

Министерство образования Республики Беларусь  
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (БГУ)**

УДК 621.37

№ госрегистрации 2003835

УТВЕРЖДАЮ

Проректор по научной работе,  
д-р хим. наук С.К. Рахманов

“        ” декабря 2003 г.

ОТЧЕТ

О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

**РАЗРАБОТКА МЕТОДОВ НЕЙРОСЕТЕВОГО МОДЕЛИРОВАНИЯ  
ПРОЦЕССОВ В СТОХАСТИЧЕСКИХ СИСТЕМАХ**

(заключительный)

г.б. НИР №533/18

Декан ф-та радиофизики  
и электроники  
д-р физ.-мат. наук, профессор

С.Г. Мулярчик

Зав. кафедрой  
системного анализа  
д-р физ.-мат. наук, профессор

В.В. Апанасович

Научный руководитель  
мл. науч. сотр., аспирант

П.В. Назаров

Куратор проекта  
ст. науч. сотр.

В.М. Лутковский

Минск 2003

## СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель работы, мл. науч. сотр,  
аспирант

П.В. Назаров  
(реферат, введение, разд. 2,  
подразд. 1.1, 1.2, 3.2, 3.3,  
заключение)

Ответственный исполнитель,  
студент 5-го курса  
Нормоконтролер

А.М. Поплетеев  
(подразд. 1.3, 3.1, 3.4)  
Т.Н. Долгая

## РЕФЕРАТ

Отчет содержит 55 страниц, 6 таблиц, 21 рисунок, 29 формул и 34 источника.

### НЕЙРОННЫЕ СЕТИ, ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ, НЕПАРАМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ, АППРОКСИМАЦИЯ, СТОХАСТИЧЕСКИЙ СИГНАЛ

Объектом исследования являются искусственные нейронные сети, а также методы нейросетевого моделирования детерминистических и случайных процессов протекающих в стохастических системах.

Целью исследования является разработка и обоснование моделей процессов в стохастических системах, а также методов идентификации систем с помощью нейросетевого моделирования.

Основные методы базируются на применении теории нейронных сетей, математической статистики, методов оптимизации, имитационного моделирования и вычислительной математики.

В процессе исследований разработаны модели резонансного переноса энергии в системах мембранных протеинов, а также флуктуационные переходы в генераторах шума.

Усовершенствована методика анализа и идентификации параметров процессов. Предложенный подход включает в себя использование имитационного и нейросетевого моделирования. Причем имитационная модель используется только единожды. Непосредственный анализ проводится с помощью нейронной сети. Полученное на экспериментальной задаче ускорение составило порядка  $10^4$  раз.

В результате исследования был разработан метод моделирования стохастических сигналов детерминистической нейронной сетью со случайными входами. Рассмотрен также подход на основе стохастических сетей. Приведены результаты моделирования процессов в полупроводниковом генераторе шума.

## СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	6
ВВЕДЕНИЕ .....	7
<b>1 НЕЙРОСЕТЕВОЙ ПОДХОД К МОДЕЛИРОВАНИЮ .....</b>	<b>9</b>
1.1 Непараметрическое моделирование .....	9
1.1.1 Модель "черного ящика" .....	9
1.1.2 Применение непараметрического моделирования .....	9
1.2 Стандартные модели нейронных сетей .....	10
1.2.1 Основы теории нейронных сетей .....	10
1.2.2 Многослойные персептроны .....	13
1.3 Импульсные и стохастические нейронные сети .....	16
1.3.1 Однонаправленные ИНС со стохастическими нейронами .....	16
1.3.2 Сети битовых потоков .....	20
1.3.3 Импульсные ИНС .....	22
1.3.4 Машина Больцмана .....	23
1.4 Выводы .....	26
<b>2 МОДЕЛИРОВАНИЕ СТАНДАРТНЫМИ ИНС .....</b>	<b>27</b>
2.1 Постановка задачи .....	27
2.1.1 Принципы идентификации систем с помощью моделирования .....	27
2.1.2 Нейросетевая аппроксимация имитационной модели .....	28
2.2 Выбор архитектуры ИНС и методика обучения .....	30
2.2.1 Генерация репрезентативной обучающей выборки .....	30
2.2.2 Выбор структуры ИНС .....	31
2.2.3 Обучение ИНС .....	32
2.3 Применение ИНС: моделирование процессов переноса энергии в системах мембранных протеинов .....	33
2.3.1 Экспериментальные объекты и методы .....	33
2.3.2 Результаты нейросетевого моделирования .....	37
2.4 Выводы .....	41
<b>3 МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ .....</b>	<b>42</b>
3.1 Подходы к моделированию стохастических процессов .....	42
3.2 Флуктуационные переходы в генераторе шума .....	43

3.3	Моделирование стохастических процессов классической ИНС.....	44
3.3.1	Схема моделирования.....	44
3.3.2	Моделирование тестового случайного сигнала .....	45
3.3.3	Моделирование сигнала генератора шума .....	48
3.4	Моделирование стохастических процессов машиной Больцмана.....	50
3.4.1	Моделирование с помощью машины Больцмана .....	50
3.4.2	Анализ результатов.....	50
3.5	Основные результаты и выводы.....	51
	ЗАКЛЮЧЕНИЕ .....	52
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	53
	ПРИЛОЖЕНИЕ 1. Нейросетевая аппроксимация детерминистического сигнала .....	56
	ПРИЛОЖЕНИЕ 2. Аппроксимация стохастического сигнала стандартной ИНС .....	58
	ПРИЛОЖЕНИЕ 3. Аппроксимация стохастического сигнала СНС .....	62

## *ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ*

ИНС	искусственная нейронная сеть
МСП	многослойный персептрон
СНС	стохастическая нейронная сеть
БП	битовые потоки
СБП	случайные битовые потоки
МБ	машина Больцмана
МБВП	машина Больцмана высшего порядка

## **ВВЕДЕНИЕ**

Моделирование является одним из наиболее распространенных способов изучения различных процессов и явлений и широко используется в научных исследованиях и инженерной практике. Различают имитационное и аналитическое моделирование. При имитационном моделировании модель воспроизводит изучаемый процесс с сохранением его физической природы, структуры и связей [1]. Под аналитическим моделированием понимают способ исследования различных процессов путем построения универсальных аналитических зависимостей и соотношений между состояниями элементов системы. Например, детерминистические объекты могут быть описаны дифференциальными уравнениями, стохастические – аппаратом марковских цепей [2].

Аналитическая модель дает наиболее полное и гибкое описание системы в рамках сделанных допущений, позволяя определить ее основные характеристики. В этом смысле аналитическая модель позволяет получить наиболее полную информацию о системе. Однако построение точной аналитической модели сложной системы часто оказывается невозможным с учетом сложности описания процессов ее функционирования. В этом случае используют имитационное моделирование, позволяющее моделировать и анализировать поведение сколь угодно сложной системы при использовании соответствующих вычислительных ресурсов. Часто говорят, что имитационная модель некоторой физической системы – это ее реализация *in silico*.

Имитационные и аналитические модели являются моделями типа "белого ящика" – т.е. в таких моделях явным образом учитываются состав и структура системы. На практике внутреннее устройство системы зачастую скрыто, но для наблюдения доступны входы и выходы. При этом для исследователя значение может иметь лишь интегральное поведение системы: какой отклик вызовет то или иное воздействие на нее. Такую систему при изучении можно заменить некоторым моделирующим элементом, воспроизводящим зависимости между входами и выходами реальной системы. Такую модель называют моделью "черного ящика" [3].

В настоящем отчете в качестве моделирующих элементов используются искусственные нейронные сети (ИНС). Являясь универсальным аппроксиматором, ИНС позволяют решать многие трудно формализуемые задачи [4,5]. Актуальность исследований в этом направлении подтверждается массой различных применений ИНС. Это автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти [6, 7, 8].

Предложенные в настоящей работе методы нейросетевого моделирования статистических и детерминистических процессов имеют важное значение для решения задачи идентификации

параметров сложных стохастических систем. Разработанные методы применялись для анализа процессов переноса энергии и биомолекулярных системах и для моделирования флуктуационных переходов в полупроводниковых генераторах шума.

В первом разделе отчета описываются типы ИНС, применяемые для моделирования процессов. Среди них классический вариант ИНС – многослойный персептрон (МСП), стохастические нейронные сети (СНС), в том числе сети Больцмана, импульсные нейронные сети [9, 10]. В этом разделе даются преимущества каждого типа, и очерчивается круг задач.

Второй раздел посвящен нейросетевому моделированию детерминистических процессов в стохастических системах. В разделе представлен метод идентификации параметров сложных систем на основе имитационного и нейросетевого моделирования и рассмотрен конкретный пример из области анализа флуоресцентных данных при изучении системы мембранных протеинов.

В третьем разделе приведены примеры моделирования стохастических процессов. Представлено два подхода к генерации случайных сигналов: путем использования детерминистических МСП со случайными входами и с помощью СНС.

# 1 НЕЙРОСЕТЕВОЙ ПОДХОД К МОДЕЛИРОВАНИЮ

## 1.1 Непараметрическое моделирование

### 1.1.1 Модель "черного ящика"

Особенность нейросетевого подхода к моделированию заключается в использовании универсальных программных или аппаратных средств, настраиваемых под конкретную задачу путем их обучения по специально подготовленному множеству. В рамках данного раздела предполагается, что ИНС моделируются на стандартном персональном компьютере, однако в равной степени они могут быть реализованы на специальных микросхемах, нейроплатах или нейрокомпьютерах [11].

Целесообразность перехода к нейросетевым алгоритмам моделирования может быть обусловлена следующими причинами:

- необходимо существенно ускорить процесс проведения вычислительного эксперимента;
- необходимо снизить затраты на этапе создания программной реализации моделей;
- не удастся привести исходную задачу к описанию последовательности операций над входными данными или другому формализованному виду, позволяющему построить алгоритм ее решения на ЭВМ.

Другими словами, искусственные нейронные сети позволяют решать многие трудно формализуемые задачи и отличаются достаточно низкими временными затратами на полный цикл их решения. В процессе построения модели ИНС приводит исходную задачу к специальному нейросетевому базису  $\{\sum \bar{a} \bar{x}\}$  с последующей классификацией входных данных по указанному базисному разложению. Следует заметить, что многие уравнения математической физики могут быть представлены в таком базисе [5]. Именно это может быть использовано в качестве отправного момента при разработке алгоритмов моделирования полупроводниковых приборов.

### 1.1.2 Применение непараметрического моделирования

Рассмотрим применения непараметрического моделирования при идентификации неизвестных параметров сложных экспериментальных систем. Такая задача является одной из основополагающих во многих областях науки. Важнейший этап при этом – адекватный анализ экспериментальных данных, носящих, как правило, статистический характер. Однако, из-за сложности проведения анализа всех доступных данных, на практике приходится прибегать к

упрощенной трактовке эмпирической информации и фокусировке на отдельных интегральных значениях [12]. Особенно остро данная проблематика характерна для изучения биомолекулярных образований, обладающих высокой степенью сложности, стохастической природой, а также зависимостью от различных физических и химических факторов среды. Одним из наиболее перспективных методов интерпретации экспериментальных данных является метод имитационного моделирования (другими словами – путем построения модели типа "белого ящика") [13]. Для построения имитационной модели сложной системы достаточно обладать информацией об элементарных процессах, происходящих в ней, и иметь представление о структуре системы, тогда как стандартное математическое моделирование предполагает наличие полного аналитического описания поведения системы и знание законов распределения всех стохастических параметров, используемых при моделировании, что редко осуществимо на практике.

В то же время, определение параметров стохастических систем с помощью имитационного моделирования затруднено, поскольку сопряжено со значительными вычислительными затратами. Как правило, идентификация системы осуществляется стандартными методами многопараметрической оптимизации [14], в которых имитационная модель выступает в качестве стохастической функции, аппроксимирующей результаты эксперимента. Очевидно, что в этом случае число пусков моделирования совпадает или даже превосходит число вычислений функции невязок. Во многих случаях высокие временные затраты не позволяют применять такой подход для идентификации систем. Этих проблем можно избежать, заменив имитационную модель моделью "черного ящика" значительно упростив при этом внутренние вычисления. Применение ИНС для замены имитационной модели позволяет значительно снизить временные затраты при поиске неизвестных параметров системы и заменить стохастический результат работы имитационной модели его гладкой аппроксимацией.

## *1.2 Стандартные модели нейронных сетей*

### *1.2.1 Основы теории нейронных сетей*

Искусственная нейронная сеть представляет собой совокупность процессорных элементов (нейронов). Искусственный нейрон состоит из набора межсоединений, сумматора и нелинейного оператора. Значение каждого входа  $X_i$  умножается на свой вес  $W_i$ , затем суммируется и из суммы вычитается порог  $\theta$ . После этого на результат действует некоторая активационная функция  $F$  и преобразованная таким образом информация подаётся на выход нейрона  $Y$ , как показано на рисунке 1 [4].

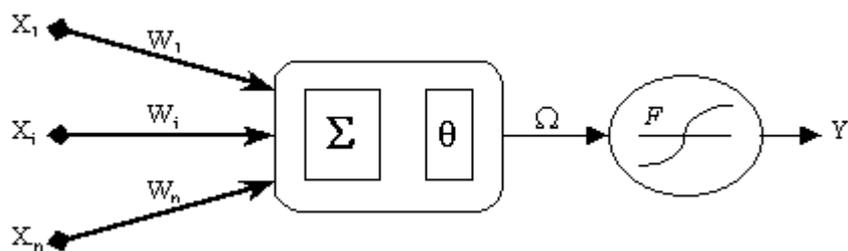


Рисунок 1 – Модель искусственного нейрона

Таким образом, выход каждого нейрона вычисляется по формуле:

$$Y = F \left\{ \sum_{i=1}^n (X_i W_i) - \theta \right\} \quad (1)$$

В дальнейшем условимся обозначать сумму произведений  $X_i W_i$  через переменную  $\Omega$ .

В качестве выходных (активационных) функций нейронов принято использовать ограниченные на множестве значений нелинейные функции, такие как: пороговая (случай сигнума) (рисунок 2а), комбинации пороговой и линейной функций, сигмоидальная функция (рисунок 2б), однотипная с ней функция гиперболического тангенса (рисунок 2в), а также радиальные базисные функции (рисунок 2г).

