# ВВЕДЕНИЕ

Нейронные сети начали применяться повсеместно совсем недавно, но уже смогли достичь огромных результатов в разных сферах. Их применяют в задачах:

1. Распознание образов и классификация. Может использоваться для картинок, звуков, символов текста и т.д.
2. Принятие решений и управление. Анализирует поступающие ситуации и принимает решения. Похожа на классификацию.
3. Кластеризация. Разбиение входных образов на классы, при том, что неизвестно количество входных сигналов и признаков класса.
4. Прогнозирование. Позволяет из последовательности извлечь скрытые зависимости и с помощью этого прогнозировать дальнейшее поведение последовательности.
5. Аппроксимация. Аппроксимирует непрерывные функции.
6. Сжатие данных и ассоциативная память. За счёт выявления значимых частей образа позволяет сжимать данные и после восстанавливать их в исходный образ.

# 1. ТРЕБОВАНИЯ К КУРСОВОМУ ПРОЕКТУ

Целью курсового проекта является исследование применения нейронных сетей в задачах аппроксимации и прогнозирования функций.

В задании дана функция, заданная в виде системы обыкновенных дифференциальных уравнений первого порядка.

***Вариант 26:***

*1,5ẋ + x= 6zexp(-z2)*

*ẏ=x - z* (1.1)

*ż=y – z/10*

В курсовом проекте необходимо рассмотреть несколько вариантов нейронных сетей, которые могут быть применены для аппроксимации и прогнозирования функции (1.1).

Всю работу можно разбить на ряд этапов:

1. Генерирование обучающей выборки. На первом этапе необходимо сгенерировать обучающую выборку, значения из которой будут использоваться для обучения, а также для проверки обобщающей способности сетей. По эталонным точкам требуется построить график функции (1.1) в трехмерном пространстве и его проекции на плоскости *XY, XZ* и *YZ*; графики зависимостей *X(t), Y(t)* и *Z(t).*

2. Проектирование нейронных сетей. На этапе проектирования нейронных сетей необходимо описать архитектуру и алгоритм обучения каждой из используемых сетей.

3. Разработка программного обеспечения. После этапа проектирования нейронных сетей необходимо разработать программное обеспечение, реализующее обучение этих сетей, а также отображение результатов их обучения.

4. Тестирование нейронных сетей. С помощью разработанного программного обеспечения требуется провести тестирование сетей, подбирая при этом их параметры таким образом, чтобы обеспечивалась минимальная погрешность аппроксимации и прогнозирования функции (1.1). Необходимо провести анализ нейронных сетей с точки зрения точности выполнения ими поставленной задачи.

# 2. ГЕНЕРИРОВАНИЕ ОБУЧАЮЩЕЙ ВЫБОРКИ

Ввиду невозможности аналитического расчета системы (1.1) для ее решения необходимо использовать приближенные численные методы, которые позволяют для заданной последовательности *t0, t1, t2, …tn* и начальных условий *X(t0), Y(t0), Z(t0)*, не определяя точного решения системы:

*1,5ẋ + x= 6zexp(-z2)*

*ẏ=x - z*

*ż=y – z/10*

Для численного решения данной системы дифференциальных уравнений будем применять алгоритм Рунге-Кутты.

Метод Рунге-Кутта заключается в рекуррентном применении следующих формул:

 \textbf{y}_{n+1} = \textbf{y}_n + {h \over 6}(\textbf{k}_1 + 2\textbf{k}_2 + 2\textbf{k}_3 + \textbf{k}_4) 

 \textbf{k}_1 = \textbf{f} \left( x_n, \textbf{y}_n \right), 

 \textbf{k}_2 = \textbf{f} \left( x_n + {h \over 2}, \textbf{y}_n + {h \over 2} \textbf{k}_1 \right), 

 \textbf{k}_3 = \textbf{f} \left( x_n + {h \over 2}, \textbf{y}_n + {h \over 2} \textbf{k}_2 \right), 

 \textbf{k}_4 = \textbf{f} \left( x_n + h, \textbf{y}_n + h\ \textbf{k}_3 \right). 

Этот метод имеет четвёртый порядок точности. Это значит, что ошибка на одном шаге имеет порядок O(h^5), а суммарная ошибка на конечном интервале интегрирования имеет порядок O(h^4) .

