

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет радиофизики и электроники

Кафедра системного анализа

МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКОЙ НЕЙРОННОЙ СЕТИ

Курсовая работа студента IV курса

Поплетеева Андрея Михайловича

Руководитель: ст. преподаватель

Лутковский В.М.

Минск 2003

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b> .....	<b>2</b>
<b>1. ОДНОНАПРАВЛЕННЫЕ НЕЙРОННЫЕ СЕТИ</b> .....	<b>3</b>
1.1. ДЕТЕРМИНИРОВАННЫЕ ИНС И ИХ ОСОБЕННОСТИ .....	3
1.1.1. Многослойные сети .....	4
1.1.2. Проблемы применения детерминированных ИНС.....	5
1.2. ОДНОНАПРАВЛЕННЫЕ ИНС СО СТОХАСТИЧЕСКИМИ НЕЙРОНАМИ.....	5
1.2.1. Алгоритмы стохастического обучения. Методы статистической физики в теории нейронных сетей .....	6
1.2.2. <i>Bit Steam Networks</i> .....	8
1.2.3. Импульсные НС ( <i>spikes NN</i> ) .....	10
1.3. ВЫВОДЫ.....	11
<b>2. МАШИНА БОЛЬЦМАНА</b> .....	<b>12</b>
2.1. АРХИТЕКТУРА СЕТИ.....	12
2.2. ОБУЧЕНИЕ МАШИНЫ БОЛЬЦМАНА .....	13
2.3. МАШИНА БОЛЬЦМАНА ВЫСОКОГО ПОРЯДКА.....	14
2.4. ПРИМЕНЕНИЯ.....	14
<b>3. МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ В ЭЛЕКТРОННЫХ ПРИБОРАХ</b> 16	
3.1. МОДЕЛИРОВАНИЕ С ПОМОЩЬЮ ДЕТЕРМИНИРОВАННОЙ ИНС .....	16
3.2. МОДЕЛИРОВАНИЕ С ПОМОЩЬЮ МАШИНЫ БОЛЬЦМАНА .....	17
<b>4. ЗАКЛЮЧЕНИЕ</b> .....	<b>19</b>
<b>ЛИТЕРАТУРА</b> .....	<b>20</b>
<b>ПРИЛОЖЕНИЕ 1</b> .....	<b>22</b>
<b>ПРИЛОЖЕНИЕ 2</b> .....	<b>24</b>

## ВВЕДЕНИЕ

Искусственные нейронные сети (ИНС) возникли в результате попыток моделирования способности человека производить обработку информации (выполнять вычисления, распознавать образы, систематизировать данные и т.д.). На протяжении длительного времени их вычислительные способности рассматривались в качестве главного критерия качества. Ограниченные возможности однослойных нейронных сетей послужили причиной практически полного прекращения исследований в этой области до 80-х годов. Возрождение интереса к ИНС объясняется не только найденными эффективными алгоритмами обучения многослойных сетей, но и осознанием уникальных возможностей решать с их помощью многие трудно формализуемые задачи [1].

Детерминированные нейронные сети, типичным представителем которых служат персептроны, уже достаточно широко применяются на практике [2-9]. При этом их обучение методом обратного распространения ошибки не всегда дает желаемые результаты, что объясняется проблемой попадания в локальный минимум [1]. Установлено, что включение стохастических элементов в такие ИНС или использование стохастических алгоритмов обучения значительно расширяет их возможности [10].

Теория стохастического обучения и сетей стохастического типа была развита в 1983–1987 гг. рядом исследователей, использовавшими аналогию между поведением нейронных ансамблей и уже изученных термодинамических систем. Принцип случайности получил наибольшее развитие в машине Больцмана, которую относят к стохастическим нейронным сетям (СНС). Следует отметить, что деление на детерминированные и стохастические нейронные сети носит условный характер. Еще Ф. Розенблатт учитывал случайный характер связей в персептронах. Однако в целом СНС недостаточно изучены и используются на практике в меньшей степени, что подтверждается пробелами в учебных изданиях [7, 11]. Они имеют ряд отличительных особенностей в сравнении с детерминированными нейронными сетями и их применение обычно ограничивалось решением проблемы локального минимума. Однако возможности СНС этим далеко не исчерпываются. В последние несколько лет возрастает интерес к изучению импульсных (spiking) ИНС, наиболее близких к биологическому прототипу.

Цель работы – исследовать возможности СНС для моделирования стохастических процессов в электронных приборах. Известно, что для решения этой задачи можно применять обычные многослойные персептроны, представляющие собой универсальные аппроксиматоры [11-14]. В рамках курсовой работы предполагается исследовать как персептроны, так и новые модели ИНС [15-19].

# 1. ОДНОНАПРАВЛЕННЫЕ НЕЙРОННЫЕ СЕТИ

После двух десятилетий почти полного забвения интерес к искусственным нейронным сетям (ИНС) быстро вырос за последние несколько лет.

Все ИНС можно разделить на два больших класса: детерминированные и стохастические. Функционирование первых полностью определяется жесткими правилами, тогда как при обучении и функционировании вторых существенную роль играет элемент случайности.

В этом разделе будут рассмотрены особенности различных примеров как детерминированных, так и стохастических НС.

## 1.1. Детерминированные ИНС и их особенности

Первой формальной моделью нейронных сетей (НС) была модель Маккаллока-Питтса, уточненная и развитая Клини [1]. Важная историческая роль этой модели заключалась в установлении того, что НС способны выполнять логические операции и вообще любые преобразования, реализуемые дискретными устройствами с конечной памятью. Эта модель легла в основу теории логических сетей и конечных автоматов и активно использовалась психологами и нейрофизиологами при моделировании некоторых локальных процессов нервной деятельности.

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На его вход поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона. На рис. 1.1 представлена модель, реализующая эту идею.

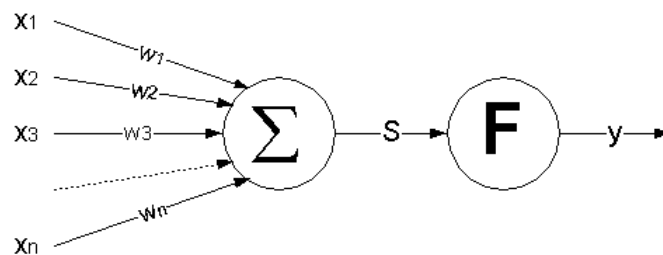


Рис. 1.1. Формальный нейрон

Здесь множество входных сигналов, обозначенных  $x_1, x_2, \dots, x_n$ , поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором  $\mathbf{X}$ , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес  $w_1, w_2, \dots, w_n$ , и поступает на суммирующий блок. Множество весов в совокупности обозначается вектором  $\mathbf{W}$ . Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, который мы будем называть  $S$ . В векторных обозначениях это может быть компактно записано следующим образом:

$$S = \mathbf{XW}$$

После этого взвешенная сумма входов преобразуется активационной функцией и дает выходной сигнал нейрона. В качестве активационной функции часто используется сигмоидная:

$$Y = \frac{1}{1 + e^{-S}}$$

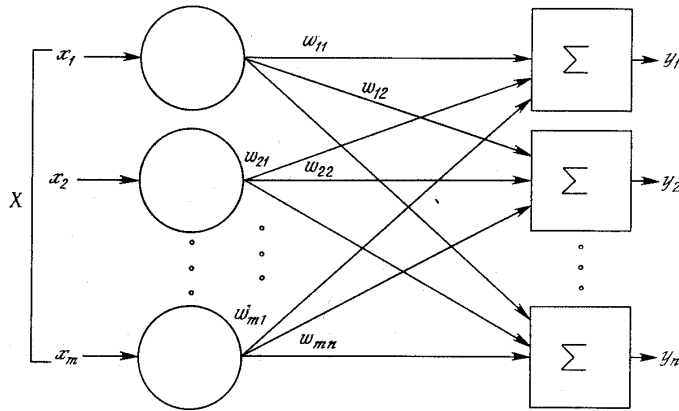


Рис. 1.2. Однослойная ИНС (перцептрон)

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, сила нейронных вычислений проистекает от соединений нейронов в сетях. Простейшая сеть состоит из группы нейронов, образующих слой, как показано на рис. 1.2.

