

Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Энергетический»

Кафедра «Электрические станции, сети и системы»

РАБОТА ПРОВЕРЕНА

Рецензент

_____/А.Г. Журавский/

«__»_____2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____/И.М. Кирпичникова/

«__»_____2017 г.

Разработка элементов искусственного интеллекта для задачи установившегося режима в условиях прогноза режимных параметров

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

ЮУрГУ-13.04.02.2017.256.00 ПЗ ВКР

Руководитель работы, доцент

_____/В.С. Павлюков/

«__»_____2017 г.

Автор работы

Студент группы П-281

_____/А.М. Данилин/

«__»_____2017 г.

Нормоконтролёр, доцент

_____/В.С. Павлюков/

«__»_____2017 г.

Челябинск 2017 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(национальный исследовательский университет)

Институт Политехнический
Факультет Энергетический
Кафедра Электрические станции, сети и системы электроснабжения
Направление Электроэнергетика и электротехника

УТВЕРЖДАЮ

Заведующий кафедрой

_____/И.М. Кирпичникова/
_____ 2017 г.

ЗАДАНИЕ

на выпускную квалификационную работу студента

Данилин Андрей Максимович

(Ф. И.О. полностью)

Группа П-281

1. Тема выпускной квалификационной работы

Разработка элементов искусственного интеллекта для задачи установившегося режима в условиях прогноза режимных параметров

утверждена приказом по университету от 28.04. 2017 г. № 835(приложение №27)

2. Срок сдачи студентом законченной работы _____

3. Исходные данные к работе

Заданы исходные схемные и режимные параметры электрической сети 220 кВ

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов):

1. Обзор существующих методов решения установившегося режима, применяемых в существующих пакетах программ, производящих подобные расчеты

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

2. Выполнение сравнительного анализа следующих методов и прогнозных моделей

3. Построение системы расчёта режима в среде Mathcad и Python по результатам исследований.

5. Перечень графического материала (с точным указанием обязательных чертежей, плакатов в листах формата А1)

Схема и алгоритм обучения элемента искусственного интеллекта – плакат.

Всего 1 листов.

6. Консультанты по работе, с указанием относящихся к ним разделов работы

Раздел	Консультант	Подпись, дата	
		Задание выдал (консультант)	Задание принял (студент)
Обзор математических методов и методов расчета режимов сетей	Павлюков В.С.		
Искусственные нейронные сети	Павлюков В.С.		
Моделирование и анализ режимов электрических сетей при помощи нейронных сетей	Павлюков В.С.		

7. Дата выдачи задания _____

Руководитель _____

(подпись)

Задание принял к исполнению _____

(подпись студента)

										Лист
Изм.	Лист	№ докум.	Подпись	Дата						

13.04.02.2017.256.00 ПЗ

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов выпускной квалификационной работы	Срок выполнения этапов работы	Отметка о выполнении руководителя
Обзор математических методов и методов расчета режимов сетей		
Искусственные нейронные сети		
Моделирование и анализ режимов электрических сетей при помощи нейронных сетей		

Заведующий кафедрой _____ /И.М. Кирпичникова/

Руководитель работы _____ / _____ /

Студент _____ / _____ /

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

АННОТАЦИЯ

Данилин А.М. Разработка элементов искусственного интеллекта для задачи установившегося режима в условиях прогноза режимных параметров. – Челябинск: ЮУрГУ, П-281,2017, с., ил., библиогр.список - наим.,
1 лист чертежей ф. А1

В данной работе исследуется применение методов искусственного интеллекта для прогнозирования режимных параметров установившегося режима существующей электрической сети. Перед этим проведен анализ методов, применяемых на практике для расчетов режимов электрических сетей.

В данной работе исследованы существующие виды нейронных сетей, выполнена их классификация, исследованы методы обучения нейронных сетей и существующие способы их применения в практической деятельности.

Разработана программа в среде Маhtcad для обучения нейронной сети и прогноза режимных параметров.

					13.04.02.2017.256.00 ПЗ			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>Разработка элементов искусственного интеллекта для задачи установившегося режима в условиях прогноза режимных параметров</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Разраб.</i>		<i>Данилин А.М.</i>						
<i>Провер.</i>		<i>Павлюков В.С.</i>					5	
<i>Реценз.</i>		<i>Журавский А.Г.</i>				<i>ЮУрГУ кафедра ЭССиСЭ</i>		
<i>Н. Контр.</i>		<i>Павлюков В.С.</i>						
<i>Утверд.</i>		<i>Кирпичникова И.М.</i>						

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	
1 ОБЗОР МАТЕМАТИЧЕСКИХ МЕТОДОВ И МЕТОДОВ РАСЧЕТА РЕЖИМОВ СЕТЕЙ.....	
1.1 Точные методы.....	
1.2 Итерационные методы.....	
2 ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ.....	
2.1 Параллель из биологии.....	
2.2 История.....	
2.3 Однослойный перцептрон.....	
2.4. Многослойные нейронные сети. Алгоритм обратного распространения ошибки.....	
2.5. Сеть Хопфилда	
2.6 Самоорганизующиеся карты Кохонена	
2.7 Генетические алгоритмы обучения НС	
3 МОДЕЛИРОВАНИЕ И АНАЛИЗ РЕЖИМОВ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ ПРИ ПОМОЩИ НЕЙРОННЫХ СЕТЕЙ.....	
ЗАКЛЮЧЕНИЕ.....	
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	

												Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ							

ВВЕДЕНИЕ

Оптимальное управление электрической системой невозможно без предварительных расчетов режимов сетей. Расчет производится для установившихся и кратковременных режимов. Установившийся режим электрических систем и цепей рассчитывают при различных способах задания исходных данных в зависимости от физической сути и цели расчета. В качестве основных задач, решаемых на основе полученных расчетов, можно выделить следующие:

- 1) потери активной мощности ΔP по конкретным элементам и суммарные;
- 2) потери реактивной мощности ΔQ по конкретным элементам и суммарные;
- 3) распределение активной и реактивной мощности P и Q по сети соответственно;
- 4) потери напряжения ΔU ;
- 5) токораспределение в электрической системе.

Решение задачи установившегося режима при помощи таких современных технологий, как искусственные нейронные сети (ИНС), нечеткая логика, опорных векторов и т.д. и их композиции составляет один из наиболее активно развивающихся предметов исследований в энергетике. Прежде всего, это связано с тем, что данные структуры являются универсальными аппроксиматорами и способны моделировать сложные нелинейные зависимости, что гарантирует возможность успешного создания нейронной сети. Комбинирование и модификации интеллектуальных вычислительных технологий дают широкие возможности для эффективного решения задачи повышения точности решения установившегося режима.

Целью работы является исследование применения элементов искусственного интеллекта для задачи установившегося режима.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

Для достижения поставленной цели в данной работе ставятся и решаются следующие задачи:

- Произвести обзор существующих методов решения установившегося режима, применяемых в существующих пакетах программ, производящих подобные расчеты;
- Выполнение сравнительного анализа следующих методов и прогнозных моделей:
 - Нейросетевая модель (ИНС) с использованием эволюционных алгоритмов;
 - Нейросетевая модель (ИНС) с использованием алгоритма обратного распространения ошибки;
 - Нейросетевая модель (ИНС) прямого распространения;
 - Рекуррентная нейронная сеть.
- Построение системы расчёта режима в среде Mathcad по результатам исследований.

Методология и методы исследования. При решении поставленных задач использовались методы нейронных сетей, экспериментальные исследования и тестирование проводились в программе, реализованной в программном комплексе Mathcad.

Теоретическая значимость результатов диссертационной работы заключается в разработанной модели ИНС. Результаты работы могут быть использованы в учебном процессе в дисциплинах, связанных с исследованием применения различных методов для решения установившихся режимов.

Практическая значимость разработанной модели состоит в том, что она может быть использована для управления режимом в региональных диспетчерских управлениях - филиалах ОАО «Системный оператор единой энергетической системы». А также в региональных сетевых компаниях, в диспетчерских управлениях отдельных предприятий, которые являются членами оптового или розничного рынков электроэнергии и мощности.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

Достоверность полученных положений и результатов, сформулированных в диссертации, подтверждается корректным использованием методов математического моделирования и прогнозирования временных рядов, искусственных нейронных сетей, методов эволюционного моделирования и результатами компьютерного моделирования.

Главные положения, выносимые на защиту:

1. Модель по решению режима на базе ИНС;
2. Итоги сравнительного разбора способов обучения ИНС и архитектур ИНС.

Апробация работы. Основные положения диссертационной работы докладывались, дискуссировались и возымели позитивную оценку на конференции кафедры.

В первой главе представлен обзор существующих методов решения задачи установившегося режима, находящих применение в программах, которыми пользуются современные сетевые компании.

Во второй главе выполнен обзор ИНС, видов сетей, основ работы, способов обучения и видов деятельности человека, в каких используются ИНС.

В третьей главе исследовано применение ИНС для нахождения режимных параметров исследуемой электрической системы.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

1 ОБЗОР МАТЕМАТИЧЕСКИХ МЕТОДОВ И МЕТОДОВ РАСЧЕТА РЕЖИМОВ СЕТЕЙ

1.1 Точные методы

Для расчета электрических систем недостаточно использовать законы Кирхгофа и Ома. В связи со сложностью конфигураций существующих сетей на основе вышесказанных законов были созданы расчетные методы, базирующиеся на балансах (небалансах) узловых токов и мощностей (в т.ч. метод узловых потенциалов и метод контурных токов) и другие редко встречающиеся в практике. Согласно этим методам можно составить системы линеаризованных (далее линейных) уравнений.

Методы решения систем линейных уравнений, описывающих или моделирующих установившийся режим можно разделить на две группы: точные (или прямые) и итерационные (или приближенные), которые можно записывать в разных системах координат (прямоугольной, полярной и других).

Если в результате определённого числа операций метод расчёта даёт возможность получить точные значения неизвестных, такой метод можно назвать точным или прямым.

Если метод расчёта допускает получить решение системы линеаризованных уравнений лишь с установленной мощностью, такой метод можно назвать итерационным или приближенным[1].

Метод последовательного исключения (метод Гаусса) – универсализированный способ решения систем линейных алгебраических уравнений (СЛАУ). Точное решение СЛАУ возможно получить, только строго следуя методу Гаусса. Алгоритмы реализации метода Гаусса различны. Наиболее употребим алгоритм единственного деления, который также называют алгоритмом с обратным ходом, при обращении матриц используется алгоритм метода Гаусса без обратного хода или схема Жордана.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

Главное достоинство метода Гаусса – получение решения в результате n необходимых операций, где n – число уравнений в системе. В этом заключается выигрыш метода Гаусса над итерационными методами, для которых необходимое число вычислений зависит от сходимости итерационного процесса.

Главный недостаток метода Гаусса – необходимость хранить в памяти ЭВМ матрицу элементов СЛАУ. Приходится применять специальные способы, учитывающие малую заполненность матрицы проводимостей, для возможности применения метода Гаусса. Из этого вытекает следующий недостаток – необходимость выделения большого объема памяти ЭВМ при вычислении режимов крупных электроэнергетических систем.

К точным методам решения линейных уравнений также следует отнести решение с помощью матрицы Z_y . Матрица Z_y – это матрица собственных и взаимных сопротивлений узлов, она является обратной по отношению к матрице собственных и взаимных проводимостей:

$$\underline{Z}_y = \underline{Y}_y^{-1}. \quad (1.1)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

Следовательно, справедливо следующее условие:

$$\underline{Z}_y \underline{Y}_y = \underline{Y}_y \underline{Z}_y = \underline{E} \quad (1.2)$$

где \underline{E} – единичная матрица.

Тогда умножив обе части матричного уравнения узловых напряжений

$$\underline{Y}_y \underline{U} = \underline{I} \quad (1.3)$$

на матрицу \underline{Z}_y получим следующее:

$$\underline{Z}_y \underline{Y}_y \underline{U} = \underline{Z}_y \underline{I} \quad (1.4)$$

или с учётом (1.2):

$$\underline{U} = \underline{Z}_y \underline{I}. \quad (1.5)$$

Формула (1.5) даёт решение уравнения узловых напряжений.

Помимо описанных выше методов существует третий метод решения уравнений узловых напряжений с помощью обратной матрицы при напряжении балансирующего узла неравном нулю.

Решение следует начинать с выражения:

$$\underline{U} - \underline{U}_B = \underline{Z}_y \underline{I} \quad (1.6)$$

или

$$\underline{U} = \underline{U}_B + \underline{Z}_y \underline{I}, \quad (1.7)$$

где \underline{U} и \underline{I} – вектор-столбцы узловых напряжений и токов в узлах, \underline{U}_B – вектор-столбец, каждый элемент которого равен напряжению балансирующего узла:

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

$$\underline{U}_B = \begin{bmatrix} U_B \\ U_B \\ U_B \end{bmatrix}. \quad (1.8)$$

При расчёте режимов систем переменного тока выражение (1.7) следует преобразовать к следующему виду:

$$\underline{U} = \underline{U}_B + \sqrt{3} \underline{Z}_y \underline{I}. \quad (1.9)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Итерационные методы

Простая итерация и метод Зейделя – простейшие из итерационных методов. Рассмотрение простой итерации важно для понимания сути применения итерационных методов расчёта установившихся режимов электрических систем.

Многие физические процессы в электроэнергетике можно описать следующей математической моделью в матричной форме:

$$\underline{A} \underline{X} = \underline{B}, \quad (1.10)$$

где \underline{A} – матрица коэффициентов, которая может моделировать сложные схемные элементы электрической системы;

\underline{B} – матрица, моделирующая, в общем случае, режимные параметры;

\underline{X} – матрица искомых режимных параметров (смешанные параметры: $\underline{U}, \underline{I}(\underline{S})$)

Следующим этапом необходим переход к алгебраической системе уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (1.11)$$

Перепишем систему уравнений в следующем виде:

$$\begin{cases} x_1 = (a_{11} + 1)x_1 + a_{12}x_2 + \dots + a_{1n}x_n - b_1, \\ x_2 = a_{21}x_1 + (a_{22} + 1)x_2 + \dots + a_{2n}x_n - b_2, \\ \dots \\ x_n = a_{n1}x_1 + a_{n2}x_2 + \dots + (a_{nn} + 1)x_n - b_n \end{cases} \quad (1.12)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

или

$$\begin{cases} x_1 = c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n - b_1, \\ x_2 = c_{21}x_1 + c_{22}x_2 + \dots + c_{2n}x_n - b_2, \\ \dots \\ x_n = c_{n1}x_1 + c_{n2}x_2 + \dots + c_{nn}x_n - b_n. \end{cases} \quad (1.13)$$

В общем виде

$$\underline{X} = \underline{C} \underline{X} - \underline{B}, \quad \text{где } \underline{C} = \underline{A} + \underline{E}, \quad (1.14)$$

где \underline{E} – единичная матрица.

Решение системы уравнений отыскивается с использованием итерационного процесса. Кратко представим алгоритм рассматриваемого итерационного процесса:

1. Задаётся вектор искомых переменных (начальных, стартовых, ..)

$$\left([\underline{X}]^{(0)}\right)^T = \left([\underline{X}_1, \underline{X}_2, \dots, \underline{X}_n]^{(0)}\right)^T, \quad (1.15)$$

где T – обозначает операцию транспонирования,

(0) – начальное приближение вектора искомых переменных.

2. Формируются текущие итерации для определения составляющих вектора неизвестных параметров. Текущие итерации являются результатом подстановки искомых переменных предыдущей итерации

$$\left([\underline{X}]^{(1)}\right)^T = \underline{X}_1^{(1)}, \underline{X}_2^{(1)}, \dots, \underline{X}_n^{(1)}, \quad (1.16)$$

где $\underline{X}_i^{(1)} = C_{i1}\underline{X}_1^{(0)} + C_{i2}\underline{X}_2^{(0)} + \dots + C_{in}\underline{X}_n^{(0)} - b_i$.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

Краткая форма для искомой переменной будет

$$\underline{X}_i^{(k)} = \sum_{j=1}^n C_{ij} X_j^{(k-1)} - b_i,$$

где $i = \overline{1, n}$; $k = 0, 1, 2, \dots$ – приближение, (1.17)
которое зависит от точности нахождения переменной.

3. Окончание итерационного цикла определяется с помощью выражения в виде следующего неравенства

$$\max \left\{ \left| X_i^{(k)} - X_i^{(k-1)} \right| \right\} < \varepsilon_x,$$

где ε_x – задаётся ранее. (1.18)

Если неравенство не выполняется, то итерационный процесс продолжается до выполнения условий неравенства.