В результате решения системы (1.1) методом Рунге-Кутта были получены эталонные значения *X(t), Y(t)* и *Z(t)* с шагом дискретизации *h = 0,01*. Для обучения нейронных сетей будем использовать 2000 полученных значений, а на оставшихся значениях будет исследоваться обобщающая способность сетей.

Первые десять значений, полученные методом Рунге-Кутта, представлены в таблице 2.1

Обучающая выборка функции Таблица 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| t | X(t) | Y(t) | Z(t) |
| 0.1 | 0,203431 | 0,111377 | 0,127765 |
| 0.2 | 0,240748 | 0,120314 | 0,138001 |
| 0.3 | 0,279668 | 0,131964 | 0,149156 |
| 0.4 | 0,320415 | 0,146395 | 0,161494 |
| 0.5 | 0,363277 | 0,163676 | 0,175285 |
| 0.6 | 0,408594 | 0,18388 | 0,1908 |
| 0.7 | 0,456745 | 0,207088 | 0,208319 |
| 0.8 | 0,508132 | 0,233392 | 0,228125 |
| 0.9 | 0,563158 | 0,262895 | 0,250508 |
| 1.0 | 0,622197 | 0,295712 | 0,275768 |

По рассчитанным точкам необходимо построить следующие графики функции:

Рис. 2.1. Проекция функции на плоскость XY

Рис. 2.2. Проекция функции на плоскость XZ

Рис. 2.3. Проекция функции на плоскость YZ

Рис. 2.4. Зависимость X(t)

Рис. 2.5. Зависимость Y(t)

Рис. 2.6. Зависимость Z(t)

Таким образом, обучающая выборка сформирована, можно переходить к следующему этапу – проектированию нейронных сетей и алгоритмов обучения.

# 3. ПРОЕКТИРОВАНИЕ НЕЙРОННЫХ СЕТЕЙ

Для решения поставленной задачи будем использовать многослойные нейронные сети, так как подобного рода сети наилучшим образом подходят для решения задач аппроксимации и прогнозирования функций.

В курсовом проекте необходимо рассмотреть следующие нейронные сети для аппроксимации и прогнозирования функции:

1. Многослойный персептрон с тремя нейронными элементами в распределительном и тремя нейронными элементами в выходном слое (рис. 3.1):

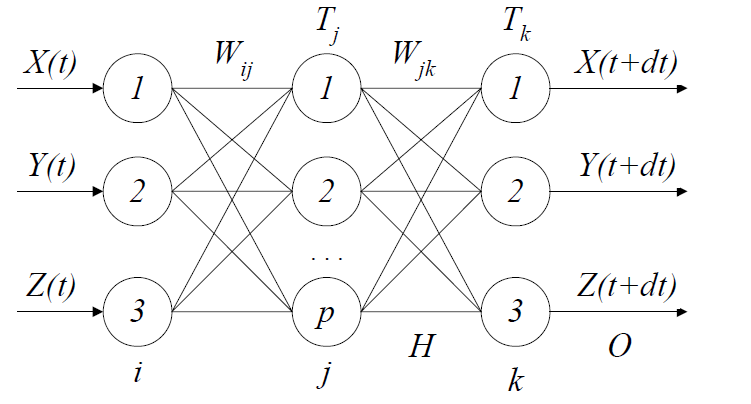


Рис. 3.1. Многослойный персептрон (3/*р*/3)

Число нейронов в скрытом слое *j* определяется экспериментально с целью достижения наилучших результатов аппроксимации и прогнозирования.

На вход сети подаются значения *X(t)*, *Y(t)* и *Z(t)*, а выходными значениями сети будут значения *X*, *Y* и *Z* в следующий момент времени – *X(t+Δt)*, *Y(t+Δt)* и *Z(t+Δt)*.