### 1.1.1. Многослойные сети

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями. Хотя существует множество сетей различных конфигураций, послойная организация нейронов копирует слоистые структуры определенных отделов мозга. Оказалось, что такие многослойные сети обладают большими возможностями, чем однослойные, и в последние годы были разработаны алгоритмы для их обучения.

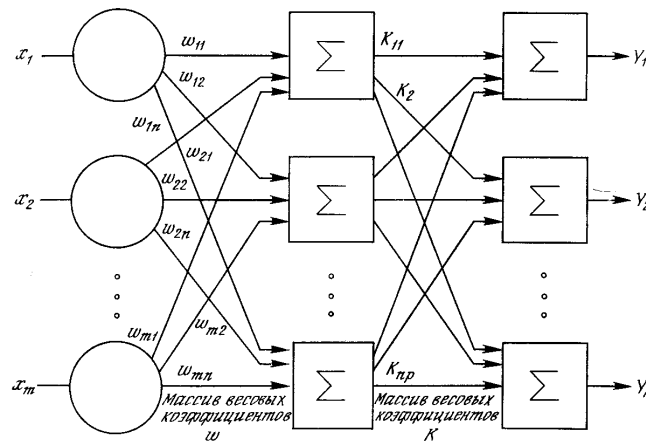


Рис. 1.3. Многослойная ИНС

Многослойные сети могут образовываться каскадами слоев. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рис. 1.3. Для обучения многослойных ИНС используется алгоритм обратного распространения ошибки и различные его вариации.

### ***1.1.2. Проблемы применения детерминированных ИНС***

ИНС используются в следующих задачах:

- Классификация образов
- Кластеризация/категоризация
- Аппроксимация функций
- Предсказание/прогноз
- Оптимизация
- Ассоциативная память

Известны и иные, более традиционные подходы к решению этих проблем, однако они не обладают необходимой гибкостью за пределами ограниченных условий. ИНС дают многообещающие альтернативные решения, и многие приложения выигрывают от их использования.

Но применение ИНС для некоторых задач также ограничено. В некоторых случаях становится невозможно правильно обучить ИНС. Причиной этого может быть паралич сети или попадание в область локального минимума ошибки.

Эти проблемы могут быть решены путем использования стохастических ИС.

### **1.2. Однонаправленные ИНС со стохастическими нейронами**

В отличие от детерминированных моделей, рассмотренных прежде, состояние нейрона стохастической ИНС зависит не только от взвешенной суммы входных сигналов, но и от некоторой случайной переменной величины.

В 1948 г. Шимбел и Рапорт сделали, по-видимому, первую систематическую попытку разработать и подвергнуть строгому анализу семейство сетей, организованных по статистическому принципу. Исходя, подобно Маккаллоку и Питтсу, из аксиоматического представления нейронов и их связей, Шимбел и Рапорт характеризуют сеть вероятностным распределением порогов, типов синапсов и источников связей. Затем выводится общее уравнение для вычисления вероятности срабатывания нейрона в данном месте в определенный момент времени в зависимости от предшествующей активности и параметров сети. Это уравнение применяется к нескольким конкретным классам сетей для определения возможности стационарной активности, а также изменений в распределении моментов срабатывания нейронов [20].

Моделирование статистически связанных сетей для исследования предполагаемых способностей к обучению было впервые с успехом выполнено Фэрли и Кларком в 1954 г. Несмотря на то что математическое исследование моделей Фэрли—Кларка или Рочестера не было проведено, они иллюстрируют удобный способ аксиоматизации сети (с помощью программы для вычислительной машины), которая открывает возможность исследования гипотез. Хотя ни один из этих экспериментов не привел к созданию достаточно совершенной системы, они все же представляют значительный исторический интерес. Механизм для обобщения образов, предложенный Фэрли и Кларком, по существу тождествен принцип построения простых перцептронов [21].

### 1.2.1. Алгоритмы стохастического обучения.

#### Методы статистической физики в теории нейронных сетей

К многослойным ИНС применяют детерминистские методы обучения. Но в некоторых случаях они не позволяют обучить ИНС. Причиной этого может быть паралич сети или попадание в область локального минимума ошибки. Стохастические методы обучения позволяют решить проблему локального минимума.

В случае стохастического обучения на каждом последующем шаге веса изменяют случайно, вычисляют ошибку и сохраняют изменения, ведущие к ее уменьшению.

Процедура стохастического обучения.

1. Параметрам ИНС присваивают произвольные малые значения. Выбирают очередной вектор обучающего множества и вычисляют фактический выход нейронной сети.
2. Вычисляют ошибку на текущем шаге:

$$\varepsilon = \sum_i (y_i^* - y_i)^2$$

3. Корректируют веса на небольшие случайные значения. Если коррекция уменьшает ошибку, то новые значения сохраняют. В противном случае оставляют старые значения.

На практике те параметры, которые ведут к увеличению ошибки, сохраняют с вероятностью, которая задается некоторой функцией распределения. В зависимости от вида этой функции распределения различают по крайней мере две модификации стохастического обучения: методы Больцмана и Коши.

Энергию сети можно определить, например, как

$$E = -\sum_{i < j} W_{ij} y_i y_j + \sum_j \Theta_j y_j \quad (2.1)$$

где  $y_i$  определяет состояние или выход бинарного нейрона. Если на нейроны не действует никакая внешняя сила, то вся система движется к минимуму энергии.

Однако каким бы образом ни была сформулирована энергия, машина стремится ее минимизировать путем случайного изменения оценки добавкой, которая принимается или отвергается. Поиск осуществляется по сравнению вероятностей распределения Больцмана.

Чтобы сеть не попала в локальный минимум, нежелательный для оптимизационных задач, применяется вероятностное правило срабатывания элементов

$$P_k = 1/[1 + \exp(-\Delta E_k/KT)] \quad (2.2)$$

где  $P_k$  – вероятность нахождения  $k$ -го элемента в единичном состоянии,  $T$  – регулируемый параметр обучения, называемый «температурой» системы, а  $K$  – постоянная, выбираемая в зависимости от задачи. Изменение энергии сети при возбуждении  $k$ -го нейрона определяется как

$$\Delta E_k = E_{k_{вык}} - E_{k_{вкл}} = \sum W_{ik} y_i - \Theta_k \quad (2.3)$$

Если оно положительно,  $k$ -й нейрон переключается на уменьшение энергии. При этом, как и во всех стохастических методах, резко падает вероятность получения ошибочного решения при попадании сети в локальный минимум. Это можно пояснить следующим образом.