Рассмотрим ещё вариант метода простой итерации на следующем примере

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3. \end{cases}$$

или в матричной форме записи (1.19)

$$[\underline{a}][\underline{X}] = [\underline{b}].$$

Разрешаем каждое из этих уравнений системы, относительно одной из переменных, причем из первого уравнения определяем x_1 , из второго уравнения определяем x_2 , из третьего уравнения – x_3 , получается эквивалентная форма уравнений относительно исходной системы:

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} - 0 \cdot x_1 - \frac{a_{12}}{a_{11}} x_2 - \frac{a_{13}}{a_{11}} x_3; \\ x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1 - 0 \cdot x_2 - \frac{a_{23}}{a_{22}} x_3; \\ x_3 = \frac{b_3}{a_{33}} - \frac{a_{31}}{a_{33}} x_1 - \frac{a_{32}}{a_{33}} x_2 - 0 \cdot x_3, \end{cases} \quad (1.20)$$

где $a_{11} \neq 0, a_{22} \neq 0, a_{33} \neq 0$.

Систему (1.20) ещё называют равносильной [4,5].

Система уравнений в матричной форме будет иметь вид

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}} \end{bmatrix} + \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad (1.21)$$

или в более компактной форме

$$[\underline{x}] = [\underline{\beta}] + [\alpha][\underline{x}],$$

здесь

$$\begin{aligned} [\underline{\beta}] &= \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}} \end{bmatrix}, \quad [\alpha] = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & 0 & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & 0 \end{bmatrix}. \end{aligned} \quad (1.22)$$

Уравнение может рассматриваться как обобщенная форма записи эквивалентной системы уравнений.

Матричные уравнения можно представить ещё в другой форме:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} - \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}^{-1} \cdot$$

$$\cdot \begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \text{ или} \tag{1.23}$$

$$[\underline{x}] = [a_{ii}]^{-1} [\underline{b}] - [a_{ii}]^{-1} [a_{(-i)}] [\underline{x}].$$

Алгоритм решения:

1. Принимаются для искомым переменных вектора \underline{x} некоторые

начальные значения:
$$\left([\underline{X}]^{(0)}\right)^T = \left([\underline{X}_1, \underline{X}_2, \dots, \underline{X}_n]^{(0)}\right)^T;$$

2. Определяются текущие значения элементов вектора $[\underline{x}]$:

$$[\underline{x}]^{(1)} = [\underline{\beta}] + [\underline{\alpha}] [\underline{x}]^{(0)};$$

3. Проверка погрешности решения:
$$\left| \underline{x}^{(0)} - \underline{x}^{(1)} \right| \leq \varepsilon_x. \tag{1.24}$$

Метод Зейделя также как и метод простой итерации, базируется на использовании уравнений, приведенных, допустим, к такому виду:

$$\begin{cases} x_1 = \frac{1}{a_{11}} (b_1 - a_{12}x_2 - \dots - a_{1n}x_n), \\ x_2 = \frac{1}{a_{22}} (b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n), \\ \dots \\ x_n = \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1}). \end{cases} \quad (1.25)$$

Однако в отличие от метода простой итерации, для вычисления i -ой переменной на k -ом шаге итерационного процесса используются значения переменных вычисленные как на предыдущем $(k-1)$ -м шаге, так и на данном.

При этом на k -ом шаге итерационного процесса система уравнений имеет вид:

$$\begin{cases} x_1^{(k)} = \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(k-1)} - \dots - a_{1n}x_n^{(k-1)}), \\ x_2^{(k)} = \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)} - \dots - a_{2n}x_n^{(k-1)}), \\ x_3^{(k)} = \frac{1}{a_{33}} (b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)} - a_{34}x_4^{(k-1)} - \dots - a_{3n}x_n^{(k-1)}), \\ \dots \\ x_{n-1}^{(k)} = \frac{1}{a_{n-1,n-1}} (b_{n-1} - a_{n-1,1}x_1^{(k)} - \dots - a_{n-1,n-2}x_{n-2}^{(k)} - a_{n-1,n}x_n^{(k-1)}), \\ x_n^{(k)} = \frac{1}{a_{n,n}} (b_n - a_{n,1}x_1^{(k)} - \dots - a_{n,n-1}x_{n-1}^{(k)}). \end{cases} \quad (1.26)$$

В матричной форме записи система уравнений представляется как

$$\underline{X} = \underline{B} + \underline{C} \underline{X}, \quad (1.27)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

где \underline{B} – вектор-столбец задающих (режимных) параметров;

\underline{C} – квадратная матрица с нулевыми диагональными элементами.

Представим матрицу коэффициентов \underline{C} в виде суммы верхней (\underline{C}_B) и нижней (\underline{C}_H) треугольных матриц, получим

$$\underline{X} = \underline{B} + \underline{C}_B \underline{X} + \underline{C}_H \underline{X}. \quad (1.28)$$

В соответствии с уравнением (1.28) итерационный процесс для системы алгебраических уравнений в матричной форме будет иметь следующую запись:

$$\underline{X}^{(k)} = \underline{B} + \underline{C}_B \underline{X}^{(k-1)} + \underline{C}_H \underline{X}^{(k)}. \quad (1.29)$$

Главное достоинство метода Зейделя выражается в простоте алгоритма и удобстве его реализации на ПК. Эффективность метода проявляется при учёте слабой заполненности матрицы узловых проводимостей, поскольку алгоритм такого учёта в методике Зейделя весьма прост.

Определяющим недостатком метода Зейделя можно назвать его медленную сходимость или даже расходимость при расчёте электроэнергетических систем с устройствами продольной компенсации, а также при расчетах предельных и неустойчивых режимов.

										Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ					

2. ИСКУСТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

Сеть нейронов, из которой состоит мозг человека, является производительной комплексной и параллельной системой переработки информации. Она может формировать свои нейроны так, чтобы организовать восприятие и распознавание изображения во много раз быстрее самых новых компьютеров. Распознавание лиц происходит в человеческом мозге за 100-120 мс, наряду с этим компьютер тратит на это несколько часов. Сегодня, как и 40 лет назад, бесспорно то, что мозг выполняет работу значительно более оперативно и принципиально другим образом, чем любая ЭВМ. Именно этот факт в течение многих лет подвигает и устремляет работы ученых по исследованию нейронных сетей.

Термин «нейронная сеть» является фактически биологическим термином, и поэтому нейронные сети должны в действительности называться Искусственными Нейронными Сетями (ИНС). Но не будем заострять на этом внимание, и, чтобы не путаться в терминах в дальнейшем, будем использовать общепринятый термин Нейронная Сеть. Реальная НС это собрание нейронов - крошечных ячеек, из которых состоит наш мозг. Сеть может состоять из нескольких миллиардов нейронов, связанных и взаимодействующих между собой различными методами. С помощью ИНС делается попытка воссоздания модели этих биологических структур, и в архитектуре и во взаимодействии. Но здесь имеется одна маленькая проблема: мы не знаем в точности, каким образом реальная НС работает. Архитектура реальной НС в корне изменяется от типа к типу и единственное на что мы можем опереться – мы знаем, каким образом устроена элементарная ячейка сети – нейрон. Подробно его строение будет рассмотрено ниже.

В отношении к применению сегодня НС в практике существуют много «за» и «против». К очевидным достоинствам НС можно отнести возможность применения их в тех областях, в которых традиционные методы не работают, например использование НС в системах, имеющих дело с постоянным притоком

						13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата			

больших объемов данных разного рода (например, в радарных и сонарных системах). Многие типы НС биологически верны, то есть с их помощью можно также изучить мыслительные процессы, которые происходят в мозгу человека. Современные разработки в области НС приводят нас к мысли, что в скором будущем, возможно, будет их применение в сортировке и обработке объектов, со скоростью превосходящей скорость любых современных компьютеров.

Будущее в этом плане представляется довольно ярким, однако сегодня у НС имеется большое количество недостатков, и к сожалению они во многом пересекаются с недостатками возможностей вычислительных машин. Одним из очевидных достоинств НС является существенное распараллеливание вычислительных процессов, однако процесс эмуляции алгоритмов НС на современных последовательных компьютерах может быть существенно заторможен, так как ЭВМ может решать в определенный момент времени только одну задачу. Соответственно скорость обработки данных при помощи НС очень низка, и поэтому НС в настоящее время исключены из списка жизнеспособных решений данной проблемы.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

2.1 Параллель из биологии

Первые шаги на пути исследования организации мозга сделал Сантьяго Рамон-и-Кахаль в 1911 году. Используя метод окраски нейронов солями серебра, который был создан несколько раньше Камилло Гольджи, Кахаль заметил, что мозг обладает клеточной структурой. Состав мозга – это не только нейроны, но и глиальные клетки, обеспечивающие автономную работу мозга. Кахаль растолковал нейроны как поляризованные клетки, получающие сигналы разветвленными отростками – аксонами (Рисунок 2.1). Окрашивание серебром предоставило возможность обнаружить большое разнообразие нейронов по их форме, развилистость дендритной части и длине аксона. Кахаль обнаружил разницу между клетками с короткими аксонами и длинными аксонами.

Нейроны служат проводниками информации вне зависимости от их строения.

Биты информации передаются по аксонам в форме электрических импульсов (потенциалов действия), амплитуда которых – около 100 мВ, а длительность 1 мс. Считается, что возникновение импульсов имеет связь с перемещением ионов натрия через клеточную мембрану из внеклеточной жидкости внутрь клетки, а точнее в ее цитоплазму. Внутриклеточная концентрация натрия примерно в 10 раз меньше его концентрации в межклеточном пространстве. Разность потенциалов сохраняется в состоянии покоя на отметке около -70 мВ. Ионы натрия медленно диффундируют в клетку, так как доступ лимитирован свойствами мембраны. Физическая и химическая стимуляция, изменяющая знак полярности мембраны, повышает ее проницаемость для ионов натрия. Приток натрия внутрь клетки еще сильнее меняет полярность мембраны, увеличивая её проницаемость.

						13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата			

Типичная структура нейрона

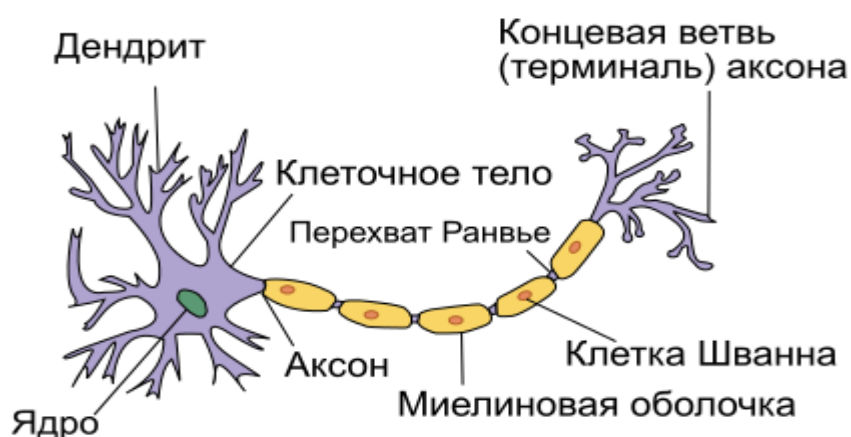


Рисунок 2.1 – Структура нейрона

После достижения порогового значения потенциала, положительная обратная связь ведёт к регенеративным сдвигам, в результате которых полярность меняется на противоположную. Таким образом, внутренность клетки заряжается положительно по отношению к внешнему пространству. Примерно спустя 1 мс проницаемость падает, и трансмембранный потенциал релаксируется к значению состояния покоя – 70 мВ. После каждого подобного изменения нейрон сохраняется на несколько мс рефрактным, то есть проницаемость в этот период имеет постоянное значение. Этот самый факт лимитирует частоту генерации импульсов – не более 200 Гц. Аксоны, имеющие сходство с проводами, иначе проводят ток, нежели провода. Активное сопротивление сравнительно велико, а мембранное сопротивление имеет маленькое значение. Положительный заряд затухает через 1-2 мм.

Электрическим импульсам необходимо восстанавливаться с целью преодоления порой расстояния в несколько см. Потребность заново увеличивать ток лимитирует предельную скорость распространения импульса по аксону до 100 м / с.

Связь нейронов друг с другом осуществляется химическими передатчиками – нейромедиаторами – выходящими из отростков нейронов в синапсах.

						13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата			

Молекулы медиатора проникают из клеточных пузырьков в синаптическую щель – отверстие размером 20 нм между мембранами клеток. Это происходит в том случае, когда потенциал действия добирается до отростка аксона. При достижении предела возбуждения, начинается управляемое образование молекул нейромедиатора. Освобожденные частицы нейромедиатора соединяются с рецепторами в постсинаптической мембране, что приводит к перемене её проницаемости. Когда перемена заряда подвигает мембранный потенциал к порогу образования импульса, эффект можно считать возбуждающим. Произвольный синапс приносит сравнительно малый эффект на динамичность аксона нейрона. Если каждый нейрон будет суммировать около 1000 входов, то интенсивность выхода определится. Нейрофизиологи выявили главную роль синапсов в обучении мозга примерно в начале XX века. Проходя через синапсы мозговые сигналы либо усиливаются, либо ослабляются. Верхняя часть человеческого мозга является обширной нейронной поверхностью толщиной в 3 мм и с площадью примерно 2200 квадратных сантиметров. Верхняя часть мозга вмещает примерно 1000 нейронов, что равно количеству звезд в нашей галактике. Произвольный нейрон имеет связь со 100 другими нейронами. Человеческий мозг охватывает примерно 1000 взаимосвязей. Примечательно, что мозг младенца и мозг взрослого индивида вмещает схожее число нейронов. Отличие состоит в том, что мозгу взрослого человека присуще организованность синаптических связей. Можно сделать вывод, что обучение мозга есть ни что иное как изменение структуры НС и настройка синапсов.

											Лист
Изм.	Лист	№ докум.	Подпись	Дата							

13.04.02.2017.256.00 ПЗ

2.2 История

В истории изучения искусственных нейронных сетей можно выделить три эпохи. Пионерами в изучении ИНС можно назвать МакКаллока и Питтса, которые в начале 40-х годов в своей работе рассмотрели идею построения аппаратов на основе пороговых функций, способных решать логические задачи. Впрочем, данная задача проектирования на поверку оказалась крайне сложной, и ее решение, которое было найдено через 20 лет, оказалось непростым, что создание автоматов, состоящих из десятков нейронов, стало просто невозможным. Вследствие этого поиск практической реализации аппаратов был связан с отысканием методов обучения схожих с нервной системой человека.

Обучаемые системы на базисе пороговых функций стали называть искусственные нейронные сети. Создание первых ИНС было произведено уже во вторую эпоху, после появления теоремы сходимости перцептрона Розенблатта в 1959 году. Уже в конце 60-х американские ученые проводили эксперименты на нескольких моделях ИНС. Подобные масштабные перцептроны в остальном мире были созданы в конце 70-х, начале 80-х годов в киевском НИИ. Схема перцептрона Розенблатта приведена на Рисунке 2.2. Перцептрон состоит трех слоев пороговых элементов. Входные данные, действуя на S-элементы, переводят рецепторы в пробужденное состояние. Связи между S-элементами и A-элементами имеют случай характер и веса. Выход со слоя A неравен нулю, только при преодолении порогового значения. Реакция слоя A приходит на слой R через связи, причем весь связей корректируется при обучении.

										Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ					

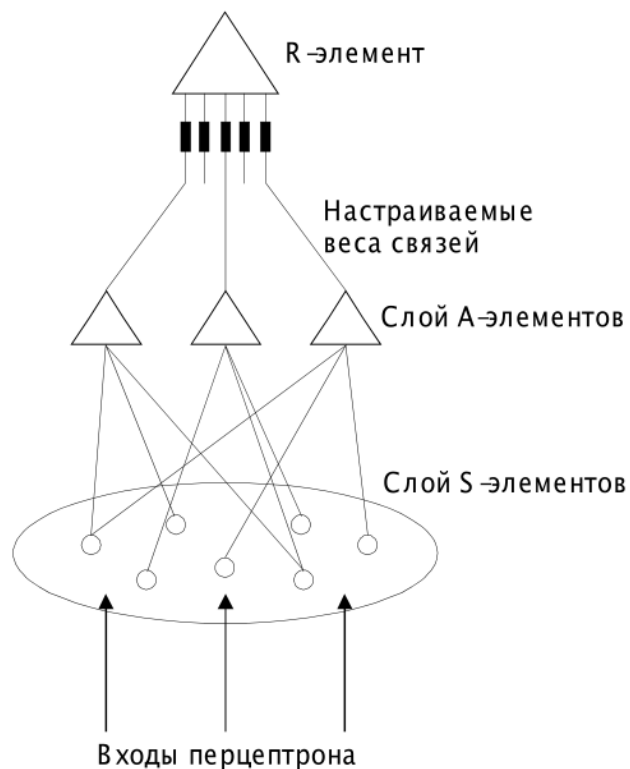


Рисунок 2.2 – Перцептрон Розенблатта

В эффекторах определяется весовая сумма пришедших импульсов. Как правило, один эффектор работает только с одним запоминаемым изображением, и решение определяется исходя из значения потенциала после синапса. Неоднородность перцептрона Розенблатта в дальнейших исследованиях была утрачена. Работа Минского и Пейперта выявила, что способности простейшего перцептрона лимитированы. Данные результаты остудили пыл ученых в области вычислительных наук. В связи с этим в изучении нейронных сетей наступил застой длиной 20 лет.

