТИЧЕСКОЙ МАШИНЫ КИРДИНА /
Горбунова Е.О. // Методы нейроинформатики. – Красноярск:
Издательство КГТУ, 1998.

Предлагается новая абстрактная модель параллельных вычислений – кинетическая машина Кирдина. Ожидается, что эта модель сыграет ту же роль для параллельных вычислений, что и нормальные алгоритмы Маркова, машины Колмогорова и Тьюринга или схемы Поста для последовательных вычислений. Неформально кинетическую машину Кирдина можно описать следующим образом. Есть банка, в которой плавают слова. В нее добавляем правила-катализаторы; одни из них, сталкиваясь со словами, способствуют их распаду, другие, встречая пару подходящих слов, способствуют их синтезу, а третьи заменяют в словах некоторые подцепочки. В работе описаны основные способы реализации вычислений и исследованы свойства простейших программ для кинетической машины Кирдина.

УДК 519

АЛГОРИТМИЧЕСКАЯ УНИВЕРСАЛЬНОСТЬ
КИНЕТИЧЕСКОЙ МАШИНЫ КИРДИНА / Горбунова Е.О. // Методы нейроинформатики. – Красноярск: Издательство КГТУ, 1998.

Рассматривается новая ансамблевая модель параллельных мелкозернистых вычислений – кинетическая машина Кирдина (КМК). Доказывается, что детерминированная КМК является эффективным

вычислителем. Приводится метод реализации частично-детерминированной КМК при помощи специально устроенной системы машин Тьюринга. Дается понятие о статистической реализации КМК.

УДК 519

ПОГРЕШНОСТИ НЕЙРОННЫХ СЕТЕЙ. ВЫЧИСЛЕНИЕ ПОГРЕШНОСТЕЙ ВЕСОВ СИНАПСОВ / Сенашова М.Ю. // Методы нейроинформатики. – Красноярск: Издательство КГТУ, 1998.

Рассматриваются нейронные сети слоистой структуры, состоящие из слоев стандартных нейронов. Изучаются ошибки, возникающие при технической реализации сетей, а также при шумах и повреждениях.

Определены максимально допустимые погрешности, возможные для весов синапсов и сигналов сети, исходя из условия, что вектор выходных сигналов сети должен вычисляться с заданной точностью. Используются два типа оценок погрешности: гарантированные интервальные оценки и среднеквадратические оценки погрешностей.

Показано, что оценки допустимых погрешностей можно получить в ходе специального процесса “обратного распространения точности”. Он состоит в функционировании сети с той же системой связей, но от выходов к входам и с заменой элементов на двойственные. Эта двойственность принципиально отличается от той, которая используется в классическом методе вычисления градиентов оценки с помощью обратного распространения ошибок (back propagation of errors).

УДК 519

НЕЙРОСЕТЕВЫЕ МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ В ЗАДАЧАХ ПРОГНОЗА КЛИМАТИЧЕСКИХ ХАРАКТЕРИСТИК И ЛЕСОРАСТИТЕЛЬНЫХ СВОЙСТВ ЛАНДШАФТНЫХ ЗОН / Царегородцев В.Г., Погребная Н.А. // Методы нейроинформатики. – Красноярск: Издательство КГТУ, 1998.

В настоящей работе рассматриваются проблемы, возникающие при обработке таблиц экспериментальных данных. Описываются нейросетевые методы обработки информации для задач классификации и построения регрессионных зависимостей. Работа может служить первым введением в нейроинформатику для специалистов-прикладников. Подробно описан опыт применения нейросетевых технологий для решения задач прогноза, классификации и глобального моделирования изменений климата. С помощью нейросетевой транспонированной линейной регрессии,

нейросетевой нелинейной регрессии и нейросетевых классификаторов решены задачи прогнозирования неизвестных значений климатических параметров, классификации ландшафтных зон и глобального моделирования последствий изменения климата на основе таблицы данных, собранных с метеостанций Сибири и Дальнего Востока.

УДК 519

ИНТУИТИВНОЕ ПРЕДСКАЗАНИЕ НЕЙРОСЕТЯМИ
ВЗАИМООТНОШЕНИЙ В ГРУППЕ / Доррер М.Г. // Методы
нейроинформатики. – Красноярск: Издательство КГТУ, 1998.

В работе решается задача моделирования и прогнозирования системы взаимоотношений в группе на основе состояния и поведения исследуемых. Задача решается при помощи нейросетевых технологий с использованием нейроимитатора MultiNeuron на базе психологических и социологических методик – социометрии и минимизированного психологического опросника. Показано, что искусственная нейронная сеть позволяет с удовлетворительной точностью предсказывать отношения типа «человек-человек» и «человек-группа», а также может служить вспомогательным механизмом при разработке психологических методик.

УДК 519

АППРОКСИМАЦИЯ МНОГОМЕРНЫХ ФУНКЦИЙ
ПОЛУТОРАСЛОЙНЫМ ПРЕДИКТОРОМ С ПРОИЗВОЛЬНЫМИ
ПРЕОБРАЗОВАТЕЛЯМИ / Доррер М.Г. // Методы
нейроинформатики. – Красноярск: Издательство КГТУ, 1998.

В работе описаны теоретические предпосылки и программная реализация искусственной нейронной сети, обладающей важным новым свойством: генерацией структуры «от простого к сложному». Описан итерационный процесс наращивания объема сети. Использованы два условия на ограничение требуемого объема - по достижению требуемой точности или по критерию сравнения константы Липшица для сети и выборочной константы Липшица.

УДК 519

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ
ПРИ РЕШЕНИИ АНАЛИТИЧЕСКИХ ЗАДАЧ В ГИС /
Питенко А.А. // Методы нейроинформатики. – Красноярск:
Издательство КГТУ, 1998.

Описан подход к использованию нейросетевых технологий для решения аналитических задач в ГИС. Дано изложение некоторых из них и показаны области применения. Представлен макет программной системы.

УДК 519

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ
ДЛЯ ПРОВЕДЕНИЯ УЧЕБНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ /
Жуков Л.А. // Методы нейроинформатики. – Красноярск:
Издательство КГТУ, 1998.

Описан опыт работы автора по организации и проведению занятий по нейроинформатике в Сибирском государственном технологическом университете (СибГТУ). В работе приводится анализ творческих возможностей школьников старших классов и студентов для проведения исследований с использованием нейроинформационных технологий. Приведены результаты выполнения некоторых учебно-исследовательских работ. Показаны относительная простота и доступность нейроинформатики и нейроинформационных технологий для студентов и школьников старших классов.

УДК 519

ПРОИЗВОДСТВО ПОЛУЭМПИРИЧЕСКИХ ЗНАНИЙ ИЗ
ТАБЛИЦ ДАННЫХ С ПОМОЩЬЮ ОБУЧАЕМЫХ
ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ / Царегородцев В.Г. // Методы нейроинформатики. – Красноярск: Издательство КГТУ, 1998.

Рассматриваются возможности извлечения полуэмпирических теорий из таблиц данных при помощи нейронных сетей. Вводится понятие логически прозрачной нейронной сети, формируется набор критериев логической прозрачности, строится набор правил по приведению нейронной сети к логически прозрачному виду. Предлагается технология вербализации логически прозрачной сети – перевода на естественный язык алгоритма решения сетью задачи. Приводится пример решения классической задачи предсказания результатов выборов президента США.