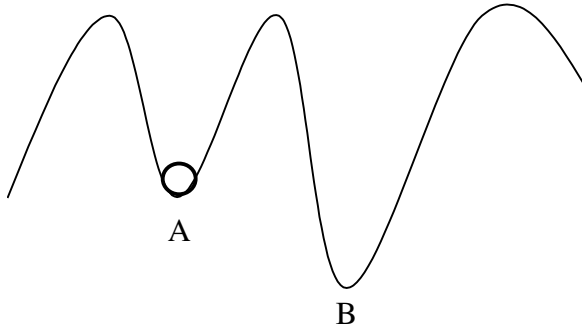


Рис. 2.2. Металлический шарик (аналогия)

В качестве аналогии рассмотрим металлический шарик в коробке с искривленным рельефом дна (рис. 2.2). Встряхивая коробку случайным образом можно перебросить шарик из одного минимума в другой.

Лучше начинать с сильных встряхиваний и постепенно уменьшать их амплитуду. Этот алгоритм получил наименование *имитации отжига* по аналогии с процессом отжига

кристаллических материалов (медленное охлаждение металла для получения низкоэнергетической кристаллической решетки). В процессе отжига распределение энергетических уровней описывается распределением Больцмана. Процесс достижения равновесия может быть длительным при низкой температуре и быстрым при высокой. Поэтому обучение начинают при высоких «температурах», а затем постепенно ее уменьшают и получают лучшее решение.

При высоких «температурах» вероятность в (2.2) приближается к единице и нахождение системы в высокоэнергетическом или в низкоэнергетическом состоянии практически равновероятны. При уменьшении «температуры» вероятность высокоэнергетических состояний уменьшается по сравнению с низкоэнергетическими. В результате система переводится в состояние с минимальной энергией. Если возникает тепловое равновесие, то относительные вероятности нахождения системы в двух глобальных состояниях подчиняются распределению Больцмана.

Показано, что охлаждение должно быть очень медленным. Для попадания в глобальный минимум скорость уменьшения температуры от начальной температуры  $T_0$  до текущего значения  $T(t)$  должна быть обратно пропорциональна логарифму времени  $t$ :

$$T(t) = T_0 / \log(1+t) \quad (2.4)$$

Используя при случайном поиске параметров нейронной сети распределение Коши вместо распределения Больцмана, можно увеличить вероятность больших шагов:

$$P(x) = T(t) / [T^2(t) + x^2] \quad (2.5)$$

Оба распределения представлены на рис. 2.3.



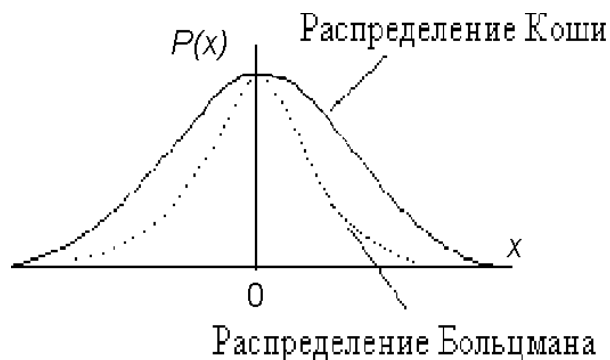


Рис. 2.3. Распределения приращений весов

$$x_c = \eta T(t) \operatorname{tg}[P(x)] \quad (2.6)$$

где  $\eta$  – скорость обучения.

Несмотря на улучшение, достигаемое с помощью метода Коши, время обучения может оказаться все еще слишком большим по сравнению с методом обратного распространения ошибки. Для ускорения процесса обучения может быть использован метод *искусственной теплоемкости*, определяемой как средняя скорость изменения температуры с целевой функцией. Этот метод также использует в качестве прототипа термодинамические процессы.

При отжиге металла происходят фазовые переходы, связанные с дискретными изменениями уровней энергии. При каждом фазовом переходе может иметь место резкое изменение величины, называемой теплоемкостью. Изменения теплоемкости происходят из-за попадания системы в локальные энергетические минимумы. Искусственные нейронные сети проходят аналогичные фазы в процессе обучения. На границе фазового перехода искусственная теплоемкость может скачкообразно измениться. Эта псевдотеплоемкость определяется как средняя скорость изменения температуры с целевой функцией. В минимумах, когда небольшое изменение температуры приводит к большим изменениям целевой функции, происходит скачкообразное уменьшение искусственной теплоемкости. Чтобы гарантировать сходимость, скорость уменьшения температуры должна замедляться. При всех других температурах можно без риска уменьшать ее быстрее.

В примере с шариком в коробке сильная начальная встряска делает среднюю величину целевой функции фактически не зависящей от температуры, т. е. теплоемкость близка к константе. Аналогично при очень низких температурах система «замерзает» в точке минимума, так что теплоемкость снова близка к константе.

### 1.2.2. Bit Steam Networks

В СНС информация обычно нечеткая и зашумленная и каждому нейрону не обязательно очень точно учитывать свои входы. Выход сети должен не зависеть от состояний отдельных нейронов, а определяться общим состоянием сети. Это можно осуществить путем построения полностью распределенного представления в сети. Распределенное представление сделает сеть устойчивой в

Искусственная температура как функция времени принимает вид

$$T(t) = T_0 / (1+t)$$

что позволяет резко сократить время обучения. Можно также получить аналитическое выражение для величины изменения синаптической связи  $x_c$ , решая выражение (2.5) относительно шага  $x$ :

зашумленной среде и улучшит ее способность к обобщению. Это предположение приводит к естественно<у представлению данных с помощью битовых потоков, в которых действительные значения представляются изменением частоты единиц и нулей.[15, 22]

Идея представления битовыми потоками (БП) была использована для разработки стохастического нейрона. Такой дизайн привлекателен тем, что он является полностью цифровым и при этом не требует больших площадей полупроводника для реализации.

На самом деле, отдельный нейрон занимает не больше места, чем простой цифровой счетчик. Результатом является очень быстрая и компактная НС с легко анализируемой функциональностью. Это также означает, что, в отличие от аналоговых реализаций, большие сети могут быть построены без накопления шумовой ошибки.

Результат простоты устройства – некоторое отклонение от функциональности стандартного сигмоидного нейрона. Ограничение весовых коэффициентов позволяет сети в процессе обучения сформировать полностью распределенное представление.

Все сигналы, обрабатываемые БП-нейронами – это действительные значения, представленные стохастическими битовыми потоками (СБП). СБП – это просто случайная двоичная последовательность, которая принимает значения 1 или 0. Можно различными способами использовать СБП для представления аналоговых величин. Основной метод – сопоставить величине вероятность того, что БП равна единице. Такой БП называется последовательностью Бернулли.

В монополярном представлении действительное значение  $r \in [0, 1]$  представляется последовательностью Бернулли с вероятностью  $p = r$ . Чтобы представить действительное значение в диапазоне  $[-1, 1]$  с помощью монополярного представления, нужен дополнительный бит знака. В биполярном представлении действительная величина  $r \in [-1, 1]$  представляется последовательностью Бернулли с вероятностью  $p = (r+1)/2$ .

В монополярном представлении для проведения операций умножения нужен только простой логический элемент И. Для биполярного представления эта операция также может быть легко реализована элементом ИСКЛЮЧАЮЩЕЕ ИЛИ.

БП-нейрон – это устройство, которое проводит очень простые операции над своими входными потоками. Все входные потоки комбинируются с соответствующими весовыми потоками и затем взвешенные биты суммируются. Конечный результат сравнивается с пороговым значением. Если сумма не меньше порога, выход нейрона приравняется 1, иначе 0.

Соответственно двум представлениям, существуют две различные версии СБП-сетей. Биполярному представлению соответствует XNOR-СБП, а монополярному – AND-СБП.

Эти варианты сетей имеют несколько различные свойства. AND-СБП лучше подходят для сетей прямого распространения и рекуррентных сетей, которые используют обучение, а XNOR-СБП удобны для имитации отжига в рекуррентных сетях.