Оживление интереса к ИНС благоприятствовал ряд обстоятельств:

- трудности с реализацией логических систем искусственного интеллекта, создаваемых в рамках японского проекта ЭВМ 5-го поколения;
- появление принципиально новых идей в отношении организации памяти и обучения нейронных сетей;

- заметное отставание темпов роста производительности ЭВМ от стремительного увеличения потребности в высокопроизводительных вычислениях.

В 80-х годах благодаря энергетическому подходу Хопфилда, алгоритму обратного распространения для обечения ИНС созданному Вербосом наступила третья эпоха в исследовании нейронных сетей. Алгоритм получил широкое распространение благодаря работам Румельхарта в 1986 году.

С середины 80-х гг. последовательно повышается внимание к моделированию ИНС. Главная черта этих ИНС – распараллеливание операций, вследствие этого моделирование сетей на ПК требует больших временных затрат. По этой причине стали появляться специальные аппараты для ускорения процесса моделирования – нейрокомпьютеры. Нейрокомпьютинг – это научная дисциплина, направленная на разработку ЭВМ шестого поколения, состоящих из множества параллельных элементов (нейронов). Нейронная сеть образована из связанных нейронов. Они не нуждаются во внешнем управлении, производят одинаковые вычислительные операции. Высокая продуктивность системы из большого количества работающих параллельно элементов. Производство нейрокомпьютеров существует в большинстве развитых стран. Решения крупного ряда интеллектуальных задач попросту не возможно без нейрокомпьютеров. Над разработками в этой области трудятся множество крупных компаний. В их числе – Intel , DEC , IBM , Motorola . Крайне перспективными направлениями стало внедрение нейрокомпьютеров в системы управления мобильными объектами. Так в Финляндии и США проводят разработки по управления химическими реакторами с использованием нейрокомпьютеров. Для РФ данные разработки неактуальны в силу морального устаревания таких реакторов.

						13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата			

2.3 Однослойный перцептрон

Однослойный перцептрон является базовым понятием в теории НС, но однако до сих пор не существует одного отдельного определения этого понятия. В настоящее время имеется большое количество алгоритмов, базирующихся на перцептронах, но очень часто бывает, что эти алгоритмы обладают довольно малым количеством схожих черт. Основной областью применения однослойных перцептронов является распознавание простейших образов.

Структура однослойного перцептрона очень проста. Имеется определенное количество входов $X_1 \dots X_N$, вес каждого из которых определяется весовой функцией $W_1 \dots W_N$. Веса входных сигналов также называют синаптическими весами, так как они реализуют функцию аналогичную функции синапсов в реальной НС. Функцией отдельного перцептрона, схема которого представлена на Рисунке 2.3, является вычисление взвешенной суммы элементов входного сигнала, вычитая при этом значение смещения b , и пропуск результата через жесткую пороговую функцию, выход которой равен +1 или -1. В зависимости от значения выходного сигнала принимается решение: «+1» – входной сигнал принадлежит классу А, «-1» – входной сигнал принадлежит классу В. График передаточной функции порогового элемента изображен на Рисунке 2.4. Схема решающих областей, созданных в пространстве входных сигналов, для случая, когда размерность входного сигнала равна 2, показана на Рисунке 2.5. В данном случае разделяющая поверхность представляет собой прямую на плоскости. Входные сигналы над разделяющей линией относятся к классу А, под линией – к классу В. Уравнение, задающее разделяющую прямую, зависит от значений синаптических весов и смещения.

										Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ					

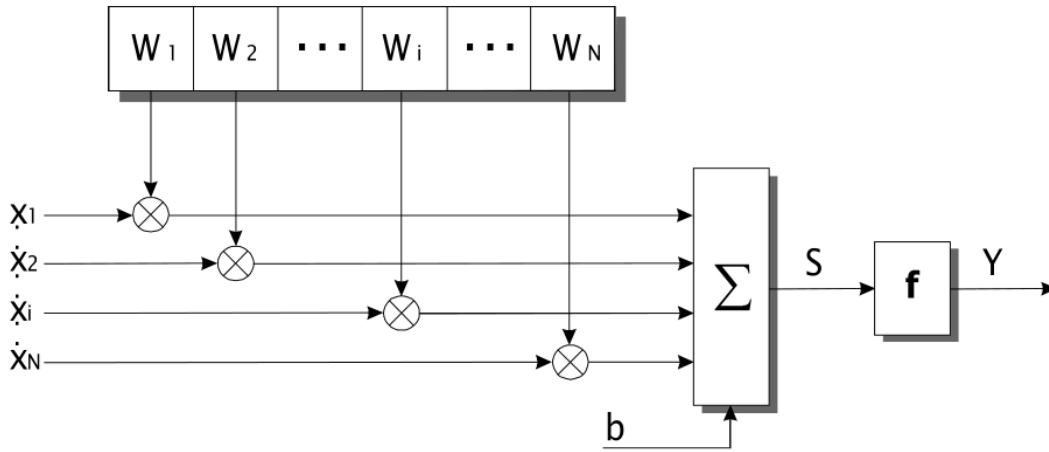


Рисунок 2.3 – Схема перцептрона

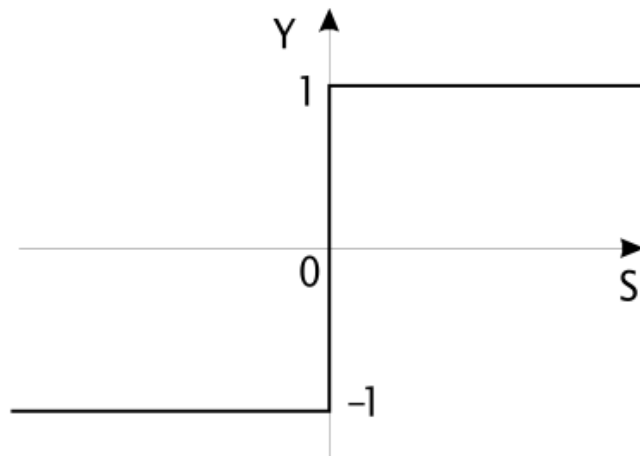


Рисунок 2.4 – Передаточная функция

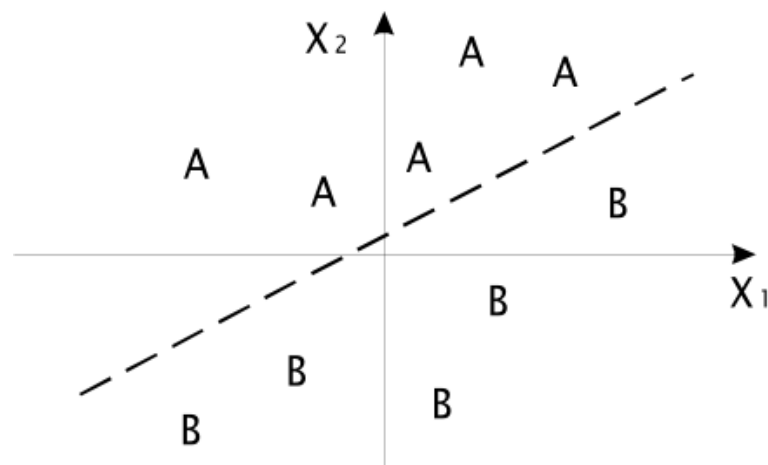


Рисунок 2.5 – Схема решающих областей

Алгоритм обучения однослойного перцептрона основан на изменении коэффициентов передачи синаптических весов, для более эффективного выполнения определенной задачи вследствие «ослабления» весового характера одних входов перцептрона и «усиления» других. Рассмотрим алгоритм обучения однослойного перцептрона более подробно с целью выяснения алгоритма программной реализации данного процесса.

1. *Инициализация синаптических весов и смещения:*

Значения синаптических весов $W_i(0)$ и смещение нейрона b устанавливаются равными некоторым малым случайным числам, где $W_i(t)$ – вес связи от i -го элемента входного сигнала ($0 \leq i \leq N$) к нейрону в момент времени t .

2. *Предъявление сети нового входного и желаемого выходного сигналов:*

Входной сигнал $X = (X_0, X_1 \dots X_{N-1})$ предъявляется нейрону вместе с желаемым выходным сигналом d .

3. *Вычисление выходного сигнала нейрона:*

$$y(t) = f\left(\sum_{i=0}^{N-1} W_i(t) X_i(t) - b\right) \quad (2.1)$$

4. *Адаптация (настройка) значений весов:*

$$W_i(t+1) = W_i(t) + r[d(t) - y(t)]X_i(t), \quad 0 \leq i \leq N-1 \quad (2.2)$$

$$d(t) = \begin{cases} +1, & \text{если } A \\ -1, & \text{если } B \end{cases}$$

где r – шаг обучения (должен быть меньше 1),

$d(t)$ – желаемый выходной сигнал.

Если сеть принимает правильное решение, то синаптические веса не модифицируются.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

5. Переход к шагу 2.

При программной реализации данного алгоритма размерности входа и выхода ограничены только возможностями вычислительной машины, на которой производится моделирование НС. При аппаратной реализации – технологическими возможностями. Типы входных сигналов могут быть бинарными (дискретными) или аналоговыми (непрерывными). Емкость сети на базе однослойного перцептрона совпадает с количеством нейронов.

Очевидными преимуществами НС на базе однослойного перцептрона в системах распознавания изображения являются простота программной и аппаратной реализации, а также простота и высокое быстродействие алгоритма обучения. Однако явным недостатком такой НС, как видно из Рисунка 2.5, является примитивность разделяющих поверхностей, что дает возможность решать только самые простые задачи распознавания. Многослойные перцептроны дают возможность строить более сложные разделяющие поверхности и поэтому имеют более широкое применение при решении задач распознавания.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

2.4 Многослойные нейронные сети. Алгоритм обратного распространения ошибки

Многослойная НС или многослойный перцептрон впервые были предложены и исследованы в 60-х годах Розенблаттом, Минским, Пейпертом и др. Но в дальнейшем исследования в данной области практически прекратились вследствие отсутствия эффективного алгоритма их обучения. Лишь в середине 1980-х несколькими исследователями независимо друг от друга был предложен алгоритм обучения многослойных перцептронов, основанный на вычислении градиента функции ошибки. Алгоритм был назван "обратным распространением ошибки". Стоит заметить, что в настоящее время алгоритм обратного распространения применяется конкретно для многослойных НС с последовательными связями.

На Рисунке 2.6 изображена укрупненная схема трехслойной НС, где нейроны помечены кружками, а связи между ними – стрелками. Для многослойных НС с последовательными связями характерно, что нейроны в таких сетях делятся на определенные группы с общим входным сигналом – слои, причем на каждый нейрон текущего слоя подаются все выходные сигналы с предыдущего слоя. В случае, если текущий слой является первым, то на каждый нейрон в его составе подаются все элементы внешнего входного сигнала.

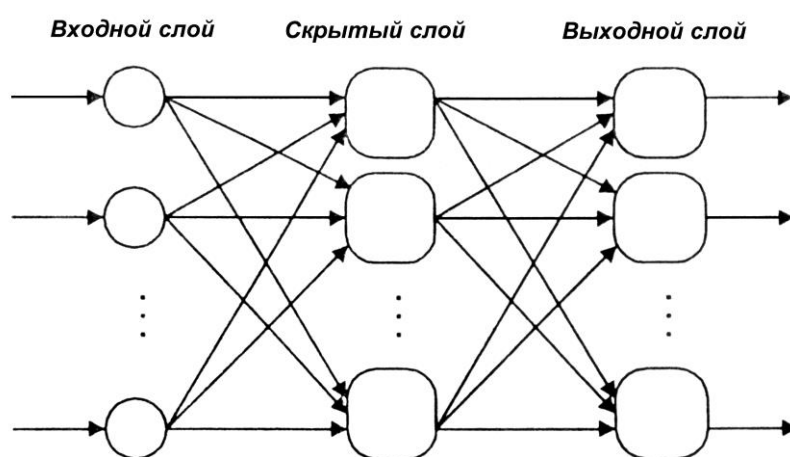


Рисунок 2.6 – Многослойная (двухслойная) сеть прямого распространения

Нейроны выполняют взвешенное суммирование элементов входных сигналов. К сумме элементов входных сигналов, домноженных на соответствующие синаптические веса, прибавляется смещение нейрона. Над результатом суммирования выполняется нелинейное преобразование - функция активации (передаточная функция). Значение функции активации есть выход нейрона.

Как видно из Рисунка 2.6 организация многослойных перцептронов гораздо более сложная, чем у однослойного. Таким образом встает вопрос о рациональности их применения при столь существенном недостатке. Основным преимуществом многослойных перцептронов является возможность их применения в задачах более высокого порядка сложности, чем могут позволить возможности однослойного перцептрона, за счет усложнения формы разделяющей прямой на плоскости входного изображения. Примеры архитектур многослойных НС и вид разделяющей прямой при неизменном входном изображении приведены на Рисунке 2.7.

Структура сети	Типы классифицируемых регионов	Классификация по операции XOR	Заполнение классификационного региона	Конечный вид классификационных регионов
	Два региона отделенных окружностью			
	Открытые выпуклые либо закрытые выпуклые регионы			
	Регионы произвольной формы			

Рисунок 2.7 – Примеры архитектур многослойной НС

Как уже отмечалось выше, применение многослойных НС на практике стало возможным благодаря разработке эффективного метода их обучения.

Однако на первый взгляд метод обратного распространения представляет собой довольно сложный для понимания и программирования алгоритм. Чтобы наиболее наглядным образом объяснить суть данного метода ненадолго вернемся к однослойному перцептрон.

В однослойной НС каждый нейрон регулирует его веса согласно тому, насколько ожидаемый результат на его выходе совпадает с желаемым результатом, что математически может быть выражено через формулу (2.3), называемую правилом дельты перцептрона :

$$\Delta W_i = X_i \delta, \quad (2.3)$$

где $\delta = (\text{желаемый выход}) - (\text{реальный выход})$.

Но это правило не может быть использовано в случае расширения сети до нескольких слоев, которое требуется для решения нелинейных задач, так как при настройке одного весового коэффициента внутри сети мы должны учесть, какое влияние это окажет на работу всей сети в целом. Чтобы это сделать, мы должны учесть производную функции ошибки относительно изменяемого веса. Однако на данном шаге возникает проблема неудовлетворения функции перцептрона условиям непрерывности и дифференцируемости на всем диапазоне определения. Поэтому наиболее популярными альтернативными функциями в данном случае являются сигмоидальные функции. Вид математического выражения наиболее простой сигмоидальной функции представлен в формуле (2.4), а ее график изображен на Рисунке 2.8.

$$f(s) = \frac{1}{1 - e^{-s}}, \quad (2.4)$$

										Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ					

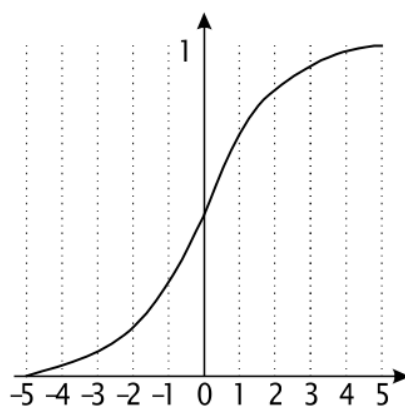


Рисунок 2.8 – Сигмоидальная функция

Как видно из рисунка график функции пролегает в диапазоне от 0 до 1 по оси Y и от -5 до +5 по оси X, а также является непрерывным и легко дифференцируемым. Таким образом формула вычисления весовой функции для (2.5) имеет вид:

$$\Delta W_i = \frac{\eta}{P} \sum_{p=1}^P X_{ip} \delta_p. \quad (2.5)$$

Это весьма сложный вид формулы, и при определенном приближении ее можно несколько упростить:

$$\Delta W_i = \eta X_i \delta, \quad (2.5)$$

$$\text{где } \delta = Y_p(1 - Y_p)(d_p - Y_p). \quad (2.6)$$

Теперь мы получили обобщенную формулу, которая используется всецело в многослойной НС, однако мы не выяснили каким образом происходит расчет ошибки в скрытых слоях. Скрытые слои НС регулируются различными значениями δ , и мы должны знать эффект на выходе нейрона скрытого слоя в зависимости от изменения веса. Поэтому мы должны знать производную функцию ошибки относительно изменяемого веса. Опустим вывод математического выражения правила дельты для данного случая, в случае надобности за более подробным описанием можно обратиться к специальной

литературе. Доказано, что для нейрона q в скрытом слое p правило дельты будет выражено следующим образом:

$$\delta_p(q) = Y_p(q) [1 - Y_p(q)] \sum W_{p+1}(q, i) \delta_{p+1}(i). \quad (2.7)$$

Таким образом именно δ_{p+1} член формулы определяет понятие «обратный» в названии метода – обратное распространение. Каждое значение дельта δ для скрытого слоя определяется предыдущим его значением, рассчитанным ранее. То есть для, например, 3-хслойной сети дельта выходного слоя будет рассчитываться по формуле (2.6), полученное значение будет использоваться для расчета дельты скрытого слоя по формуле (2.7). Отсюда название метода «обратное распространение» означает распространение ошибки от выходного слоя назад через сеть.

