Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Томский государственный университет систем управления и радиоэлектроники»

Кафедра электронных приборов

Основы оптоинформатики

РЕШЕНИЕ ЗАДАЧ:
ОБ ОБРАТНОМ РАСПРОСТРАНЕНИИ ОШИБОК В НЕЙРОННОЙ СЕТИ
ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ КОХОНЕНА
ОПТИМИЗАЦИИ НЕЙРОННОЙ СЕТИ ХОПФИЛЬДА
ОБУЧЕНИЯ ВЕРОЯТНОСТНОЙ НЕЙРОННОЙ СЕТИ

Методические указания к практическим работам

для студентов направления «Фотоника и оптоинформатика»

2012

Слядников, Евгений Евгеньевич

Решение задач: об обратном распространении ошибок в нейронной сети, обучения нейронной сети Кохонена, оптимизации нейронной сети Хопфильда, обучения вероятностной нейронной сети: методические указания к лабораторной работе для студентов направления «Фотоника и оптоинформатика» / Е.Е. Слядников; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Томский государственный университет систем управления и радиоэлектроники, Кафедра электронных приборов. - Томск : ТУСУР, 2012. - 19 с.

Цель данной работы: научить умению моделировать, исследовать и оценивать, самообучающиеся и самоорганизующиеся нейросетевые системы в оптике; научить владению методами, способами и средствами технологии передачи, приёма, обработки, хранения и отображения информации с помощью искусственного интеллекта.

.

Предназначено для студентов очной и заочной форм, обучающихся по направлению «Фотоника и оптоинформатика» по курсу «Основы оптоинформатики».

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Томский государственный университет систем управления и
радиоэлектроники»

Кафедра электронных приборов

УТВЕРЖДАЮ
Зав.кафедрой ЭП
_____С.М. Шандаров

«___» _____ 2012 г.

Основы оптоинформатики

РЕШЕНИЕ ЗАДАЧ:

ОБ ОБРАТНОМ РАСПРОСТРАНЕНИИ ОШИБОК В НЕЙРОННОЙ СЕТИ

ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ КОХОНЕНА

ОПТИМИЗАЦИИ НЕЙРОННОЙ СЕТИ ХОПФИЛЬДА

ОБУЧЕНИЯ ВЕРОЯТНОСТНОЙ НЕЙРОННОЙ СЕТИ

Методические указания к практическим работам

для студентов направления «Фотоника и оптоинформатика»

Разработчик

д-р физ.-мат. наук, проф. каф.ЭП
_____Е.Е. Слядников
«____»_____2012 г

2012

Содержание

1. ВВЕДЕНИЕ

Данная курсовая работа посвящена разработке методического пособия по курсу «Основы оптоинформатики», целью которого является раскрытие основных понятий и изучение основных моделей нейронных сетей с глубиной, достаточной для того, чтобы можно было реализовать такую сеть на том языке программирования, который окажется предпочтительнее.

В работе рассматриваются основные модели нейронных сетей, важные для понимания основ изучаемого предмета, и обсуждаются связи между нейронными сетями и традиционными понятиями из области искусственного интеллекта.

Актуальность проблемы:

В последние несколько летнаблюдаетсявзрыв интереса к нейронным сетям, которые успешно применяются в самых различных областях – бизнесе, медицине, технике, геологии, физике. Нейронные сети вошли в практику везде, где нужно решать задачи прогнозирования, классификации или управления. Такой впечатляющий успех определяется несколькими причинами:

- Богатые возможности. Нейронные сети - исключительно мощный метод моделирования, позволяющий воспроизводить чрезвычайно сложные зависимости;
- Простота в использовании. Нейронные сети учатся на примерах. Пользователь нейронной сети подбирает представительные данные, а затем запускает алгоритм обучения, который автоматически воспринимает структуру данных. При этом от пользователя, конечно, требуется какой-то набор эвристических знаний о том, как следует отбирать и подготавливать данные, выбирать нужную архитектуру сети и интерпретировать результаты, однако уровень знаний, необходимый для успешного применения нейронных сетей, гораздо скромнее, чем, например, при использовании традиционных методов статистики.

Нейронные сети привлекательны с интуитивной точки зрения, так как они основаны на примитивной биологической модели нервных систем. В будущем развитие таких нейробиологических моделей может привести к созданию действительно мыслящих компьютеров.

2. ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

Задача: определение основных элементов нейронной сети.

Цели: понять, как объяснить простыми словами, что такое нейронная сеть (н.с.);

что такое элемент, его связи и функции активности;

как представить связи в матричной форме;

как объяснить простыми словами, что обозначает обучение сети.

Искусственные нейронные сети представляют собой устройства параллельных вычислений, состоящие из множества взаимодействующих простых процессоров. Такие процессоры обычно исключительно просты, особенно в сравнении с процессорами, используемыми в персональных компьютерах. Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам, и, тем не менее, будучи соединенными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи.

Хотя н.с. могут быть реализованы в виде быстрых аппаратных устройств, большинство исследований выполняется с использованием программного моделирования на обычных компьютерах. Например, программная реализация н.с. может использоваться для составления плана кредитных выплат индивидуума, обращающегося в банк за займом.

Хотя решение на основе нейронной сети может выглядеть и вести себя как обычное программное обеспечение, они различны в принципе, поскольку большинство реализаций на основе н.с. “обучается”, а не программируется: сеть учится выполнять задачу, а не программируется непосредственно.

Решения на основе н.с. становятся все более совершенными и, несомненно, в будущем наши возможности по разработке соответствующих устройств возрастут за счет лучшего понимания их основополагающих принципов. Но уже сегодня имеется немало впечатляющих разработок.

База приложений нейронных сетей просто огромна:

- выявление фальшивых кредитных карт;
- прогнозирование изменений на фондовой бирже;
- составление кредитных планов;

- оптическое распознавание символов;

- профилактика и диагностика заболеваний человека;

- наблюдение за техническим состоянием машин и механизмов;

- автоматическое управление движением автомобиля;

- принятие решений при посадке поврежденных летательных аппаратов и т.д.

Будущее нейронных сетей кажется вполне ясным, и сегодня это та область знаний, о которой должны иметь определенное представление все научные специалисты, работающие в области компьютерных технологий, равно как и многие инженеры, и научные работники смежных специальностей [1].

2.1.Основные компоненты нейронных сетей

Нейронная сеть является совокупностью *элементов*, соединенных некоторым образом так, чтобы между ними обеспечивалось взаимодействие. Эти элементы, называемые также *нейронами* или *узлами*, представляют собой простые процессоры, вычислительные возможности которых обычно ограничиваются некоторым правилом комбинирования входных сигналов и правилом активизации, позволяющим вычислить выходной сигнал по совокупности входных сигналов. Выходной сигнал элемента может посылаться другим элементам по взвешенным *связям*, с каждой из которых связан *весовой коэффициент или вес.* В зависимости от значения весового коэффициента передаваемый сигнал или усиливается, или подавляется. Элемент н.с. схематически показан на рис. 2.1.

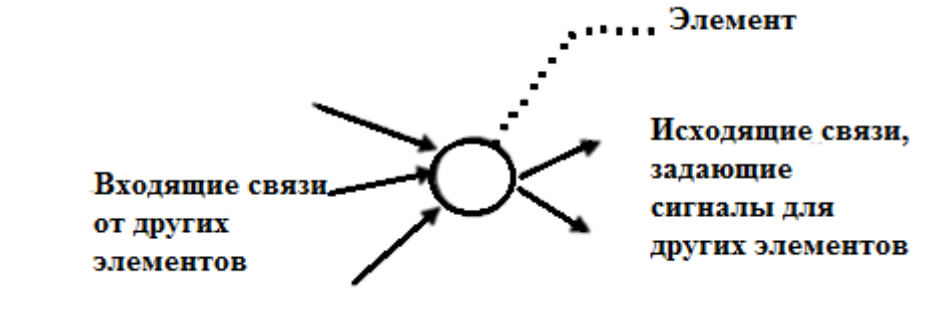


Рисунок 2.1 – Отдельный элемент нейронной сети.

Структура связей обычно определяется в два этапа:

- сначала разработчик системы указывает, какие элементы должны быть связаны и в каком направлении,

- затем в процессе фазы обучения определяются значения соответствующих весовых коэффициентов.

Существует множество различных типов нейронных сетей, но все они обладают рядом общих характеристик:

- множество простых процессоров

- структура связей

- правило распространения сигналов в сети

- правило комбинирования входящих сигналов

- правило вычисления сигнала активности

- правило обучения, корректирующие связи [1].

2.2. Обучение нейронной сети

Качество работы нейронной сети сильно зависит от предъявляемого ей в процессе обучения набора учебных данных. Учебные данные должны быть типичными для задачи, решению которой обучается сеть. Обучение часто оказывается уникальным процессом, когда приемлемые решения многих проблем могут быть получены только в процессе многочисленных экспериментов. Разработчикам решения на основе нейронной сети требуется следующее:

- Выбрать соответствующую модель сети
- Определить топологию сети (т.е. число элементов и их связи)
- Узнать параметры обучения

Часто разработчику необходимо выполнить и предварительную подготовку данных. Такая предварительная подготовка может быть совсем простой, - например, перевод с помощью масштабирования значений всех признаков в диапазон от 0 до 1, - а может включать использование и более сложных статистических процедур [1].

3. КЛАССИФИКАЦИЯ ОБРАЗЦОВ

3.1.Множество простых процессоров

С каждым процессором связывается набор входящих связей, по которым к данному элементу поступают сигналы от других элементов сети, и набор входящих связей, по которым сигналы данного элемента передаются другим элементам.

Некоторые элементы предназначены для получения сигналов из внешней среды и поэтому называются *входными элементами*, а некоторые для вывода во внешнюю среду результатов вычислений, и называются *выходными элементами.*

3.2.Структура связей

Структура связей отражает то, как соединены элементы сети.

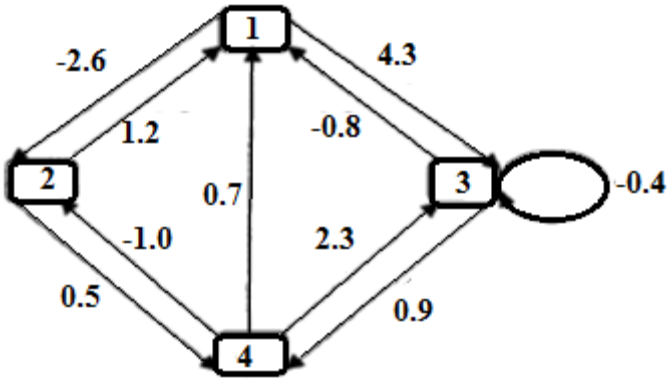
В одной модели каждый элемент может быть связан со всеми другими элементами сети, в другой модели элементы могут быть организованы в некоторой упорядоченной по уровням иерархии, где связи допускаются только между элементами в смежных слоях, а в третьей- могут допускаться обратные связи между смежными слоями или внутри одного слоя, или же допускаться посылка сигналов элементами самим себе.