В [15] показано, что СВП-сети по своим возможностям подобны сетям из обычных сигмоидных нейронов, при этом возможно обучение таких сетей для вычислений большого масштаба. Кроме того, для СВП-сетей возможна аппаратная реализация алгоритма обратного распространения ошибки.

### 1.2.3. Импульсные НС (*spikes NN*)

Большинство существующих моделей НС, которые используют непрерывные сигналы, имеют весьма отдаленное отношение к биологическим НС, которые кодируют сигналы посредством последовательностей импульсов (*spike trains*). Импульсные НС интересны как с биологической точки зрения, так и в качестве инструмента, хорошо подходящего для обработки частично зашумленных сигналов.

В табл. 1 приведены особенности обычной и импульсной НС.

Существует несколько способов кодирования сигналов импульсными последовательностями. Один из вариантов приведен на рис. 1.4. Здесь входные импульсы модулируются по времени и амплитуде (слева), формируют постсинаптические импульсы (центр) и, наконец, интегрируются сомой (справа). Когда потенциал (сплошная линия) достигает порога (пунктирная линия), генерируется импульс и потенциал уменьшается. Там же изображен пример потенциала.

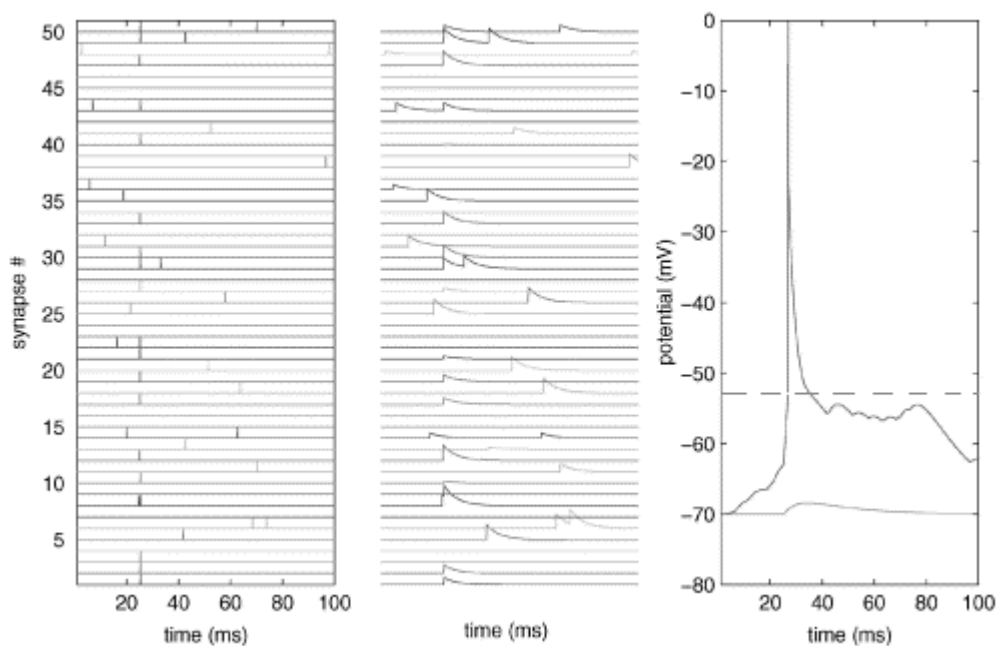


Рис. 1.4. Работа импульсной НС.

Табл. 1. Сравнение характеристик обычной и импульсной НС

Обычная НС	Импульсная НС
Простая модель нейрона	Сложная модель нейрона
Не учитывается время	Учитывается зависимость от времени
Активность представляется непрерывной функцией.	Активность преобразуется в последовательность импульсов.
Почти полносвязные	Малое количество связей
Используются все нейроны	Используется несколько нейронов
В основном, обучение с учителем	Самообучение

### 1.3. Выводы

По сравнению с методом обратного распространения ошибки, в котором веса корректируются в направлении минимизации ошибки, методы стохастического обучения работают значительно дольше, так как многие шаги делаются в неверном направлении. Очевидно, что при этом длительность поиска оптимальных параметров нейронной сети зависит от начальных условий и очень быстро растет при увеличении числа настраиваемых параметров. Поэтому стремятся сузить область поиска до разумных пределов.

Тем не менее, стохастический метод обучения достаточно универсален: он применим как к уже изученным нейронным сетям, так и ко многим другим. Более того, основное преимущество стохастических НС и соответствующих алгоритмов обучения – высокая вероятность нахождения глобального, а не локального минимума энергетического ландшафта.

## 2. МАШИНА БОЛЬЦМАНА

В данном разделе рассматриваются стохастические методы контролируемого обучения и стохастическая ИНС, названная машиной Больцмана.

### 2.1. Архитектура сети

В общем случае в машине Больцмана случайность используют как при обучении, так и при моделировании. Рассматриваемая модель состоит из случайно соединенных нейронов особого вида (рис. 2.1), при этом любая совокупность нейронов может быть использована для ввода информации и определена как входная, тогда как любая другая совокупность нейронов может использоваться для вывода информации. Такую ИНС можно рассматривать как стохастический вариант сети Хопфилда, но с двунаправленными связями, пропускающими информацию в обоих направлениях.

Нейроны, обозначенные на рис. 2.1 кружками, могут находиться в одном из двух состояний: +1 (включен) и -1 (выключен). Вероятность перехода в состояние -1 или +1 определяется взвешенной суммой сигналов  $s$  на входах нейронов:

$$P\{x=1\} = F_c(2s), \quad P\{x=-1\} = 1 - F_c(2s), \quad \text{где } F_c(s) = [1 + \exp(-s/c)]^{-1}$$

Параметр  $c$  в выражении, описывающем сигмоидальную активационную функцию, определяет ее крутизну. Если сеть состоит из  $n$  нейронов, то ее состояние будет определяться вектором  $x(t)$  с компонентами  $x_i(t)$ , описывающими состояние каждого индивидуального нейрона. Как и в сети Хопфилда, веса связей между нейронами симметричны:  $w_{ij} = w_{ji}$  и  $w_{ii} = 0$  для всех  $i$ .

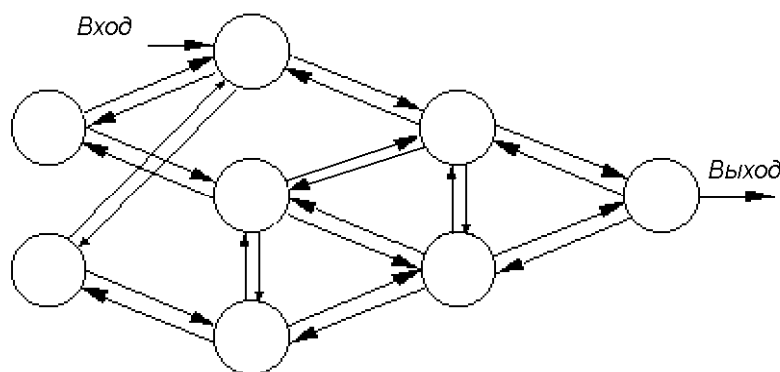


Рис. 2.1. Машина Больцмана

Изменения состояний нейронов могут происходить синхронно или не синхронно по некоторому правилу. Алгоритм несинхронного изменения состояний отдельных нейронов может быть следующим:

```

i:= random (n);
si :=  $\sum_{j=1}^n w_{ij}x_j(t) - \vartheta_i$ ;
if random(1) < Fc(2si) then xi(t+1):=1 else xi(t+1):= - 1;
for j:=1 to n do
  if i ≠ j then xj(t+1):= xj(t)
end.