2. Многослойный персептрон с шестью нейронными элементами в распределительном и тремя нейронными элементами в выходном слое (рис. 3.2):

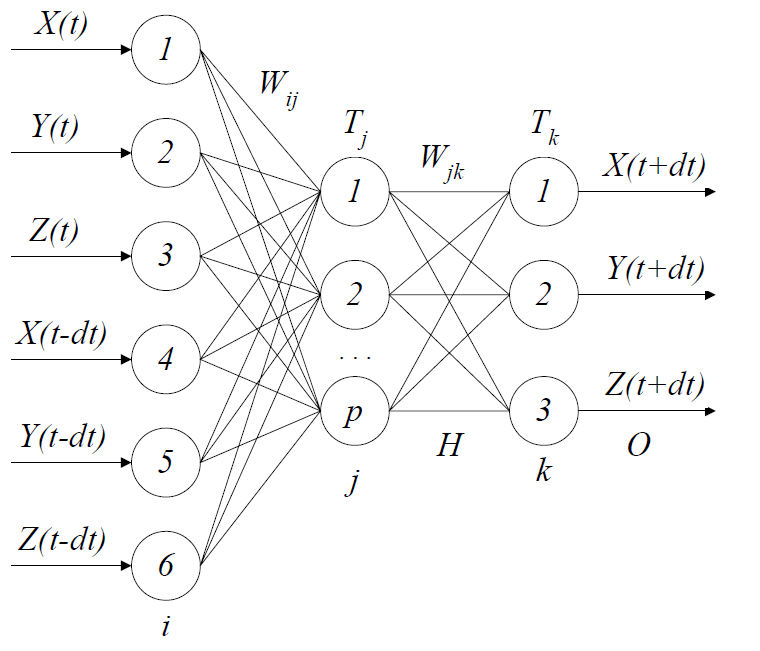


Рис. 3.2. Многослойный персептрон (6/*р*/3)

Здесь количество нейронных элементов в скрытом слое *j* также изначально неизвестно, оптимальное их количество определяется экспериментальным путем.