При пороговой функции (2) нейрон остаётся неактивным до тех пор, пока суммарный сигнал с его входов не достигает порогового значения. Когда этот порог достигнут, нейрон возбуждается и даёт на выходе 1.

$$Y = \begin{cases} 0, & \Omega \leq \theta \\ 1, & \Omega > \theta \end{cases} \quad (2)$$

Пороговая функция наиболее точно моделирует нелинейную передаточную характеристику биологического нейрона и благодаря своей простоте представляет нейронным сетям большие возможности. При классификации образов, ИНС с такой активационной функцией разбивает пространство на многогранники [9,10].

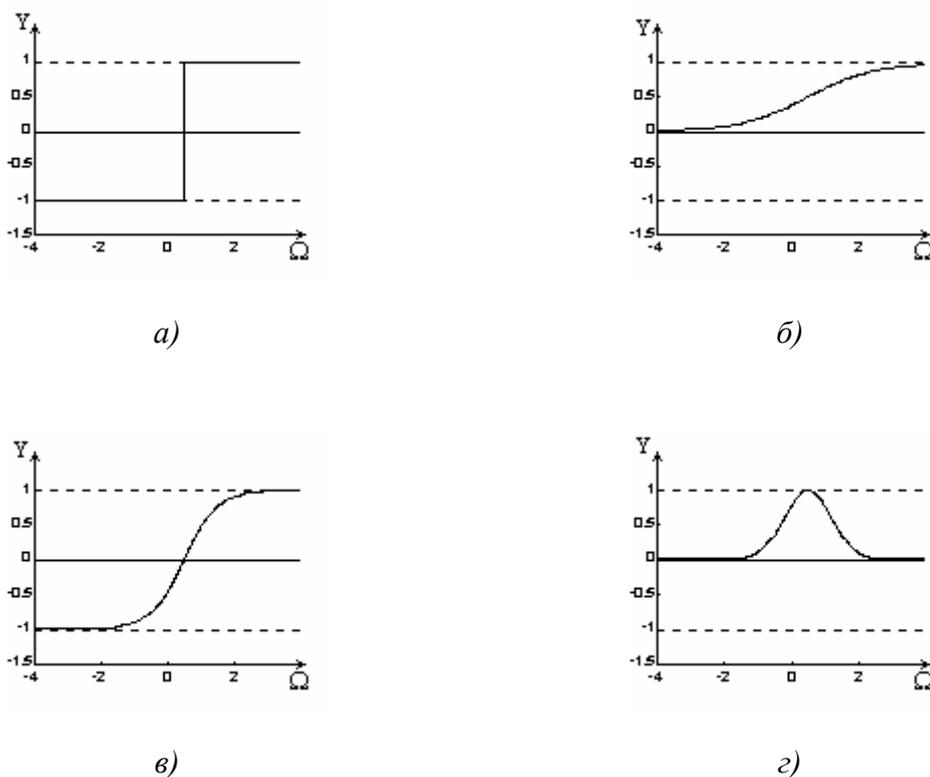


Рисунок 2 – Наиболее распространённые виды активационных функций (порог равен 0.5); а) пороговая; б) сигмоидальная; в) гиперболический тангенс; г) радиальная базисная.

При использовании сигмоидальной (иначе логистической) функции (3) диапазон изменений величины  $X$  сужается таким образом, что при любых значениях  $X$  значения  $Y$  принадлежат некоторому конечному интервалу. К тому же эта функция удобно дифференцируема, что немаловажно в алгоритмах обучения [9,12].

$$Y = \frac{1}{1 + e^{-(\Omega - \theta)}} \quad (3)$$

Функция гиперболического тангенса является расширением логистической функции на интервал  $(-1,1)$  и имеет вид

$$Y = 1 - \frac{2}{1 + e^{2(\Omega - \theta)}} \quad (4)$$

Наиболее интересным и перспективным на сегодняшний день (особенно в вопросах интерполяции и аппроксимации) является класс т.н. *радиальных базисных функций* (РБФ), к которым принято относить функции, имеющие глобальный экстремум и ведущие себя монотонно по мере удаления от него. Например, к радиальным базисным функциям относится функция вида:

$$Y = e^{-\frac{1}{2}(\Omega-\theta)^2} \quad (5)$$

В данной работе использован класс ИНС называемый многослойным персептроном (МСП). Рассмотрим структуру этих сетей и способы их применения.

### 1.2.2 Многослойные персептроны

Многослойным персептроном называют нейронную сеть, изображённую на рисунке 3, нейроны в которой расположены слоями, причём нейроны каждого слоя связаны только с нейронами предыдущего слоя.

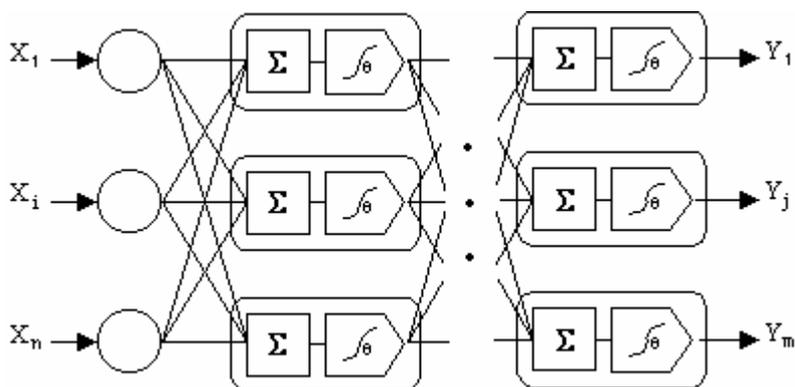


Рисунок 3 – Многослойный персептрон

Для сетей этого класса существует довольно хороший "классический" метод обучения – так называемый метод обратного распространения ошибки (*Back Propagation Error*), который относится к разряду методов градиентного спуска. Он сыграл очень важную роль в пробуждении интереса к нейронным сетям в начале 80-х, а его модификации и по сей день остаются лучшими для рассматриваемого класса сетей.

С использованием процедуры обратного распространения решаются задачи, в которых для некоторого входного вектора  $X(x_1, \dots, x_n)$  сеть на выходе формирует требуемый вектор  $Y(y_1, \dots, y_n)$ . На этапе обучения используется ряд пар векторов  $(X, Y^*)$ , называемых обучающей парой, где  $Y^*$  – целевой вектор.

Используя обучающие пары, сеть подстраивает свои веса таким образом, чтобы адекватно реагировать на входной вектор  $X$ . Рассмотрим работу алгоритма обратного распространения ошибки в случае, когда в качестве активационной функции выступает сигмоидальная функция.

При инициализации сети случайным образом задаются начальные веса и сдвиги сети. Алгоритм обучения сети обратного распространения включает в себя следующие шаги:  
 Выбрать очередную обучающую пару  $(X, Y^*)$  из обучающего множества и подать входной вектор  $X$  на вход сети.

Вычислить выход сети  $Y$ .

Вычислить разность между реальным (вычисленным) выходом сети и требуемым выходом (целевым вектором обучающей пары).

Подкорректировать веса сети так, чтобы минимизировать ошибку.

Повторить шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемой величины.

Шаги 1 и 2 используются как на этапе обучения сети, так и при функционировании уже обученной сети.

Вычисления в сети выполняются послойно. На шаге 3 каждый из выходов сети  $Y$  вычитается из соответствующей компоненты целевого вектора с целью получения ошибки. Эта ошибка используется на шаге 4 для коррекции весов сети, причем величина изменений определяются алгоритмом обучения.

Шаги 1 и 2 можно рассматривать как "проход вперед", а 3 и 4 — как проход назад, так как сигнал ошибки распространяется обратно по сети и используется для подстройки весов. Эти два прохода можно выразить математически.

На входе имеем вектор  $X$ , на основе которого вычисляется выходной вектор  $Y$ . Вектор  $Y$  вычитается из целевого вектора  $Y^*$  с целью получения ошибки  $\varepsilon$ :

$$\varepsilon = Y^* - Y \quad (6)$$

Величина  $\Omega$  каждого нейрона первого слоя вычисляется как взвешенная сумма входов нейрона. Затем активационная функция  $F$  сжимает  $\Omega$  и дает величину  $OUT$  для каждого нейрона в этом слое.

Полученное выходное множество  $OUT$  является входом для следующего слоя. Процесс повторяется слой за слоем, пока не будет получено заключительное множество сети.

На этапе обратного прохода происходит подстройка весов выходного слоя. Так как для каждого нейрона выходного слоя задано целевое значение, то подстройка весов легко осуществляется с помощью дельта-правила. Внутренние слои не имеют целевых значений и называются скрытыми слоями.

Процесс подстройки одного веса от нейрона  $p$  в скрытом слое  $j$  к нейрону  $q$  в выходном слое строится следующим образом. Выход  $OUT$  слоя  $k$ , вычитаясь из целевого значения  $Y^*$ , дает ошибку, которая умножается на производную сжимающей функции (в нашем случае  $OUT(1-OUT)$ ), вычисленную для этого нейрона слоя  $k$ , давая, таким образом, величину

$$\delta = OUT(1-OUT)(Y^* - OUT) \quad (7)$$

Затем  $\delta$  умножается на величину  $OUT$  нейрона  $j$ , из которого выходит рассматриваемый вес. Это произведение в свою очередь умножается на коэффициент обучения  $\eta$  ( $0,01 \leq \eta \leq 1$ ) и результат прибавляется к весу.

$$\Delta w_{pq,k} = \eta \delta_{q,k} \cdot OUT_{p,q} \quad (8)$$

где  $\delta_{q,k}$  - величина  $\delta$  для нейрона  $q$  в выходном слое  $k$ ;  $OUT_{p,q}$  - величина выхода для нейрона в скрытом слое  $j$ .

$$w_{pq,k}^{(n+1)} = w_{pq,k}^n + \Delta w_{pq,k} \quad (9)$$

где  $w_{pq,k}^n$  - величина веса от нейрона в скрытом слое  $k$  к нейрону  $q$  в выходном слое на шаге  $n$ ,  $w_{pq,k}^{(n+1)}$  - величина веса на шаге  $n+1$  после коррекции. Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.

Рассмотрим один нейрон в скрытом слое, предшествующем выходному слою. При проходе вперед этот нейрон передает свой выходной сигнал нейронам в выходном слое через соединяющие их веса.

Во время обучения эти веса функционируют в обратном порядке, пропуская величину  $\delta$  от выходного слоя назад к скрытому слою. Каждый из этих весов умножается на величину  $\delta$  нейрона, к которому он присоединен в выходном слое. Величина  $\delta$ , необходимая для нейрона скрытого слоя, получается суммированием всех таких произведений и умножением на производную сжимающей функции:

$$\delta_{p,j} = OUT_{p,j} (1 - OUT_{p,j}) \sum_q \delta_{q,k} w_{pq,k} \quad (10)$$

Когда значение  $\delta$  получено, веса между входным слоем и скрытым слоем  $j$  могут быть скорректированы с помощью формул (9) и (10), в которых индексы необходимо модифицировать в соответствии со слоем. То есть процесс обучения представляет собой вычисление  $\delta$  для каждого нейрона в данном слое и коррекцию всех весов данного слоя.

Для ускорения обучения сети алгоритм обучения модифицируют путем введения нейронного смещения и стабилизирующего множителя, условно называемого "импульсом".

Введение нейронного смещения позволяет сдвигать начало отсчета передаточной функции и по сути является процедурой аналогичной подстройке порога персептронного нейрона. Смещение вводится посредством добавления к каждому слою нейронов дополнительного нейрона, на который подается сигнал, равный +1, а не выходу нейрона предыдущего слоя. В процессе обучения вес данного нейрона корректируется так же, как и остальные веса нейронов.

Введение импульса позволяет ускорить обучение сети при использовании алгоритма обратного распространения. Этот метод заключается в добавлении к коррекции веса члена, пропорционального величине предыдущего изменения веса. Как только происходит коррекция, она запоминается и служит для модификации всех последующих коррекций. Уравнение коррекции принимает следующий вид:

$$w_{pq,k}^{(n+1)} = \alpha \Delta w_{pq,k}^n + \eta (\delta_{q,k} OUT_{pj}) \quad (11)$$

Затем вычисляется изменение веса:

$$w_{pq,k}^{n+1} = w_{pq,k}^n + \Delta w_{pq,k}^{(n+1)} \quad (12)$$

где  $\alpha$  - коэффициент импульса, обычно  $\approx 0.9$  [4].

Описанный метод обучения МСП применяется в данной работе для построения модели "чёрного ящика". На входы МСП подавался сигнал с входа системы (или имитационной модели системы), а с выхода снимался сигнал, соответствующий выходному сигналу системы.

### 1.3 Импульсные и стохастические нейронные сети

#### 1.3.1 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 Рапопорт сделали 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 Рапопорт 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 [15].

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 персептронов [16].

*Алгоритмы стохастического обучения.* 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 на случайную величину, вычисляют ошибку и сохраняют изменения, ведущие к уменьшению последней. Процедура стохастического обучения состоит в следующем.

1) Параметрам ИНС присваивают произвольные малые значения.

Выбирают очередной вектор обучающего множества и вычисляют фактический выход нейронной сети.

Вычисляют ошибку на текущем шаге:

$$\varepsilon = \sum_i (y_i^* - y_i)^2 \quad (13)$$

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

$$E = -\sum_{i < j} W_{ij} y_i y_j + \sum_j \Theta_j y_j \quad (14)$$

где  $y_i$  определяет состояние или выход бинарного нейрона. Если на нейроны не действует никакая внешняя сила, то вся система стремится 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:

$$P_k = 1/[1+\exp(-\Delta E_k/KT)] \quad (15)$$

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

$$\Delta E_k = E_{k \text{ выкл}} - E_{k \text{ вкл}} = \sum W_{ik} y_i - \Theta_k \quad (16)$$

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 примером.