Представим еще несколько вариантов сигмоидальных функций, которые применяются в настоящее время в качестве передаточных функций нейронов:

1. Однопараметрические:

- Функция Ферми (экспоненциальная сигмоида):

$$f(s) = \frac{1}{1 + e^{2s}}, \quad (2.8)$$

где s – выход сумматора нейрона.

- Рациональная сигмоида:

$$f(s) = \frac{s}{|s| + \alpha}. \quad (2.9)$$

- Гиперболический тангенс:

$$f(s) = th \frac{s}{\alpha} = \frac{e^{\frac{s}{\alpha}} - e^{-\frac{s}{\alpha}}}{e^{\frac{s}{\alpha}} + e^{-\frac{s}{\alpha}}}. \quad (2.10)$$

2. Многопараметрические:

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

$$f(s) = \rho_1 \frac{s}{s + \rho_2} + \rho_3. \quad (2.11)$$

В однопараметрических функциях значение функции зависит от аргумента и только от одного варьируемого параметра, в многопараметрических – параметров может быть несколько. Наиболее эффективной передаточной функцией является рациональная сигмоида. Для вычисления гиперболического тангенса требуется больше всего тактов работы процессора.

Для облегчения программирования метода обратного распространения ошибки приведем краткий алгоритм его программной реализации.

1. Инициализация сети.

Весовым коэффициентам и смещениям сети присваиваются малые случайные значения из диапазонов и соответственно.

2. Определение элемента обучающей выборки.

Текущие входы $(X_0, X_1 \dots X_{N-1})$, должны различаться для всех элементов обучающей выборки. При использовании многослойного персептрона в качестве классификатора желаемый выходной сигнал $(D_0, D_1 \dots D_{N-1})$ состоит из нулей за исключением одного единичного элемента, соответствующего классу, к которому принадлежит текущий входной сигнал.

3. Вычисление текущего выходного сигнала.

Текущий выходной сигнал определяется в соответствии с традиционной схемой функционирования многослойной нейронной сети.

4. Настройка синаптических весов.

Синаптические веса настраиваются в соответствии с правилами метода обратного распространения, начиная с выходных нейронов сети, а затем проходя сеть в обратном направлении до первого слоя.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

Существует большое количество модификаций алгоритма обратного распространения, разработанных в последние десять лет исследований в данной области. Модификации связаны с использованием различных функций ошибки, процедур определения направления и величины шага, рассмотрим некоторые из них.

Различные виды функций ошибок:

- интегральные функции ошибки на всей совокупности обучающих примеров;
- функции ошибки целых и дробных степеней.

Процедуры определения величины шага на каждой итерации:

- дихотомия;
- инерционные соотношения;
- отжиг.

Процедуры определения направления шага:

- с использованием матрицы производных второго порядка (метод Ньютона и др.);
- с использованием направлений на нескольких шагах.

В качестве недостатков метода обратного распространения стоит упомянуть о низкой скорости обучения вследствие того, что на каждом шаге итерации происходят изменения значений параметров сети, улучшающие работу лишь с одним примером обучающей выборки. Классический метод обратного распространения относится к методам с линейной сходимостью, и для увеличения скорости сходимости необходимо использовать матрицы вторых производных функции ошибки.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

2.5 Сеть Хопфилда

Биофизик Джон Хопфилд работал в области организации ассоциативной памяти, начиная с 1974 года. Он является автором энергетического подхода к организации НС как динамической системы, рассеивающей энергию взаимодействуя с окружающей средой. Хопфилд рассматривал электрохимические явления, возникающие в результате взаимодействия между нейронами в сети. Им было предложено, что каждая физическая система, чей энергетический уровень зависит от суммы уровней каждого элемента данной системы, может быть представлена как динамическая память с автогенерацией адресов (ассоциативная память). Кроме того, физическая система может рассматриваться как память в случае, если энергетический уровень каждого ее элемента потенциально зависит от уровня элементов, преобладающих в этой системе и имеющих более устойчивое состояние. Хопфилд рассматривал две модели сети в зависимости от типа входов и выходов нейрона, к которому происходит обращение: дискретные и непрерывные. Первая модель является существенно упрощенной, так как нейрон принимает на входе бинарный сигнал и выдает также бинарный сигнал. Для второй модели Хопфилд попытался создать наиболее вероятную модель биологического функционирования нейрона, в котором входной и выходной сигнал являются непрерывными. Модели, созданные Хопфилдом, также могут применяться не только для организации ассоциативной памяти, но также для классификации\категоризации и решения некоторых задач оптимизации.

Одна из первых предложенных моделей сети Хопфилда используется как ассоциативная память. Исходными данными для расчета значений синаптических весов сети являются векторы - образцы классов. Сеть функционирует циклически. Выход каждого из нейронов подается на входы всех остальных нейронов (Рисунок 2.9).

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

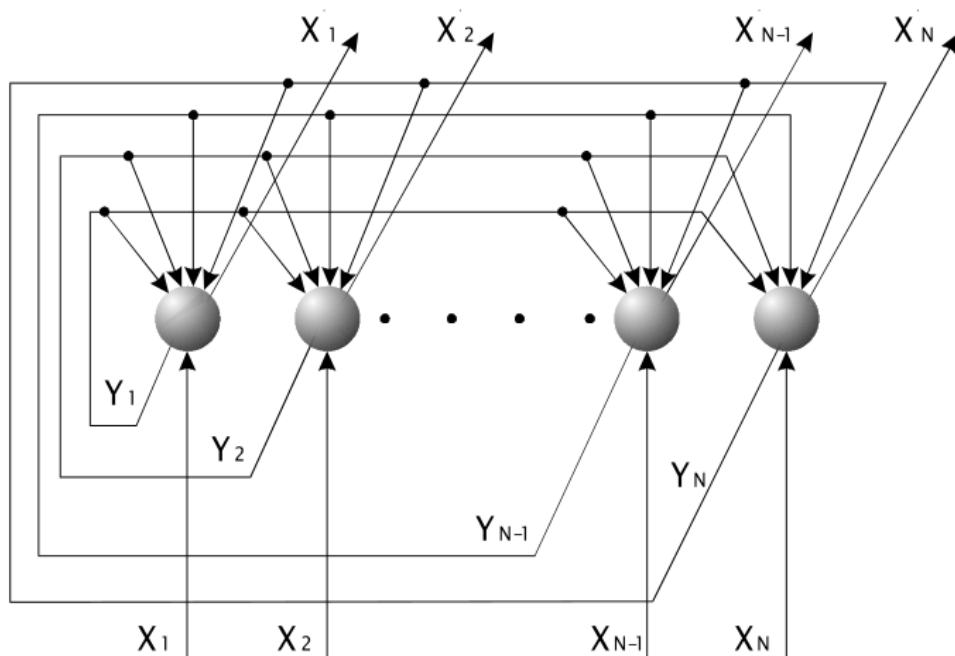


Рисунок 2.9 – Сеть Хопфилда

Нейроны сети имеют жесткие пороговые функции. Итерации сети завершаются после того, как выходные сигналы нейронов перестают меняться. Передаточная функция сети Хопфилда изображена на Рисунке 2.10.

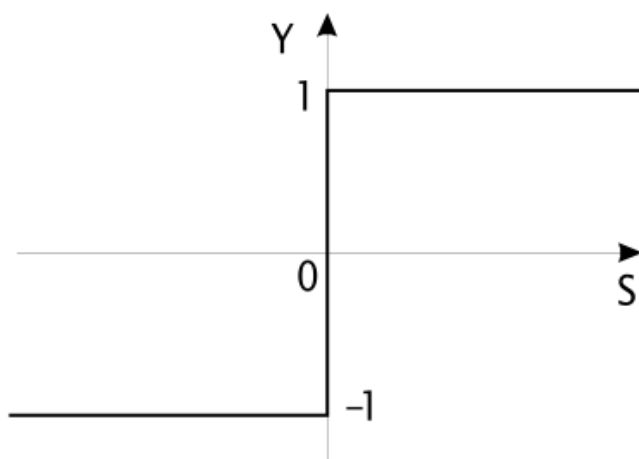


Рисунок 2.10 – Передаточная функция

Емкость сети ограничена количеством нейронов, из которых она состоит, то есть сеть, содержащая N нейронов, может запомнить не более чем $M = 0,15 \cdot N$ образов. При этом запоминаемые образы не должны быть полностью коррелированы, мерой корреляции при этом может служить формула (2.12):

$$K = \sum_{j,k}^M \sum_{i=1}^N x_i^j x_i^k, j \neq k, \quad (2.12)$$

где N – размерность входного сигнала,

M – количество векторов-образцов,

x_i^j – i -й элемент j -го вектора-столбца.

Формирование синаптических весов сети реализуется по формуле:

$$w_{i,j} = \begin{cases} \sum_{s=1}^M x_i^s x_j^s, & i \neq j \\ 0, & i = j \end{cases}, \quad (2.13)$$
$$1 \leq i, j \leq N$$

где $w_{i,j}$ – i -й синаптический вес j -го нейрона.

Функционирование сети производится в соответствии с формулой:

$$y_j(0) = x_j, 1 \leq j \leq N$$
$$y_j(t+1) = f\left(\sum_{i=1}^N w_{i,j} y_i(t)\right), 1 \leq j \leq N. \quad (2.14)$$

где y_j – выход j -го нейрона.

Для данной формулы, верно, то, что функционирование сети прекращается в случае выполнения условия:

$$y_j(T) = y_j(T-1). \quad (2.15)$$

Таким образом, суммарный выходной сигнал сети будет равен:

$$\hat{x}_j = y_j(T). \quad (2.16)$$

В данной сети одним из основных параметров будет изменение энергетической функции сети, в данном случае ее падение:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{i,j} x_i x_j. \quad (2.17)$$

Одним из основных недостатков сети является ее малая емкость, полезное значение которой также уменьшается в два раза вследствие того, что наряду с запомненными образами в сети хранятся и их негативы. Возможность применения сети в задачах распознавания образов также ограничено тем, что размерность ее

									Лист
Изм.	Лист	№ докум.	Подпись	Дата					

13.04.02.2017.256.00 ПЗ

входных сигналов должна совпадать с размерностью выходных. Также недостатком сети можно считать квадратичный рост числа синапсов в зависимости от увеличения размерности входного сигнала.

Для увеличения емкости сети и увеличения ее качества в задаче распознавания образов используют мультипликативные нейроны. Сети, состоящие из таких нейронов, называются сетями высших порядков.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

2.6 Самоорганизующиеся карты Кохонена

Задача категоризации объектов по их общим признакам является трудной, но необходимой задачей. Самоорганизующиеся карты, созданные Кохоненом в промежутке между 1979 и 1982 годами, представляют собой наиболее приемлемую структуру нейронной сети, находящуюся между сетями Хопфилда и сетью Адалина.

Функционирование самоорганизующихся карт Кохонена подчинено одной главной цели: создание двухмерной карты характеристик входного изображения, причем сохранение в этой карте не только самих характеристик, но также и последовательность их поступления на входы сети. Другими словами, алгоритм Кохонена дает возможность строить нейронную сеть для разделения векторов входных сигналов на подгруппы. Входные сигналы - вектора действительных чисел - последовательно предъявляются сети. Желаемые выходные сигналы не определяются. После того, как было предъявлено достаточное число входных векторов, синаптические веса сети определяют кластеры. Кроме того, веса организуются так, что топологически близкие узлы чувствительны к похожим внешним воздействиям (входным сигналам). Одной из основных характеристик карт Кохонена является то, что они могут обучаться без учителя, самоорганизуя структуру поступающих сигналов, легко взаимодействуя при этом с другими типами сетей в задачах категоризации и классификации. Они позволяют нам построить набор особенностей входных векторов, в котором сконцентрирована интересующая нас в рамках данной проблемы информация, используя в этом случае нелинейный метод, через который карта и создается. Нелинейность метода объясняется присутствием в комплексе с входным слоем адаптивного слоя, который со временем (в течение обучения) меняет свою структуру, преобразуясь в пространственно заданную карту особенностей входных векторов. Другими словами, положение элементарной единицы в слое Кохонена является показателем очередной особенности входного сигнала со статистической точки зрения, то есть определенной через анализ плотности распределения различных характеристик. Количество активизированных единиц в слое Кохонена напрямую

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

связано с количеством сгруппированных кластеров характеристик входных векторов.

Таким образом, самоорганизация, при обучении без учителя, представляет собой сжатие данных в двухмерную карту без потери информационности входного сигнала, что в принципе делает эту модель сети более способной к выживанию с биологической точки зрения, по сравнению с другими моделями. Так как в человеческом мозге нейроны располагаются так, что некоторые внешние физические воздействия вызывают ответную реакцию нейронов из определенной области мозга. Например, в той части мозга, которая отвечает за восприятие звуковых сигналов, нейроны группируются в соответствии с частотами входного сигнала, на которых они резонируют. Хотя строение мозга в значительной степени предопределяется генетически, отдельные структуры мозга формируются в процессе самоорганизации. Алгоритм Кохонена в некоторой степени напоминает процессы, происходящие в мозге. С точки зрения вычислительных процессов карты Кохонена могут рассматриваться как нелинейное обобщение результатов анализа основных составляющих входного сигнала. В этом контексте они демонстрируют себя как наиболее приемлемая модель нейронной сети применительно к задачам от распознавания изображений до адаптивного управления роботами.

Сеть состоит из M нейронов, образующих прямоугольную решетку на плоскости (Рисунок 2.11). Элементы входных сигналов подаются на входы всех нейронов сети. В процессе работы алгоритма настраиваются синаптические веса нейронов. Для реализации алгоритма необходимо определить меру соседства нейронов (меру близости).

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

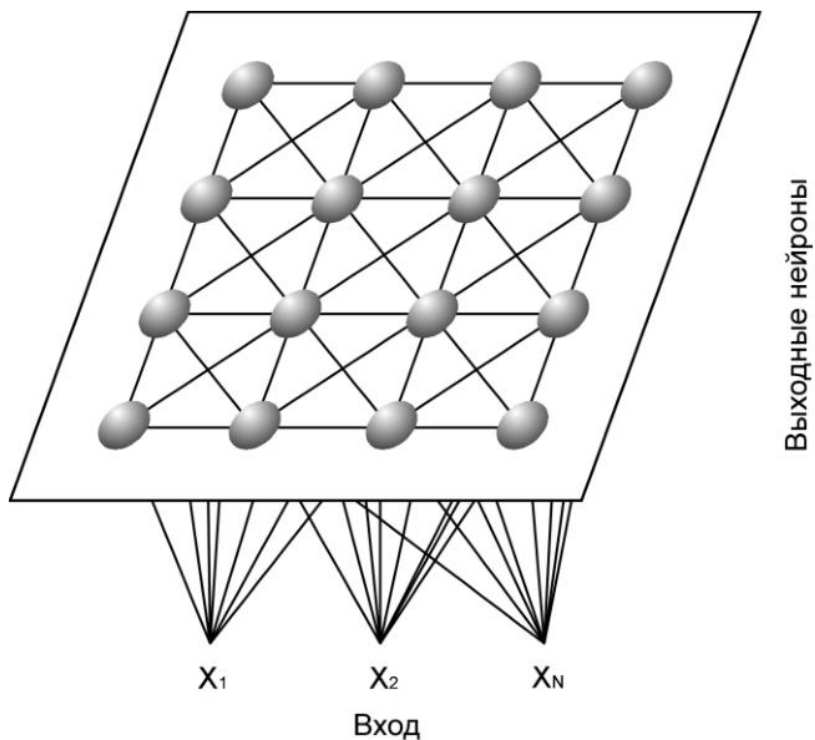


Рисунок 2.11 – Нейронная сеть

На Рисунке 2.12 показаны зоны топологического соседства нейронов на карте признаков в различные моменты времени. $NE_j(t)$ – множество нейронов, которые считаются соседями нейрона j в момент времени t . Зоны соседства уменьшаются с течением времени.