На вход сети помимо значений *X(t)*, *Y(t)* и *Z(t)* будем подавать значения

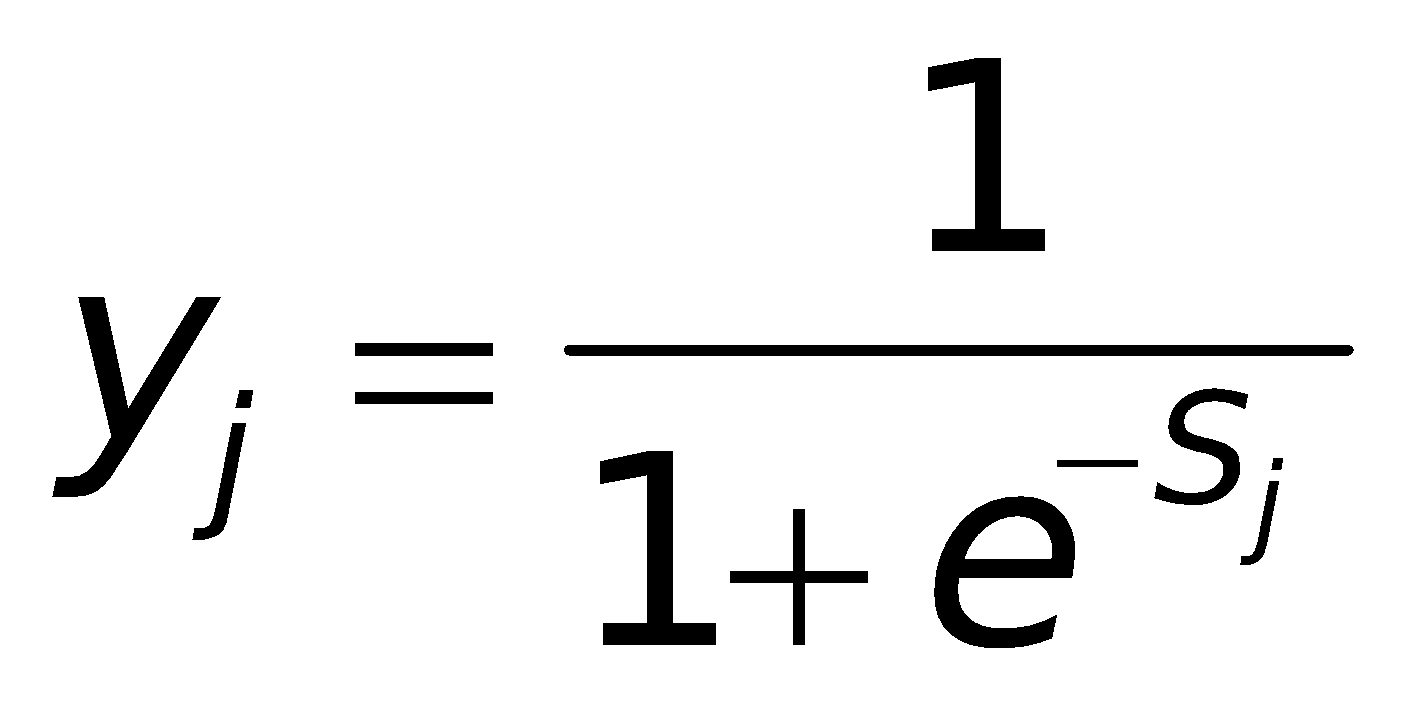
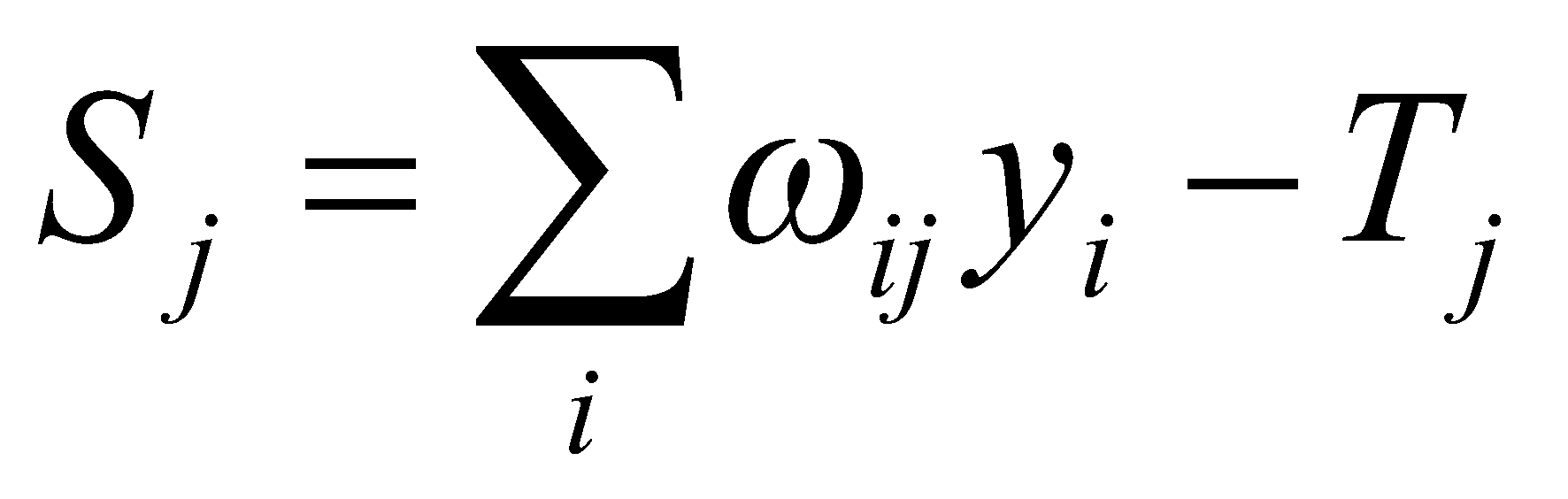
*X(t-Δt)*, *Y(t-Δt)* и *Z(t-Δt)*. Такая форма обучения называется обучением с запаздыванием. На выходе сети будем получать значения *X*, *Y* и *Z* в момент времени *t+Δt*: *X(t+Δt)*, *Y(t+Δt)* и *Z(t+Δt)*.

В курсовом проекте необходимо использовать две таких нейронных сети: для ряда *X(t)*, для ряда *Y(t)* и для ряда *Z(t).* Число нейронов во входном слое, а также размерность промежуточного слоя подбираются так, чтобы погрешность аппроксимации и прогнозирования была минимальной.