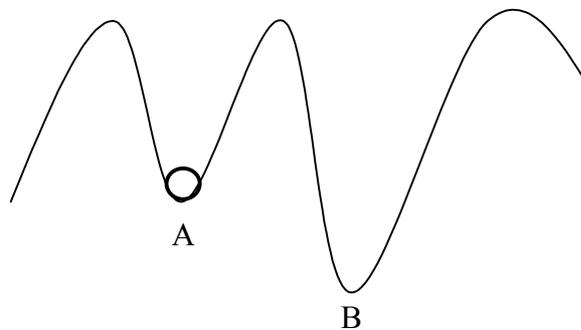


Рисунок 4 – Проблема локального минимума

В качестве аналогии рассмотрим металлический шарик в коробке с искривленным рельефом дна (рисунок 4). 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 (15) 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  $t$ :

$$T(t) = T_0 / \log(1+t) \tag{17}$$

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

$$P(x) = T(t) / [T^2(t) + x^2] \tag{18}$$

Оба распределения представлены на рисунке 5.

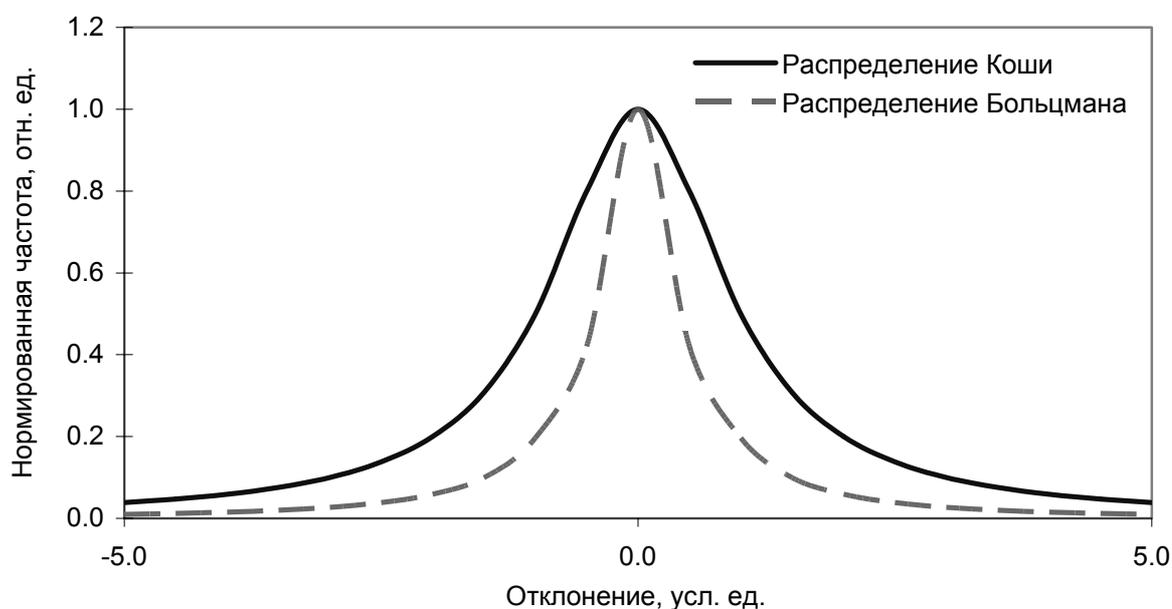


Рисунок 5 – Распределения приращений весов

В случае использования распределения Коши, 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 (18) относительно шага  $x$ :

$$x_c = \eta T(t) \operatorname{tg}[P(x)] \tag{19}$$

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.

### 1.3.2 Сети битовых потоков

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 зашумленной среде и улучшит ее способность к обобщению. Это предположение приводит к естественному представлению данных с помощью битовых потоков (bit-stream), в которых вещественные значения представляются изменением частоты единиц и нулей [17,18].

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 ИНС 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 вещественные 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 значение БП равно единице. Такой БП называется последовательностью Бернулли.

В монополярном представлении вещественное значение  $r \in [0, 1]$  представляется последовательностью Бернулли с вероятностью  $p = r$ . Чтобы представить вещественное 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 приравнивается 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 [17] показано, что СБП-сети по своим возможностям подобны сетям из обычных сигмоидных нейронов, при этом возможно обучение таких сетей для вычислений большого масштаба. Кроме того, для СБП-сетей возможна аппаратная реализация алгоритма обратного распространения ошибки.

### 1.3.3 Импульсные ИНС

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

В таблице приведены особенности обычной и импульсной ИНС.

Таблица 1 – Сравнение характеристик обычной и импульсной ИНС

Обычная ИНС	Импульсная ИНС
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

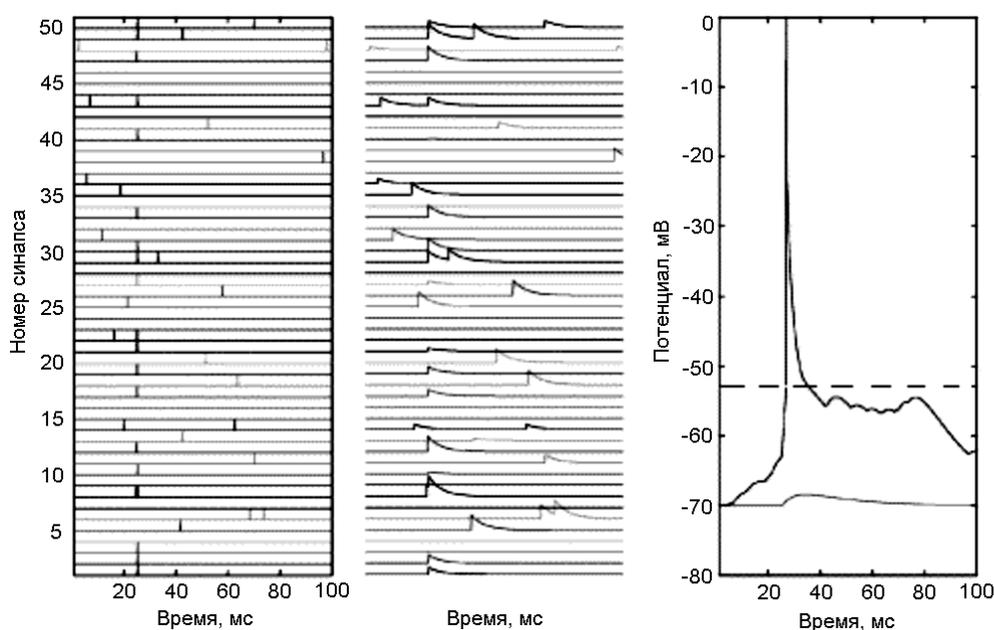


Рисунок 6 – Работа импульсной ИНС [19].

Существует несколько способов кодирования сигналов импульсными последовательностями. Один из вариантов приведен на рисунке 6. 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.

### 1.3.4 Машина Больцмана

В данном разделе рассматривается стохастическая ИНС, названная машиной Больцмана (МБ), ее разновидности и алгоритмы обучения.

*Архитектура сети.* В общем случае в машине Больцмана случайность используют как при обучении, так и при моделировании. На рисунке 7 представлена рассматриваемая модель, состоящая из случайно соединенных нейронов особого вида, 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.

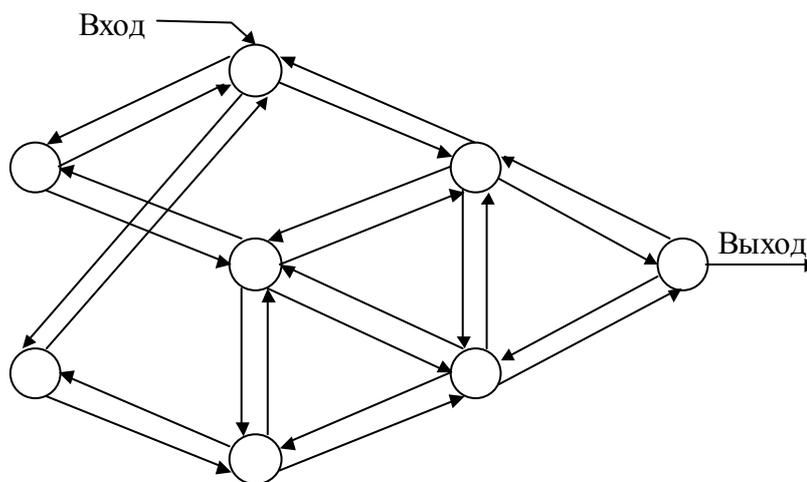


Рисунок 7 – Машина Больцмана

Нейроны, обозначенные на рисунке 7 кружками, могут находиться в одном из двух состояний: +1 (включен) и -1 (выключен). Вероятность перехода в состояния -1 или +1 определяется взвешенной суммой сигналов  $s$  на входах нейронов:

$$P\{x=1\} = F_c(2s), \quad P\{x=-1\} = 1 - F_c(2s), \quad \text{где } F_c(s) = [1 + \exp(-s/c)]^{-1} \quad (20)$$

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  $i$ .

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. Поэтому обычно рассматривают асимптотическую сходимость последовательности состояний к некоторому предельному состоянию.

*Процедура обучения машины Больцмана*, описанная в [20], 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 ИНС не получили широкого распространения.

*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 [21].

Можно доказать сходимость 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 [21] также показано, что МБВП обладает свойствами сходимости, которые не содержатся в МБ со скрытыми нейронами, в частности, в обычной МБ второго порядка.

*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 [4]. 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.

Одной из областей 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 применения 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 различать 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 перцептронных задач, таких, как задача «исключающего ИЛИ», обнаружение симметрии во входном образе и т. д., а также для распознавания речи.

#### **1.4 Выводы**

Для моделирования процессов в стохастических системах могут применяться как стандартные модели ИНС, так и стохастические сети (машина Больцмана). При этом возможности СНС изучены недостаточно.

## 2 МОДЕЛИРОВАНИЕ СТАНДАРТНЫМИ ИНС

### 2.1 Постановка задачи

#### 2.1.1 Принципы идентификации систем с помощью моделирования

Метод идентификации параметров систем с помощью имитационного моделирования был разработан для определения физических и химических параметров сложных систем, которые не могут быть полностью описаны аналитическими выражениями. Рассмотрим идею идентификации параметров на следующем общем примере. Сложная экспериментальная система  $\Theta$  может быть описана вектором параметров  $P = (p_1, p_2, p_3, \dots)$ . Эти параметры могут рассматриваться в качестве входов системы  $\Theta$ . Результатом проведения некоторого числа экспериментов над  $\Theta$  с различными входными параметрами является вектор выходных величин  $F$ . В этом случае система может рассматриваться как оператор, выполняющий следующее преобразование:

$$\Theta(p_1, p_2, p_3, \dots) = \Theta(P) = F \quad (21)$$

Обычно некоторые входные параметры известны. Обозначим их  $P_0$ , к примеру, пусть  $P_0 = (p_1, p_2)$ . Другие параметры должны быть определены в ходе идентификации. Обозначим их  $P_X$ ; пусть  $P_X = (p_3, p_4, \dots)$ . Таким образом вектор входов системы содержит комбинацию известных и неизвестных параметров  $P = (p_1, p_2, p_3, \dots) = (P_0, P_X)$ . Определение  $P_X$  является целью нашего анализа.

Предположим, что для системы  $\Theta$  можно построить адекватную имитационную модель, которая выполняет преобразование (22) над теми же физическими параметрами  $P$ .

$$\Xi(P) \equiv \Xi(P_0, P_X) = F^* \quad (22)$$

где  $F^*$  – смоделированные выходные значения, аппроксимирующие экспериментальные данные  $F$ .

Определение неизвестных параметров  $P_X$  как правило проводят путем минимизации отклонений между  $F$  и  $F^*$ . Блок-схема алгоритма анализа представлена на рисунке 8.

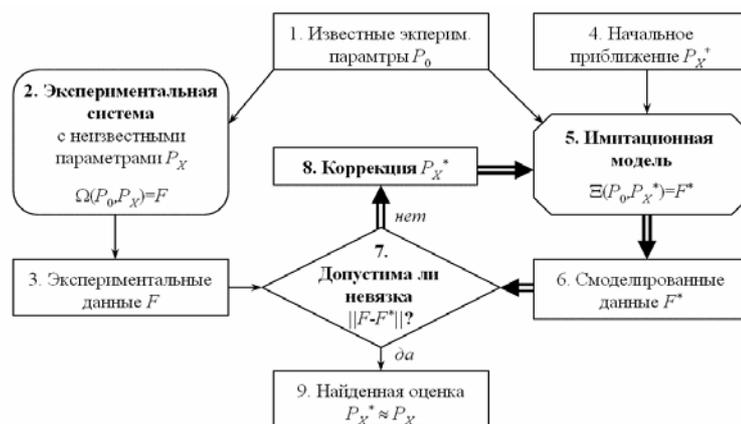


Рисунок 8 – Схема идентификации параметров системы с помощью имитационного моделирования.

При этом можно выделить следующие шаги:

1. Экспериментально получают выходные значения  $F$  (блоки 1-3).
2. Создается адекватная модель  $\Xi$  системы  $\Theta$ , выполняющая операцию (2) (блок 5).
3. Делаются начальные приближения  $P_X^*$  искомых параметров  $P_X$  (блок 4).
4. Алгоритм оптимизации изменяет параметры  $P_X^*$ , минимизируя функцию невязки  $\|F^* - F\|$  (блоки 6-8 и 5).
5. В результате получают параметры  $P_X^*$ , являющиеся оценками экспериментальных параметров  $P_X$  (блок 9).

Как было отмечено ранее, основной проблемой такого подхода являются высокие временные затраты при подгонке параметров. Эта проблема может быть решена путем замены имитационной модели моделью "черного ящика" – нейронной сетью.

### 2.1.2 Нейросетевая аппроксимация имитационной модели

Субенко [22] и Hornik [23] независимо показали что непрерывные гладкие функции могут быть асимптотически аппроксимированы линейной комбинацией и суперпозицией сигмоидальных функций, т.е. многослойным персептроном.

Большинство зависимостей в химии и физике могут быть представлены гладкими кривыми (если конкретная экспериментальная реализация такой зависимости будет носить стохастический характер – будем рассматривать средние значения). Это дает возможность аппроксимировать такие зависимости, а значит и имитационную модель  $\Xi$ , многослойным персептроном. Обозначим аппроксимирующий нейросетевой оператор через  $\Psi$ . Он выполняет следующее преобразование:

$$\Psi(P_0, P_X) = F^{**} \quad (23)$$

где  $F^{**}$  – нейросетевая аппроксимация выходных значений системы (точнее, ее имитационной модели).