Каждая сеть определяется тремя параметрами:

1. Элемент, от которого исходит данная связь.
2. Элемент, которому данная связь направлена.
3. Число, указывающее весовой коэффициент (т.е. вес связи).

Отрицательное значение веса соответствует подавлению активности соответствующего элемента, а положительное значение – усилению его активности.

Структура связей обычно представляется в виде весовой матрицы W, в которой каждый элемент w_{ij} представляет величину весового коэффициента для связи, идущей от элементаiк элементу j.Для описания структуры связей может использоваться не одна, а несколько весовых матриц, если элементы сети оказываются сгруппированными в слои [2].



$$W=\begin{bmatrix} 0.0 & -2.6 & 4.3 & 0.0 \\ 1.2 & 0.0 & 0.0 & 0.5 \\ -0.8 & 0.0 & -0.4 & 0.9 \\ 0.7 & -1.0 & 2.3 & 0.0 \end{bmatrix}$$

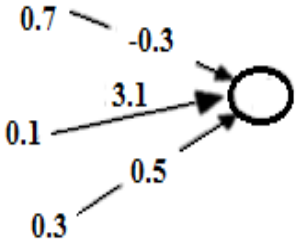
Рисунок 3.2 – Матрица, описывающая сетевые связи.

3.3.Правило распространения сигналов в сети

В обычных компьютерных программах используются условия, выполнение которых определяет начало и конец различных процессов. То же самое верно и для нейронных сетей. Каждая конкретная модель сети предполагает наличие некоторого правила обновления состояния элементов сети и посылки сигнала другим элементам. При этом в одних моделях моменты обновления элементов выбираются случайным образом, в других же моделях обновление некоторых групп элементов допускается только после определенных групп других элементов.

3.4.Правило комбинирования входящих сигналов

Довольно часто входящие сигналы элемента предлагается комбинировать путем суммирования их взвешенных значений. Пример этого показан на рис.3.4.



$$net_j = \sum_{i=1}^{n} x_i w_{ij}$$

$$net_j = (0.7 \times -0.3) + (0.1 \times 3.1) + (0.3 \times 0.5) = 0.25$$

или в векторном представлении

$$\begin{bmatrix} 0.7 & 0.1 & 0.3 \end{bmatrix} \begin{bmatrix} -0.3 \\ 3.1 \\ 0.5 \end{bmatrix}$$

Рисунок 3.4 – Типичный метод суммирования сигналов, направленных конкретному элементу.

net_j - результат комбинированного ввода элемента j, x_i - выход элемента i, а n–число задействованных связей.

Используются и другие формы комбинирования входящих сигналов. Часто встречающийся метод – рассмотрение квадрата разности между значением силы связи и значением передаваемого по связи сигнала с последующим суммированием таких разностей для всех входящих связей данного элемента [3].

3.5.Правило вычисления сигнала активности

Для всех элементов имеется правило вычисления выходного значения, которое предполагается передать другим элементам или во внешнюю среду. Это правило называется *функцией активности*, а соответствующее выходное значение называют *активностью* соответствующего элемента. Активность может представляться либо некоторым действительным значением произвольного вида, либо действительным значением из некоторого ограниченного интервала значений, или же некоторым значением из определенного дискретного набора значений. На вход функции активности поступает значение комбинированного ввода данного элемента [3].

3.6. Функция выбора решений

Чтобы описать, каким образом работает нейронная сеть, рассмотрим тривиальную задачу и используем для решения этой задачи самый простой тип сети.

Задача состоит в выработке правил классификации самолетов для бомбардировщиков и истребителей в зависимости от их максимальной скорости и максимального взлетного веса. Такие правила могут быть заданы формально:

Если вес> 0,80 и скорость <0,55, то бомбардировщик

Если вес < 0,90 и скорость >0,25, то истребитель.

Эти правила используют дискретные граничные значения, разделяющие пространство всех значений на прямоугольные области.

Разделения, порожденные этими правилами, вполне успешно классифицируют самолеты, представленные на диаграмме, но оказываются не слишком гибкими, если этими правилами придется классифицировать новый самолет.

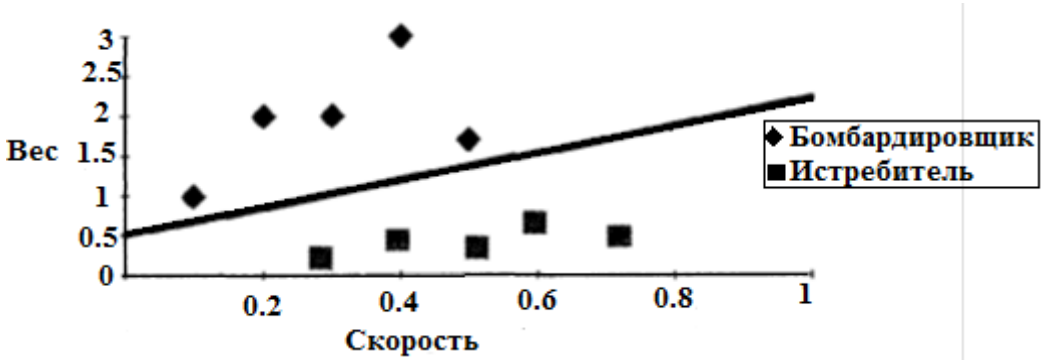


Рисунок 3.6 – Разделение абстрактных данных на два класса.

Альтернативный подход к использованию правил заключается в выводе функции классификации путем построения прямой, разделяющий два класса. Для нового самолета нам нужно просто указать точку на плоскости, соответствующую известным значениям максимальной скорости и максимального взлетного веса и посмотреть, по какую сторону от прямой будет расположена эта точка.

В данном случае рассмотрены два признака: скорость и вес, поэтому можно представить данные в виде изображения на плоскости.

Выход заключается в использовании функции выбора решения. Уравнение прямой, разделяющей два типа самолетов, записывается в следующем виде:

$$x_2 = 1,5x_1 + 0,5$$

где x_1 представляет скорость, а x_2 - вес. Это уравнение можно использовать для создания функции выбора решения:

$$f(x_1, x_2) = -x_2 + 1{,}5x_1 + 0{,}5,$$

d = истребитель, если $f(x_1, x_2) \geq 0$

d = бомбардировщик, если $f(x_1, x_2) < 0$.

Предлагаемую функцию выбора решения можно моделировать с помощью нейронной сети и даже реализовать в виде аппаратных средств.

На рис.3.7 показана сетевая модель для данной функции выбора решения:

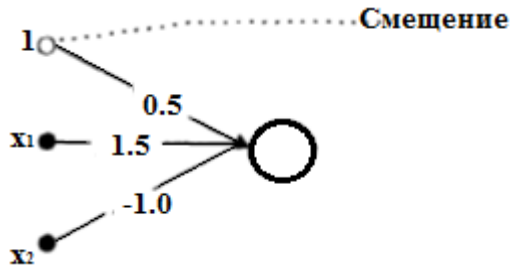


Рисунок 3.7 – Простая нейронная сеть.

Сетевой ввод для центрального элемента находится путем умножения переменных ввода, x_1 и x_2, на соответствующие их взвешенным связям коэффициенты с последующим суммированием результата.

Указанное на схеме значение смещения тоже добавляется в сумму, в результате чего получается значение комбинированного ввода для данного элемента.

$$net_j = w_o + \sum_{i=1}^{n} x_i w_{ij}$$

где net_j представляет значение комбинированного ввода, w_o - смещение, связываемое с элементом, значение активности которого считается всегда равным 1, x_i - значение активности i – го элемента, а w_{ij} - вес связи, ведущей от элемента с номеромі к элементу с номером j [4].

3.6.1.Тождественная функция

Функция активности для входных элементов может быть тождественной функцией, это означает, что значение активности оказывается в точности равным комбинированному вводу.

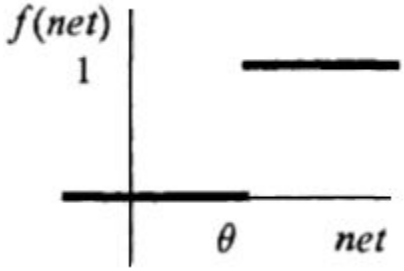
Входные элементы обычно предназначены для распределения вводимых сигналов между другими элементами сети, поэтому для входных элементов обычно требуется, чтобы исходящий от элемента сигнал был таким же, как и входящий. В отличие от других элементов сети, входные элементы имеют только по одному входному значению.

Например, каждый входной элемент может получать сигнал от одного соответствующего ему датчика, размещенного на фюзеляже самолета. Один этот элемент связывается со многими другими элементами сети, так что данные, полученные от одного датчика, оказываются распределенными между многими элементами сети.

Поскольку входные элементы предназначены исключительно для того, чтобы распределять сигналы, получаемые из внешней среды, многие исследователи вообще не считают входные элементы частью нейронной сети.

3.6.2.Пороговая функция

В большинстве моделей н.с. используются нелинейные функции активности. Пороговая функция ограничивает активность значения 1 или 0 от значения комбинированного ввода в сравнении с некоторой пороговой величиной Θ.



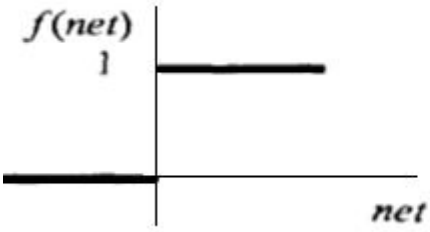
$$f(net) = \begin{cases} 1, \text{ если } net_j \geq \theta, \\ 0, \text{ если } net_j < \theta. \end{cases}$$

Рисунок 3.6.2 – Пороговая функция.

Чаще всего удобнее вычесть пороговое значение из значения комбинированного ввода и рассмотреть пороговую функцию в ее математически эквивалентной форме.

Сдвиг w_0 в данном случае оказывается отрицательным, а значение комбинированного ввода вычисляется по формуле:

$$net_j = w_o + \sum_{i=1}^{n} x_i w_{ij}$$



$$f(net) = \begin{cases} 1, \text{ если } net_j \geq 0, \\ 0, \text{ если } net_j < 0. \end{cases}$$

Рисунок 3.6.3 – Пороговая функция с учетным смещением

Сдвиг обычно интерпретируется как связь, исходящая от элемента, активность которого всегда равна 1. Комбинированный ввод в данном случае можно представить в виде:

$$net_j = \sum_{i=1}^{n} x_i w_{ij}$$

где x_0 всегда считается равным 1.