```

Заметим, что при  $c > 0$  вероятности оказаться в состояниях  $-1$  или  $+1$  для каждого из нейронов конечны. Это означает, что в принципе не существует устойчивого состояния. Рассматривают асимптотическую сходимость последовательности состояний к некоторому предельному состоянию. Для анализа динамики смены состояний машины Больцмана применяют математический аппарат цепей Маркова.

## 2.2. Обучение машины Больцмана

Процедура обучения машины Больцмана, описанная в [23], состоит из следующих шагов:

1. Вычислить закрепленные вероятности.
  - 1.1. придать входным и выходным нейронам значения обучающего вектора;
  - 1.2. предоставить сети возможность искать равновесие;
  - 1.3. записать выходные значения для всех нейронов;
  - 1.4. повторить шаги от 1.1 до 1.3 для всех обучающих векторов;
  - 1.5. вычислить вероятность  $P^*$ , т.е. по всему множеству обучающих векторов вычислить вероятность того, что значения обоих нейронов равны единице.
2. Вычислить незакрепленные вероятности.
  - 2.1. предоставить сети возможность "свободного движения" без закрепления входов или выходов, начав со случайного состояния;
  - 2.2. повторить шаг 2.1 много раз, регистрируя значения всех нейронов;
  - 2.3. вычислить вероятность  $P_{ij}$ , т.е. вероятность того, что значения обоих нейронов равны единице.
3. Скорректировать веса сети следующим образом:  $\Delta W_{ij} = n[P_{ij}^+ - P_{ij}^-]$ , где  $\Delta W_{ij}$  - изменение веса  $W_{ij}$ ,  $n$  - коэффициент скорости обучения.

Алгоритм обучения рассмотренной сети имеет типичные недостатки, связанные с неполным равновесием при ограниченном времени сбора статистик. Из-за стохастичности обучение занимает гораздо больше времени по сравнению с градиентными методами, поскольку многие шаги выполняются в неверном направлении. Имеющаяся аппаратная реализация, однако, смягчает этот недостаток по крайней мере для небольших сетей.

Для практической реализации сеть имеет ряд особенностей. Во-первых, обучающий алгоритм локален и каждый синапс должен иметь несколько цепей для подбора статистики. Из этого следует низкая точность подбора весов. Однако для ряда простых проблем достаточно 3–7-битовой точности. Во-вторых, алгоритм требует случайных событий, т. е. каждый нейрон нуждается в независимом их источнике. Еще одной проблемой является реализация двунаправленных связей, которые не так просты, как однонаправленные цепи.

Все это затрудняет применение рассмотренной модели, поэтому стохастические НС не получили широкого распространения.

### **2.3. Машина Больцмана высокого порядка**

В обычной машине Больцмана имеются скрытые нейроны, функция энергии задается членами первого и второго порядка переменных, т.е. соединениями до второго порядка и аппроксимирующее распределение является граничным распределением видимых нейронов.

При ближайшем рассмотрении теоретической базы обычной МБ обнаруживается отсутствие ответов на фундаментальные вопросы. В частности, для заданного образцового распределения не установлена уникальность аппроксимирующего распределения, получаемого при обучении. Если существует несколько относительных минимумов, получаются различные решения, различные веса в зависимости от стартовой точки, т.е. начальных весов и параметров обучающего правила. Имея структуру, заданную несколькими скрытыми нейронами и связями, невозможно охарактеризовать выученное распределение.

Эти вопросы удовлетворительно решаются с помощью машины Больцмана высокого порядка (МБВП).

МБВП – это разновидность обычной МБ, в которой имеются соединения высокого порядка и нет скрытых нейронов [17].

Можно доказать сходимостъ обучающего алгоритма для МБВП и уникальность выученного распределения, которое соответствует максимально правдоподобной оценке весов соединений модели.

МБВП (без скрытых нейронов) и обычная МБ второго порядка со скрытыми нейронами являются двумя способами улучшения емкости МБ второго порядка без скрытых нейронов. МБ второго порядка без скрытых нейронов дает вероятностную модель, которая соответствует ограниченному классу распределений. Она не может решать задачи, в которых существенное значение имеют корреляции высоких порядков. Следовательно, необходимо добавить скрытые нейроны или соединения высокого порядка, которые в некоторой степени эквивалентны. Фактически, любое распределение может быть представлено как граничное распределение видимых нейронов МБ со скрытыми нейронами и, подобно этому, любое распределение может быть записано как распределение, заданное МБВП без скрытых нейронов.

В [17] также показано, что МБВП обладает свойствами сходимости, которые не содержатся в МБ со скрытыми нейронами, в частности, в обычной МБ второго порядка.

### **2.4. Применения**

Машина Больцмана используется для задач, в которых прямые математические вычисления невозможны или слишком длительны, но их можно реализовать при помощи нейронной сети. В машине Больцмана получил свое наибольшее развитие принцип случайности, впервые использованный Метрополисом [1]. Он позволяет выводить ИНС из локальных минимумов энергетического рельефа, означающих ложное решение. В дальнейшем

стохастический фактор был использован не только для преодоления локальных минимумов, но и для всего процесса обучения.

Примером применения стохастических нейронных сетей служат оптимизационные задачи. Обычные методы в таких случаях не работают или работают неудовлетворительно. Тогда вместо точного решения задачи ищется оценка, приемлемая в условиях дефицита времени и информации. Поиск глобальной оценки и есть сфера машины Больцмана. Оценка ищется по минимуму энергии, которая определяется в соответствии с априорной информацией или целями.

Введение отличной от нуля температуры в динамику нейросети улучшает ее свойства при использовании в качестве ассоциативной памяти, так как система перестает “чувствовать” мелкие локальные минимумы, отвечающие ложным образам. Однако за это приходится платить неточностями при воспроизведении образов вследствие отсутствия полной стабилизации системы в точке минимума.

Известны также примеры применения машины Больцмана для решения классических персептронных задач, таких, как задача «исключающего ИЛИ», обнаружение симметрии во входном образе и т. д., а также для распознавания речи.



### 3. МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ В ЭЛЕКТРОННЫХ ПРИБОРАХ

Большинство полупроводниковых приборов (диодов и транзисторов) работает в динамически неравновесных режимах. При этом в переходах возникают шумы, обусловленные различными факторами:

- неодинаковое число электронов и дырок, проходящее через переход в единицу времени. Это высокочастотные дробовые шумы;
- тепловой шум сопротивлений эмиттера, базы, коллектора;
- поверхностными явлениями у переходов. Это низкочастотные шумы;
- флуктуациями концентраций подвижных носителей заряда из-за нерегулярности процесса рекомбинации. Это низкочастотные рекомбинационные шумы.

Шумы в транзисторах имеют стохастический характер и ограничивают их чувствительность к слабому сигналу. Для повышения стабильности и чувствительности прибора шумы стремятся минимизировать.

Тем не менее, шумы полупроводниковых приборов находят и полезное применение в различных генераторах случайных сигналов, которые совершенно необходимы в активно развивающейся науке криптографии.

Пример сигнала, снятого с шумового диода показан на рис. 3.1.

Представляет интерес возможность моделирования такого сигнала с помощью нейронных сетей. В данном разделе исследуется возможность применения для решения этой задачи двух различных видов НС: обычного многослойного персептрона и стохастической НС (машины Больцмана).