Следовательно, при подгонке параметров можно использовать не имитационную модель  $\Xi$ , а ее нейросетевую аппроксимацию. Предложенный подход продемонстрирован на рисунке 9. ИНС является в этом случае моделью типа "черного ящика" экспериментальной системы  $\Theta$ .

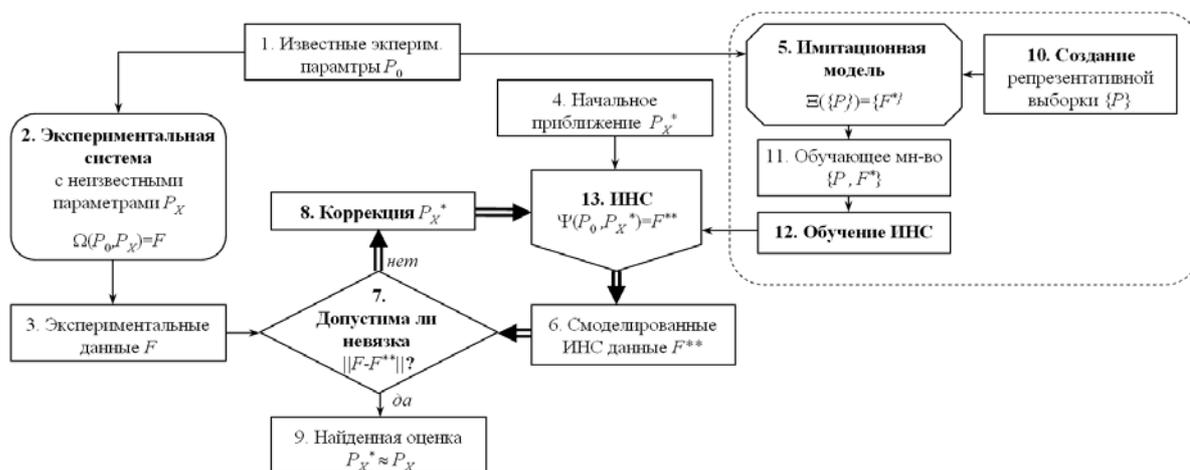


Рисунок 9 – Схема идентификации параметров системы с помощью нейросетевой модели типа "черного ящика". Блок выделенный пунктиром выполняется 1 раз для каждой имитационной модели.

Следующие шаги могут быть выделены в этом алгоритме:

1. Экспериментально получают выходные значения системы  $F$  (блоки 1-3).
2. Строится адекватная модель  $\Xi$  системы  $\Theta$  (блок 5).
3. Генерируется репрезентативное множество точек  $\{P\}$  пространства входных параметров (блок 10) и для каждой точки вычисляются выходные значения имитационной модели  $\{F^*\}$ . Эти множества формируют обучающее множество ИНС  $\{P, F^*\}$ . (блок 11).
4. ИНС обучается на множестве примеров  $\{P, F^*\}$  (блок 12).
5. Делаются начальные приближения  $P_X^*$  искомых параметров  $P_X$  (блок 4).
6. Алгоритм оптимизации изменяет параметры  $P_X^*$ , минимизируя функцию невязки  $\|F^{**} - F\|$  (блоки 6-8 и 13).
7. В результате получают параметры  $P_X^*$ , являющиеся оценками экспериментальных параметров  $P_X$  (блок 9).

Перед началом работы необходимо, используя имитационную модель, создать обучающую выборку и обучить ИНС. Причём, и генерацию обучающей выборки, и обучение приходится повторять при любом изменении модели, в том числе, при её коррекции либо усложнении.

## 2.2 Выбор архитектуры ИНС и методика обучения

### 2.2.1 Генерация репрезентативной обучающей выборки

Будучи замененной аппроксимирующей нейронной сетью, имитационная модель используется только для начального обучения ИНС. Таким образом этап создания обучающей выборки становится наиболее трудоемким из-за больших вычислительных затрат на получение выходных значений имитационной модели. Для того чтобы получить адекватную аппроксимацию чрезвычайно важно обучать сеть на репрезентативной выборке. Таким образом, для того чтобы увеличить эффективность обучения, необходимо вводить такой алгоритм выбора точек, который дает максимально разбросанные точки в пространстве параметров системы при минимальном их количестве. Причем желательно, чтобы число обучающих точек могло быть легко увеличено на произвольную величину.

В этой работе следующая схема была разработана и применена для выбора обучающих точек. Далее предполагаем, что все параметры нормированы на отрезок  $[0,1]$ .

*Шаг 1.* Генерация "граничных" узловых точек. Для каждого параметра выбирались: минимальное, максимальное и среднее (в статистическом смысле) значение, и брались все возможные их комбинации. Размер этой выборки для 3, 4 и 5 варьируемых параметров составлял 27, 81 и 243 соответственно.

*Шаг 2.* Генерация основного обучающего множества. На этом шаге значения параметров выбираются по разработанному алгоритму:

1. Пусть  $n$  – размерность пространства,  $N$  – число уже выбранных узлов. Задаётся некоторая константа  $a \in [0,1]$ , определяющая равномерность заполнения пространства узловыми точками. Значение  $a=0$  соответствует неупорядоченному заполнению.

2. Случайным образом из диапазона допустимых значений выбирается набор параметров (точка многомерного пространства параметров).

3. Производится обход всех ранее сгенерированных узлов с вычислением расстояния  $d$  до нового узла.

4. Если выполняется условие

$$\min(d) > \frac{a}{\sqrt[n]{N}} \quad (24)$$

где  $d$  – евклидово расстояние между рассматриваемым узлом и одним из ранее сгенерированных, то узел принимается.

5. Проверяется критерий останова. Если он не выполнен – алгоритм возвращается на шаг 2.

Ниже приведён результат работы такого алгоритма для  $n=2$ . Видно, что использование алгоритма позволило заполнить пространство значительно равномернее. Кроме того, заполнение осталось случайным и легко может быть продолжено.

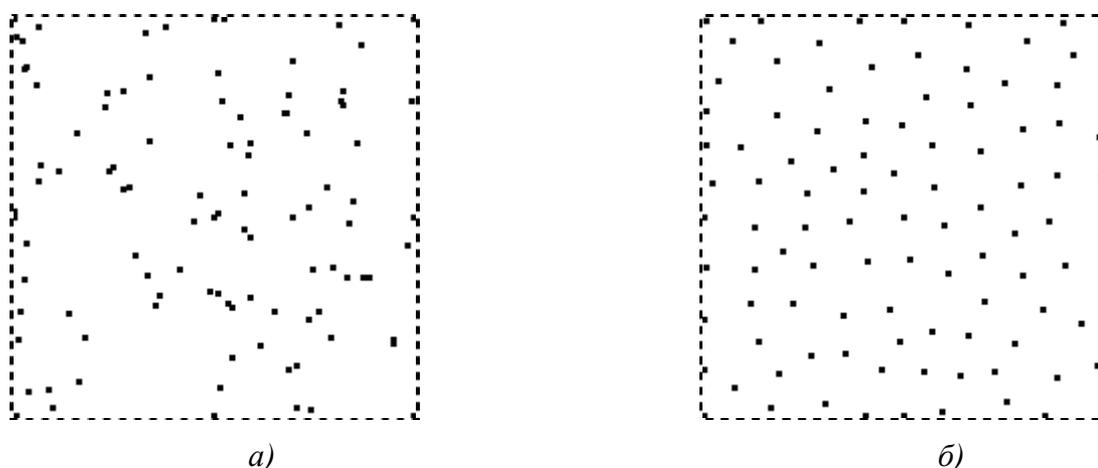


Рисунок 10 – Результат выбора 100 узловых точек для функции 2-х переменных: при  $a=0$  (полностью случайный) (а) и при  $a=0.8$  (б).

Очевидно, что применение предложенной схемы позволяет получить равномерное заполнение пространства параметров. Характер заполнения остается стохастическим, что позволяет без труда добавить новые точки в обучающее множество. Более того, при таком подходе в обучающем множестве отсутствуют точки с одинаковыми координатами, что позитивно сказывается на обобщающих свойствах обученной сети.

### 2.2.2 Выбор структуры ИНС

При обучении нейронных сетей, в частности многослойных персептронов, существует две не решённые (в общем случае) проблемы:

1. Проблема выбора оптимальной конфигурации сети, т.е. числа скрытых слоёв нейронов и количества нейронов в каждом слое.
2. Проблема выбора начальных весовых коэффициентов. Как правило, начальные веса задаются случайным образом. В то же время эффективность алгоритмов обучения в значительной степени зависит от того, насколько удачным оказался выбор начальных весовых коэффициентов.

Для аппроксимации имитационных моделей было решено использовать трехслойные персептроны. Они показали наибольшую гибкость в сочетании с хорошими обобщающими свойствами. Поиск оптимального числа нейронов проводился методом локальных вариаций (использование более сложные методов не имело смысла из-за ограниченности решений). Поиск начинался из области малых размеров слоёв (например, по 4 нейрона в каждом) и постепенно сдвигался к оптимальному значению. Для вычисления невязки в узле осуществлялось обучение

сети на небольшой, максимально репрезентативной выборке. Число итераций обучения равнялось числу настраиваемых весовых коэффициентов.

Начальные значения весовых коэффициентов выбирались подбором. При этом производилось несколько пусков алгоритма обучения с различными исходными весами (в опытах – порядка 100 пусков) и выбиралась конфигурация с лучшим результатом работы на некотором контрольном множестве данных.

Для каждой сети производилось 200 итераций. Операция поиска оптимальных весов выполнялась параллельно на нескольких машинах. В результате ошибка обучения снижалась в сравнении со средним значением в ~2.1 раза (и до 4.4 раз по отношению к наиболее неудачному выбору).

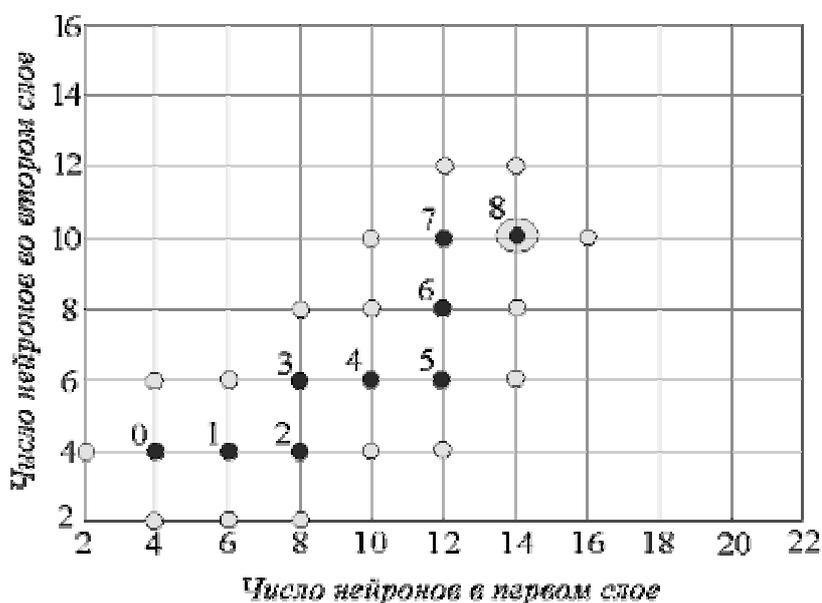


Рисунок 11 – Оптимизация числа нейронов трехслойного персептрона

Результаты работы нейросетевой аппроксимации приведены в следующем разделе.

### 2.2.3 Обучение ИНС

При обучении МСП должны удовлетворяться два достаточно противоречивых условия. С одной стороны отклонение между ожидаемыми и полученными выходами сети должно минимизироваться. С другой стороны – сеть не должна терять свою обобщающую способность, обучаясь экспериментальному "шуму" [24]. Избыточно длительное обучение способствует попаданию методы обучения в локальный минимум, ухудшая аппроксимацию в точках, не принадлежащих к обучающей выборке. Этот эффект принято называть переобучением [24, 25]. Для избежания этого применяют специальную стратегию, основанную выделении из обучающего множества и небольшого контрольного множества, на котором сеть не обучается, но тестируется

после каждой итерации обучения. Если ошибка на контрольном множестве не уменьшается в течение некоторого времени – обучение прекращают. Были проведены численные эксперименты для установления оптимального размера контрольного множества. Приняв все обучающее множество за 100 %, были получены следующие результаты. Для 3-параметрической модели процентное отношение контрольного множества составило 20 %; для 4-параметрической – 15 % и для 5- и 6-параметрических моделей – 10 %. Применение описанного в пункте 2.2.1 метода генерации обучающего множества дает возможность получать наиболее удаленные точки, выбирая для контрольного множества последние сгенерированные элементы.

## 2.3 Применение ИНС: моделирование процессов переноса энергии в системах мембранных протеинов

### 2.3.1 Экспериментальные объекты и методы

Представленный выше подход к моделированию был протестирован на задаче моделирования резонансного переноса энергии между флуоресцентными метками вирусного протеина M13, внедренного в липидную мембрану. Рассмотрим модель экспериментальной системы.