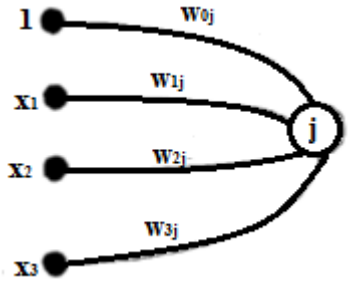


Рисунок 3.6.4–Для удобства компонент смещения часто интерпретируется как связь с элементом предыдущего слоя в предположении, что активность этого элемента равна1.

3.6.3.Сигмоидальная функция

Наиболее часто используемой функцией активности является сигмоидальная функция. Выходные значения такой функции непрерывно заполняют диапазон от 1 до 0. Примером может служить логистическая функция, показанная на рис. 3.6.3:

$$f(net) = \frac{1}{1+\exp(-net)}$$

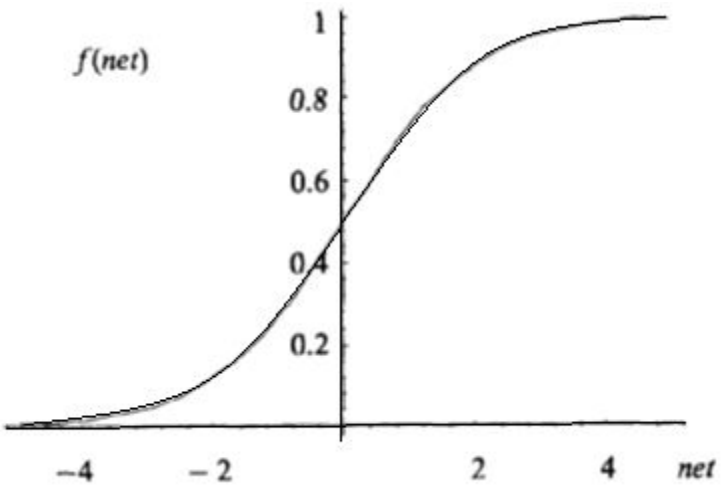


Рисунок 3.6.3 – Сигмоидальная функция.

3.7. Корректировка весов

Из рисунка 3.6 должно быть ясно,что имеется множество прямых, которые могут быть выбраны в качестве границы, разделяющей данные, поэтому имеется целое множество весовых значений, дающих подходящее решение.

Если требуемый выход i – го элемента обозначить t_j, а наблюдаемый на самом деле - σ_j, то ошибка E_p для образца рможет быть определена по формуле:

$$E_p = \frac{1}{2}\sum_j (t_j - \sigma_j)^2$$

Полная ошибка будет равна:

$$E = \sum_p E_p$$

Активность любого элемента зависит от комбинированного ввода этого элемента, а значит, от весовых значений, влияющих на этот элемент [4].

3.8. Сеть обратного распространения ошибки

Для корректировки весов можно использовать дельта – правило:

$$\Delta w_{ij} = \eta \delta_j x_i , \qquad \delta_j = (t_j - \sigma_j) \qquad (1)$$

где t_j обозначает требуемое значение для элемента j, σ_j - его реальный ввод, x_i - сигнал, приходящий от элементаi, η - норма обучения, а Δw_{ij} - величина, на которую изменяется вес для связи, идущий от элементаiк элементу j.

Это правило очень просто получается в случае линейного элемента, когда вывод определяется следующей формулой:

$$\sigma_j = \sum_j x_i w_{ij} \ (2)$$

Используя цепное правило, можно выразить производную поверхности ошибок в зависимости от веса в виде произведения, характеризующего изменение ошибки в зависимости от вывода элемента, и изменение вывода в зависимости от связанных с элементом весовых коэффициентов:

$$\frac{\partial E}{\partial w_{i\,j}} = \frac{\partial E}{\partial \sigma_j} \frac{\partial \sigma_j}{\partial w_{i\,j}} \ (3)$$

$$\frac{\partial E}{\partial \sigma_j} = -\delta_j \ (4), \qquad \frac{\partial \sigma_j}{\partial w_{ij}} = x_i \ (5), \quad \text{откуда получаем:} \quad -\frac{\partial E}{\partial w_{i\,j}} = \delta_j x_i \ (6)$$

Принимая во внимание тот факт, что вес должен изменяться в направлении, противоположном направлению вектора градиента, и умножая на норму обучения, приходим к равенству (1).

3.9.Правило обучения. Корректирующая связь

Одно из главных преимуществ н.с. заключается в том, что они предполагают наличие правил, с помощью которых сеть может программироваться автоматически.

Например, можно рассмотреть следующую функцию, реализующую вышеперечисленное определение XOR.

Типичной формой обучения является управляемое обучение, когда для каждого набора данных, подающегося в процессе обучения на вход сети, соответствующий выходной набор известен. Обычно в начале обучения весовые коэффициенты устанавливаются равными случайным малым значениям, так что в первый раз при

предъявлении сети учебного образца оказывается весьма маловероятным, чтобы сеть произвела верный вывод.

Расхождение между тем, что даст сеть, и тем, что для данного учебного набора должно быть получено на самом деле, составляет ошибку, которая может использоваться для корректировки весов.

Примером правила коррекции ошибок является дельта-правило, выражаемое также правилом Видроу – Хоффа [2].

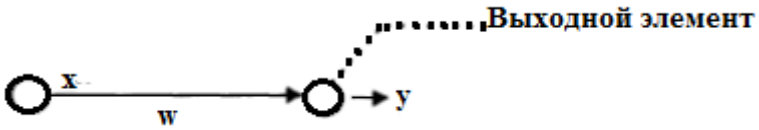


Рисунок3.9 – Взвешенная связь двух элементов.

На рис.3.9 выходной элемент имеет активность y, а истинный вывод должен быть равным t. Ошибка задается следующей формулой:

$$\delta = \mathrm{t} - \mathrm{y}$$

Сигнал, приходящий к выходному элементу, обозначен через x. В соответствии сдельта – правилом, необходимо внести коррекцию $\Delta \mathrm{w}$, вычисляемую по формуле:

$$\Delta \mathrm{w} = \eta \delta \mathrm{x}$$

где η обозначает действительное число, называемое *нормой обучения.* Новый весовой коэффициент устанавливается равный сумме значений старого веса и коррекции:

$$\mathrm{w} = \mathrm{w} + \Delta \mathrm{w}$$

В алгоритме обучения выполнению операций типа XOR, используется обобщенная версия дельта – правил [5].

В начале обучения весовые коэффициенты устанавливаются равными малым случайным значениям, например, из диапазона {-0,3, +0,3}. В процессе обучения на вход сети подаются образец за образцом, и в результате их обработки весовые коэффициенты корректируются до тех пор, пока для всех видимых образцов ошибки не станут меньше некоторого приемлемого достаточно малого значения. В завершение процесса сеть тестируется на данных, не представленных в фазе обучения: в результате можно оценить,

насколько хорошо сеть работает с данными, которые в процессе обучения были ей неизвестны.

3.10. Линейные и нелинейные проблемы

Для задачи классификации, например, выяснения типа самолетов, если прямая или гиперплоскость может разделить все образцы на соответствующие им классы, то проблема является *линейной.*

Если же для решения проблемы разделения образцов на классы требуется несколько прямых или гиперплоскостей, то проблема называется *нелинейной.*

Широко известным примером нелинейной проблемы является проблема моделирования отношения XOR. Отношение XOR при выводе дает 1 только тогда, когда в точности одно из вводимых значений равно 1, иначе вывод оказывается равным 0.

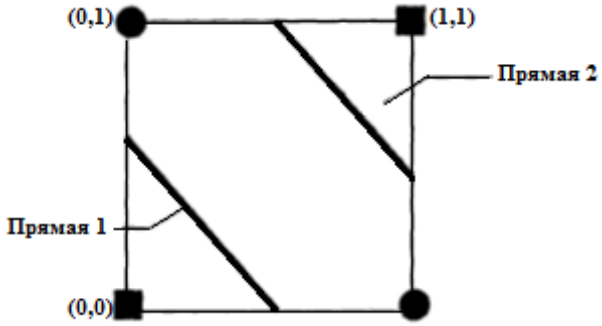


Рисунок 3.10.1 – Проблема XOR, решаемая с помощью двух разделяющих прямых.

Проблема XORявляется нелинейной, поэтому для ее решения с помощью нейронной сети необходимо строить две или больше прямых для разделения данных.

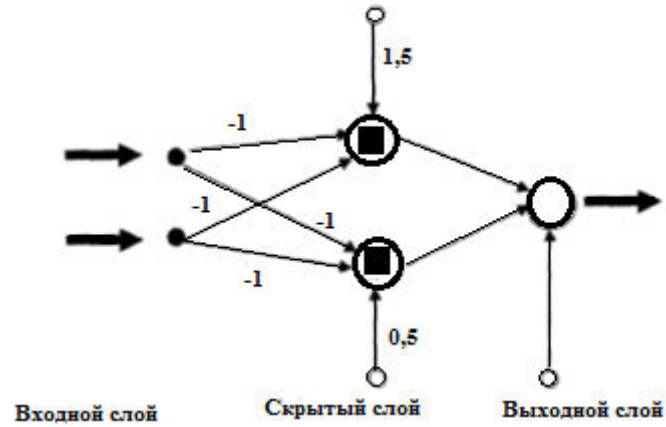


Рисунок 3.10.2 – Нейронная сеть для моделирования отношения XOR.

В сети на рис. 3.10.2 элементы разделены на три слоя. Элементы первого слоя являются элементами данных в сеть. Эти элементы отличаются от элементов последующих слоев тем, что они не имеют функции преобразования. Второй слой называется *скрытым* слоем – скрытые элементы связаны только с другими элементами и не имеют непосредственных связей с внешней средой. Выходной слой предназначен для передачи ответа сети во внешнюю среду[5].

4. КЛАСТЕРИЗАЦИЯ ОБРАЗЦОВ

Задача: описание обучения без управления.

Цели: понять, что такое обучение без управления и что такое принцип кластеризации образцов;

принципы построения самоорганизующейся сети Кохонена, чтобы иметь возможность реализовать такую сеть на предпочитаемом языке программирования.

Ранее было рассмотрено управляемое обучение, когда нейронная сеть обучается классифицировать образцы в соответствии с инструкциями: целевой выходной образец дает информацию сети о том к какому классу следует научиться относить входной образец.