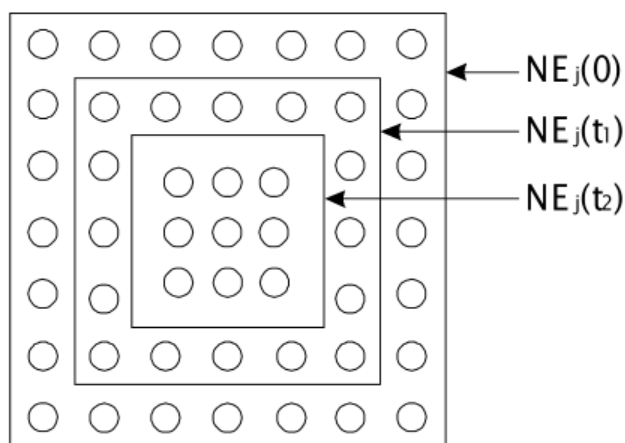


Рисунок 2.12 – Зоны топологического соседства

Приведем алгоритм Кохонена формирования карт признаков входных векторов:

1. *Инициализация сети.*

Весовым коэффициентам сети присваиваются малые случайные значения. Общее число синаптических весов равно $(M \cdot N)$. Начальная зона соседства на карте признаков для момента времени $t=0$ показана на рис.2.12.

2. *Поддача на сеть нового входного сигнала.*

3. *Вычисление расстояния Кохонена между входными векторами и нейронами.*

Расстояние d_j от входного сигнала до j -го нейрона определяется по формуле (2.18):

$$d_j = \sum_{i=1}^N (x_i(t) - w_{i,j}(t))^2, \quad (2.18)$$

где x_i – i -й элемент входного сигнала в момент времени t ,

$w_{i,j}(t)$ – вес связи от i -го элемента к нейрону j в момент времени t .

4. *Выбор нейрона с наименьшим расстоянием.*

Выбирается нейрон j^* , для которого расстояние d_j наименьшее.

5. *Настройка весов нейрона j^* и его соседей.*

Производится подстройка весов для нейрона j^* и всех нейронов из его зоны соседства NE. Новые значения весов получаются по формуле (2.19):

$$w_{i,j}(t+1) = w_{i,j}(t) + r(t) (x_i(t) - w_{i,j}(t)), \quad (2.19)$$

где $r(t)$ – шаг обучения, положительное число меньше единицы, которое уменьшается с течением времени.

6. *Возвращение к шагу 2.*

Основным достоинством самонастраивающихся карт Кохонена является их способность функционировать в условиях помех на входном сигнале, так как число классов является фиксированным, настройка весов производится медленно и заканчивается после окончания процесса обучения. Однако неприятной

стороной в данном вопросе является то, что число кластеров должно быть заранее известным.

Самоорганизующиеся карты Кохонена могут быть использованы для проектирования многомерных данных, аппроксимации плотности и кластеризации. Эта сеть успешно применялась для распознавания речи, обработки изображений, в робототехнике и в задачах управления. Одним из наиболее известных способов определения нейрона с наименьшим расстоянием Кохонена является применение сети MAXNET, которая используется совместно с самонастраивающимися картами.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

2.7 Генетические алгоритмы обучения НС

Начало теории генетических алгоритмов (ГА) положил Дж. Холланд в первом издании своей знаменитой книги в 1975 году. Правда генетическими они стали называться позднее, а в 1975 году Холланд называл их репродуктивными планами (*reproductive plan*) и рассматривал прежде всего как алгоритмы адаптации. Почти два десятилетия исследований ушли на доказательство грани могущества теории ГА в области глобальной оптимизации. Но не стоит забывать и о неплохих результатах применения ГА в теории адаптации и обучения.

ГА основываются на синтетической теории эволюции, говорящей о механизмах наследования признаков в популяциях животных. Кроме того ГА опирается на гипотезу селекции: вероятность того, что в потомстве будет высокая приспособленность, тем выше чем выше приспособленность у особи-родителя. Зачастую особи с низкой приспособленностью либо не участвуют в создании нового поколения, либо исключаются из популяции. Поэтому следует очевидный вывод: родителями становятся только особи с лучшим фенотипом, а не случайные особи из популяции, неудачные решения отбрасываются из задачи.

Стоит определить те характеристики, которые ГА берет из биологии:

- идею о преимущественном размножении наиболее приспособленных особей;
- способы передачи генетической информации в чередующихся поколениях (генетические операторы);
 - понятийный аппарат;
 - идею коллективного поиска экстремума при помощи популяции особей;
 - способы представления генетической информации.

Рассмотрим два парадигма теории ГА : представление генетической информации и генетические операторы, более подробно.

Первая проблема построения алгоритмов генетической эволюции - это кодировка информации, содержащейся в модели НС. Коды называют "хромосомами". Для фиксированной топологии (архитектуры) НС эта информация полностью содержится в значениях синаптических весов (W) и

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

смещений (В). Вектор (W,B) рассматривается как хромосома. Возможны более сложные способы кодирования информации. Для простоты изложения приведем пример бинарного кодирования, используемый при моделировании эволюции гаплоидных популяций. Будем считать, что каждая переменная x_i кодируется определенным фрагментом хромосомы, состоящим из фиксированного количества генов (Рисунок 2.13). Все локусы хромосом диаллельны- то есть в любой позиции фрагмента может стоять как ноль, так и единица. Рядом стоящие фрагменты не отделяют друг от друга какими-либо маркерами, тем не менее, при декодировании хромосомы в вектор переменных на протяжении всего моделируемого периода эволюции используется одна и та же маска картирования.

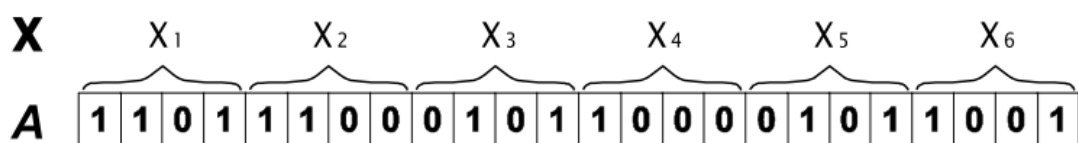


Рисунок 2.13 – Зоны топологического соседства

Хотя мы постоянно говорим о декодировании, на самом деле прямая операция, понимаемая как операция кодирования вектора переменных X в хромосому A , в ГА не применяется. Хромосомы образуются случайным способом, путем последовательного комплектации разрядов, сразу в бинарном виде, и всякие будущие изменения в популяции влияют сначала генетический уровень, а только потом производится анализ последствий в изменение этого фенотипа. В общем, для декодирования генетической информации из двоичной системы в десятичную систему подходит абсолютно любой двоично-десятичный код, но зачастую пользуются кодом Грея. От кода Грея совершаем переход к двоично-десятичному коду, а от него – к целым натуральным числам.

Одним из главных итогов изучения ГА на настоящий момент стало понимание прекрасной взаимной комплиментации триады генетических операторов (ГО) «кроссовер – мутация – инверсия». Воздействуя с некоторой вероятностью на генотипы родительских особей, каждый из них, с одной стороны, обеспечивает передачу потомству жизненно важных признаков, а с другой – поддерживает на протяжении эволюционно значимого периода

достаточно высокий уровень изменчивости. Выщепление в потомстве новых, отличных от родительских, фенотипических признаков открывает для популяции дополнительные возможности для адаптации, то есть способствует сохранению ею поисковой способности.

В рассмотренном на Рисунке 2.14 примере оператор мутации, подобно точечным мутациям в природе, трактуется как замена состояния гена на противоположное состояние. Становится очевидным, что разница между потомком и родителем зависит от разряда произошедшей в гене мутации. Инверсия ведет к изменению порядка следования фрагментов хромосом у потомка в отличие от хромосом родителей. В свою очередь кроссовер открытый Холландом при моделировании эволюции популяций [20], ведет к тому, что хромосома потомка содержит части хромосом обоих родителей. Т.е. из-за кроссоверов популяции обмениваются генами и поиск носит коллективный характер.

На Рисунке 2.14 у всех трех операторов место приложения, помеченное значком ∇, выбирается случайно.

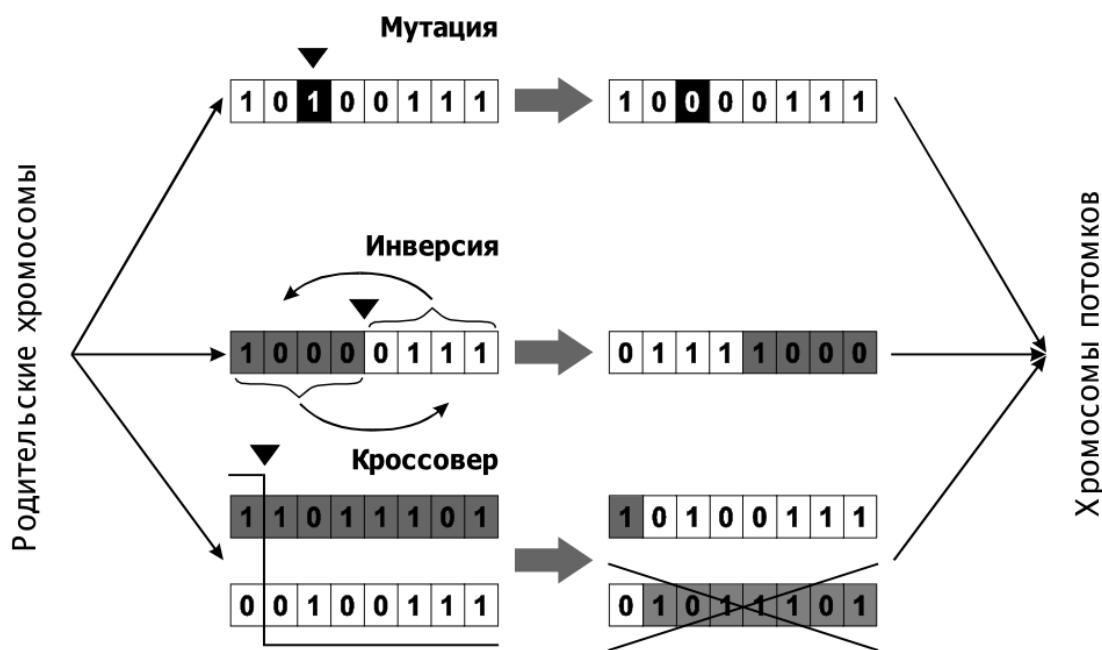


Рисунок 2.14 – Пример

Иногда, говоря о триаде ГО, подчеркивают способность кроссовера и инверсии к глобальному поиску, в то время как мутацию отождествляют со

средствами локальной настройки решения, отводя ей фоновую роль. Такой распределение ролей представляется спорным, так как мутация может породить потомка далеко за пределами локального экстремума, в котором находится родитель, с другой стороны, кроссовер, проведенный над гометами родителей, расположенных в общем экстремуме, наверняка породит потомков в этом же экстремуме. Важно другое – ни кроссовер, ни мутация не опираются в процессе генерирования потомка на знание локального рельефа поверхности целевой функции. В этом смысле их можно считать глобальными.

Исходя их теории НС, генетический алгоритм можно использовать для настройки весовых коэффициентов исследуемой сети. Т.е. производить обучение нейронной сети с помощью ГА. Преимуществом данного вида обучения является, например, то, что алгоритмы очень эффективны в поиске глобальных минимумов адаптивных рельефов, так как в них исследуются большие области допустимых значений параметров НС. (Градиентные алгоритмы дают возможность находить только локальные минимумы.) Другая причина того, что ГА не застревают в локальных минимумах - случайные мутации, которые аналогичны температурным флуктуациям метода имитации отжига. В качестве другого преимущества можно выделить, что ГА дают возможность оперировать дискретными значениями параметров нейронных сетей. Это упрощает разработку цифровых аппаратных реализаций сетей. При обучении на компьютере НС, не ориентированных на аппаратную реализацию, возможность использования дискретных значений параметров в некоторых случаях может приводить к сокращению общего времени обучения.

В рассмотренном на Рисунке 2.15 примере ГА используется для настройки численных параметров нейроконтроллера, когда алгоритм обратного распространения не пригоден из-за того, что требуются данные об ошибке на эффекторном слое НС. Для получения которой в схеме обучения потребовалось бы предусмотреть обращенный нейроэмулятор объекта, роль которого сводилась бы к преобразованию ошибки на выходе объекта в выходной сигнал на выходе нейроконтроллера, что существенно усложнило бы процедуру обучения.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

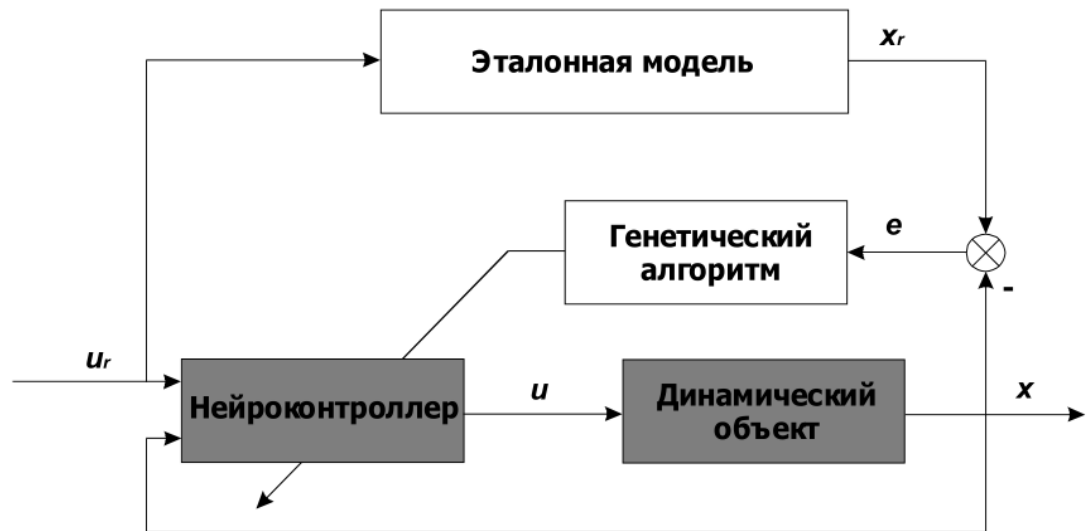


Рисунок 2.15 – Пример

Генетический же алгоритм настраивает параметры нейроконтроллера не по ошибке в управлении u , а по ошибке в выходе объекта, сравнивая его с выходом эталонной модели. Последняя воплощает в себе наши представления о желаемом поведении объекта, то есть играет приблизительно ту же роль, что и библиотека тренировочных шаблонов в процедуре обучения.

Еще одним преимуществом применения ГА является достаточно высокая скорость обучения, хотя скорость сходимости градиентных алгоритмов в среднем выше, чем генетических. А среди очевидных недостатков можно выделить сложность ГА для понимания и программной реализации. В рамках "генетического" подхода в последнее время разработаны многочисленные алгоритмы обучения НС, различающиеся способами представления данных нейронной сети в "хромосомах", стратегиями репродукции, мутаций, отбора.

3 МОДЕЛИРОВАНИЕ И АНАЛИЗ РЕЖИМОВ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ ПРИ ПОМОЩИ НЕЙРОННЫХ СЕТЕЙ

Многие задачи в электроэнергетике основываются на традиционных математических моделях, использующих узловые уравнения в разных формах представления [1,2] и системах координат. Упомянутые математические модели в большинстве случаев сводятся к решению систем уравнений больших порядков в окрестности режимных параметров широко распространенными методами нулевого, первого и, реже, второго порядков. Матричные системы, представляющие модели режимов электроэнергетических систем, могут быть линейными или четкими, но в большинстве случаев нелинейными. Если модель линейная, то решение ее численными методами получается точным. В данной задаче рассматривается линеаризованная модель в окрестности вектора номинальных напряжений. Соответственно, такие модели имеют приближенное решение с некоторой заданной погрешностью. Более того, такие модели перед решением исследуются на сходимость итерационного процесса.

Для анализа условий сходимости можно исследовать нормы матриц коэффициентов, которые определяют достаточные условия сходимости численных методов.