*Резонансный перенос энергии.* Идея спектроскопии резонансного переноса энергии базируется на диполь-дипольном безизлучательном переносе энергии между флуоресцентными молекулярными зондами. Теория такого переноса была разработана Ферстером [26] и улучшена далее Страйером [27]. Суть метода заключается в следующем: изучаемые макромолекулы метятся флуоресцентными зондами двух типов – донорами и акцепторами [28], при этом спектр абсорбции акцептора должен перекрываться со спектром эмиссии донора. Доноры возбуждаются внешним излучением, после чего часть энергии переходит на акцепторы путем резонансного переноса. Вероятность переноса энергии от донора к акцептору в уединенной донорно-акцепторной паре может быть вычислена по формуле (25):

$$P_{ET} = \frac{1}{1 + (r/R_0)^6} \quad (25)$$

где  $r$  – расстояние между донором и акцептором,  $R_0$  – так называемое расстояние Ферстера, соответствующее 50 % переносу энергии [26]. Если система содержит  $N_a$  акцепторов, выражение для переноса энергии становится более сложным:

$$p_{ET} = \frac{\sum_{i=1}^{N_a} (R_0/r_i)^6}{1 + \sum_{i=1}^{N_a} (R_0/r_i)^6} \quad (26)$$

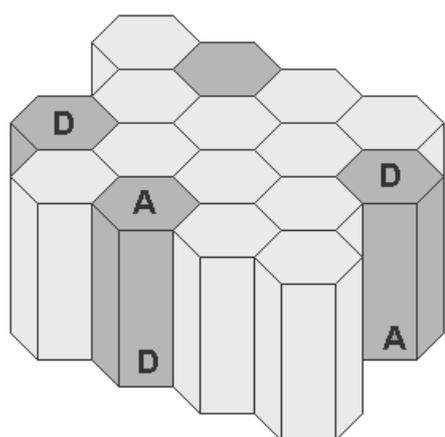
Средняя вероятность переноса энергии в системе содержащей  $N_d$  доноров и  $N_a$  акцепторов, которую называют также эффективностью переноса энергии, может быть вычислена тривиальным образом:

$$E = \frac{1}{N_d} \sum_{j=1}^{N_d} (p_{ET})_j \quad (27)$$

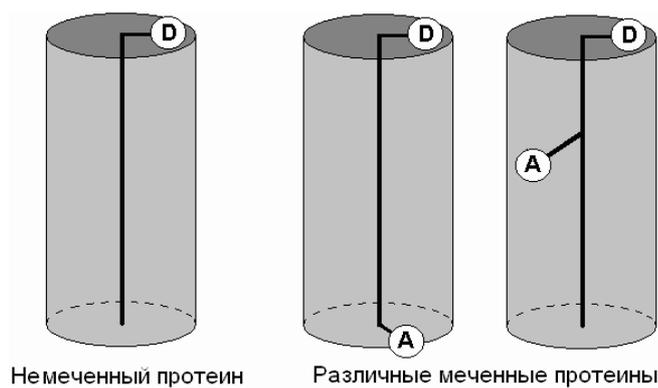
Наблюдая процесс переноса энергии можно сделать вывод о взаимном расположении доноров и акцепторов в системе.

*Пространственная модель протеин-липидной системы.* Внедренный в мембрану белок бактериофага M13, инфицирующего *Escherichia Coli* является весьма удобной системой для изучения фундаментальных аспектов протеин-липидных и протеин-протеиновых взаимодействий. Длина этого протеина составляет 50 аминокислот, образующих преимущественно  $\alpha$ -спираль. Этот протеин изучался в течение последних 10 лет различными биофизическими методами [29].

При изучении методом резонансного переноса энергии аминокислота Trp в позиции 26 использовалась в качестве донора. А качестве акцептора в белок искусственно внедрялась флуоресцентная метка AEDANS [30]. Поскольку вероятность внедрения была ниже 100 % в системе существовало 2 типа протеинов: не меченные, содержащие лишь один донор, и меченные, содержащие как донор так и акцептор. Для изучения такой сложной системы была введена следующая упрощенная пространственная модель. Биологическая мембрана была аппроксимирована двумерной периодической структурой с гексагональной упаковкой липидов в которой распределены протеины M13. Такая структура представлена на рисунке 12а. Площадь занятая каждым протеином была принята равной площади липида. Было предположено что расстояние между двумя ближайшими молекулами в решетке равно 8 Å, а толщина двойного липидного слоя – 30 Å. Протеины аппроксимировались структурами представленными на рисунке 12б с постоянным положением донора (D) и переменным акцептора (A).



a)



б)

Рисунок 12 – Модель мембраны (a) и мембранного протеина (б) с флуоресцентными метками.

*Модель переноса энергии.* Входные параметры модели представлены в таблице 2. Их диапазоны выбраны таким образом, чтобы покрыть все физически оправданные значения возможные на практике.

Таблица 2 – Входные параметры модели

Параметр	Описание	Диапазон
1. Поверхностная плотность меченых протеинов	Отношение площади занятой меченными протеинами (содержащими и донор и акцептор) к полной площади мембраны.	0.0001 – 0.1
2. Поверхностная плотность не меченых протеинов	Отношение площади занятой не меченными протеинами (содержащими и донор и акцептор) к полной площади мембраны	0.0001 – 0.1
3. Позиция акцептора	Номер аминокислотной позиции к которой прикрепляется акцептор.	1 – 50
4. Коэффициент ассоциации протеинов	Вероятность того, что выбранный протеин находится рядом с другим.	0 – 1
5. Размер молекул	Минимальное расстояние между центрами 2-х ближайших молекул (липидов или протеинов)	5 – 10 Å
6. Расстояние Ферстера	Расстояние между донором и акцептором, при котором происходит 50 % перенос энергии.	1 – 100 Å

В экспериментальной ситуации параметры 1-3 и 6 известны (хотя и с некоторой погрешностью), следовательно, они могут рассматриваться как  $P_0$  в обозначениях подраздела 2.1.

Тем не менее, в общем случае разделение параметров представленных в таблице 2 на  $P_0$  и  $P_X$  зависит от конкретной ситуации. Например, для 3-х параметрической модели параметры 1 и 2 могут использоваться как  $P_X$ , а параметр 3, известный более точно, как  $P_0$ . Следует отметить, что 3-х параметрическая модель использовалась только для проверки методологии. Для 4-х параметрической модели коэффициент ассоциации протеинов становится объектом интереса ( $P_X$ ), в то время как параметры 1-3 – известны ( $P_0$ ).

В качестве выходного значения системы бралась эффективность переноса энергии ( $F$  в обозначениях подраздела 2.1). Вследствие того, что моделирование носит полу-имитационный характер (имитационная модель пространственной структуры и математическая модель процесса) выходное значение системы содержало случайную шумовую составляющую. Поэтому моделирование запускалось несколько раз с усреднением результата. Блок-схема моделирования представлена на рисунке 13.



Рисунок 13 – Схема отдельного моделирования переноса энергии в системе мембранных протеинов.

Моделирование осуществлялось следующим образом:

1. Устанавливались параметры системы (блок 1).
2. Пространственная модель мембраны с внедренными протеинами создавалась в соответствии с входными параметрами. Координаты и ориентация протеинов предоставляли информацию о положении доноров и акцепторов в системе (блок 2).
3. Для каждого донора (обозначенного как  $i_D$ ) вычислялись расстояние до всех акцепторов и вероятность переноса энергии с этого донора на любой из них (формула (26)) (блок 3-5).
4. Средняя вероятность переноса энергии давала эффективность переноса энергии для всей системы.

Шаги 2-4 (и блоки 2-6 в блок-схеме) повторялись несколько раз для уменьшения случайной составляющей эффективности. В наших вычислениях использовалось  $10^4/n_D$  пусков моделирования, где  $n_D$  – число доноров в системе.

### 2.3.2 Результаты нейросетевого моделирования

*Конфигурация ИНС.* Параметры системы линейным преобразованием переводились в диапазон [0,1] перед подачей на входы сети. Оптимальное число нейронов МСП выяснялось с помощью численных экспериментов, результаты которых приведены в таблице 3.

Таблица 3 – Оптимальное число нейронов в ИНС

<b>Число входных параметров</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Номера используемых параметров	1-3	1-4	1-5	1-6
Число нейронов в первом слое сети	13	15	18	20
Число нейронов во втором слое сети	10	13	16	20

После определения оптимального числа нейронов сеть обучалась, как это было описано в разделе 2.2.3. Результирующая ошибка после обучения составляла порядка 2 %.

*Временные затраты.* Все вычисления проводились с помощью MATLAB® 6.1 (Neural Networks Toolbox) на ПК на базе процессора Intel Pentium III-850. Временные затраты представлены в таблице 4.

Таблица 4 – Временные затраты при нейросетевой аппроксимации модели переноса энергии

<b>Число параметров</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Время создания обучающей выборки	11 ч	22 ч	56 ч	110 ч
Время обучения	6 мин	10 мин	14 мин	20 мин
Время единичного нейросетевого моделирования	$6.0 \times 10^{-4}$ с	$7.0 \times 10^{-4}$ с	$8.0 \times 10^{-4}$ с	$10^{-3}$ с
Среднее время имитационного моделирования	40 с	40 с	40 с	40 с
Среднее ускорение при моделировании	$6.7 \times 10^4$	$5.7 \times 10^4$	$5.0 \times 10^4$	$4.0 \times 10^4$

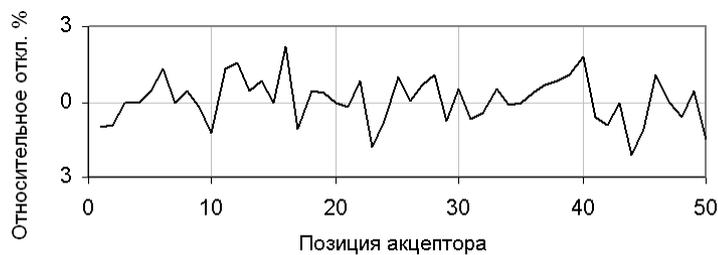
Из таблицы 4 видно, что процесс генерации обучающих пар является наиболее трудоемким. Следует, однако, отметить, что этот процесс выполняется автоматически (без привлечения исследователя) и лишь единожды для каждой модели. Ускорение, получаемое при замене имитационной модели на нейросетевую, составляет порядка  $5 \times 10^4$ .

Типичный пример нейросетевого моделирования представлен на рисунке 14. Здесь показана зависимость эффективности переноса энергии от позиции акцептора. Причиной колебаний эффективности является спиральная структура протеина. Относительное отклонение ИНС аппроксимации от результатов имитационного моделирования не превышает 3 %. Это указывает на достаточную точность аппроксимации, поскольку статистический разброс результатов моделирования составляет порядка 1-2 %.

*Состоятельность аппроксимации.* Для доказательства состоятельности аппроксимации был проведен вычислительный эксперимент с 4-х параметрической моделью. При этом модифицировался один из параметров – коэффициент ассоциации протеинов. Затем анализировались отклонения нейросетевого предсказания (аппроксимации) от результатов моделирования. На рисунке 15 представлены: зависимость эффективности переноса энергии от коэффициента ассоциации для аппроксимации и моделирования, относительные отклонения, автокорреляционная функция отклонений и оценка их плотности распределения.



а)

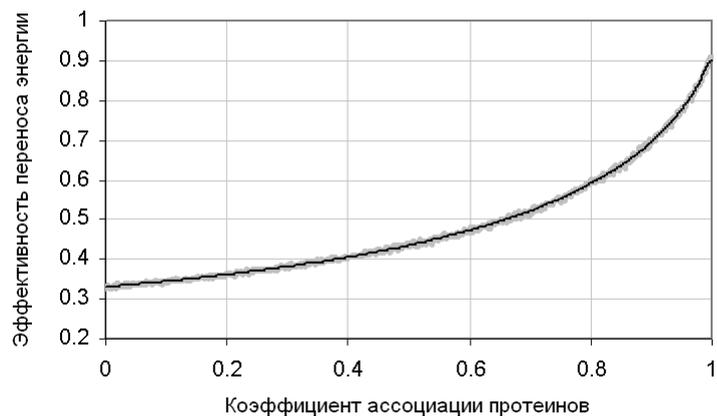


б)

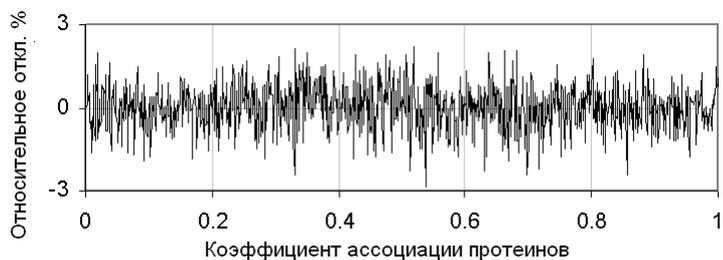
Рисунок 14 – ИНС аппроксимация результатов имитационного моделирования (а) и относительные отклонения при этом (б). На рисунке (а) кружками показаны значения полученные методом имитационного моделирования, линией – ИНС аппроксимации.

Поскольку автокорреляционная функция близка к  $\delta$ -функции, можно сказать что поведение отклонений стохастично, и, следовательно, эти отклонения являются результатом погрешности имитационного моделирования. Распределение отклонений близко к гауссову с математическим ожиданием равным 0.

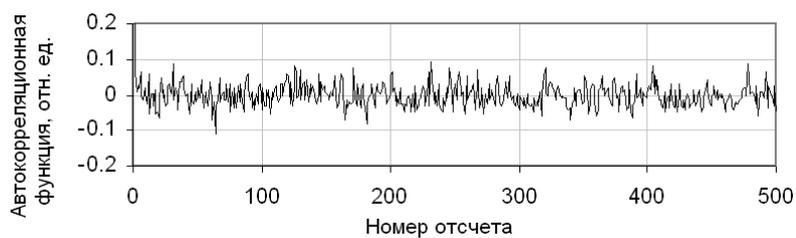
Следует отметить, что МСП с сигмоидальными активационными функциями дает гладкую аппроксимацию зашумленных результатов имитационного моделирования. Поэтому операция подгонки параметров оперирует с менее случайной функцией невязки  $\|F - F^{**}\|$  содержащей меньшее число локальных "шумовых" минимумов.



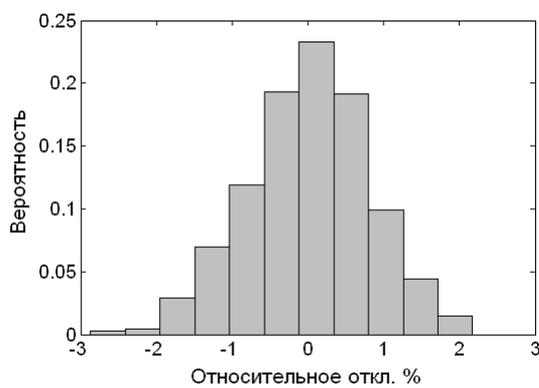
а)



б)



в)



г)

Рисунок 15 – Состоятельность ИНС аппроксимации. На рисунке (а) толстой серой линией показан результат имитационного моделирования, тонкой черной – нейросетевое предсказание. Ниже показаны относительные отклонения (б) и их автокорреляционная функция (в). Оценка распределения отклонений представлена на (г).