При обучении без управления таких инструкций нет, и сети приходится проводить кластеризацию образцов самостоятельно. Все образцы одного кластера должны иметь что-то общее – они будут оцениваться, как подобные.

Алгоритмы кластеризации выполняют такие операции с образцами данных. Группы в дальнейшем будем называть *кластерами* и предполагать, что разделение образцов на кластеры должно удовлетворять следующим двум требованиям:

- образцы внутри одного кластера должны быть в некотором смысле подобны;

- кластеры, подобные в некотором смысле, должны размещаться близко один от другого.

На рис.4.1 показано размещение на двумерной плоскости данных, которые естественным образом организуются в три кластера: соответствующая образцу точка

попадает в определенный кластер, если она располагается близко к точкам этого кластера в сравнении с точками, принадлежащими другим кластерам.

Мерой близости двух точек обычно является квадрат евклидова расстояния между ними, вычисляемый по формуле:

$$d_{pq} = \sum_{t}^{n} (x_{pi} - x_{qi})^2 ,$$

где d_{pq} обозначает квадрат евклидова расстояния между точкой ри точкой q, x_{pi} - i- я координата образца p, а n–значение размерности.

Рисунок 4.1 – Данные, формирующие три кластера.

4.1.Алгоритм кластеризации

Алгоритм кластеризации представляет собой статистическую процедуру выделения групп из имеющегося набора данных. Существует немало алгоритмов кластеризации самого разного уровня сложности. Один из самых простых подходов заключается в том, чтобы предложить существование определенного числа кластеров и произвольным образом выбрать координаты для каждого из прототипов.

Затем каждый вектор из набора данных связывается с ближайшим к нему прототипом, и новыми прототипами становятся центроиды всех векторов, связанных с исходным прототипом.

Можно сформулировать некоторые основные свойства алгоритма кластеризации:

- автоматическое определение числа прототипов;

- сравнение прототипов;
- представление характерных признаков прототипа

4.2.Самоорганизующаяся карта признаков – сеть Кохонена

Самоорганизующаяся карта признаков (сеть SOFM – Self-OrganizingFeatureMap)имеет набор входных элементов, число которых соответствует размерности учебных векторов, и набор выходных элементов, которые служат в качестве прототипов. Базовая архитектура сети SOFM показана на рис.4.2.

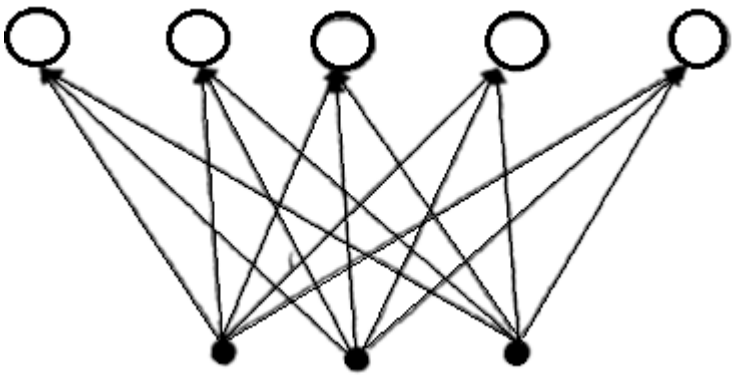


Рисунок 4.2 –Сеть, имеющая три входных и пять кластерных элементов, каждый элемент входного слоя связан с каждым элементом кластерного слоя.

Входные элементы предназначены только для того, чтобы распределять данные вектора между выходными элементами сети. Выходные элементы называются *кластерными элементами*. Так как число входных элементов соответствует размерности вводимых векторов, а каждый входной элемент связан со всеми кластерными элементами, общее число влияющих на кластерный элемент весовых значений тоже оказывается равным размерности входных векторов.

Часто удобно интерпретировать весовые значения кластерного элемента как значения координат, описывающих позицию кластера в пространстве входных данных [6].

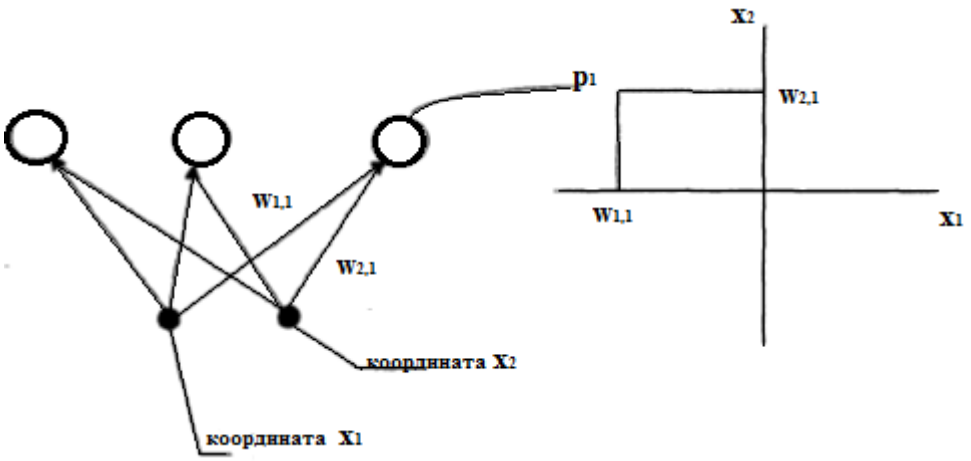


Рисунок 4.3 – Самоорганизующаяся карта признаков.

4.3.Обучение сети SOFM

Карта признаков проходит два этапа обучения. На первом этапе элементы упорядочиваются так, чтобы отражать пространство входных элементов, а на втором этапе происходит уточнение их позиций. Процесс представляется путем использования двумерных данных и построения соответствующей поверхности.

Например, входные векторы выбираются случайным образом на основе однородного распределения в некотором квадрате, и начинает обучение карты.

В определенные моменты в ходе обучения строятся изображения карты путем использования соответствия, показанного на рис. 4.3. Элементы соединяются линиями, чтобы показать их относительное размещение. Конечным результатом обучения является карта, покрывающая все входное пространство [6].

4.4.Дополнительные сведения о сети SOFM

В определенных случаях в качестве меры сходства векторов можно использовать угол между ними. На рис.4.4 вектор*a* ближе к прототипу p_1в смысле евклидова расстояния, но ближе к прототипу p_2, если в качестве меры сходства выбрать значение угла.

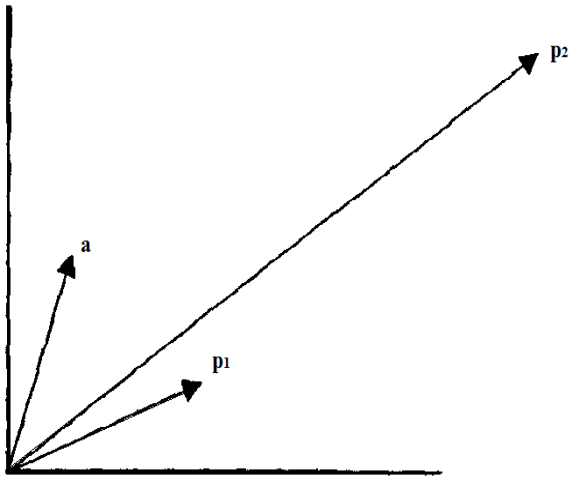


Рисунок 4.4 – Три вектора, приведенные в примере.

Обучение сети SOFM, в которой в качестве меры сходства используется скалярное произведение векторов.

Формула обновления весов должна быть следующей:

$$w_j(n+1) = \frac{w_j(n) + \eta x}{\left\| w_j(n) + \eta x \right\|}$$

Таким образом, для элемента- победителя к его весовому вектору добавляется часть этого вектора, а затем полученный в результате вектор нормализуется.

5. АССОЦИАЦИЯ ОБРАЗЦОВ

Задача: описание ассоциативной памяти.

Цели: понять принципы построения дискретной сети Хопфилда и двунаправленной ассоциативной сети, чтобы иметь возможность реализовать эти сети.

5.1. Введение

Все знают, что такое ассоциация: слово одного языка может ассоциироваться со словом другого языка, фотография друга ассоциируется с его именем и можно даже умудриться ассоциировать некоторое туманное изображение с определенным реальным объектом.

Идея заключается в том, чтобы выбрать нужный образец из памяти, даже если у нас нет всей необходимой информации для начала поиска сохраненного образца.

Например, если вы хотите найти книгу в библиотеке, но не помните ее название. При этом если вы знаете имя автора и описание того, чему книга посвящена, этого уже достаточно, чтобы найти ассоциируемый с этой информацией объект.

Когда сохраняемая в памяти пара ассоциируемых образцов создается одинаковыми образцами, память называется ассоциативной, а если образцы являются разными, то память называется гетероассоциативной [7].

5.2. Дискретная сеть Хопфилда

Сеть Хопфилда является автоассоциативной сетью, ведущей себя подобно памяти, которая может вспомнить сохраненный образец даже о подсказке, представляющей собой искаженную помехами версию нужного образца.

Дискретная сеть Хопфилда имеет следующие характеристики:

- один слой элементов;
- каждый элемент связывается со всеми другими элементами, но не связывается с самим собой;
- за один шаг обновляется только один элемент, в отличие от сети с обратным распространением ошибок, где все элементы слоя могут изменяться одновременно;
- элементы обновляются в случайном порядке, но в среднем каждый элемент должен обновляться в одной и той же мере;
- вывод элемента ограничен значениями 0 и 1.

Сеть Хопфилда является рекуррентной в том смысле, что для каждого входного образца выход сети повторно используется в качестве ввода до тех пор, пока не будет достигнуто устойчивое состояние.

Пример сети Хопфилда показан на рис. 5.2. Удобно считать, что сеть Хопфилда не имеет входных элементов, так как входной вектор просто определяет начальные значения активности элементов. Значение активности элемента получается на основе использования некоторого правила активизации. Каждый элемент сети Хопфилда имеет состояние, характеризующееся значением активности, которое должен посылать данный элемент другим элементам, а состояние сети в любой момент времени задается вектором состояний всех ее элементов.

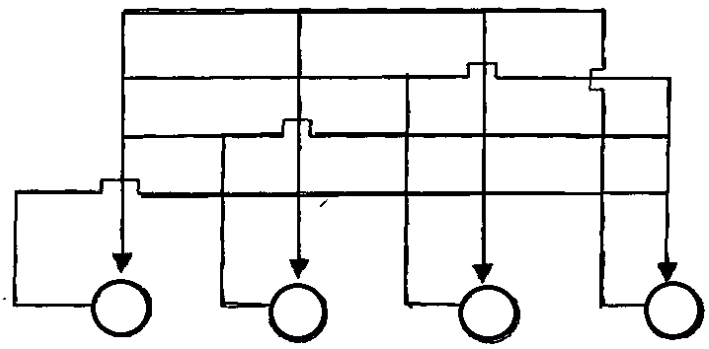


Рисунок 5.2 – Дискретная сеть Хопфилда с четырьмя элементами.