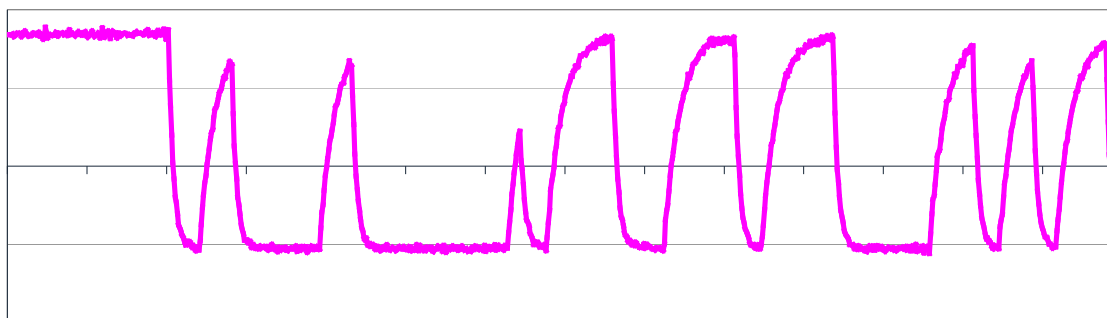


Рис. 3.1. Сигнал, снятый с шумового диода

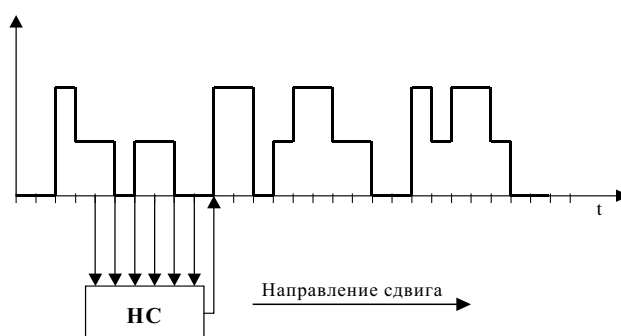
#### 3.1. Моделирование с помощью детерминированной ИНС

В данном варианте задача сводится к уже достаточно освещенной в литературе (например, [24-26]) задаче предсказания сигнала многослойным персептроном.

Алгоритм предсказания временных рядов с помощью детерминированной НС заключается в следующем..

1. Сигнал квантуется по времени и дискретизируется на  $N$  уровней.
2. Строится НС. На ее входы подается предыстория (от 3 до нескольких сотен отсчетов), с выхода снимается следующее значение.

3. НС обучается на отрезке моделируемого ряда.
4. Производится собственно моделирование. На входы сети подается некоторый начальный набор значений. С выхода снимается предсказанное значение.
5. Выходное значение сети включается в предысторию, т.е. подается на один из входов НС. Вычисляется выход сети и процедура повторяется. Схема процесса приведена на рис. 3.2.



*Рис. 3.2.* Схема моделирования сигнала с помощью детерминистической НС.

В данной работе для моделирования использовалась трехслойная НС с пятью входами. Обучение сети производилось методом обратного распространения ошибки. Сигнал дискретизировался на 8 уровней в интервале  $[-1; 1]$ .

Моделирование проводилось несколько раз, т.к. в зависимости от начальных весов связей НС результаты обучения оказывались различными.

Обучающая выборка содержала 200 пар. После 2000 итераций обучения минимальная среднеквадратичная ошибка имела значение 0.875187. Исходный текст программы для среды Matlab приведен в приложении 1.

Исходный и смоделированный сетью сигнал, а также их распределения приведены на рис. 3.3.

### **3.2. Моделирование с помощью машины Больцмана**

Как уже упоминалось, обучение машины Больцмана требует значительных вычислительных и временных ресурсов и является отдельной непростой задачей. Поэтому в данной работе приведен только пример, показывающий принципиальную возможность использования СНС для моделирования стохастических сигналов.

Для моделирования сигнала с помощью машины Больцмана предлагается использовать схему, аналогичную приведенной выше для детерминированных сетей.

Созданная машина Больцмана имела пять входных нейронов для предыстории, один выходной и пять скрытых нейронов. Моделировался двухуровневый сигнал со значениями 1 и 0. В процессе вычисления выхода сети искусственная температура изменялась от 16 до 0.5 с коэффициентом 0.5.

Один из сгенерированных сетью сигналов изображен на рис. 3.4

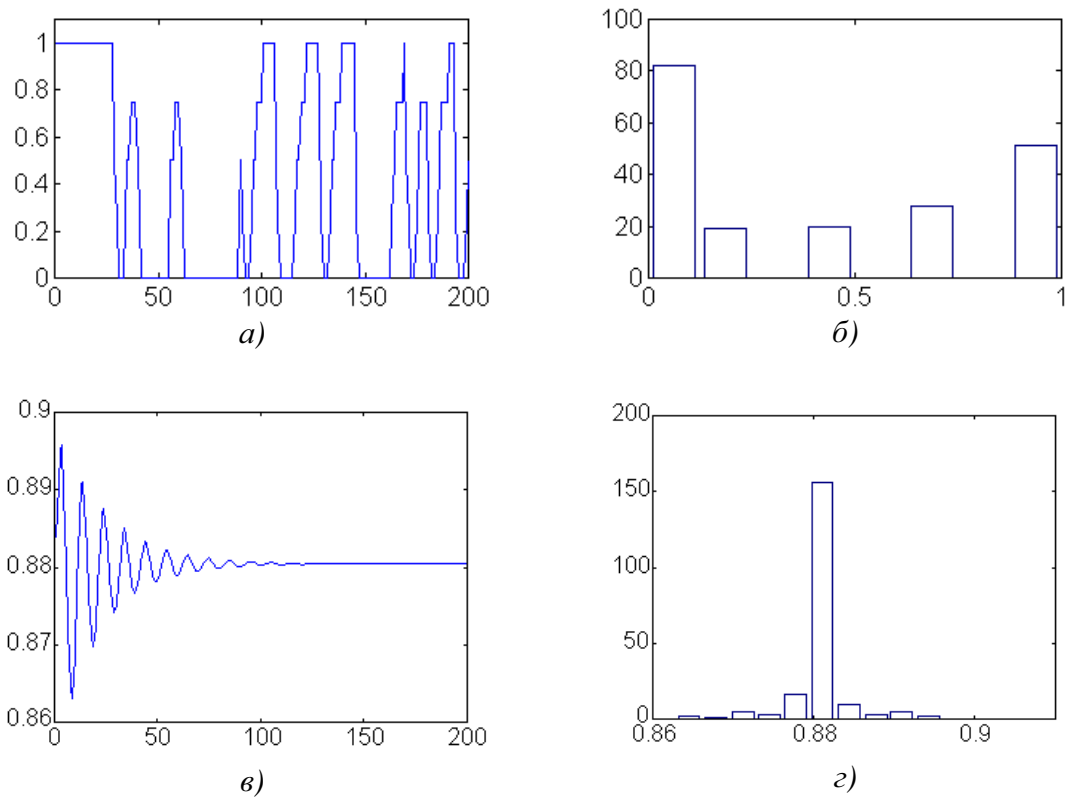


Рис. 3.3. Моделирование сигнала с помощью многослойного персептрона.  
 а) исходный сигнал; б) распределение исходного сигнала;  
 в) смоделированный сигнал; г) распределение смоделированного сигнала

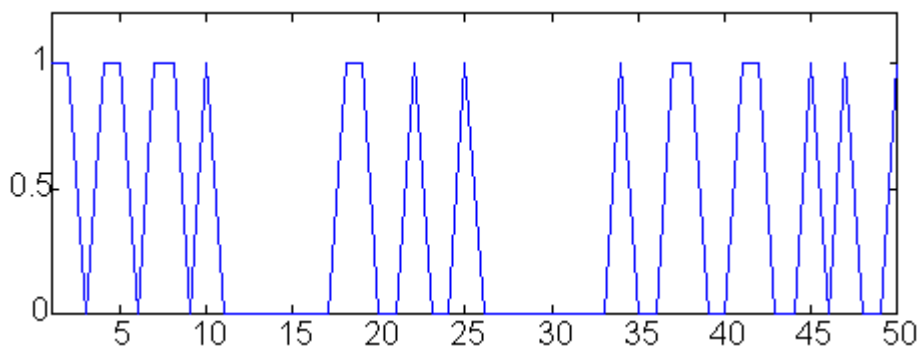


Рис. 3.4. Пример сигнала, сгенерированного стохастической НС

Исходный текст программы для моделирования стохастического сигнала с помощью машины Больцмана в среде Matlab приведен в приложении 2.