Если рассмотреть пример, приведенной ниже системы уравнений,

$$[\underline{Y}][\underline{U}] = \begin{bmatrix} 1,0 & 10,0 \\ 10,0 & 101,0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 11,1 \\ 111,0 \end{bmatrix}, \quad (3.1)$$

то она имеет решение $u_1 = 11,0$ и $u_2 = 0,0$. Если для системы (3.1) незначительно изменить правую часть

$$\begin{bmatrix} 1,0 & 10,0 \\ 10,0 & 101,0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 11,1 \\ 111,0 \end{bmatrix}, \quad (3.2)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

то составляющие вектора решения принимают соответствующие значения $u_1 = 1,0$ и $u_2 = 1,0$. Для приведенных примеров нормы матриц коэффициентов будут иметь следующие значения:

$$\|Y\|_1 = \max_i \sum_j |y_{ij}| = \max_i(10, 0, 101, 0) = 101, 0, \quad (3.3)$$

$$\|Y\|_2 = \max_j \sum_i |y_{ij}| = \max_j(10, 0, 101, 0) = 101, 0, \quad (3.4)$$

$$\|Y\|_3 = \sqrt{1,0^2 + 10,0^2 + 10,0^2 + 101,0^2} = 101,99. \quad (3.5)$$

Для сходимости итерационного процесса [3-6] нормы должны быть $\|Y\| < 1$.

Окончательная роль в ответе на поставленный вопрос принадлежит исследованию собственных значений матрицы коэффициентов.

Обусловленность матрицы может оцениваться числом обусловленности

$$\rho = \frac{|\lambda_1|}{|\lambda_2|}, \quad (3.6)$$

показывающим отношение модулей максимального и минимального собственных чисел. Из теории [3,6] известно, чем ближе число ρ к единице, тем лучше обусловлена матрица коэффициентов в выражениях (3.1, 3.2). Для матрицы коэффициентов системы уравнений (3.1, 3.2) собственные числа равны $\lambda_1 = 101,9$ и $\lambda_2 = 0,0098$ и число обусловленности будет соответственно $\rho = 10400$, что подтверждает неустойчивость сходимости численного процесса нахождения решения (3.1, 3.2).

Для обеспечения сходимости численного процесса, если матрица коэффициентов в уравнении (3.1) имеет неположительную определенность [4,5], следует перейти к положительно-определенной матрице, применяя операцию первой трансформацию Гаусса [6]. Учитывая данное действие, перепишем матричное уравнение (3.1) в комплексных координатах и получим следующее выражение

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

$$[\underline{Y}]^T [\underline{Y}][\underline{U}] = [\underline{Y}]^T [\underline{J}]$$

или

$$[\tilde{\underline{Y}}][\underline{U}] = [\tilde{\underline{J}}], \quad (3.7)$$

где $[\tilde{\underline{Y}}] = [\underline{Y}]^T [\underline{Y}]$, $[\tilde{\underline{J}}] = [\underline{Y}]^T [\underline{J}]$, T — символ транспонирования.

Для решения матричной системы уравнений (3.7), применим нетрадиционный подход [7], используя технологию элементов искусственного интеллекта. В данной работе в качестве технологии решения систем уравнений используется нейросетевой подход, который непрерывно развивается во многих областях науки и практики. Нейросетевая технология в постановке и решении задачи будет использовать следующую структуру.

Сенсорный слой или вход сети будет принимать вектор начальных приближений переменных, которые могут быть равны любому случайному числу, но в обозначенной задаче равны номинальному напряжению сети. Данный входной вектор для нейронной сети является искомым параметром. На сенсорный слой дополнительно подаются сигналы и от правой части, которые моделируют режимные параметра электрической сети(3.1). Элементы векторов входных переменных, с использованием которых происходит обучение сети, подвергают действию препроцессирования под пределы существования или изменения активационной функции.

Синаптические связи, охватывающие сенсорный слой с нейронами скрытого слоя, моделируют столбцы не транспонированной матрица коэффициентов системы (3.1). Каждый элемент сенсорного слоя связан перекрестным образом с нейронами скрытого слоя, который является единственным для рассматриваемой задачи в случае положительно определенной матрицы коэффициентов. Это свойство матрицы исследуется до разработки архитектуры нейронной сети. Количество элементов скрытого слоя в предварительном варианте было установлено равным размеру вектора

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

неизвестных. Четко обоснованные рекомендации для моделирования числа скрытых слоев и количества в них нейронов на сегодняшний день отсутствуют. Сенсоры, представляющие правую часть (3.1), с помощью синапсов также перекрестно связаны со всеми элементами скрытого слоя. В скрытом слое выполняются две операции. Первая операция связана с суммированием сигналов, передаваемых от сенсорного слоя, вторая - с обработкой суммарных сигналов входных векторов при помощи активационной функции, в качестве которой принята сигмоида [7].

Внутренний или скрытый слой связан перекрестно с выходным слоем, моделирующим $[y]^T$. Выходные данные активировались линейной функцией [7].

Алгоритм обучения нейронной сети

На Рисунке 3.1 показан пример многослойной сети (Multiple layer perceptron – MLP), состоящей из входного, промежуточного (или скрытого) и выходного слоев или кратко, входа, скрытых слоев и выхода. Вход состоит из ячеек, которые не являются нейронами, а лишь выполняют функции источников сигналов разной природы. Для каждой ячейки задан идентификатор $e_i, i = \overline{1, n}$, n – число ячеек входов. В рассматриваемом примере нейронная сеть имеет две входные ячейки с номерами e_1, e_2 . Две входные ячейки в зависимости от исследуемой задачи являются источниками соответствующих сигналов pow_1 и pow_2 ($pow(i), i = \overline{1, u}$), u – количество источников на входе нейронной сети. Две ячейки скрытого слоя, которые представляют нейронный слой, соответственно имеют номера ne_j^k ($ne(j)^k, j = \overline{1, m}$), m – число нейронов в скрытом слое, $k = \overline{1, sl}$, sl – количество скрытых слоев персептрона. Скрытый слой нейронов в сети связан с ячейками выхода. Эти ячейки также не являются нейронами и идентифицированы номерами o_i ($o(i), i = \overline{1, b}$), b – число ячеек на выходе нейронной сети.

Обозначения соединений между ячейками, в общем, задано идентификаторами вида w , верхний и нижний индексы которых зависят от их

										Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ					

топологии в сети. Связи отображают веса или весовые коэффициенты между ячейками нейронной сети, которые отражают влияние одной ячейки на другую.

Входные ячейки задают для нейронной сети исходные параметры. Кроме того имеются еще входы смещения, которые передают соответствующие сигналы как в ячейки скрытых слоев, так и в ячейки выхода сети через свои весовые коэффициенты. Ячейки скрытых слоев и выходные представляют собой функции. Результаты суммирования данных или сигналов в слоях обрабатываются функциями активации (или сжатия), например, сигмоидой и выдаются с выхода ячеек в другой скрытый слой или в ячейки выхода сети.

Функция сигмоиды показана на Рисунке 3.2.

Далее детализируем взаимодействия всех слоев нейронной сети.

На Рисунке 3.3 показан фрагмент нейронной сети, представленной на Рисунок 3.1. Этот фрагмент охватывает вход сети и связанный с ним промежуточный или скрытый слой ячеек, которые выполняют функции нейронов.

Входные ячейки только задают исходные величины сигналов для всей сети. Природа сигналов может быть различной в зависимости от исследуемой задачи. Сигналы от входа передаются к первому слою посредством связей, характеризующихся весовыми коэффициентами.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

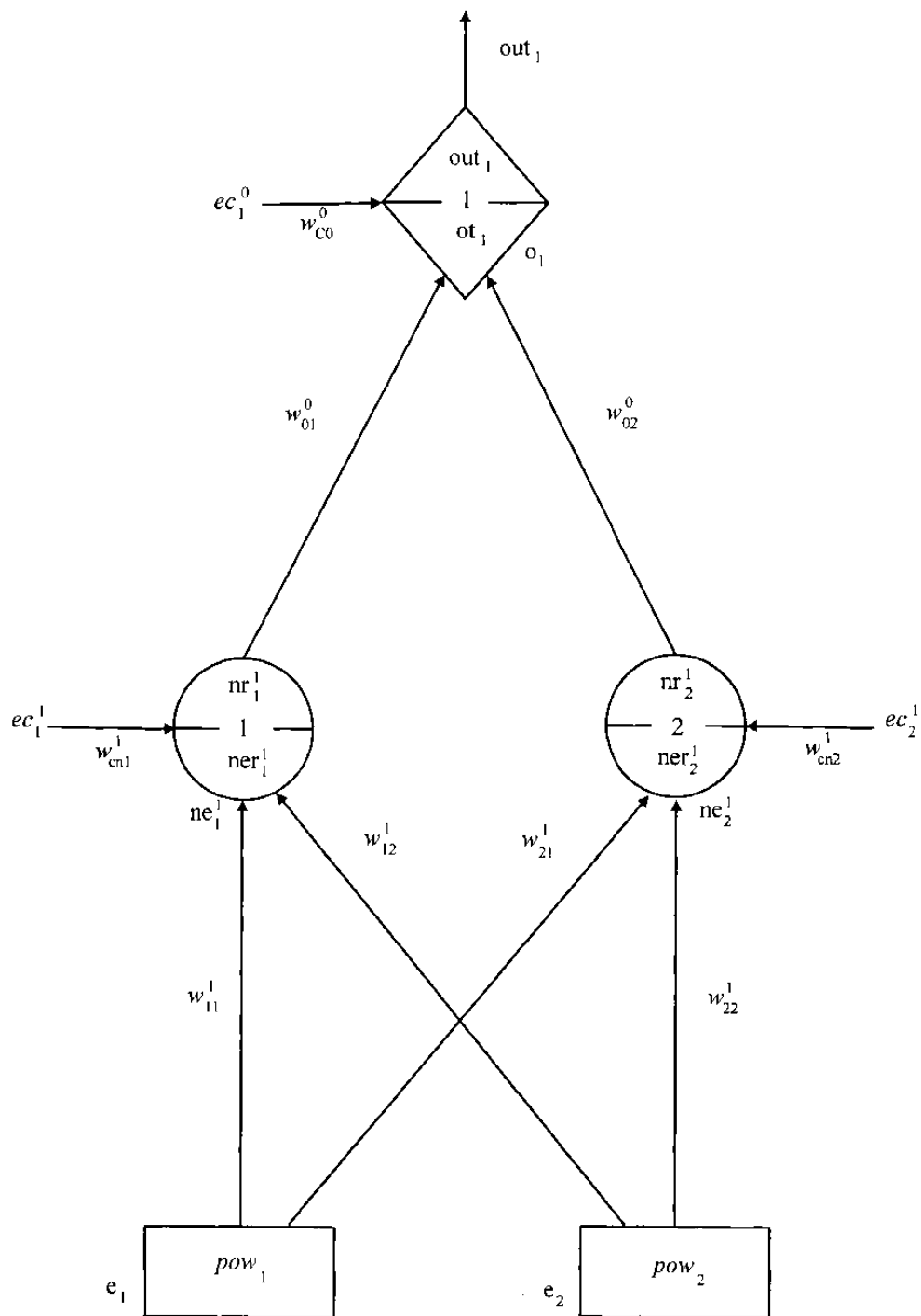


Рисунок 3.1 – Многослойный персептрон

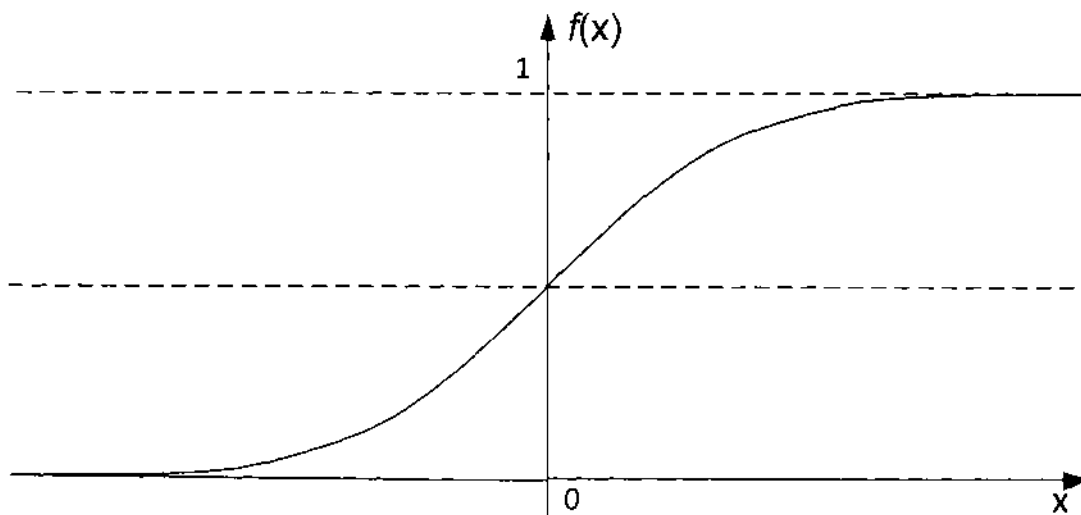


Рисунок 3.2 – Сигмоидальная функция

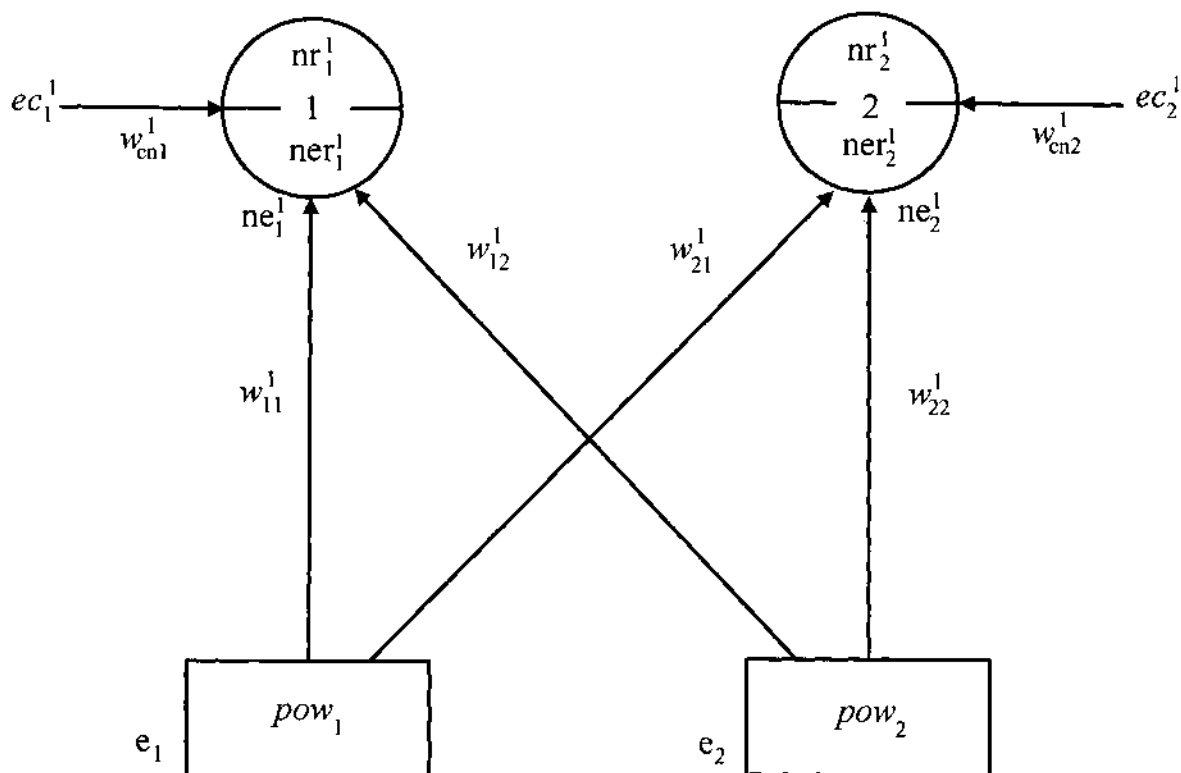


Рисунок 3.3 – Входной и скрытый слой нейронной сети

Ячейки (нейроны) слоя, связанного с входом, получает комбинированные сигналы, которые моделируются следующим уравнением, записанным в общем

ВИДЕ:

$$ner_j^k = \sum_{j=1}^n pow_i w_{ji}^k, \quad (3.8)$$

где w_{ji}^k – весовой коэффициент связи нейрона и ячейки сигнала входа сети.

К ячейкам скрытого слоя через связи с весовыми коэффициентами w_{cm}^k , ($l = \overline{1, cm}$, cm – количество сигналов смещения скрытого слоя) присоединяются корректирующие источники сигналов (могут отсутствовать), которые называются сигналами смещения.

На рис. 1 иллюстрируется взаимодействие скрытого слоя с выходом нейронной сети.