## 2.4 Выводы

Использование обученной ИНС вида МСП для замены имитационной модели приводит к значительному выигрышу по времени при решении задачи идентификации параметров сложной системы. В продемонстрированном примере ускорение составляло порядка  $5 \times 10^4$ . Более того, МСП генерирует гладкую аппроксимацию стохастических результатов моделирования, что уменьшает уровень зашумленности функции невязки. Это упрощает процедуру применения стандартных методов многопараметрической оптимизации для определения неизвестных параметров системы. Было показано, что отклонение результатов нейросетевого предсказания от результатов имитационного моделирования обусловлено большей частью погрешностью самого моделирования.

Тем не менее предложенный подход обладает несколькими ограничениями. Он работает только в том случае, когда число варьируемых параметров системы не превышает 6-ти (так называемое "проклятие размерности"). Кроме того, создание обучающего множества является весьма длительным с точки зрения машинного времени, хотя и может быть достаточно легко распараллелено в вычислительном кластере.

В заключение следует отметить, что метод нейросетевого моделирования может применяться не только в биофизике, но и в любой области науки, работающей со сложными экспериментальными системами (физика, кибернетика, экономика).

## 3 МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ

### 3.1 Подходы к моделированию стохастических процессов

Моделирование стохастического сигнала принципиально отличается от моделирования детерминистического по причине невозможности случайного сигнала. Таким образом, отпадает возможность использования метода обратного распространения ошибки для обучения. Для решения данной задачи можно воспользоваться одним из двух различных подходов.

Первый состоит в использовании классической, детерминистической нейронной сети (например, МСП) в качестве миксера и преобразователя базовых случайных величин в моделируемый сигнал. На входы сети при этом подаются некоррелированные равномерно распределенные случайные величины из диапазона  $[0,1]$ , а с выхода снимается искомый сигнал. Для обучения такой сети следует использовать особый механизм, описанный в следующем разделе. Следует однако учитывать, что сигнал на выходе такой сети будет всегда  $\delta$ -коррелированным, поскольку при генерации отклика многослойный персептрон не использует информацию о своих предыдущих состояниях.

Второй подход заключается в использовании стохастической нейронной сети (например, машины Больцмана). В этом случае сеть работает как некоторый генератор шума, используя для генерации отсчета не только текущие входы, но и свое состояние в течение нескольких последних итераций. Этот метод более сложен в применении чем предыдущий, но в перспективе он позволяет получить сколь угодно сложный сигнал с заданными статистическими параметрами, спектром и корреляционной функцией. К сожалению, СНС недостаточно изучены и используются на практике в меньшей степени, что подтверждается пробелами в литературных источниках.

Примером, иллюстрирующим второй подход, служит моделирование марковского случайного процесса. Следует отметить, что математический аппарат цепей Маркова применяют для анализа динамики смены состояний машины Больцмана.

Изменения состояний нейронов в МБ могут происходить синхронно или асинхронно по некоторому правилу. Алгоритм асинхронного изменения состояний отдельных нейронов может быть следующим:

```
i := random (n);  
si :=  $\sum_{j=1}^n w_{ij} x_j(t) - \mathcal{G}_i$ ;  
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);
```

### 3.2 Флуктуационные переходы в генераторе шума

Рассмотрим экспериментальную систему, в которой протекает стохастический процесс, подлежащий моделированию. В качестве такой системы был выбран полупроводниковый генератор шума. Большинство полупроводниковых приборов (диодов и транзисторов) работает в динамически неравновесных режимах. При этом в переходах возникают шумы, обусловленные различными факторами:

1. неодинаковое число электронов и дырок, проходящее через переход в единицу времени. Это высокочастотные дробовые шумы;
2. тепловой шум сопротивлений эмиттера, базы, коллектора;
3. поверхностными явлениями у переходов. Это низкочастотные шумы;
4. 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 полупроводникового генератора шума приведен на рисунке 16.

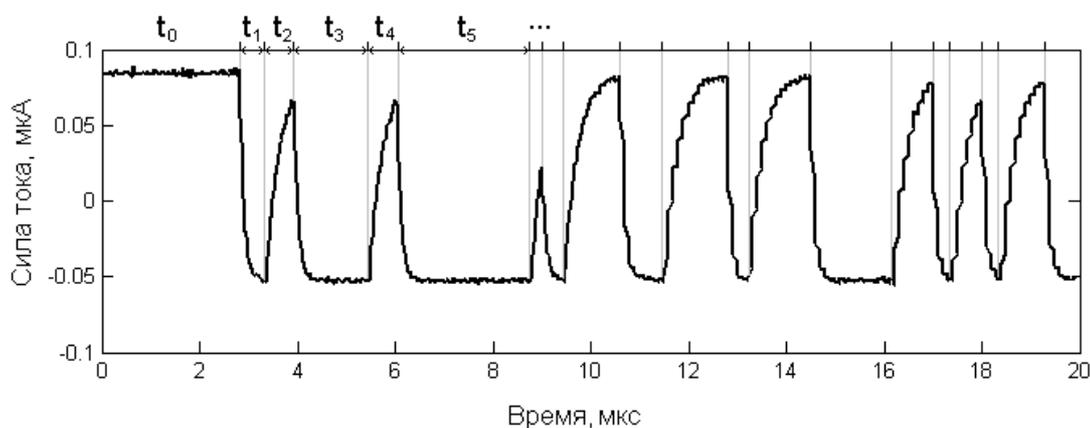


Рисунок 16 – Реальный сигнал, снятый с полупроводникового генератора шума

Напрямую этот сигнал аппроксимировать достаточно сложно. Что бы упростить ситуацию необходимо выделить основные логические элементы сигнала, то есть провести его декомпозицию. В качестве одного из таких элементов можно использовать времена  $t_0, t_1, t_2, t_3, \dots$  между событиями переходов в новое состояние (рисунок 16). Ряд временных интервалов полностью задает экспериментальный сигнал, поскольку, как видно из рисунка, поведение сигнала в стационарных состояниях и в процессе самих переходов достаточно тривиально. Представление

участка сигнала из интервала от 0 до 20 мкс в дискретно-событийной форме проиллюстрировано на рисунке 17. Аппроксимация этого слабокоррелированного сигнала и будет являться основной задачей раздела.

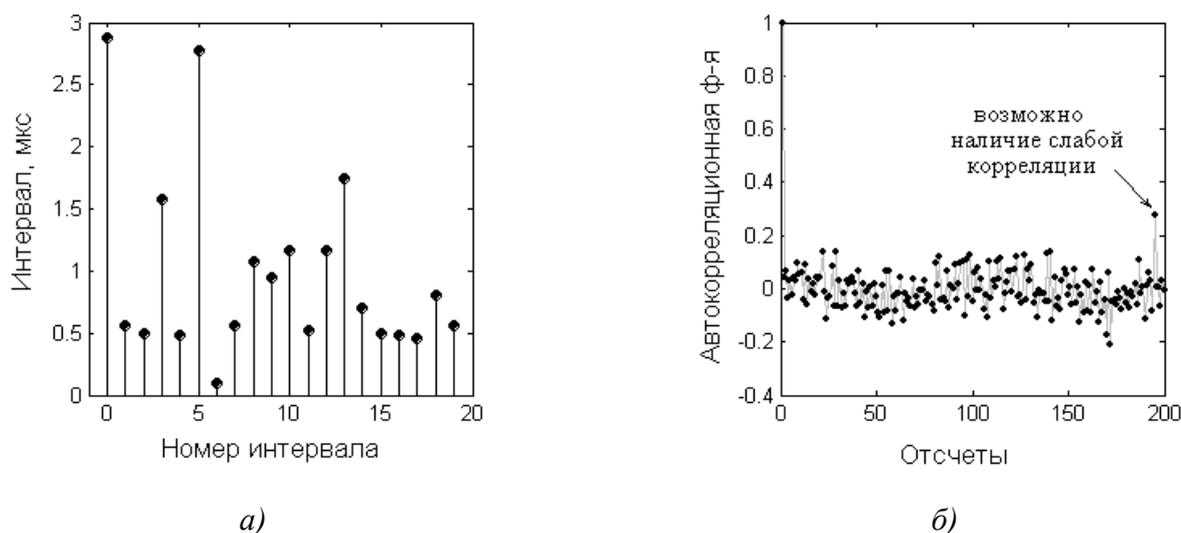


Рисунок 17 – Интервалы между переходами полупроводникового генератора шума в новые состояния (а) и автокорреляционная функция этих интервалов (б).

### 3.3 Моделирование стохастических процессов классической ИНС

#### 3.3.1 Схема моделирования

Предлагаемый метод аппроксимации строится на том утверждении, что две случайные последовательности имеющие одинаковые статистические характеристики могут быть рассмотрены как эквивалентные. Таким образом, при обучении сети следует сравнивать не значения полученных выходов с реальными, а характеристики выходного сигнала с характеристиками обучающего. В качестве "генератора случайности" в ИНС можно использовать входы, на которые будут подаваться базовые случайные величины (равномерно распределённые в диапазоне от 0 до 1).

Нейросетевое преобразование при аппроксимации случайного сигнала можно записать следующим образом.

$$\Psi(p_T, \{x(t)\}) = y_{ann}(t) \stackrel{стат.}{\approx} y(t) \quad (28)$$

где  $p_T$  – входные параметры системы,  $\{x(t)\}$  – набор базовых случайных величин,  $y_{ann}(t)$  – аппроксимация экспериментального сигнала  $y(t)$ .

В качестве статистических параметров в этой работе были использованы: математическое ожидание, среднеквадратическое отклонение, минимальное значение, максимальное значение и оценка функции распределения (гистограмма). Из этих величин составлялся вектор, который впоследствии умножался на весовые коэффициенты (поскольку не все элементы были одинаково статистически точны) и использовался в качестве дескриптора случайного процесса.

На рисунке 18 представлена упрощенная схема обучения такой сети.

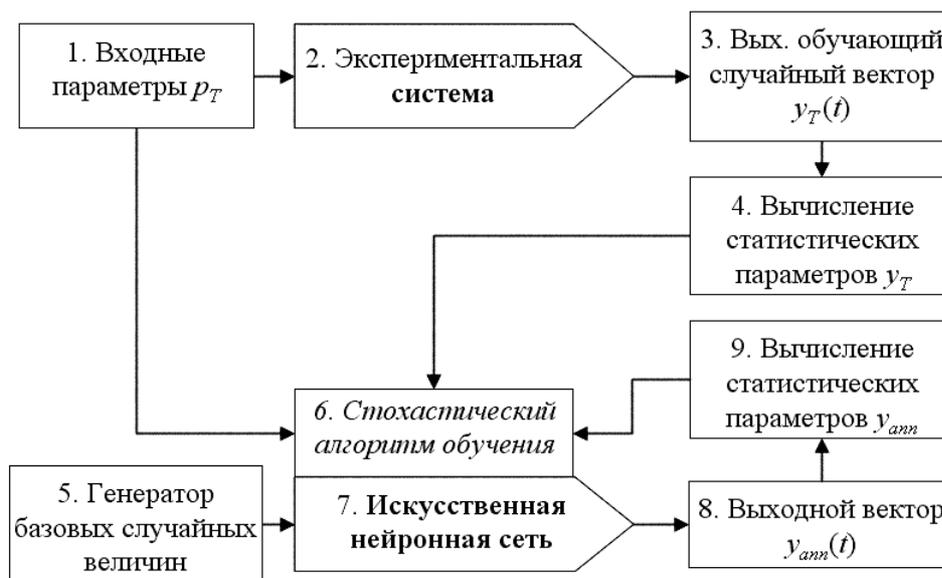


Рисунок 18 – Общая схема построения нейросетевой аппроксимации стохастического сигнала.

В качестве алгоритма обучения можно использовать любой не градиентный подход (симплекс метод, Монте-Карло, Нелдера-Мида, генетический алгоритм). В нашем случае использовался простейший метод Монте-Карло, состоящий в добавлении к весам случайных значений. В том случае если это приводило к уменьшению ошибки, изменения принимались. В противном случае восстанавливались старые значения и опять изменялись.

### 3.3.2 Моделирование тестового случайного сигнала

В качестве тестовой задачи был выбран случайный сигнал с распределением вида

$$y(t) = 0.1 \cdot n^2(t) \tag{29}$$

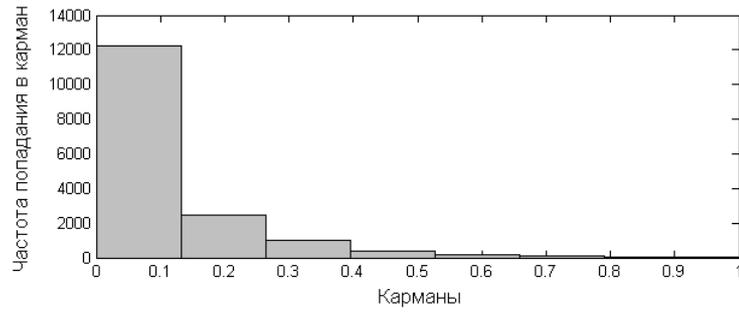
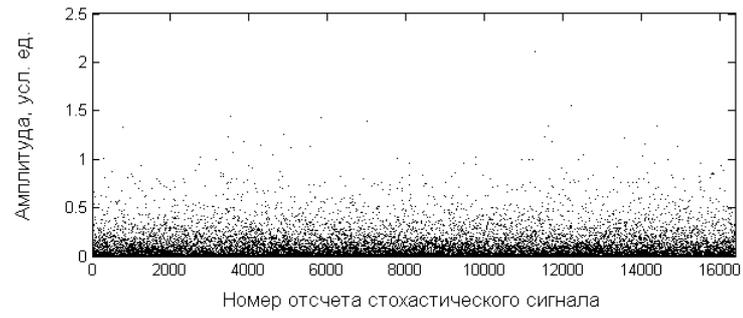
где  $n(t)$  – нормально распределенный случайный сигнал с  $m=0, \sigma=1$ . Нейронная сеть, аппроксимирующая его, содержала 4 входа и по 8 нейронов в первом и втором слое. Таким образом, в качестве  $\{x_i\}$  выступало 4 равномерно распределенных случайных величины. Сеть

обучалась простейшим стохастическим алгоритмом в три подхода по 1000 итераций в каждом. Размер обучающей и генерируемой выборки составлял 16384 отсчетов.