#### 4. ЗАКЛЮЧЕНИЕ

Результаты работы показывают, что обычные детерминистические сети малопригодны для моделирования стохастических сигналов. В этом случае одна из основных полезных особенностей НС – способность к обобщению – превращается в ее недостаток. Пытаясь “обобщить” стохастический входной сигнал, сеть просто выходит на некоторый постоянный уровень, что ярко демонстрируется рис. 3.3, в.

Использование для решения описанной задачи стохастических НС представляется более перспективным. К сожалению, реализация этого способа осложняется тем фактом, что обучение СНС требует больших временных и вычислительных ресурсов. Тем не менее, вполне возможно сохранить на диск конфигурацию сети (веса, пороги), обученной для моделирования конкретного сигнала, и в дальнейшем использовать НС без предварительного обучения.

В работе показана принципиальная возможность моделирования стохастического сигнала с помощью СНС.

Рассмотренный подход представляется многообещающим, и требует дальнейшей проработки. Прежде всего, следует реализовать алгоритмы обучения СНС и исследовать возможность сети моделировать сигнал с такими же характеристиками, что и использованный при обучении.

Финансовая поддержка проводимых исследований осуществлялась из средств НИР 533/18. Результаты работы докладывались на 60-й научной конференции студентов и аспирантов БГУ.

## ЛИТЕРАТУРА

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Пер. с англ. Ю. А. Зуева и В. А. Точенова. М.: Мир, 1992. 184 с.
2. Bishop M. Neural Networks for Pattern Recognition. Oxford: Clarendon Press, 1997. 477 p.
3. Ежов А. А., Шумский С. А. Нейрокомпьютинг и его применение в экономике и бизнесе / Под ред. проф. В. В. Харитоновна. М.: МИФИ, 1998. 224 с.
4. Головкин В. А. Нейроинтеллект: теория и применение. Кн. 1: Организация и обучение нейронных сетей с прямыми и обратными связями. Брест: Изд-во БПИ, 1999. 264 с.
5. Головкин В. А. Нейроинтеллект: теория и применение. Кн. 2: Самоорганизация, отказоустойчивость и применение нейронных сетей. Брест: Изд-во БПИ, 1999. 228 с.
6. Aart E. H. I., Ten Eikelder H. M. M. Neural networks. Eindhoven: TUE, 1999.
7. Галушкин А. И. Теория нейронных сетей. Кн. 1: Учеб. пособие для вузов / Под общ. ред. А. И. Галушкина. М.: ИПРЖР, 2000. 416 с.
8. Осковский С. Нейронные сети для обработки информации / Пер. с пол. И. Д. Рудинского. М.: Финансы и статистика, 2002. 344 с.
9. Лутковский В.М. Нейронные сети. Конспект лекций. Мн.: БГУ, 2003. 100 с.
10. Hangartner R.D., Cull P. Probabilistic computation by Neuromine Networks // BioSystems. 2000. Vol. 58. P. 167–176.
11. Медведев В. С., Потемкин В. Г. Нейронные сети. MATLAB 6 / Под общ. ред. В. Г. Потемкина. М.: ДИАЛОГ-МИФИ, 2002. 496 с.
12. Cybenko G. Approximations by superpositions of a sigmoidal function // Math Control Signals Systems. 1989. V. 2. P. 303–314.
13. Hornik K. Approximations capabilities of multilayer feedforward networks // Neural Networks. 1991. Vol. 4. P. 251–257.
14. Belli M.R., Conti M., Crippa P., Turchetti C. Artificial neural networks as approximators of stochastic processes // Neural Networks. 1999. Vol. 12. P. 647–658.
15. Zhao J., Shawe-Taylor J., Van Daalen M. Learning in Stochastic Bit Stream Neural Networks. // Neural Networks. 1996. Vol. 9. P. 991–998.
16. Rodriguez R. ....Noisy Spiking Neurons and Networks: ... // BioSystems. 1998. Vol. 48. P. 187–194.
17. Abizuri F.X., D'Anjou A., Grana M., Lozano J.A. Convergence Properties of High Order Boltzmann machines // Neural Networks. 1996. Vol. 9. P. 1561–1567.
18. Albizuri, F.X. Danjou, A. Grana, M. Torrealdea, J. Hernandez, M.C. The high-order Boltzmann machine: learned distribution and topology // IEEE Transactions on Neural Networks, 1995. V 6, №: 3, P. 767-770.
19. Thathachar, M.A.L. Arvind, M.T. Global Boltzmann perceptron network for online learning of conditional distributions // IEEE Transactions on Neural Networks, 1999, Volume: 10, Issue: 5, P. 1090-1098.
20. Shimbel, A. Rapoport, P. A Statistical Approach to the theory of the Central Nervous System // Bull. Math. Biophys, 1948. Issue 10, P. 41-55
21. Farley, B.G. Byerk, W.A. Simulation of Self-Organising systems by Digital Computer. // Trans. IRE Proffes. Group on Inform. Theory, Sept. 1954.
22. Card, H. C. Dynamics of stochastic artificial neurons // Neurocomputing, 2001. Issue: 41, P. 173-182.
23. Hinton O.E., Sejnowski T.J. 1986. Learning and relearning in Boltzmann machines. In Parallel distributed processing, vol.- 1. pp. 282-317. Cambridge, MA: MIT Press.

24. *Гордеев А. Н.* Применение нейросетевых алгоритмов к анализу временных рядов. Сб. тр. 57-й науч. конф. студ. и асп. В 3 ч. Мн.: БГУ. 2000. Ч.1. С. 280–283.
25. *Головко В. А., Савицкий Ю. В.* Использование нейронных сетей для прогнозирования хаотических временных рядов. Вестн. Брестского государственного технического университета. Сер. Физика, математика, химия. 2001. № 5. С. 78–82.
26. *Головко В. А., Савицкий Ю. В., Маньяков Н. В. и др.* Методы анализа хаотических процессов. Вестн. Брестского государственного технического университета. Сер. Физика, математика, химия. 2001. № 5. С. 75-78.