В качестве входных данных сети Хопфилда можно использовать двоичные: +1 - “включено” и -1 – “выключено”. Комбинированный ввод элемента вычисляется по формуле:

$$net_j = \sum_{i=1}^{n} s_i w_{ij}$$

где s_i обозначает состояние элемента с номером i.

Если комбинированный ввод оказывается равным нулю, то элемент остается в состоянии, в котором он пребывал перед обновлением.

Сеть работает очень просто. Входной вектор задает начальное состояние всех элементов. Элемент для обновления выбирается случайным образом. Выбранный элемент получает взвешенные сигналы от всех остальных элементов и изменяет свое состояние. Выбирается другой элемент и все повторяется. Сеть достигает предел, когда ни один из ее элементов, будучи выбранным для обновления, не меняет своего состояния.

Сеть Хопфилда ведет себя как память, и процедура сохранения отдельного вектора представляет собой вычисление прямого произведения вектора с ним самим. В результате этой процедуры создается матрица, задающая весовые значения для сети Хопфилда, в которой все диагональные элементы должны быть установлены равными нулю. Таким образом, весовая матрица, соответствующая сохранению вектора x, задается формулой:

$$W = x^T x$$

5.3. Двунаправленная ассоциативная память

Сетью, имеющей много общего с сетью Хопфилда, является двунаправленная ассоциативная память, которая является гетероассоциативной рекуррентной сетью. Сеть сохраняет пары образцов и может восстановить образец, когда ассоциированный с ним образец предлагается ей в качестве подсказки.

В этой сети два слоя элементов – по одному для каждого из образцов пары – и оба слоя соединяются двумя связями (рис.5.3).

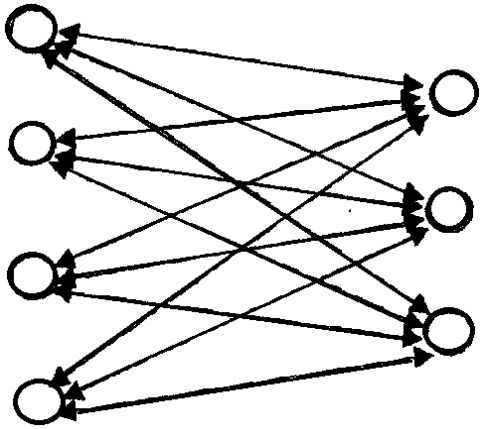


Рисунок 5.3 – Двунаправленная ассоциативная память.

Ассоциативная сеть действует подобно памяти:

- автоассоциация связывает образец с самим собой;
- гетероассоциация представляет собой связывание двух разных образцов, один может использоваться в качестве подсказки, по которой из памяти извлекается другой образец;
- когда автоассоциативное обучение происходит путем пропускания сигналов через узкий канал типа скрытого слоя автоассоциативной сети с прямой связью, происходит в некотором смысле сжатие данных;
- сеть Хопфилда может использоваться для автоассоциации, а двунаправленная ассоциативная память – для гетероассоциации;
- весовые значения дискретной сети Хопфилда и двунаправленной ассоциативной памяти могут быть найдены с помощью простых матричных вычислений, поэтому такие сети не требуют длительного обучения;
- от числа элементов ассоциативной сети зависит число образцов, которые могут быть сохранены.

6. РЕКУРРЕНТНЫЕ СВЯЗИ

Задача: описание сетей с рекуррентными связями.

Цели: понять, для чего предназначены рекуррентные сети и в чем заключаются их преимущества по сравнению с другими сетями при решении задач определенного типа.

6.1. Последовательность

Последовательность – это цепочка образцов, имеющих отношение к одному и тому же объекту. Например, последовательность может состоять из букв, образующих слово, или из слов, образующих предложение. Иметь дело с последовательностями не так уж просто, поскольку они могут иметь разную длину. Одним из способов обработки предложений с помощью фиксированного числа входных элементов является использование скользящего окна, как показано на рис.6.1. Недостатком этого подхода является то, что последовательность необходимо делить на сегменты, в результате чего зависимость между находящимися далеко друг от друга словами теряется.

Чтобы сразу обрабатывать все предложение, потребуется нейронная сеть с архитектурой, допускающей обработку самого длинного из ожидаемых предложений. Эта проблема решается путем использования сети с рекуррентными связями [7].

Рисунок 6.1 – Предложение “Большая собака нашла большую кость”.

6.2. Обратное распространение ошибки

Сеть с обратным распространением ошибок, является сетью с прямой связью, что означает ограничение направления связей: все связи должны идти в одном направлении – от входного слоя к выходному. Но сеть с обратным распространением ошибок не

обязательно должна быть сетью с прямой связью, она может иметь и рекуррентные связи, когда элемент посылает сигналы себе или другим элементам того же слоя или более низких слоев.

Алгоритм работы рекуррентной сети с обратным распространением ошибок представляет собой непосредственную модификацию соответствующего алгоритма работы сети с прямой связью. На рис. 6.2 показана сеть со связями, по которым сигналы возвращаются обратно от выходного слоя к входному, а рис. 6.3 иллюстрирует работу этой сети в течение двух шагов во времени.

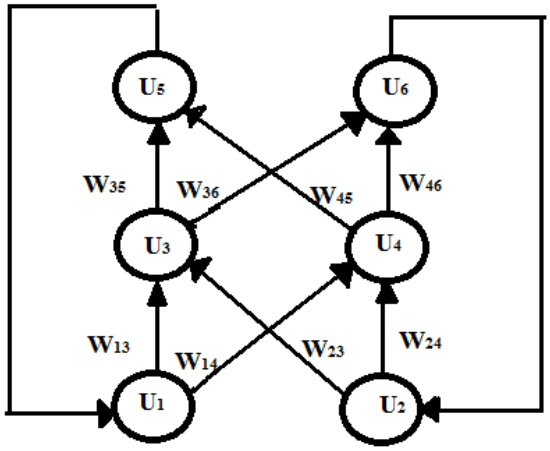


Рисунок 6.2 – Рекуррентная сеть с обратным распространением ошибок.

В ходе обучения на вход сети подается образец и выполняется прямой проход. Для каждого шага времени рассматривается своя копия сети и для каждого шага времени вычисляются ошибки точно так же, как в стандартной сети с обратным распространением ошибок. Изменения весовых значений, вычисленные для каждого экземпляра сети, перед **корректировкой суммируются. Наборы весовых значений всех экземпляров сети должны оставаться одинаковыми.**

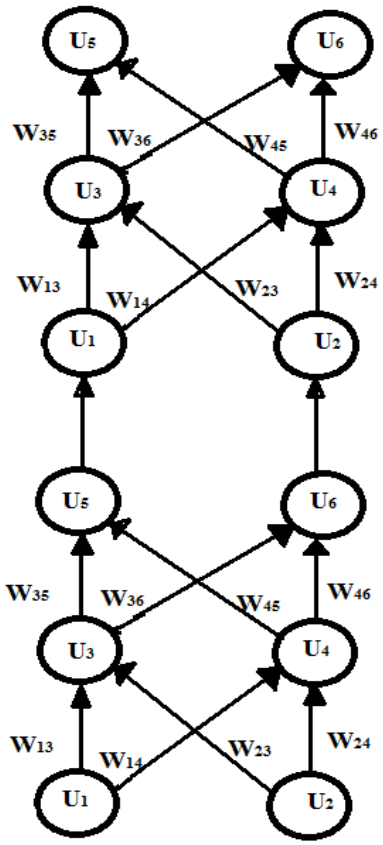


Рисунок 6.3 – Развернутая версия сети, показанной на рис.6.2.

6.3. Простая рекуррентная сеть

В своем докладе Джордан представил одну из наиболее ранних форм рекуррентной сети. Сеть Джордана имеет связи, идущие обратно от выходного к входному слою, и некоторые элементы входного слоя имеют связи, возвращающиеся обратно к этим элементам. Архитектура сети показана на рис. 6.4. Сеть способна обучиться выполнению задач, зависящих от последовательности состояний. Такая сеть может быть обучена с помощью алгоритма обратного распространения ошибок.

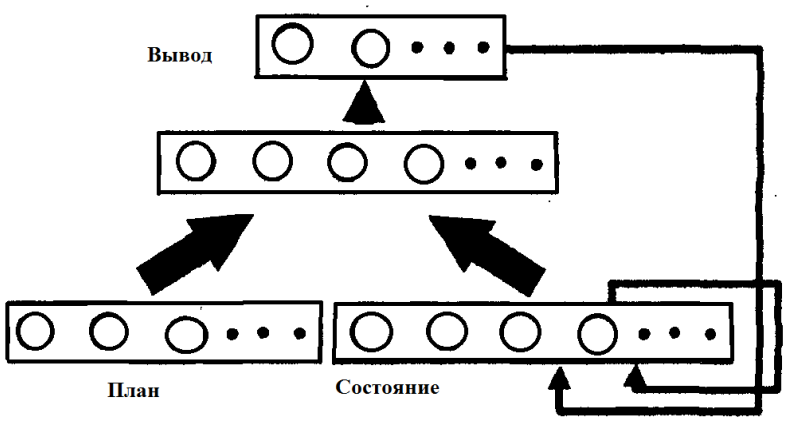


Рисунок 6.4 – Сеть Джордана.

Из – за рекуррентных связей в сети Джордана поведение сети формируется введенными ранее входными данными. Сеть обладает некоторыми свойствами краткосрочной памяти. Еще одной очень популярной рекуррентной сетью с подобными свойствами краткосрочной памяти является так называемая простая *рекуррентная сеть*.

7. ДРУГИЕ МОДЕЛИ СЕТЕЙ

Задача: описание других моделей сетей.

Цели: понять, что такое метод модельной “закалки” и как он используется в

нейронных сетях;

принципы построения вероятностной нейронной сети;

что такое “машина Больцмана”.

7.1. Сети, использующие статистический подход, вероятностная нейросеть

До сих пор все рассмотренные нами модели нейронных сетей использовали *детерминированные* алгоритмы обучения. “Детерминированные” здесь означает, что при необходимости обновления весовых значений или значения активности элемента соответствующие величины изменений могут быть определены в результате непосредственного прямого вычисления. Даже в сети Хопфилда, где обновляемый элемент выбирается случайно, сами изменения являются детерминированными, поскольку если элемент уже выбран, новое значение его активности может быть выполнено вполне однозначно. В противоположность этому, для *стохастического* алгоритма обучения в любой момент времени мы остаемся сневедении относительно того, как изменится

состояние сети. Другими словами, по информации о текущем состоянии предсказать следующее состояние оказывается невозможным [8].