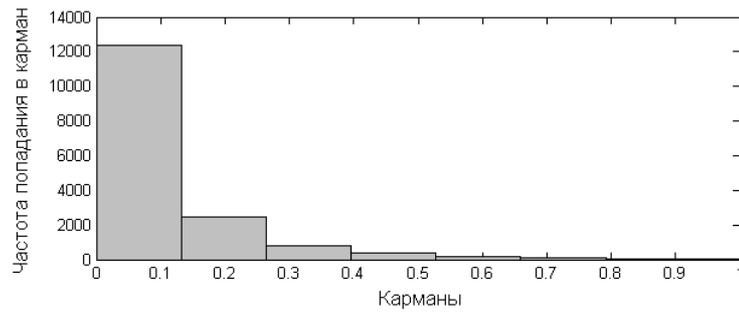
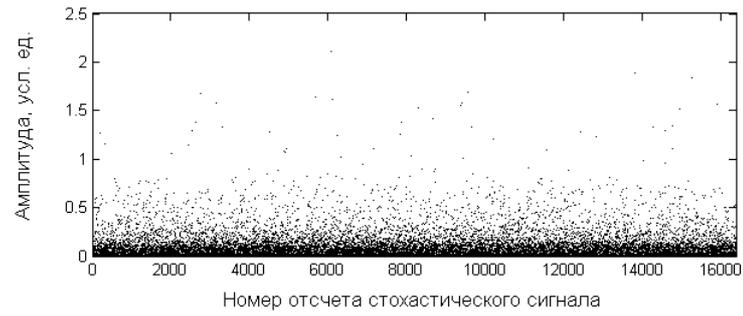
В таблице 5 и на рисунке 19 представлены результаты нейросетевого моделирования сигнала (29).

Таблица 5 – Параметры обучающего и сгенерированного сигналов

<b>Параметр</b>	<b>Значения для обучающего сигнала</b>	<b>Значения для сгенерированного сигнала</b>
среднее	0.0997	0.1062
среднеквадр. отклонение	0.1417	0.1413
минимальное значение	0.0000	0.0000
максимальное значение	2.1095	2.1106
карман гистограммы 1	1.0000	1.0000
карман гистограммы 2	0.1980	0.1967
карман гистограммы 3	0.0798	0.0653
карман гистограммы 4	0.0309	0.0288
карман гистограммы 5	0.0122	0.0165
карман гистограммы 6	0.0081	0.0081
карман гистограммы 7	0.0038	0.0019
карман гистограммы 8	0.0018	0.0004
остальные карманы (9-16)	0.0017	0.0023



*a)*



*б)*

Рисунок 19 – Обучающий (*a*) и сгенерированный (*б*) случайные сигналы. Гистограммы искусственно усечены до области  $[0,1]$ , как наиболее информативной.

### 3.3.3 Моделирование сигнала генератора шума

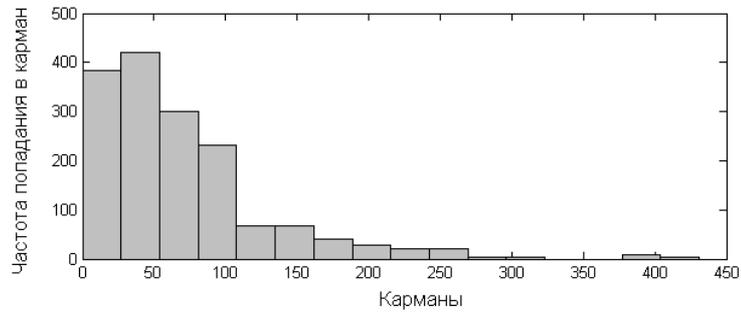
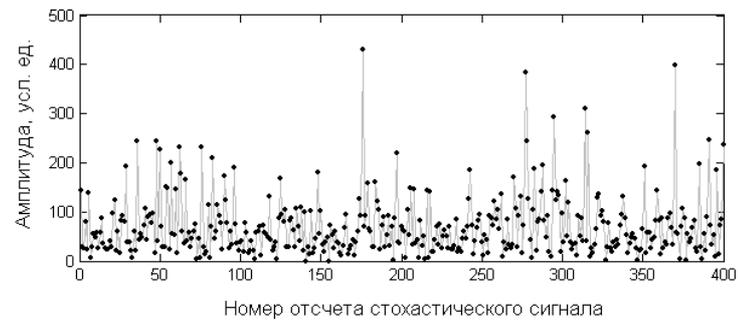
Рассмотрим нейросетевую аппроксимацию флуктуационных переходов в полупроводниковом генераторе шума (подраздел 3.2). При аппроксимации в качестве  $\{x_i\}$  выступало 4 равномерно распределенных случайных величины. Размер обучающей выборки составлял 400 отсчетов.

В таблице 6 и на рисунке 20 представлены результаты аппроксимации экспериментального сигнала. Несмотря на то, что аппроксимация достаточно адекватна, она несколько хуже, чем у тестового сигнала. Возможными причинами этого являются:

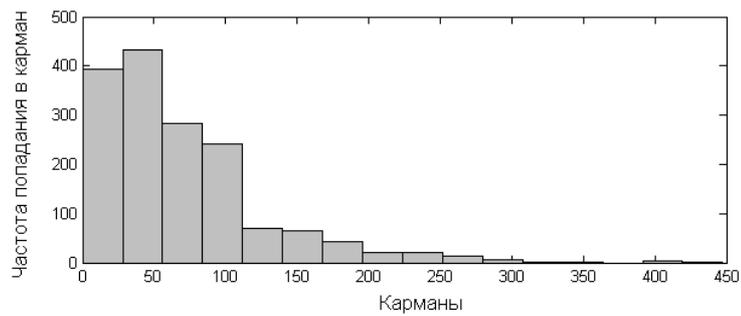
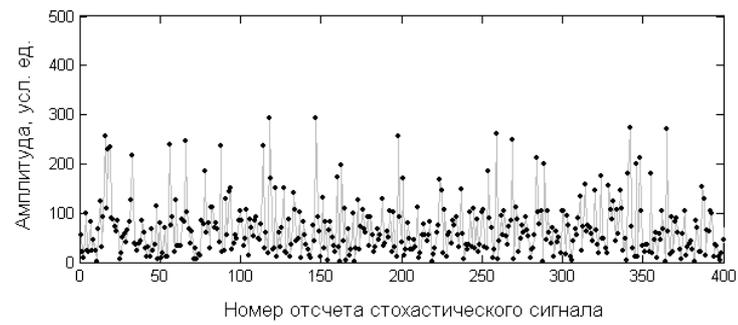
- недостаточный размер обучающей выборки;
- наличие корреляционной зависимости (рисунок 17б) и сложный характер флуктуационных процессов.

Таблица 6 – Параметры обучающего и сгенерированного сигналов

<b>Параметр</b>	<b>Значения для обучающего сигнала</b>	<b>Значения для сгенерированного сигнала</b>
среднее	69.473	69.338
среднеквадр. отклонение	61.920	59.873
минимальное значение	0.0000	0.0932
максимальное значение	431.00	447.38
карман гистограммы 1	1.0000	1.0000
карман гистограммы 2	0.7143	0.6559
карман гистограммы 3	0.5524	0.5589
карман гистограммы 4	0.1619	0.1617
карман гистограммы 5	0.1619	0.1478
карман гистограммы 6	0.0952	0.0970
карман гистограммы 7	0.0667	0.0508
карман гистограммы 8	0.0476	0.0462
остальные карманы (9-16)	0.0951	0.0669



*a)*



*б)*

Рисунок 20 – Экспериментальный (а) и сгенерированный (б) случайные сигналы.

### 3.4 Моделирование стохастических процессов машиной Больцмана

#### 3.4.1 Моделирование с помощью машины Больцмана

Как уже упоминалось, обучение машины Больцмана требует значительных вычислительных и временных ресурсов и является самостоятельной 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 рисунке 21.

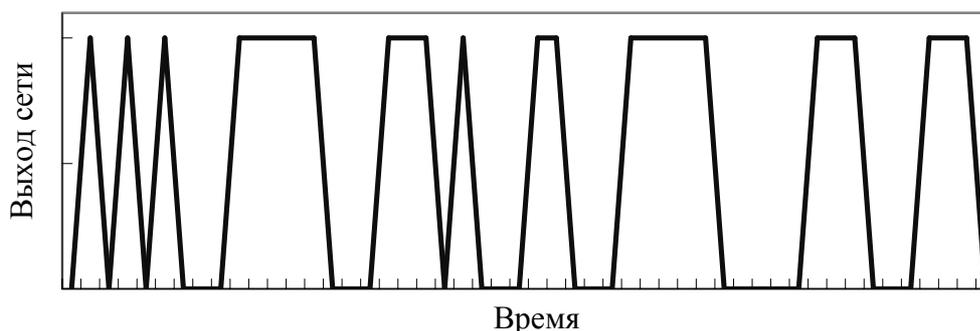


Рисунок 21 – Пример сигнала, сгенерированного СНС

Исходный текст программы для моделирования стохастического сигнала с помощью машины Больцмана в среде Matlab приведен в приложении 3.

#### 3.4.2 Анализ результатов

На практике возникает необходимость получения и обработки больших массивов данных с целью отбора из большой партии отдельных экземпляров генераторных диодов с заданными характеристиками. Известно, что объем информации, подлежащей обработке, можно уменьшить, используя различные методы сжатия [31]. Это позволит решить проблемы, связанные с хранением и передачей данных. Но возникает проблема дальнейшей обработки таких упакованных сигналов.

Известно, что в некоторых случаях можно обрабатывать сжатые данные без предварительной их распаковки. Интересно также, что для этого можно использовать нейронную сеть.

Основная идея метода заключается в следующем [32]. В большинстве случаев нет необходимости работать с полной выборкой, обычно бывает достаточно знать ее основные характеристики: дисперсию, математическое ожидание, распределение плотности и т.д. Известно, что классический метод Хаффмана для сжатия использует сведения о распределении (таблицу частот) исходного сигнала [31], причем таблица частот должна храниться вместе с упакованными данными.

Следовательно, можно упаковать реализацию сигнала методом Хаффмана, тем самым, уменьшив объем занимаемой ею памяти, а затем работать непосредственно с сохраненной таблицей частот. При этом остается возможность при необходимости распаковать данные для более тщательного анализа.

В частности, для обработки таблицы частот можно использовать ИНС. Например, соответствующим образом обученная ИНС может проводить анализ распределения сигнала и классифицировать его.

### ***3.5 Основные результаты и выводы***

В ходе проведенных экспериментов было показано, что детерминистические ИНС достаточно хорошо справляются с аппроксимацией некоррелированных случайных последовательностей. В то же время они неприменимы в случае, когда требуется аппроксимировать коррелированный сигнал с непостоянным спектром.

Использование для решения описанной задачи стохастических ИНС представляется весьма 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 ИНС без предварительного обучения.

В работе показана принципиальная возможность моделирования стохастического сигнала с помощью СНС, а также обработки и анализа полученных данных в сжатом виде без предварительной распаковки.

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.

## ЗАКЛЮЧЕНИЕ

Результаты проведенных исследований показывают целесообразность использования нейросетевых моделей процессов на этапе идентификации параметров сложных стохастических систем. Применение ИНС позволяет значительно ускорить анализ процессов, а также сделать его более устойчивым к экспериментальным шумам. В процессе работы были усовершенствованы методы моделирования детерминистических процессов, а также разработаны новые подходы к моделированию стохастических процессов с использованием как МСП, так и СНС.

Построенные алгоритмы нейросетевого моделирования будут применяться для изучения флуктуационных переходов в полупроводниковых приборах на кафедре системного анализа БГУ, а также для анализа фотофизических процессов в сложных биомолекулярных системах (мембранные протеины) в лаборатории биофизики Вагенингенского университета, Нидерланды (Laboratory of Biophysics, Wageningen University, The Netherlands).

Главный результат проведенных исследований состоит в следующем:

- усовершенствован метод идентификации параметров процессов и систем с помощью нейросетевой аппроксимации, обеспечивающей значительное ускорение вычислений;
- разработан метод моделирования стохастических процессов с помощью детерминистических нейронных сетей;
- в ходе работы была показана принципиальная применимость СНС, в частности машины Больцмана, для моделирования процессов флуктуационных переходов в шумовых полупроводниковых приборах.

Полученные результаты докладывались на четырех научных конференциях и были включены в сборники трудов:

- 60-й научной конференции студентов и аспирантов БГУ, Минск;
- European School on "Ultrafast processes in photochemistry and photobiology", Torun, Poland;
- 3-ей республиканской конференции молодых ученых и студентов "Современные проблемы математики и вычислительной техники", Брест;
- 8-ой научной конференции студентов и аспирантов РБ "НИРС-2003", Минск.

Кроме того, наиболее важные результаты проведенных исследований опубликованы в виде статей в рецензируемых изданиях [33, 34].

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Апанасович В. В., Тихоненко О. М. Цифровое моделирование стохастических систем. Минск: Университетское. 1986. 127с.
2. Большаков И. А., Гуткин Л. С., Левин Б. Р., Стратонович Р. Л. Математические основы современной радиоэлектроники. М.: Советское радио. 1968. 208 с.
3. Энциклопедия кибернетики. В 2-х томах. Т.2. Киев. Главная ред. Укр. Сов. энциклопедии. С.530.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Пер. с англ. Ю. А. Зуева и В. А. Точенова. М.: Мир. 1992. 184 с.
5. Лутковский В. М. Нейронные сети. Конспект лекций. Мн: БГУ. 2003.
6. Bishop C. M. Neural Networks for Pattern Recognition. Oxford: Clarendon Press. 1997. 477 p.
7. 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.
8. Головкин В. А. Нейроинтеллект: теория и применение. Кн. 2: Самоорганизация, отказоустойчивость и применение нейронных сетей. Брест: Изд-во БПИ, 1999. 264 с.
9. Van Schaik A. Building blocks for electronic spiking neural networks // Neural Networks. 2001. Vol. 14. P. 617–628.
10. Hangartner R. D., Cull P. Probabilistic computation by Neuromine Networks // BioSystems. 2000. Vol. 58. P. 167–176.
11. Теория нейронных сетей. Кн.1 Учебное пособие для вузов / Ред. А. И. Галушкин. М.: ИПРЖР. 2000. 416 с.
12. Loura L. M. S., et al. . Fluid-fluid membrane microheterogeneity: a fluorescence resonance energy transfer study. Biophysical Journal, 2001, 80. P. 776-788.
13. Andrews L., Demidov A., Eds. Resonance Energy Transfer. John Wiley & Sons Ltd Inc: New York. 1999.
14. Flecher R. Practical Methods of Optimisation. John Willey & Sons Inc., New York. 1987.
15. Shimbela A., Rapoport P. A Statistical Approach to the theory of the Central Nervous System // Bull. Math. Biophys. 1948, Issue 10. P. 41-55.
16. Farley B. G., Byerk W. A. Simulation of Self-Organizing systems by Digital Computer. // Trans. IRE Proffes. Group on Inform. Theory, Sept. 1954.
17. Zhao J., Shawe-Taylor J., Van Daalen M. Learning in Stochastic Bit Stream Neural Networks. // Neural Networks. 1996. Vol. 9. P. 991–998.