Ячейка o_i , ($i = \overline{1, \lambda}$), λ – количество выходов нейронной сети. Данная ячейка получает результат работы скрытого слоя после активации комбинированных сигналов ячеек ne_j^k по связям w_{ob}^k , ($b = \overline{1, \eta}$), η – количество связей выхода нейронной сети со скрытым слоем. Результат, полученный на выходе сети, как и сигналы на выходе скрытого слоя, обрабатываются активационной функцией вида сигмоиды

$$f(x) = \frac{1}{1 + e^{(-x)}}. \quad (3.9)$$

Данную функцию еще называют – передаточной функцией.

Кратко пояснив принципиальную схему нейронной сети, рассмотрим алгоритм ее работы, используя схемы на Рисунках 3.4 и 3.5, сопровождая численными расчетами.

									Лист
Изм.	Лист	№ докум.	Подпись	Дата					

13.04.02.2017.256.00 ПЗ

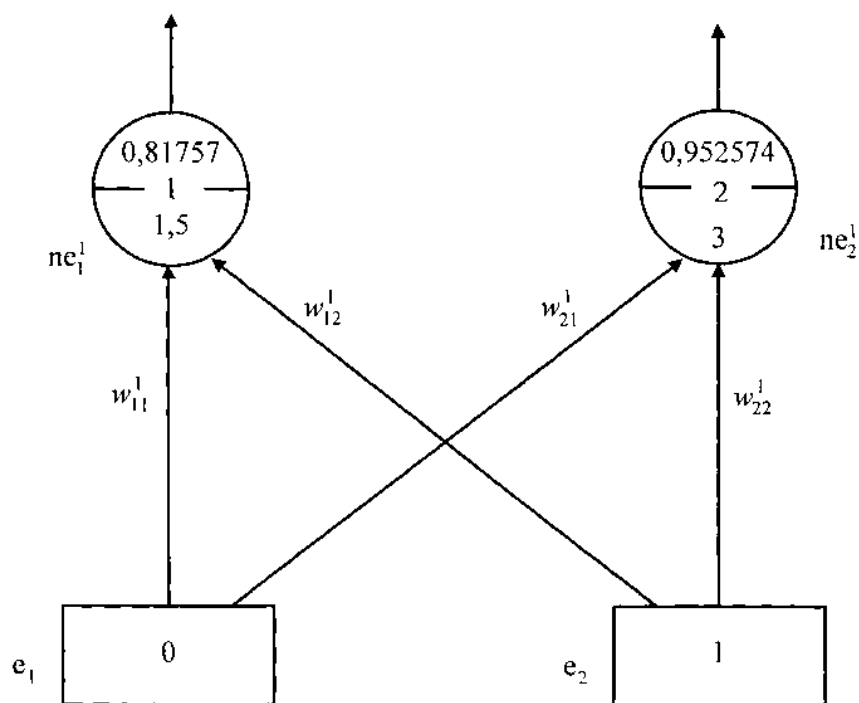


Рисунок 3.4 – Взаимодействие входных сигналов со скрытым слоем нейронов

Заметим, что на Рисунке 3.1 показано прохождение сигналов от входа к выходу или в прямом направлении. Подобные сети называются радиально-базисными, которые имеют широкое применение в разных научно-практических задачах как аппроксиматоры, но требующих значительного времени для их обучения с целью получения положительного результата.

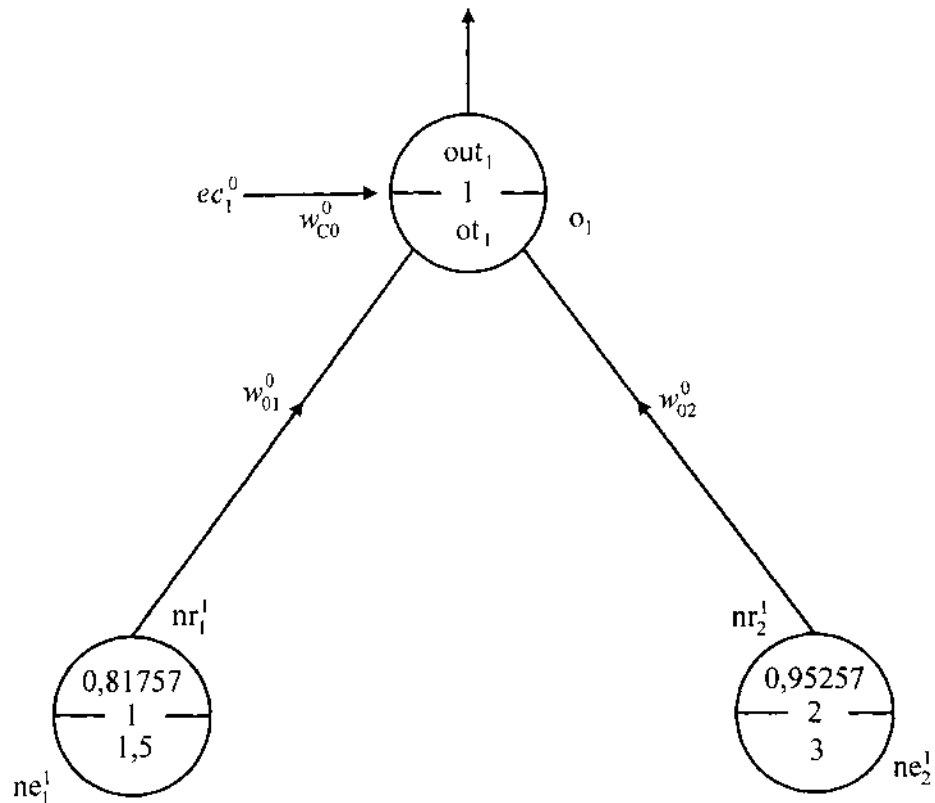


Рисунок 3.5 – Реакция скрытого слоя с выходом сети

Поэтому для оптимизации времени получения результата разработан итерационный способ, состоящий из развития вычислительного процесса вперед, то есть от входа сети к выходу, а при достижении сигнала выходной ячейки с последующей активацией его. Проверяется правильность работы нейронной сети. Если будет ошибка в работе сети при решении задачи, то можно применить алгоритм обратного распространения для минимизации ошибки совместно с коэффициентом обучения сети. Заметим, что каждая итерация или эпоха состоит из двух шагов (этапов): первый шаг заключается в продвижении от входа к выходу нейронной сети (в прямом направлении) и движении в противоположном направлении от выхода к входу с выполнением соответствующих операций, изложенных ниже в процедуре текущего шага алгоритма.

Рассмотрим пример функционирования радиально-базисной нейронной сети и ее обучения. Цель обучения состоит в подборе таких значений весов для

всех слоев сети, чтобы при заданном входном векторе сигналов (режимных параметров) $[pow]$ на выходе получить значения сигналов (сигнала) out_i , которые с требуемой точностью будут совпадать с ожидаемыми значениями исследуемых данных.

Алгоритм первой (текущей) эпохи начинается с процедуры развития вперед:

–вычисляется реакция для входа скрытого слоя (Рисунок 3.4) с использованием следующего матричного выражения

$$\begin{bmatrix} ner_1^1 \\ ner_2^1 \end{bmatrix} = \begin{bmatrix} ner_{c1}^1 \\ ner_{c2}^1 \end{bmatrix} + \begin{bmatrix} w_{11}^1 & w_{12}^1 \\ w_{21}^1 & w_{22}^1 \end{bmatrix} \cdot \begin{bmatrix} pow_1 \\ pow_2 \end{bmatrix}, \quad (3.10)$$

где $ner_{c1}^1 = w_{cn1}^1 \cdot ec_1^1, ner_{c2}^1 = w_{cn2}^1 \cdot ec_2^1$ – влияние источников смещения ec_{ci} ($i=1,2$) через соответствующие связи w_{cj}^k на нейроны j скрытого k слоя – ner_j^k .

Результаты, с учетом подстановки исходных данных, будут

$$\begin{bmatrix} ner_1^1 \\ ner_2^1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 \\ 1 \cdot (-1) \end{bmatrix} + \begin{bmatrix} 1 \cdot 0 & 0,5 \cdot 0 \\ (-1) \cdot 0 & 2 \cdot 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1,5 \\ 3,0 \end{bmatrix}. \quad (3.11)$$

–выполняется действие передачи предыдущих результатов входа ячейки на выход скрытого слоя после воздействия функции активации $f(ner_j^k)$

$$\begin{bmatrix} f(ner_1^1) \\ f(ner_2^1) \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + \exp(-ner_1^1)} \\ \frac{1}{1 + \exp(-ner_2^1)} \end{bmatrix} = \begin{bmatrix} 0,81757 \\ 0,952574 \end{bmatrix}. \quad (3.12)$$

–осуществляется действие перемещения сигналов с выхода скрытого слоя на вход ячейки выхода сети, представленное фрагментом Рисунка 3.1 с помощью иллюстрации на Рисунках 3.6.

–вход ячейки o_1 выхода нейронной сети получает результат работы двух ячеек скрытого слоя nr_1^1 и nr_2^1 по весам связи w_{o1}^0 и w_{o2}^0 соответственно:

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

$$ot_1 = nr_1^1 + nr_2^1. \quad (3.13)$$

–результат ot_1 корректируется сигналом смещения ec_1^0 и полученная величина

$$ot_1' = ot_1 + ec_1^0 \quad (3.14)$$

–обрабатывается функцией активации на выходе нейронной сети по следующему выражению

$$[f(ot_1')] = \left[\frac{1}{1 + \exp(-ot_1')} \right], \quad (3.15)$$

приводящему к следующему результату

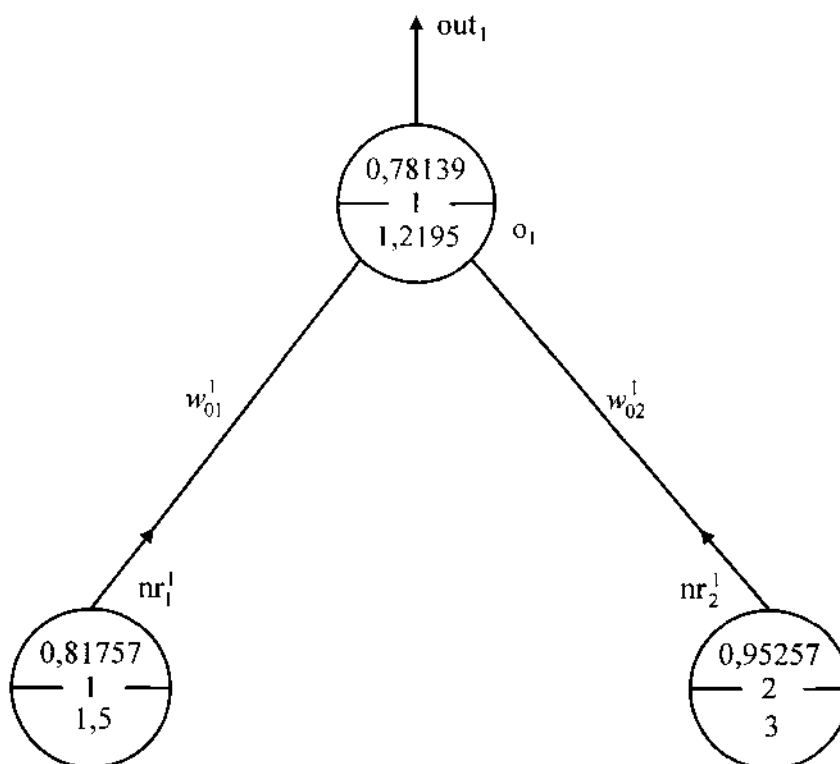


Рисунок 3.6 – Активация выхода сети

$$out_1 = [f(ot_1')] = 0,78139, \quad (3.16)$$

На этой операции заканчивается прямой проход первой или текущей эпохи

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

по нейронной сети, после чего вычисляется ошибка или реакция нейронной сети на исходные данные.

Правильная реакция нейронной сети на входной сигнал (или на вектор входного сигнала) является $O_{in} = 1$ (задано по условию задачи). Реальное значение выходного сигнала, рассчитанного сетью, которое обозначается идентификатором out_1 (Рисунок 3.6) равно 0,78139.

Ошибка работы сети, определяемая по следующему выражению

$$err = 0,5(O_{in} - out_1)^2, \quad (3.17)$$

составила 0,024.

Если результат не удовлетворяет поставленной задаче, то для уточнения или дальнейшего обучения нейронной сети применяют специальные приемы, например упомянутый выше, алгоритм обратного распространения ошибок. Функции данного алгоритма будут представлять обратный ход текущей эпохи.

Алгоритм обратного распространения (Backpropagation algorithm) – это самый на сегодняшний день, изученный и эффективный способ для обучения нейронных сетей. Из названия алгоритма следует, что ошибка распространяется от выхода (или в общем случае от выходного слоя) к входному слою, то есть в направлении, противоположном направлению прохождения сигнала при нормальном функционировании сети. Начиная с выходных ячеек, рассчитываются значения ошибок, далее продолжают их вычисления для всех скрытых слоев по соответствующим аналитическим выражениям. Затем проводятся действия уточнения весов между выходным и скрытым слоем, между скрытыми слоями (если сеть многослойная), скрытым слоем и входом с использованием коэффициента ρ , называемого коэффициентом обучения. Рекомендуется начинать тестирование сети с небольшого значения и далее постепенно его увеличивать.

Продолжение алгоритма будет складываться из следующих этапов:

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

– рассчитываются значения ошибок для ячеек выходного слоя или выходной ячейки по формуле

$$\delta_0 = (O_{in} - out_1)out_1(1 - out_1) = (1 - 0,78139)0,78139(1 - 0,78139) = 0,0373; \quad (3.18)$$

– определяются величины ошибок скрытого слоя(или слоев) с использованием выражения в общем виде

$$\delta_j^k = (w_{kj}^k \delta_0)nr_j^k (1 - nr_j^k), \quad (3.19)$$

на основании которого величины коррекций для первой и второй ячеек соответственно составят:

$$\begin{aligned} \delta_1 &= (w_{01}^0 \delta_0)nr_1^1 (1 - nr_1^1) = ((-1,0) \cdot 0,0373) \cdot 0,952574(1,0 - 0,952574) = -0,0016851; \\ \delta_2 &= (w_{02}^0 \delta_0)nr_2^1 (1 - nr_2^1) = (1,5 \cdot 0,0373) \cdot 0,81757(1,0 - 0,81757) = 0,0083449; \end{aligned} \quad (3.20)$$

– производится уточнение весовых коэффициентов между выходами (выходами) и скрытым слоем с учетом коэффициента обучения $\rho = 0,5$:

$$\begin{aligned} w_{01}^{0*} &= w_{01}^0 + \rho \delta_0 nr_1^1 = -1 + (0,5 \cdot 0,0373 \cdot 0,952574) = -0,9882; \\ w_{02}^{0*} &= w_{02}^0 + \rho \delta_0 nr_2^1 = 1,5 + (0,5 \cdot 0,0373 \cdot 0,81757) = 1,51525; \end{aligned} \quad (3.21)$$

–обновляются вес связи сигнала смещения для выходной ячейки

$$w_{c0}^{0*} = w_{c0}^0 + \rho \delta_0 ec_1^0 = 1 + (0,5 \cdot 0,0373 \cdot 1) = 1,01865; \quad (3.22)$$

из представленных выше двух пунктов следует, что для весовой связи коэффициент уменьшился. Смещение обновляется для изменения возбуждения, которое увеличилось. Далее рассчитываются веса связей к входным ячейкам;

– рассчитываются новые веса между скрытым слоем и входными ячейками по следующему общему выражению:

$$\begin{aligned} w_{ij}^* &= w_{ij} + \rho \delta_0 u_i, \\ w_{11}^{1*} &= w_{11} + \rho \delta_2 row_1 = 1,0 + (0,5 \cdot 0,0083449) \cdot 1,0 = 1,0, \\ w_{12}^{1*} &= w_{12} + \rho \delta_2 row_2 = 0,5 + (0,5 \cdot 0,0083449) \cdot 1,0 = 0,50417, \\ w_{22}^{1*} &= w_{22} + \rho \delta_1 row_2 = 2,0 + (0,5 \cdot (-0,016851)) \cdot 1,0 = 1,99916, \\ w_{21}^{1*} &= w_{21} + \rho \delta_1 row_1 = -1,0 + (0,5 \cdot (-0,016851)) \cdot 0 = -1,0. \end{aligned} \quad (3.23)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

–корректируются веса связей сигналов смещений скрытого слоя (слоев)

$$\begin{aligned} w_{cn1}^* &= w_{cn1}^1 \cdot (\rho \delta_2 ec_1^1) = 1,0 + (0,5 \cdot 0,0083449 \cdot 1,0) = 1,00417, \\ w_{cn2}^* &= w_{cn2}^1 \cdot (\rho \delta_1 ec_2^1) = 1,0 + (0,5 \cdot (-0,0016851) \cdot 1,0) = 0,99915. \end{aligned} \quad (3.24)$$

Обновление весов для заданных (выбранных) обучающих сигналов завершается.