7.2. Метод модельной “закалки”

Метод модельной “закалки” является методом, часто используемым при решении проблем оптимизации. Решение рассматриваемой проблемы представляется в виде решении задачи минимизации некоторой функции стоимости.

Метод модельной “закалки” базируется на аналогии с процессом закалки металла. Когда металл подвергается закалке, он нагревается почти до точки плавления, а потом медленно охлаждается до комнатной температуры. Процесс закалки делает металл более гибким, так что становится возможным придать ему нужную форму без изломов. Когда металл нагрет до высокой температуры, атомы двигаются хаотически. Если металл быстро охладить, атомы застынут в случайных положениях. Если же металл охладить медленно, атомы будут стремиться выстроиться регулярным образом. Поэтому основой всего процесса является управление графиком снижения температуры.

Аналогия с функцией оптимизации показана на рис. 7.1:

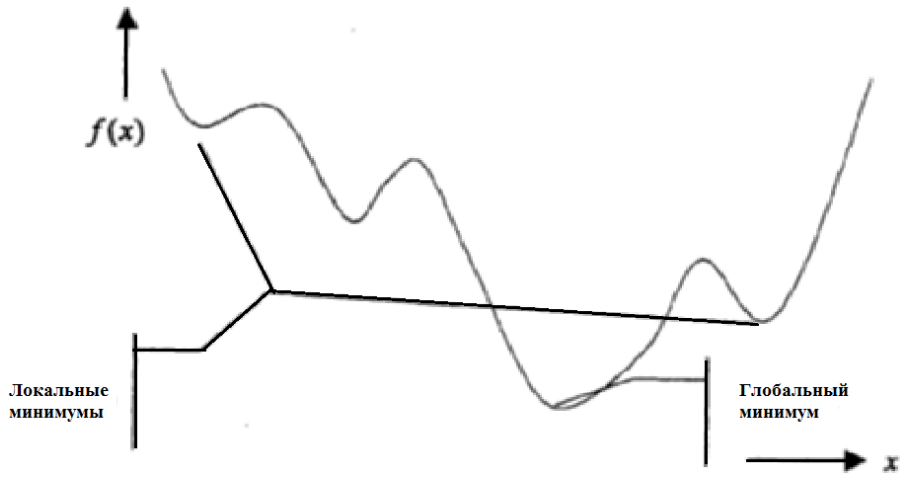


Рисунок 7.1 – Локальные минимумы и глобальный минимум.

Метод модельной “закалки” сначала ухудшает работу сети, чтобы потом ее усовершенствовать.

Единого и окончательного алгоритма модельной “закалки” нет, но принцип, на котором базируется метод модельной “закалки”, заложен в основу алгоритма

Метрополиса. Основной процедурой этого алгоритма является случайный выбор части системы для изменения. Изменения всегда принимаются, если уменьшается глобальная энергия системы, а если наблюдается рост энергии, то изменения принимаются с вероятностью *p*, задаваемой формулой

$$p = \exp(-\frac{\Delta E}{T})$$

где ΔE - изменение энергии, а T обозначает температуру [8].

7.3. Машина Больцмана

Машина Больцмана представляет собой нейронную сеть, использующую идею модельной “закалки” для обновления состояния сети. В своей базовой форме машина Больцмана является сетью Хопфилда, использующей для обновления состояния элемента сети стохастический процесс. Энергия сети Хопфилда была определена следующей формулой:

$$E = -\frac{1}{2}\sum_j \sum_i s_j s_i w_{ij}$$

где s обозначает состояние элемента сети. Если элементj изменяет состояние на величину Δs_j, то изменение энергии будет равно

$$\Delta E = -\Delta s_j \sum_i s_i w_{ij}$$

Изменение энергии может также быть выражено в виде:

$$\Delta E = -2 s_j \sum_i s_i w_{ij}$$

Для вычисления вероятности принятия предложенного изменения состояния обычно используется функция – сигмоид:

$$p = \frac{1}{1 + \exp(-\frac{\Delta E}{T})}$$

Поэтому вероятность перехода элемента в новое состояние оказывается равной:

$$p = \frac{1}{1 + \exp(2s_j \frac{\sum s_i w_{ij}}{T})}$$

Машина Больцмана использовалась для решения ряда проблем оптимизации, включая проблему коммивояжера.

Машина Больцмана может иметь и скрытые элементы. Видимые элементы. Видимые элементы могут быть разделены на входные и выходные. Пример соответствующей архитектуры изображен на рис. 7.3:

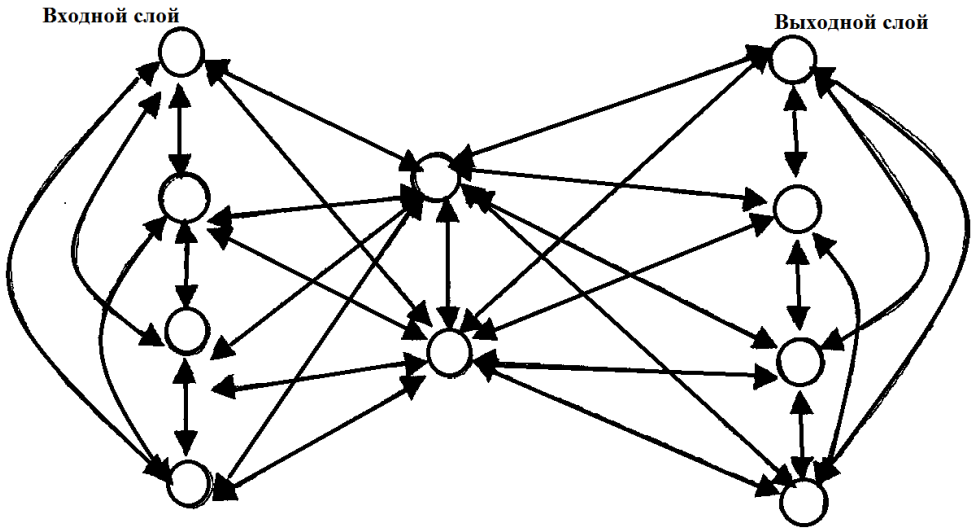


Рисунок 7.3 – Пример машины Больцмана с входными и выходными слоями.

Элементы имеют двунаправленные связи со всеми другими элементами, кроме элементов входного слоя, которые не связаны непосредственно с элементами выходного слоя. Сеть с такой архитектурой можно подвергнуть управляемому обучению. В ходе обучения выходные и входные элементы закрепляются. По завершении обучения закрепляются только входные элементы, и сеть выполняет некоторое число итераций до тех пор, пока в результате не установятся значения выходных элементов.

Процесс обучения управляемой сети Больцмана разделяется на две фазы: фазу фиксации, в ходе которой входные и выходные элементы закрепляются входным и целевым образцами, и фазу свободного выполнения, в ходе которой закрепляются только входные элементы. Сетевая статистика собирается на протяжении обеих фаз и используется затем для того, чтобы обновить весовые значения [9].

7.4. Алгоритм минимизации функции

Рассмотрим алгоритм модельной “закалки” для минимизации действительнозначной функции $f(x)$ двоичного вектора x:

- случайным образом выбираются начальный вектор *x* и начальное значение*T*;

- создается копия *x* с именем x_{new} и случайным образом выбирается компонента x_{new} для изменения. Бит выбранной компоненты переключается;

- вычисляется изменение энергии;

- если изменение энергии меньше нуля, устанавливается $x = x_{new}$. Иначе с помощью функции однородного распределения плотности вероятностей выбирается случайное число между 0 и 1. Если случайное число меньше, чем $\exp(-\Delta E/T)$, устанавливается $x = x_{new}$.

- если было выполнено заданное число *(M)* изменений вектора *x*, для которых значениеf уменьшилось, или было выполненоN изменений *x* с момента последнего изменения температуры, то устанавливается $T = \alpha T$.

- если минимальное значениеf не уменьшается уже в течение некоторого определенного фиксированного числа L последних итераций, то процесс останавливается, иначе происходит возврат к п. 2, и вычисления продолжаются.

Здесь M меньше N, а L обычно намного больше N. Понижение температуры управляется константой α, значение для которой, как правило, выбирается из диапазона от 0.8до 0.9999. Начальное значение*T* выбирается так, что для всех изменений энергии имеет место неравенство $\exp(-\Delta E/T) \geq 0.9999$.

8. СВЯЗЬ С ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ

Задача: введение в символьную парадигму искусственного интеллекта и связь с нейронными сетями.

Цели: понять, каково сегодняшнее состояние и цели изучения искусственного интеллекта;

главные составляющие интеллектуальных систем;

суть основных проблем, возникающих при разработке систем

понимания речи.

8.1. Введение

Нейронные сети могут применяться при решении самых разных технических проблем. Архитектура искусственных нейронных сетей не идет ни в какое сравнение с человеческим мозгом по сложности и размерам, но именно такие сети составляют абстрактные модели мозга.

Поэтому сразу же возникает вопрос: можно ли с помощью искусственных сетей реализовать “искусственный интеллект”? Искусственный интеллект является одной из важнейших дисциплин в области информатики, имеющей глубокие связи с математикой, психологией и даже философией.

Цель изучения искусственного интеллекта в широком смысле может быть описана как создание машин, выполняющих задачи, с которыми хорошо справляется человек, но которые нелегко запрограммировать с использованием традиционного подхода к вычислениям.

Исследователи искусственного интеллекта сосредоточились главным образом на том, что обычно называют символьной парадигмой, а также “традиционным искусственным интеллектом” или “классическим искусственным интеллектом”. Теперь существует новая форма искусственного интеллекта, основанная на коннекциях, т.е. нейронных сетях.

Чтобы развивать новый искусственный интеллект, нам нужно понять кое – что из традиционного искусственного интеллекта и, возможно, свести обе эти парадигмы вместе “под знаменем” некоторого общего искусственного интеллекта [10].

8.2. Знания и представления

Интеллект представляет собой концепцию, не поддающуюся строгому определению. Человеческий интеллект складывается из множества компонентов, среди

которых способность к обучению, органы чувств, позволяющие взаимодействовать с внешней средой, и объем знаний, который вообще не поддается оценке.