## ПРИЛОЖЕНИЕ 1

```
% Предсказание стохастического сигнала многослойным персептроном
%
% (с)2003, А. Поплетеев

nDiscr = 8; %Количество уровней дискретизации
count = 200; %Количество обучающих пар
inCount = 5; %Число входов
outCount = 1; %Число выходов

%Параметры обучения сети
updateEpochs = 25;
maxEpochs = 20;
errorGoal = 0.1;
learningRate = 0.03;

%Чтение обучающего множества из файла
[inp, out] = readIset(inCount, outCount, count, nDiscr);

%Инициализация и обучение сети
[w1,b1,w2,b2,w3,b3] =
initff(inp,inCount,'logsig',3*inCount,'logsig',outCount,'logsig');
params = [updateEpochs maxEpochs errorGoal learningRate];
[w1,b1,w2,b2,w3,b3,te,tr] =
trainbp(w1,b1,'logsig',w2,b2,'logsig',w3,b3,'logsig',inp, out, params);
disp('Press any key...');
pause;

%Проверка обучения
plot(out);
title('Исходный сигнал');
axis([0 count 0 1.1]);
disp('Press any key...');
pause;

hist(out,nDiscr);
title('Распределение исходного сигнала');
disp('Press any key...');
pause;

nnOut = gen_mlp(w1,b1,'logsig',w2,b2,'logsig',w3,b3,'logsig',inp(:,1), 50)
plot(nnOut);
title('Сгенерированный сигнал');
disp('Press any key...');
pause;

hist(nnOut,nDiscr);
title('Распределение сгенерированного сигнала');

function res = discrete(x, n)
% res = discrete(x, n)
% Params:
% x - исходная аналоговая величина
% n - количество уровней квантования
%
% Интервал [-1; 1] разбивается на n уровней.
% Возвращается значение, к которому x находится ближе
%
% (с)2003, А. Поплетеев
%
nn=n/2;
res=round(nn*x)/nn;
```

```

function nnOut = gen_mlp(w1,b1,f1,w2,b2,f2,w3,b3,f3,inp0,n);
%   nnOut = gen_mlp(w1,b1,f1,w2,b2,f2,w3,b3,f3,inp0);
%   Производит моделирование сигнала многослойным персептроном.
%   inp0 - начальные входы
%   n - количество генерируемых значений
%   (c) 2003, А. Поплетеев
%
nnOut=zeros(n,1);

inCount = size(inp0, 1);
inp = inp0;
for i=1:n,
    out = simuff(inp, w1,b1,f1,w2,b2,f2,w3,b3,f3)
    for j=1:inCount-1,
        inp(j)=inp(j+1);
    end;
    inp(inCount)=out;
    nnOut(i)=out;
end;

function [inp, out] = readlset(inCount, outCount, pairCount, nDiscr);
%
%   [inp, out] = readlset(inCount, outCount, pairCount);
%   Считывает из файла и возвращает обучающие пары входов/выходов
%   Params:
%   inCount    Количество входов
%   outCount   Количество выходов
%   pairCount  Количество обучающих пар
%   Results:
%   inp        Матрица входов
%   out        Матрица выходов
%
%   ©2003, А. Поплетеев

%Чтение обучающего множества из файла
finp = fopen('c:\progra~1\matlab\work\inputs.dat');
if (finp== -1),
    disp('Can't read learning set inputs');
end;
fout = fopen('c:\progra~1\matlab\work\outputs.dat');
if (fout== -1),
    disp('Can't read learning set outputs');
end;

inp_tmp = fread(finp, 'float32');
out_tmp = fread(fout, 'float32');
fclose(finp);
fclose(fout);

inp = zeros(pairCount, inCount);
for i=1:pairCount,
    for j=1:inCount,
        inp(i,j)=inp_tmp((i-1)*inCount+j);
    end;
end;
inp=inp-min(min(inp));
inp=inp/(max(max(inp)));
inp=discrete(inp, nDiscr)';

out = zeros(pairCount, outCount);
for i=1:pairCount,
    for j=1:outCount,
        out(i,j)=out_tmp((i-1)*outCount+j);
    end;
end;
out=out-min(min(out));
out=out/(max(max(out)));
out=discrete(out, nDiscr)';

```



## ПРИЛОЖЕНИЕ 2

```
%Моделирование стохастического сигнала с помощью машины Больцмана
%
% (с) 2003, А. Поплетеев
%

nToGenerate = 50; %Количество генерируемых сетью отсчетов
inCount = 5;      %Количество входных нейронов
outCount = 1;     %Количество выходных нейронов
hidCount = 5;     %Количество скрытых нейронов
count = inCount+outCount+hidCount;

%Загрузка весов и порогов из файлов
w = loadwghs('c:\progra~1\matlab\work\weights.dat');
b = loadthrs('c:\progra~1\matlab\work\threshld.dat');

%Задание случайной предыстории
ins = zeros(inCount,1);
for i=1:inCount,
    ins(i) = round(rand);
end;

disp('Testing...');
output = zeros(nToGenerate, outCount);

params = [16 0.5 0.5 50]; %Параметры НС
for i=1:nToGenerate,
    %Вычисление выхода сети
    outs = simub(ins,w,b,outCount,params);
    %Сдвиг массива предыстории
    for j=1:inCount-1, ins(j)=ins(j+1); end;
    ins(inCount)=outs(1);
    outputs(i,:)=outs;
end;

plot(outputs);
axis([1 nToGenerate 0 1.2]);

function w = loadwghs(fName);
%Загружает из указанного файла и возвращает веса сети.
%
%    w = loadwghs(fName);
%

fid = fopen(fName);
if fid==-1,
    disp('Can''t open file');
end;
w = fread(fid, 'float32');
fclose(fid);

n = sqrt(size(w,1));
w = reshape(w, n, n);

function b = loadthrs(fName);
%Загружает из указанного файла и возвращает пороги нейронов сети.
%
%    b = loadthrs(fName);
%

fid = fopen(fName);
if fid==-1,
    disp('Can''t open file');
end;
```

```

b = fread(fid, 'float32');
fclose(fid);

function [news] = bupdate(fixedCount, s, w, b, c)
%bupdate осуществляет асинхронное обновление машины Больцмана
%
%   [news] = bupdate(fixedCount, s, w, b, c)
%
%   fixedCount - количество неизменяемых нейронов
%   c         - температура
%
%   Возвращает вектор нового состояния сети
%

count = size(w,1);
news = s;
%Выбирается случайный нейрон
n = round((count-fixedCount-1)*rand)+fixedCount+1;
%Взвешенная сумма входов
z = w(n,:)*news-b(n);

r = rand;
if r>bsigm(z,c),
    news(n)=1;
else
    news(n)=0;
end;%if

function r = bsigm(z,c)
%bsigm Активационная функция машины Больцмана
%
%   r = bsigm(z,c)
%
%   z - взвешенная сумма входов
%   c - температура
%
r = 1/(1+exp(-z/c));

function [out] = simub(x,w,b,outCount,params)
%simub Вычисляет выход машины Больцмана
%
%   [out] = simub(x,w,b,outCount,params)
%   x - входной вектор (столбец)
%   w - матрица весов
%   b - столбец порогов
%   outCount - количество выходных нейронов
%   params - параметры МБ:
%       params(1) - начальная температура
%       params(2) - коэффициент уменьшения температуры
%       params(3) - конечная температура
%       params(4) - обновлений при каждом значении температуры
%   Возвращает:
%       out - выходы сети
%
%   See also: learnb
%
% (c) 2003, А. Поплетеев
%

N=params(4);
inpCount = size(x,1);
count = size(w,1);

%Инициализация состояний нейронов
s=zeros(count,1);
for i=1:count,
    if rand>0.5, s(i)=1; else s(i)=0; end;

```

```

end; %for
s(1:inpCount)=x;

p = zeros(count);

%Поиск равновесия путем имитации отжига.
c = params(1);
while c>params(3),
    for i=1:N,
        s = bupdate(inpCount, s, w, b, c);
    end;%for i
    c = c*params(2);
end;%while

p = zeros(count);
for n=inpCount+1:inpCount+outCount,
    z = w(n,:)*s-b(n);
    if bsigm(z,c)>0.5,
        p(n)=1;
    else
        p(n)=0;
    end;
end;

out = p(inpCount+1:inpCount+outCount);

```