18. Card H. C. Dynamics of stochastic artificial neurons // *Neurocomputing*, 2001. Issue: 41. P. 173-182.
19. Rodriguez R. Noisy Spiking Neurons and Networks. // *BioSystems*. 1998. Vol. 48. P. 187–194.
20. Hinton O. E., Sejnowski T. J. Learning and relearning in Boltzmann machines. In *Parallel distributed processing*, Vol. 1. Cambridge, MA: MIT Press. 1986. P. 282-317.
21. 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.
22. Cybenko G. Approximations by superpositions of a sigmoidal function. *Math. Contr. Signals Syst.* 1989, 2. P. 304-314.
23. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators. *Neural Networks* 1989, 2. P. 359–366.
24. Stegemann J. A., Buenfeld N. R. A glossary of basic neural network terminology for regression problems. *Neural Comp. Appl.* 1999, Vol. 8. P. 290–296.
25. Tetko I. V., Livingstone D. J., Luik A. I. Neural network studies. 1. Comparison of overfitting and overtraining. *J. Chem. Inf. Comput. Sci.* 1995, 35. P. 826–833.
26. Förster T. In *Delocalized excitation and energy transfer, Modern quantum chemistry*; Sinanoglu, O., Ed.; Academic Press: New York. Vol. 3. 1965.
27. Stryer L. Fluorescence energy transfer as a spectroscopic ruler. *Annu. Rev. Biochem.* 1978, 47. P. 819–846.
28. Lakowicz J. R. *Principles of fluorescence spectroscopy*; Kluwer Academic/Plenum Publishers: New York, 1999. 698 p.
29. Meijer A. B., Spruijt R. B, Wolfs C. J. A. M, Hemminga M. A. Membrane-anchoring interactions of M13 major coat protein. *Biochemistry* 2001. Vol. 40. P. 8815–8820.
30. Spruijt R. B., Meijer A. B., Wolfs C. J. A. M., Hemminga M. A. Localization and rearrangement modulation of the N-terminal arm of the membrane-bound major coat protein of bacteriophage M13. *Biochim. Biophys. Acta* 2000, 1509. P. 311–323.
31. Ватолин Д. С. Алгоритмы сжатия изображений. М.: МГУ, 1999. 76 с.
32. Макарова Е. В., Поплетеев А. М., Назаров П. В., Лутковский В. М. Применение нейронных сетей для обработки упакованных данных в задачах эмиссионной спектроскопии. Сб. научн. работ студ. высших учебн. зав. Республики Беларусь "НИРС-2002". Мн.: Мин. образования РБ. 2003. С.419–423.
33. Nazarov P. V., Apanasovich V. V., Lutkovski V. M., Hemminga M. A., Koehorst R. B. M. Neural Network Simulation of Energy Transfer Processes in a Membrane Protein System. *Advances in Soft Computing: Neural Networks and Soft Computing*, Springer. 2003. P. 873-888.

34. Nazarov P. V., *et. al.* Artificial neural network modification of simulation-based fitting: application to a protein-lipid system. *Journal of Chemical Information and Computer Sciences*. 2004. In press.

## ПРИЛОЖЕНИЕ 1. НЕЙРОСЕТЕВАЯ АППРОКСИМАЦИЯ ДЕТЕРМИНИСТИЧЕСКОГО СИГНАЛА

Листинг программы NNModel42.m реализующей моделирование процесса переноса энергии в системе мембранных протеинов. Данные имитационного моделирования хранятся в файле SProtMod42\_NE\_d.mat

```
%*****
%* APPLICATION OF ANN TO MODEL SIMULATING
%*****
% nA ->|      |
% nD ->|      |
% R0 ->| ANN | -> I
% kAO ->|     |
% h_A ->|     |
%
clear all;

nD_R=[10,1000];
nA_R=[10,500];
R0_R=[5,50];
k_R=[0,1];
h_R=[-20,50];

I_R=[-0.01,0.17];
%E_R=[-0.05,1.1];

nTr=3^5+2300;
nTst=50;

load('d:\nazarov\matlab\DATA\SProtMod42_NE_d.mat');
%Creating Tr-pairs ([min mean max])
for iTr=1:(3^5)
    TX(1,iTr)=nA(iTr);
    TX(2,iTr)=nD(iTr);
    TX(3,iTr)=R0(iTr);
    TX(4,iTr)=kAO(iTr);
    TX(5,iTr)=h_A(iTr);

    TY(1,iTr)=(I(iTr)-I_R(1))/I_R(2);
end;
clear nA nD R0 kAO h_A;

load('d:\nazarov\matlab\DATA\SProtMod32_NE2d.mat');%_x100
%Creating Tr-pairs
for iTr=(3^5+1):nTr;
    n=iTr-3^5;
    TX(1,iTr)=nA(n);
    TX(2,iTr)=nD(n);
    TX(3,iTr)=R0(n);
    TX(4,iTr)=kAO(n);
    TX(5,iTr)=h_A(n);

    TY(1,iTr)=(I(n)-I_R(1))/I_R(2);
% TY(1,iTr)=(E(iTr)-E_R(1))/E_R(2);
end;
```

```

%Creating Tst-pairs
for iTst=1:nTst
    n=n+1;
    TstX(1,iTst)=nA(n);
    TstX(2,iTst)=nD(n);
    TstX(3,iTst)=R0(n);
    TstX(4,iTst)=kAO(n);
    TstX(5,iTst)=h_A(n);

    TstY(1,iTst)=(I(n)-I_R(1))/I_R(2);
    % TstY(1,iTst)=(E(n)-E_R(1))/E_R(2);
end;
%-----
nInp=5;
nOutp=1;
nL1=20;
nL2=20;
nL3=nOutp;
F1='logsig';
F2='logsig';
F3='logsig';

Range=[nA_R;nD_R;R0_R;k_R;h_R];
net = newff(Range,[nL1 nL2, nL3],{F1 F2 F3});

load NNModel01d_20 net

net.trainFcn='trainlm';
net.trainParam.epochs = 500;
net.trainParam.show = 1;
net.trainParam.goal = 1e-10;
net = train(net,TX,TY);

save NNModel01d net

tic
Y1 = sim(net,TX);
Y2 = sim(net,TstX);
toc

figure(1);clf;
subplot(2,1,1);
plot(1:nTr,TY, '.',1:nTr,Y1);
title('TRAINING: System-I and NNetwork-I');
axis([0,nTr,0,1]);

subplot(2,1,2);
plot(1:nTr,TY-Y1);
title('TRAINING: Errors');
axis([0,nTr,-0.5,0.5]);

e_tr=mean(abs(TY-Y1)./TY)

figure(2);clf;
subplot(2,1,1);
plot(1:nTst,TstY, '.',1:nTst,Y2);
title('TESTING: System-I and NNetwork-I');
axis([0,nTst,0,1]);
subplot(2,1,2);
plot(1:nTst,TstY-Y2);
title('TESTING: Errors');
axis([0,nTst,-0.5,0.5]);
e_tst=mean(abs(TstY-Y2)./TstY)

```

## ПРИЛОЖЕНИЕ 2. АППРОКСИМАЦИЯ СТОХАСТИЧЕСКОГО СИГНАЛА СТАНДАРТНОЙ ИНС

1. Листинг программы `nn_mod_rand_stat.m` реализующей моделирование тестового стохастического сигнала

```
clear all;

n=2^12;
n_itr=100;
n_runs=200;

%=====
% Experimental data
x=0.1*(randn(1,n).^2);
h=hist(x,8);h=normal(h);
F=[mean(x),std(x),min(x),max(x),h];
W=[4,4,1,1,1,1,1,1,1,1,1,1];
%=====
% ANN
nInp=2;
nOutp=1;
nL1=4;
nL2=4;
nL3=nOutp;
F1='logsig';
F2='logsig';
F3='purelin';
Range=zeros(nInp,2);Range(:,2)=1;

ERROR=1e10;
for i_runs=1:n_runs

net = newff(Range,[nL1 nL2 nL3],[F1 F2 F3]);

%=====
% Train

net0=net;
r=rand(nInp,n);
y=sim(net,r); h=hist(y,8);h=normal(h);

Fn=[mean(y),std(y),min(y),max(y),h];

e_min=sum(W.*abs(Fn-F));

a=0.1;
for itr=0:n_itr
    net.IW{1}=net.IW{1}+a*randn(nL1,nInp);
    net.LW{2}=net.LW{2}+a*randn(nL1,nL2);
    net.LW{6}=net.LW{6}+a*randn(nL3,nL2);

    net.b{1}=net.b{1}+a*randn(nL1,1);
    net.b{2}=net.b{2}+a*randn(nL2,1);
    net.b{3}=net.b{3}+a*randn(nL3,1);

    y=sim(net,r); h=hist(y,8);h=normal(h);
```

```

Fn=[mean(y),std(y),min(y),max(y),h];

e=sum(W.*abs(Fn-F));

if (e<e_min)
    net0=net;
    e_min=e;
else
    net=net0;
end

if (mod(itr,10)==0)
    disp(sprintf('%d: Min error = %f. Current a = %f',itr,e_min,a));
end;
a=a/1.001;
end;

if ERROR>e_min
    NET=net0;
    disp(sprintf('RUN #d: New NET found. Error = %f.',i_runs,e_min));
    ERROR=e_min;
    save nn_mod_rand_01 NET x
end;

end;

y=sim(NET,r);h=hist(y,8);h=normal(h);
F
Fn=[mean(y),std(y),min(y),max(y),h]

figure(1);clf
subplot(2,1,1);
plot(x); V=axis;
subplot(2,1,2);
hist(x,8);

figure(2);clf
subplot(2,1,1);
plot(y);axis(V) ;
subplot(2,1,2);
hist(y,8);

figure(3);clf
hold on;
plot(F,'b');
plot(Fn,'r');
hold off;

```

2. Листинг программы nn\_mod\_rand\_experimental.m реализующей моделирование флуктуационных переходов шумового диода. Экспериментальные данные (временные интервалы) хранятся в файле load nn\_mod\_rand\_experimental\_data.mat.

```

clear all;

% load experimental data
load nn_mod_rand_experimental_data x

% increase the size of training signal
% (to increase the size of simulated signal)
x=[x,x,x,x];

n=length(x);

n_itr=500;
karman=16;

% normilization of the signal
nrm=1/max(x);
x=x*nrm;

%=====
% Experimental data parameters

h=hist(x,karman);h=normal(h);

F=[mean(x),std(x),min(x),max(x),h];

W=ones(1,karman+4);
W(1:4)=[4,4,1,1];

%=====
% ANN
nInp=4;
nOutp=1;
nL1=4;
nL2=4;
nL3=nOutp;
F1='logsig';
F2='logsig';
F3='purelin';
Range=zeros(nInp,2);Range(:,2)=1;

net = newff(Range,[nL1 nL2 nL3],[F1 F2 F3]);

%=====
% Train

net0=net;

%load nn_mod_rand_experimental_01 net0
%net=net0;

r=rand(nInp,n);
y=sim(net,r); h=hist(y,karman);h=normal(h);

Fn=[mean(y),std(y),min(y),max(y),h];

e_min=sum(W.*abs(Fn-F));

```

```

a=1;
for itr=0:n_itr
    net.IW{1}=net.IW{1}+a*randn(nL1,nInp);
    net.LW{2}=net.LW{2}+a*randn(nL2,nL1);
    net.LW{6}=net.LW{6}+a*randn(nL3,nL2);

    net.b{1}=net.b{1}+a*randn(nL1,1);
    net.b{2}=net.b{2}+a*randn(nL2,1);
    net.b{3}=net.b{3}+a*randn(nL3,1);

    y=abs(sim(net,r)); h=hist(y,karman);h=normal(h);

    Fn=[mean(y),std(y),min(y),max(y),h];

    e=sum(W.*abs(Fn-F));

    if (e<e_min)
        net0=net;
        e_min=e;
    else
        net=net0;
    end

    if (mod(itr,10)==0)
        disp(sprintf('%d: Min error = %f. Current a = %f',itr,e_min,a));
    end;
    a=a/1.01;
end;

x=x/nrm;
y=abs(sim(net0,r));h=hist(y,karman);h=normal(h);
y=y/nrm;
F
Fn=[mean(y),std(y),min(y),max(y),h]

%-----

figure(1);clf
subplot(2,1,1);
plot(x(1:n)); V=axis;
V(1)=0;V(2)=n;
axis(V);
subplot(2,1,2);
hist(x,karman);

figure(2);clf
subplot(2,1,1);
plot(y);axis(V);
subplot(2,1,2);
hist(y,karman);

figure(3);clf
hold on;
plot(F,'b');
plot(Fn,'r');
hold off;

```

### ПРИЛОЖЕНИЕ 3. АППРОКСИМАЦИЯ СТОХАСТИЧЕСКОГО СИГНАЛА СНС

Листинг программы nn\_mod\_rand\_boltz.m реализующей моделирование стохастического сигнала машиной Больцмана.

```
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('weights.dat');
b = loadthrs('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);

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

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

newS = s;

%=====
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
%

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 bsign(z,c)>0.5,
        p(n)=1;
    else
        p(n)=0;
    end;
end;

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

```

---