Традиционный искусственный интеллект заставляет нас верить, что теоретически возможно создать мозг – воплощение искусственного интеллекта – следует лишь создать правильный алгоритм, и мозг тут как тут. В рамках традиционного искусственного интеллекта изучение интеллектуальных систем концентрировалось вокруг нескольких ключевых вопросов – это представление данных, способность рассуждать и способность системы автоматически адаптироваться к изменению условий.

Интеллектуальная система воплощает знание. Когда мы говорим о знаниях, это могут быть фактические знания, необходимые, например, для участия в викторине, или практические знания, применяемые при замене автомобильного колеса. Это могут быть знания, называемые навыками, необходимые, например, при езде на велосипеде, или же это могут быть миллионы битов информации, которую мы называем здравым смыслом, позволяющим, например, не пролить воду из чашки при питье. Представление знания может быть явным и неявным.

Явное знание может быть установлено и инспектировано, например, в форме фактов.

Яблоко есть плод.

Кот есть животное.

Неявное знание передать непросто. Например, ребенку можно разъяснить общие принципы езды на велосипеде, включая необходимость вращать педали и направлять переднее колесо в сторону движения, но процедура не может быть записана так, чтобы ребенок мог ее прочитать, а затем использовать для того,чтобы поехать на велосипеде. В действительности знание того, как ехать на велосипеде, приобретается в результате проб и ошибок.

Традиционный искусственный интеллект складывается из обработки символов, а знания представляются с помощью символьных структур. Представления знания имеет несколько форм и различных уровней.

Исследователи искусственного интеллекта используют знания о мозге как руководство для того, чтобы строить искусственный интеллект, но теорию того, как человек представляет знания, формализовать очень тяжело. Существуют когнитивные

теории того, как представляет знания мозг человека, и именно такие теории служат основой для символьных представлений [11].

8.3. Рассуждения

Человек знает, как вести себя в новых ситуациях, поскольку он может анализировать свои знания и принимать адекватные решения, основанные на прошлом опыте и на прогнозировании возможных последствий таких решений. Символьная парадигма пытается отчасти имитировать такое поведение с помощью программ, включающих набор правил представления соответствующих умозаключений.

Совсем не сложно запрограммировать систему, работающую в соответствии с основными правилами логических заключений, но прежде, чем машина сможет демонстрировать нечто подобное реальному интеллекту, придется найти ответы на некоторые важные вопросы. Правила воплощают знания, а поэтому и вопросы оказываются связанными со знаниями.

Рассмотрим специальную задачу, требующую только знания устройства автомобиля. Экспертная система вовлекается в следующий диалог.

Клиент: Зачем на автомобилях устанавливаются ручные тормоза?

Экспертная система: Чтобы остановить движение автомобиля.

Клиент: Автомобиль ведь не всегда движется, когда ручной тормоз выключен?

Экспертная система: Это зависит от горизонтальности поверхности.

Клиент: Что дает энергию, заставляющую колеса автомобиля вращаться?

Экспертная система: Двигатель.

Клиент: Таким образом, если двигатель работает, то колеса будут вращаться?

Экспертная система: Если сцепление включено.

Клиент: что случится, если двигатель будет работать со включенным ручным тормозом и включенным сцеплением?

Экспертная система: Предполагается, что вы сначала должны отключить ручной тормоз.

Клиент: Да, но предположим, что я этого не сделал.

Экспертная система: ???

Ответ “Не знаю” на последний вопрос является приемлемым, если вы не эксперт по автомобилям, но выглядит смешным, если предполагается, что вы эксперт [12].

Понимание того, сколько требуется знаний для адекватного выполнения соответствующей задачи, является проблемой разработчика экспертной системы.

РЕШЕНИЕ ПРАКТИЧЕСКИХ ЗАДАЧ

Задачи

Задача№1

Вычислите комбинированный сетевой ввод для элемента на рис.4.1 и соответствующее выходное значение при использовании пороговой функции и выходного вектора $[0.7;\quad 2.5]$[7].

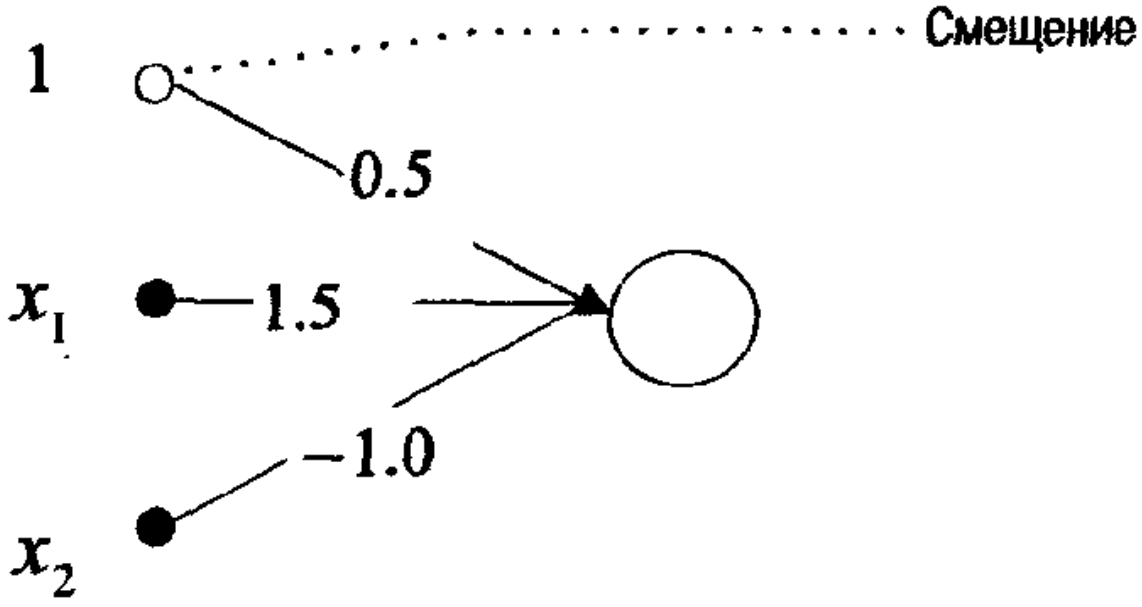


Рисунок 4.1 - Входные элементы н.с. со смещением

Задача№2

Вычислите выходное значение, использовав в качестве функции активности сигмоидальную функцию. Входной вектор $[0.7;\quad 2.5]$.[7].

Задача№3

Вычислите комбинированный ввод для сети с архитектурой показанной на рис.2.1, но с набором весовых значений $[-0.2; \quad 0.03; \quad 1.2]$и входным вектором $[0.7; \quad 2.5]$. [7].

Задача№4

Найти весовые коэффициенты для модели нейронной сети, подобно показанной на рис.2.1 и представляющей уравнение: $2\mathbf{x_2} = -4\mathbf{x_1} + 8$.[7].

Задача№5

На рисунке показаны скрытый и выходной слои сети с прямой связью. Вычислите ошибку для скрытого элемента **U** при условии, что значение его активности для обрабатываемого сетью образца равна 0.64. [7].

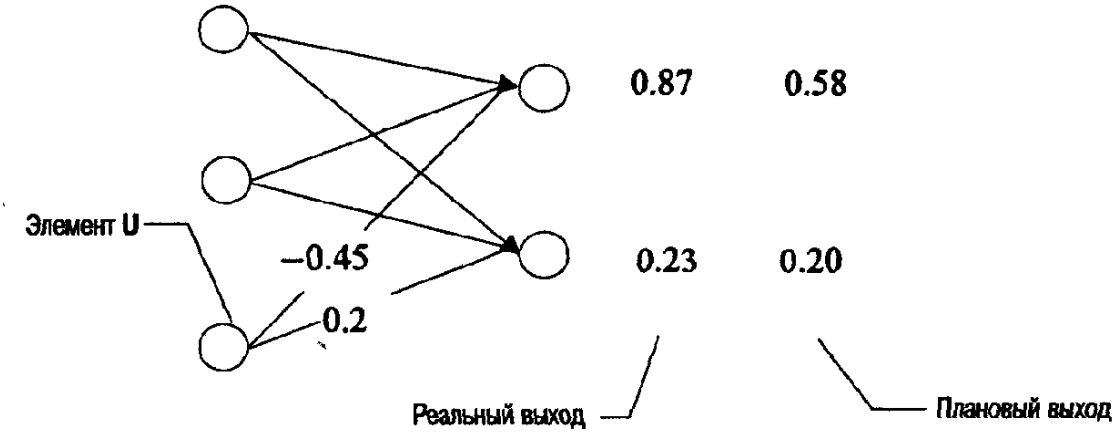


Рисунок 4.2 - Сеть с прямой связью

Задача№6

Найдите квадрат евклидового расстояния между векторами: $\mathbf{p} = [-2.3; \quad 1.4]$, $\mathbf{x} = [4.5; \quad 0.6]$.[7].

Задача№7

Найдите квадрат евклидового расстояния между векторами: $\mathbf{p} = [0.4; \quad 0.3; \quad 1.1; \quad 0.9]$,

$\mathbf{x} = [0.6; \quad 0.7; \quad -0.5; \quad 1.1]$.[7].

Задача№8

Сколько эпох потребуется для того, чтобы значение η уменьшилось до 0.1, при условии использования следующего правила обновления: $\eta(0) = 0.9$,

$\eta(n+1) = \eta(n) - 0.001$.[7].

Задача№9

Сеть SOFM имеет вид двумерной сетки размером 10x10, а радиус изначально задан равным 6. Выясните, как много элементов должны будут обновляться после 1000 эпох, если элемент-победитель размещен в правом нижнем углу сетки, а радиус меняется по правилу: $r = r - 1$, если номер текущей эпохи $mod\ 200 = 0$. Предполагается, что нумерация эпох начинается с 1. [7].

Задача№10

Определите весовую матрицу сети Хопфилда, соответствующую сохранению следующих двух векторов: $[-1; \quad 1; \quad -1]$,

$[1; \quad -1; \quad 1]$.[7].

Задача№11

Имеется следующая информация.

Если аккумулятор машины разряжен, то машина не заводится. Если машина Джона не заводится и текущее время оказывается позже 8 часов утра, то Джон опоздает на поезд. Однажды утром после 8 часов утра аккумулятор машины Джона оказался разряженным.

Используя логические правила вывода, покажите, что Джон опоздал на поезд.[7].

Задача№12

Продемонстрируйте порядок кодирования и декодирования бинарного дерева ((D(AN))(V(P(DN)))).[7].

Задача№13

Для предложенного ниже учебного множества выясните, сколько деревьев научится представлять сеть RAAM, если использовать это множество?[7].