Проверим, что алгоритм, представленный выше, действительно уменьшает ошибку на выходе сети, введя тот же обучающий сигнал ещё раз:

$$\begin{aligned} \begin{bmatrix} ner_1^1 \\ ner_2^1 \end{bmatrix} &= \begin{bmatrix} ner_{c1}^1 \\ ner_{c2}^1 \end{bmatrix} + \begin{bmatrix} w_{11}^1 & w_{12}^1 \\ w_{21}^1 & w_{22}^1 \end{bmatrix} \cdot \begin{bmatrix} pow_1 \\ pow_2 \end{bmatrix} = \\ &= \begin{bmatrix} 1,00417 \cdot 1,0 \\ 0,99915 \cdot 1,0 \end{bmatrix} + \begin{bmatrix} 1,0 & 0,50417 \\ -1,0 & 1,99916 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1,50834 \\ 2,999831 \end{bmatrix}, \end{aligned} \quad (3.25)$$

$$\begin{bmatrix} nr_1^1 \\ nr_2^1 \end{bmatrix} = \begin{bmatrix} f(ner_1^1) \\ f(ner_2^1) \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + \exp(-ner_1^1)} \\ \frac{1}{1 + \exp(-ner_2^1)} \end{bmatrix} = \begin{bmatrix} 0,81888 \\ 0,952497 \end{bmatrix}. \quad (3.26)$$

$$\begin{aligned} ot_1^1 &= ot_1 + ec_1^0 = w_{o1}^0 \cdot nr_1^1 + w_{o2}^0 \cdot nr_2^1 + w_{co}^0 \cdot ec_1^0 = \\ &= 1,51525 \cdot 0,81888 + (-0,9822) \cdot 0,952497 + 1,01865 \cdot 1 = 1,32379. \end{aligned} \quad (3.27)$$

$$out_1 = [f(ot_1^1)] = [f(1,32379)] = 0,7898. \quad (3.28)$$

$$err = 0,5(O_{ur} - out_1)^2 = 0,5(1,0 - 0,7898_1)^2 = 0,022. \quad (3.29)$$

Начальная ошибка была равна 0,023895. Текущая ошибка составляет 0,022, а это значит, что одна эпоха алгоритма обратного распределения позволила уменьшить среднюю ошибку на 0,001895.

Также были исследованы результаты работы сети с другими активационными функциями в режиме обучения. Результаты получены следующие: для функции $f(x) = th(x)$: на первой итерации ошибка составляла 0,037, на второй – 0,034, на сотой – $1,677 \cdot 10^{-3}$, на тысячной – $1,301 \cdot 10^{-4}$, на десятитысячной – $1,065 \cdot 10^{-5}$, на стотысячной – $9,061 \cdot 10^{-7}$; для функции

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

$f(x) = e^x / (1 + e^x)$: на первой итерации ошибка составляла 0,038, на второй – 0,034, на сотой – $1,763 \cdot 10^{-3}$, на тысячной – $1,333 \cdot 10^{-4}$, на десятитысячной – $1,075 \cdot 10^{-5}$, на стотысячной – $9,028 \cdot 10^{-7}$; для функции $f(x) = 1 / (1 + x^2)$: на первой итерации ошибка составляла 0,034, на второй – 0,033, на сотой – $3,152 \cdot 10^{-3}$, на тысячной – $1,608 \cdot 10^{-4}$, на десятитысячной – $1,075 \cdot 10^{-7}$, на стотысячной – $8,31 \cdot 10^{-7}$; для функции $f(x) = e^{-x^2}$: на первой итерации ошибка составляла 0,036, на второй – 0,035, на сотой – $3,137 \cdot 10^{-3}$, на тысячной – $1,588 \cdot 10^{-4}$, на десятитысячной – $1,065 \cdot 10^{-7}$, на стотысячной – $8,243 \cdot 10^{-7}$.

Ниже представлен алгоритм метода “Z” реализованный в среде Mathcad и модель нейронной сети для системы линейных алгебраических уравнений метода “Z”:

1. Задание погрешности расчета вектора узловых напряжений EU

2. Задание напряжений в базисном узле $UB = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

3. Задание начального приближения вектора искомых переменных (\underline{U}) – уменьшили в 100 раз

$$XU = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

4. Требуемое значение прогнозируемой переменной

$$ech = \begin{bmatrix} 3,83 \\ 7,39 \\ 7,05 \end{bmatrix} \rightarrow \Delta \underline{U} - \text{вектор превышения над базисным значением}$$

5. Коэффициент обучения

$$lnw = 0,5$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

6. Матрица коэффициентов системы уравнений Z – исполняет роль весовых коэффициентов, соединяющих сенсорный слой N сети с первым скрытым слоем нейронов

$$Y = \begin{bmatrix} 1,833 & -0,5 & -0,333 \\ -0,5 & 0,7 & -0,2 \\ -0,333 & -0,2 & 0,783 \end{bmatrix}$$

$$ZY = Y^{-1} = \begin{bmatrix} 0,855 & 0,77 & 0,56 \\ 0,77 & 2,23 & 0,895 \\ 0,56 & 0,895 & 1,73 \end{bmatrix}$$

7. Вектор узловых мощностей (сопряжённый)

$$S = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

8. Формирование корректируемой диагональной матрицы для узловых режимных параметров (напряжений)

$$DUKOR = \text{diag}(XU) \quad (3.30)$$

9. Формирование корректируемого вектора задающих узловых токов

$$JKOR = DUKOR * S * 100 \quad (3.31)$$

10. Рассчитывается вектор падений напряжений текущего приближения

$$DUKOR = ZY * JKOR \quad (3.32)$$

11. Определяется текущий вектор узловых напряжений

$$XUT = UB + DUKOR \quad (3.33)$$

12. Анализ погрешности расчета узловых напряжений

$$ABS |XU - XUT| < EU \quad (3.34)$$

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

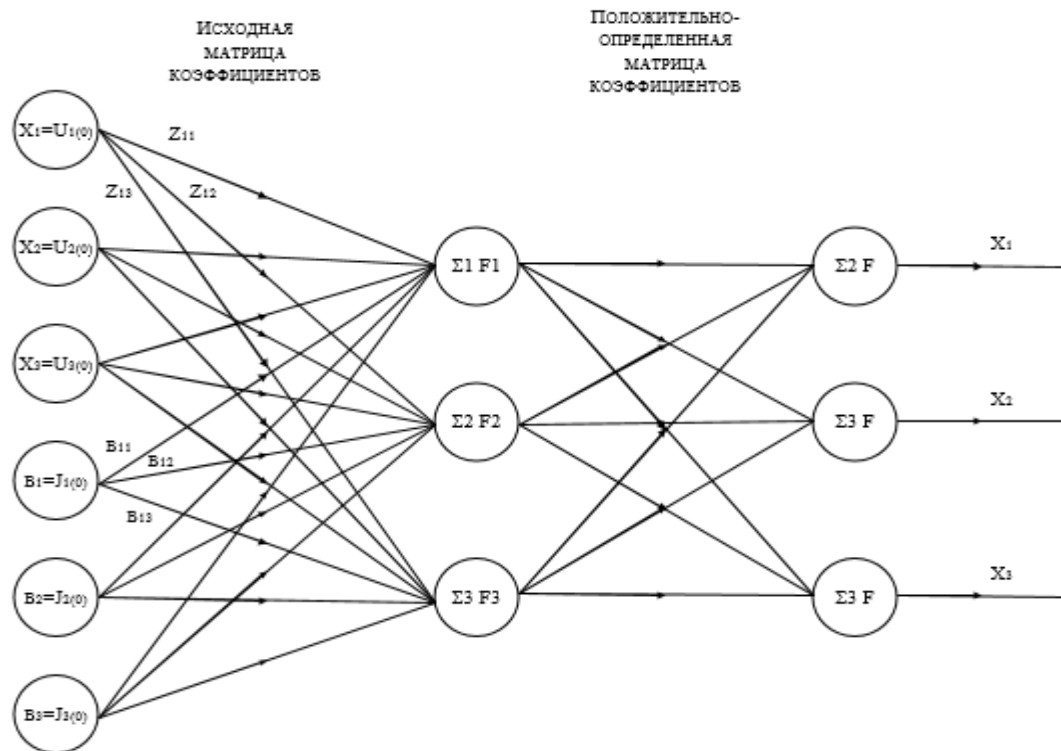


Рисунок 3.7 – Модель НС для СЛАУ – “Z”

Далее реализуем расчёт электроэнергетической сети 220 кВ в среде программирования Python 3.

Произведем расчёт электрической сети посредством программы NetWork, изменяя значения нагрузок в узлах 1 и 2. Примем за базисные значения напряжения – 220 кВ, мощность – 100 МВт, полная потеря мощности – 5 МВт. Результаты сведём в Таблицу 1.

Таблица 1 – Вводные данные

S, МВт	U1, кВ	U2, кВ	ΔS , МВт	S', о.е.	U1', о.е.	U2', о.е.	$\Delta S'$, о.е.
20,230	218,090	217,320	0,229	0,202	0,991	0,988	0,046
30,500	216,920	216,150	0,498	0,305	0,986	0,983	0,100
40,880	215,740	214,960	0,882	0,409	0,981	0,977	0,176
51,380	214,550	213,770	1,382	0,514	0,975	0,972	0,276
62,000	213,350	212,560	2,005	0,620	0,970	0,966	0,401
72,750	212,130	211,330	2,753	0,728	0,964	0,961	0,551
83,630	210,900	210,090	3,632	0,836	0,959	0,955	0,726
94,650	209,650	208,840	4,647	0,947	0,953	0,949	0,929

Далее приведен листинг программы выполненной на Python 3:

```
import numpy as np

import sys

class NN(object):

    def __init__(self, learning_rate=0.1):

        self.weights_0_1 = np.random.normal(0.0, 2 ** -0.5, (2, 3))

        self.weights_1_2 = np.random.normal(0.0, 1, (1, 2))

        self.sigmoid_mapper = np.vectorize(self.sigmoid)

        self.learning_rate = np.array([learning_rate])

    def sigmoid(self, x):

        return 1 / (1 + np.exp(-x))

    def predict(self, inputs):

        inputs_1 = np.dot(self.weights_0_1, inputs)

        outputs_1 = self.sigmoid_mapper(inputs_1)

        inputs_2 = np.dot(self.weights_1_2, outputs_1)

        outputs_2 = self.sigmoid_mapper(inputs_2)

        return outputs_2

    def train(self, inputs, expected_predict):

        inputs_1 = np.dot(self.weights_0_1, inputs)
```

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		


```

outputs_1 = self.sigmoid_mapper(inputs_1)

inputs_2 = np.dot(self.weights_1_2, outputs_1)
outputs_2 = self.sigmoid_mapper(inputs_2)
actual_predict = outputs_2[0]

error_layer_2 = np.array([actual_predict - expected_predict])
gradient_layer_2 = actual_predict * (1 - actual_predict)
weights_delta_layer_2 = error_layer_2 * gradient_layer_2
self.weights_1_2 -= (np.dot(weights_delta_layer_2, outputs_1.reshape(1,
len(outputs_1)))) * self.learning_rate

error_layer_1 = weights_delta_layer_2 * self.weights_1_2
gradient_layer_1 = outputs_1 * (1 - outputs_1)
weights_delta_layer_1 = error_layer_1 * gradient_layer_1
self.weights_0_1 -= np.dot(inputs.reshape(len(inputs), 1), weights_delta_layer_1).T
* self.learning_rate

def MSE(y, Y):
    return np.mean((y-Y)**2)

train = [
    ([0.202, 0.991, 0.988], 0.046),
    ([0.305, 0.986, 0.983], 0.100),
    ([0.409, 0.981, 0.977], 0.176),
    ([0.514, 0.975, 0.972], 0.276),

```

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

```

([0.620, 0.970, 0.966], 0.401),
([0.728, 0.964, 0.961], 0.551),
([0.836, 0.959, 0.955], 0.726),
([0.947, 0.953, 0.949], 0.929),
]

epochs = 8000
learning_rate = 0.07

network = NN(learning_rate=learning_rate)

for e in range(epochs):
    inputs_ = []
    correct_predictions = []
    for input_stat, correct_predict in train:
        network.train(np.array(input_stat), correct_predict)
        inputs_.append(np.array(input_stat))
        correct_predictions.append(np.array(correct_predict))

    train_loss = MSE(network.predict(np.array(inputs_).T), np.array(correct_predictions))
    sys.stdout.write("rProgress: { }, Training loss: { }".format(str(100 *
e/float(epochs))[:4], str(train_loss)[:5]))

    for input_stat, correct_predict in train:
        print("For input: { } the prediction is: { }, expected: { }".format(
            str(input_stat),

```

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

```

str(network.predict(np.array(input_stat)) > .5),
str(correct_predict == 1)))

```

for input_stat, correct_predict in train:

```

print("For input: {} the prediction is: {}, expected: {}".format(
    str(input_stat),
    str(network.predict(np.array(input_stat))),
    str(correct_predict == 1)))

```

```
print (network.weights_0_1)
```

```
print (network.weights_1_2)
```

На выходе получаем значения весов нейронной сети и результаты расчёта потерь сети 220 кВ. Результаты сведём в Таблицу 2.

Таблица 2 – Выходные данные

$\Delta S'$, о.е.	$\Delta S'$, о.е. Python 3
0,046	0,071
0,100	0,104
0,176	0,160
0,276	0,254
0,401	0,402
0,551	0,585
0,726	0,746
0,929	0,851

Результаты получены со средней квадратичной ошибкой равной 0,001. Погрешности обусловлены малым количеством обучающих данных, не оптимальным подбором эпох и коэффициентов обучения.

Завершая раздел, отметим, что нейронные сети без сомнения займут важное место, как в прикладном направлении, так и в интеллектуализации процессов управления в электроэнергетике.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

В ходе данной работы было проведено исследование применения элементов искусственного интеллекта для задачи установившегося режима.

В работе было отражено решение следующих задач:

- Обзор существующих методов решения установившегося режима применяемых в существующих пакетах программ, производящих подобные расчеты;

- Выполнение сравнительного анализа следующих методов и прогнозных моделей:

- Нейросетевая модель (ИНС) с использованием эволюционных алгоритмов;

- Нейросетевая модель (ИНС) с использованием алгоритма обратного распространения ошибки;

- Нейросетевая модель (ИНС) прямого распространения;

- Рекуррентная нейронная сеть.

- Построение системы расчёта режима в средах Mathcad и Python по результатам исследований;

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Идельчик В.И. Электрические системы и сети. – М.: Энергоатомиздат, 1989.
2. Электрические системы. Электрические сети. – М.: Высшая школа, 1998.
3. Стренг Г. Линейная алгебра и её применение. – М.: Мир, 1980.
4. Калиткин Н.Н. Численные методы. – М.: Наука, 1978.
5. Райс Дж. Матричные вычисления и математическое обеспечение. – М.: Мир, 1984.
6. Гантмахер Ф.Р. Теория матриц. – М.: Наука, 1966.
7. Хайкин С. Нейронные сети: полный курс. М.: ИД Вильямс, 2006.
8. СТО ЮУрГУ 04-2008 Курсовое и дипломное проектирование. Общие требования к содержанию и оформлению. – Челябинск: ЮУрГУ, 2008.
9. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский // М.: Горячая линия - Телеком, 2004.
10. Хебб Д. Организация поведения: нейропсихологическая теория. – М.: Наука, 2002.
11. Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга. – М.: Мир, 1965.
12. Кохонен Т. Самоорганизующиеся карты. М.: «БИНОМ. Лаборатория знаний», 2008.
13. BaseGroupLabs – Практическое применение нейросетей в задачах классификации
14. http://fetmag.mrsu.ru/2009-3/pdf/Forecasting_electricity_consumption.pdf

										Лист
Изм.	Лист	№ докум.	Подпись	Дата	13.04.02.2017.256.00 ПЗ					

15. Gallant S. L., Neural Network Learning and Expert Systems. - Cambridge, Mass.: MIT Press, 1994.

16. Шумков Е.А. Система поддержки принятия решений предприятия на основе нейросетевых технологий. Краснодар: КубГТУ, 2004.

17. Божич В.И., Лебедев О.Б., Шницер Ю.Л. Разработка генетического алгоритма обучения нейронных сетей // Известия ТРТУ. Тематический выпуск. 2001.

18. Чипига А.Ф., Воронкин Р.А. Обучение искусственных нейронных сетей путем совместного использования методов локальной оптимизации и генетических алгоритмов // Известия ТРТУ, 2001.

					13.04.02.2017.256.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		