(D(A(A(AN))))

((DN)(P(DN))

(V(DN))

(P(D(AN)))

((DN)V)

((DN)(V(D(AN))))

((D(AN))(V(P(DN)))))

Решение

Задача №1

Порядок элементов во входном векторе говорит о том, что $\mathbf{x_1} = 0.7$ и $\mathbf{x_2} = 0.5$. Таким образом, комбинированный ввод оказывается равным

$$0.5 + (0.7 * 1.5) + \big(2.5 * (-1)\big) = -0.95$$

Комбинированный ввод оказывается отрицательным, поэтому вывод равен 0.[7].

Задача №2

Активность в случае сигмоидальной функции вычисляется по формуле: $f\big(net_j\big) = \frac{1}{1+\exp\,(-net_j)}$. Комбинированный ввод равен -0.95, и подстановка этого значения в сигмоидальную функцию дает выходное значение 0.28. [7].

Задача №3

Если входной вектор обозначить **x**, а вектор весов – **w**, то комбинированный ввод элемента можно выразить в виде $net_j = \mathbf{xw}$, при условии, что входное вектор включает и значение активности элемента смещения. Добавляя к входному вектору значение активности элемента смещения, для комбинированного ввода в нашем случае получаем: $net_j = \big(1 * (-0.2)\big) + (0.7 * 0.03) + (2.5 * 1.2) = 2.82$. [8].

Задача №4

Для любой точки, лежащей на указанной прямой, весовые коэффициенты можно определить из уравнения: $w_0 + x_1 w_1 + x_2 w_2 = 0$. Отсюда получаем: $x_2 = -x_1 \frac{w_1}{w_2} - \frac{w_0}{w_2}$. Сравнивая члены полученного равенства с коэффициентами, указанными в условии примера, имеем:

$-\frac{w_1}{w_2} = -\frac{4}{2}$, $-\frac{w_0}{w_2} = \frac{8}{2}$. Таким образом, $w_0 = -8, w_2 = 2$.[8].

Задача №5

Сначала вычисляются ошибки для выходных элементов: $\delta_{выход\_1}$=(0.58-0.87)*(1-0.87)=-0.033, $\delta_{выход\_2}$=(0.2-0.23)*0.23*(1-0.23) = -0.005. Теперь ошибку распространения обратно к элементу **U**: δ_U=0.64*(1-0.64)*((-0.033*(-0.45))+(-0.005*0.2))=0.003. [8].

Задача №6

d(p,**x**)=$(-2.3-4.5)^2 + (1.4-0.6)^2$=46.9.

Задача №7

d(p,**x**)=$(0.4-0.6)^2 + (0.3-0.7)^2 + (1.1-(-0.5))^2 + (0.9-1.1)^2$=2.8.

Задача №8

0.1=0.9 – n0.001, =>n=(0.9-0.1)/0.001=800.

Задача №9

Если предположить, что счет начинается с 1, радиус будет уменьшаться на 1 после каждых 200 эпох. После 1000 эпох радиус окажется равным 1. Число изменяемых элементов будет равно четырем, включая элемент-победитель. [8].

Задача №10

$$W = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}[-1;\ 1;\ -1] + \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}[1;\ -1;\ 1] = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}$$

Диагональные элементы были обнулены.

Задача №11

В символьном виде информация может быть представлена следующим образом.

P: аккумулятор машины разряжен.

Q: машина не заводится.

R: время после 8 утра.

S: Джон опоздал на поезд.

Правило 1. P=>Q.

Правило 2. QandR=>S.

Известно, что P и R ИСТИНА. Задачей является доказательство S.

1. P - Дано.

2. R – Дано.

3. Q - Следует из шага 1 и правила 1, по правилу modusponens.

4. QandR – Следует из шага 3 и 2, по правилу интродукции И.

5. S – Следует из шага 4 и правила 2, правилу modusponens. [8].

Задача №12 и №13

Порядок кодирования показан в таблице 4.1,а порядок декодирования в таблице 4.2.

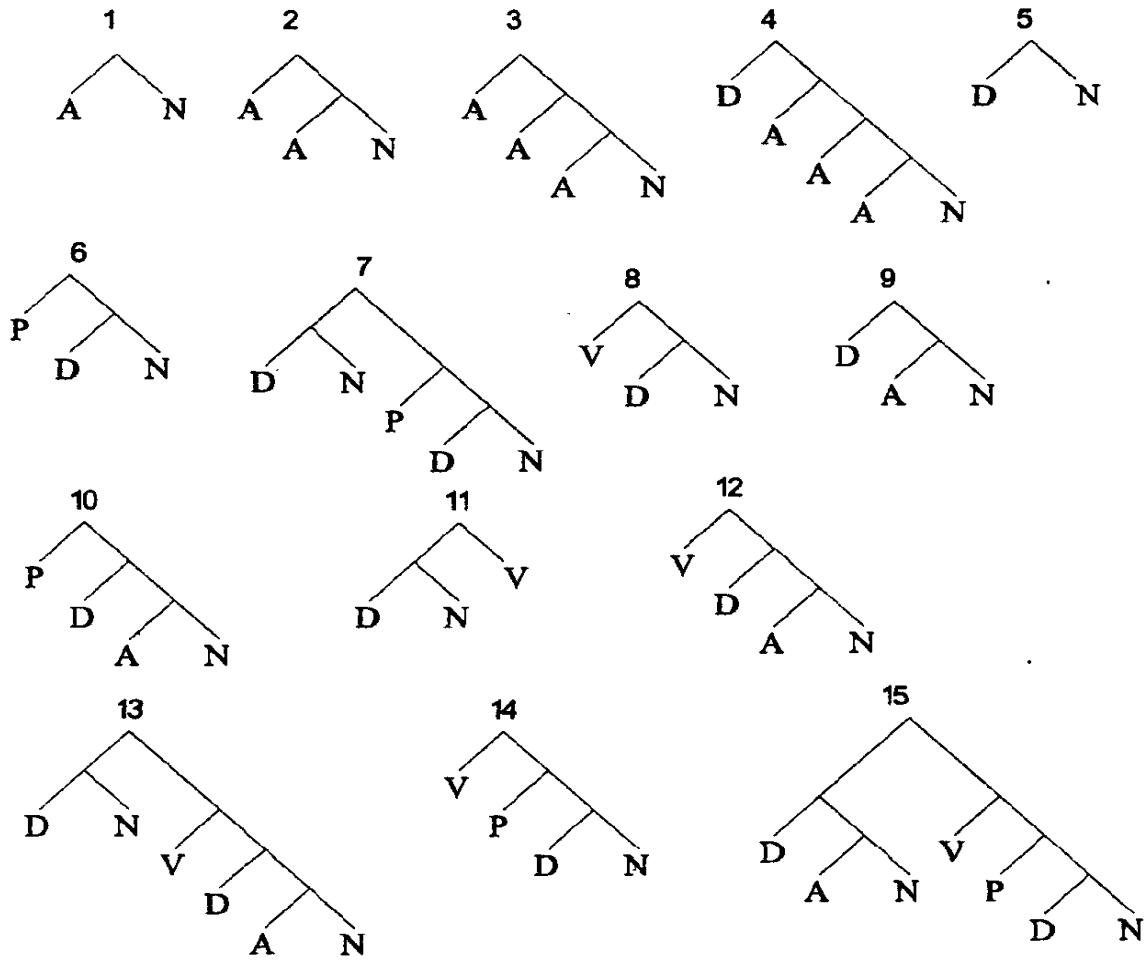
Весь соответствующий набор деревьев показан на рисунке 1. [8].

Левая ветвь	Правая ветвь	Скрытое представление
D	N	(DN)’
P	(DN)’	(P(DN))’
V	(P(DN)’	(V(P(DN)))’
A	N	(AN)’
D	(AN)’	(D(AN))’
(D(AN))’	(V(P(DN)))’	((D(AN))(V(P(DN))))’

Таблица 4.1 - Порядок кодирования бинарного дерева

Скрытое представление	Левая ветвь	Правая ветвь
(DN)’	D	N
(P(DN))’	P	(DN)’
(V(P(DN)))’	V	(P(DN)’
(AN)’	A	N
(D(AN))’	D	(AN)’
((D(AN))(V(P(DN))))’	(D(AN))’	(V(P(DN)))’

Таблица 4.2 - Порядок декодирования бинарного дерева



1. (A N)
2. (A (A N))
3. (A (A (A N)))
4. (D (A (A (A N))))
5. (P (D N))
6. ((D N)(P(D N)))
7. (V (D N))
8. (D (A N))
9. (P (D (A N)))
10. (D (A N))
11. ((D N) V)
12. (V (D (A N)))
13. ((D N) (V (D (A N))))
14. (V (P (D N)))
15. ((D (A N)) (V (P (D N))))

Рисунок 4.3 - Схема построения деревьев

9. СПИСОК ЛИТЕРАТУРЫ

1. AckleyD.H., HintonG.E. andSejnowskiT.J. AlearningalgorithmforBolzmannmachines. – CognitiveScience: 1985. - 147 – 169.
2. Галушкин А.И.,ЦыпкинЯ.З.Нейронные сети: история развития теории. — М.: ИПРЖР, 2001. -840 с.
3. Осовский С. Нейронные сети для обработки информации // Пер. с польского И.Д. Рудинского. М.: Финансы и статистика, 2002. - 344 с.
4. Copeland J. Artificial Intelligence, A Philosophical Introduction/ Oxford : Blackwell,1993.
5. Dean T., Alien J. and Aloimonos Y. Artificial Intelligence. Redwood City, CA: Benjamin/Cummings Publishing Company, Inc.,1995.
6. Каллан Р. Основные концепции нейронных сетей //Саутгемптонский институт И.Д. “Вильямс”. Москва: 2001.
7. КомарцоваЛ.Г., Максимов А.В.. Нейрокомпьютеры: Учебное пособие для вузов. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2004. – 400с.
8. Уоссермен Ф. Нейрокомпьютерная техника – Теория и практика, перевод на русский язык,Зуев Ю.А., ТоченовВ.А., 1992. – 184 с.
9. Хайкин С. Нейронные сети: полный курс, 2 издание.: Пер. с англ. – М.: Изд. дом «Вильямс», 2006. – 1104 с.
10. Медведев В.С. Потемкин В.Г. Нейронные сети MATLAB 6// Под общ.ред. К. т. н. Потемкина В.Г – ДИАЛОГ – МИФИ, 2002. – 496 с.
11. Лисс А.А., Степанов М.В. Нейронные сети и нейрокомпьютеры: Учебное пособие. – ГЭТУ. – СПб., 1997. – 64 с.
12. Горбань А.Н. Обучение нейронных сетей. – М.: СП «ParaGraph», 1990. – 160 с.