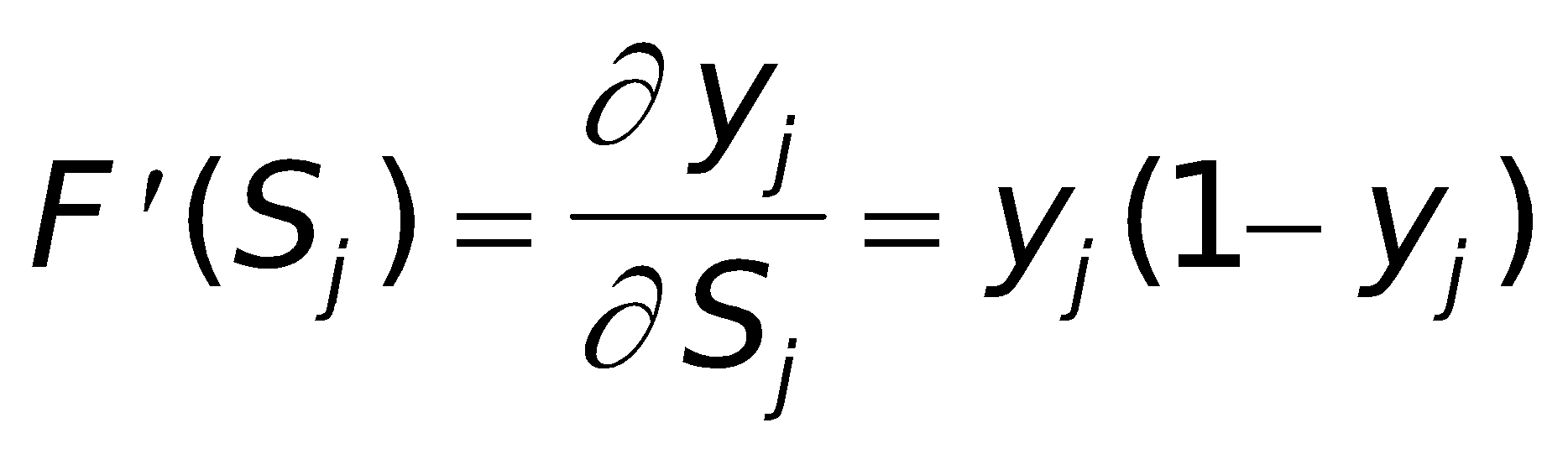
Для обучения таких сетей используется метод «скользящего окна». Суть метода «скользящего окна» состоит в том, что в каждый момент времени на вход сети подаются значения временного ряда *A(t+(1-n)Δt)..A(t)*, при этом полученное на выходе значение сравнивается со значением *A(t+Δt)*, где *n* – размерность «окна».

Для программной реализации использовалась сигмоидная функция активации.

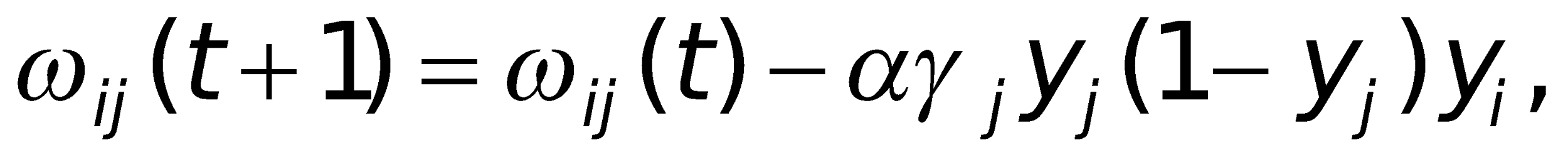
Выходное значение *j*-го нейронного элемента определяется следующим   
образом:

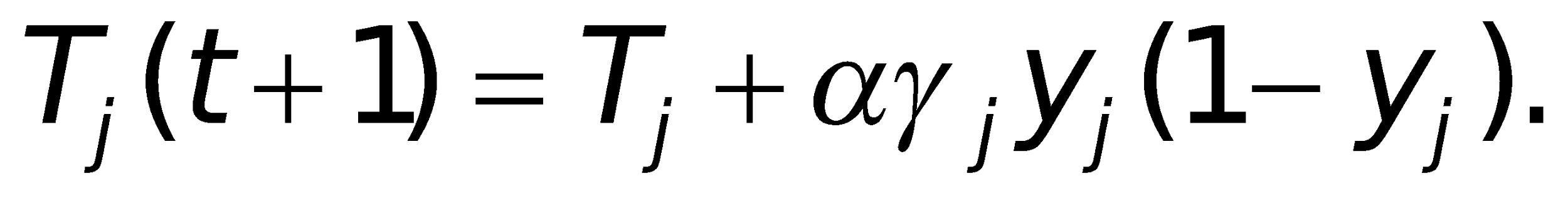
*, *

Тогда

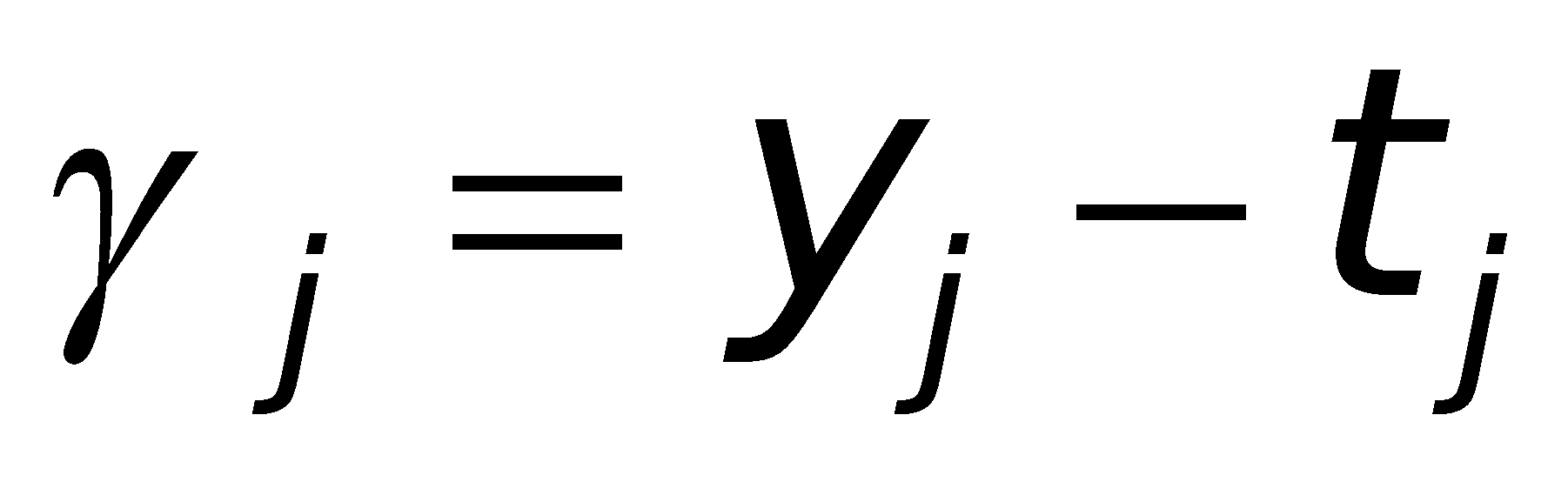
**

В результате обобщенное дельта правило для сигмоидной функции активации можно представить в следующем виде:

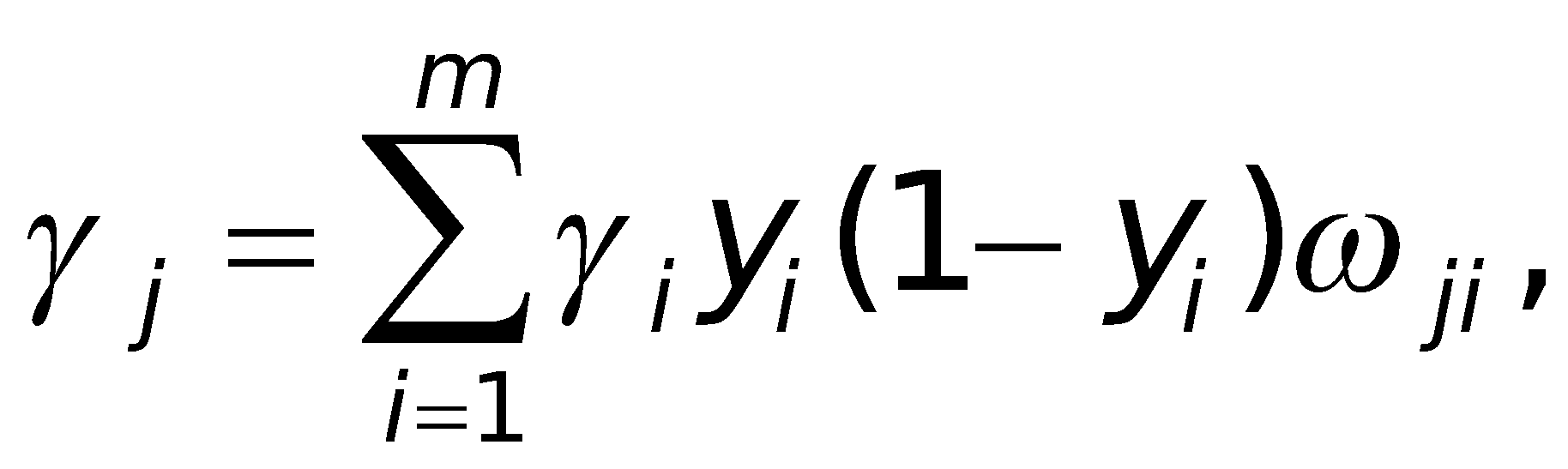
**

**

Ошибка для *j*-го нейрона выходного слоя определяется, как

*.*

Для *j*-го нейронного элемента скрытого слоя:



# 4. ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ ОБУЧЕНИЯ

Для обучения сети будем использовать алгоритм обратного распространения ошибки. Он является эффективным средством обучения нейронных сетей и представляет собой следующую последовательность шагов:

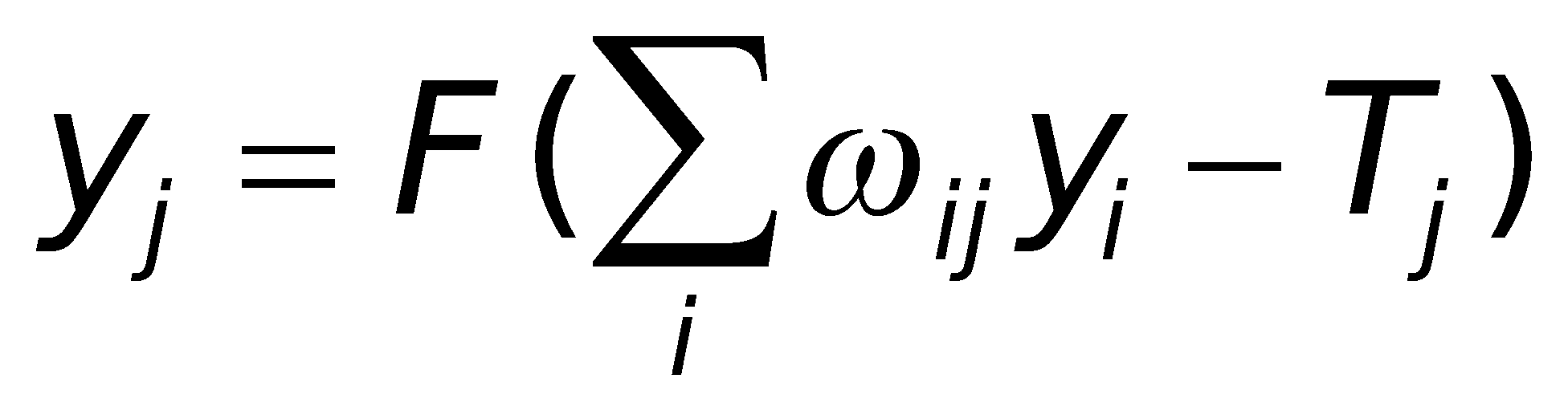
1. Задается шаг обучения α (0> α>1) и желаемая среднеквадратичная

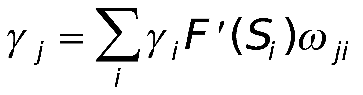
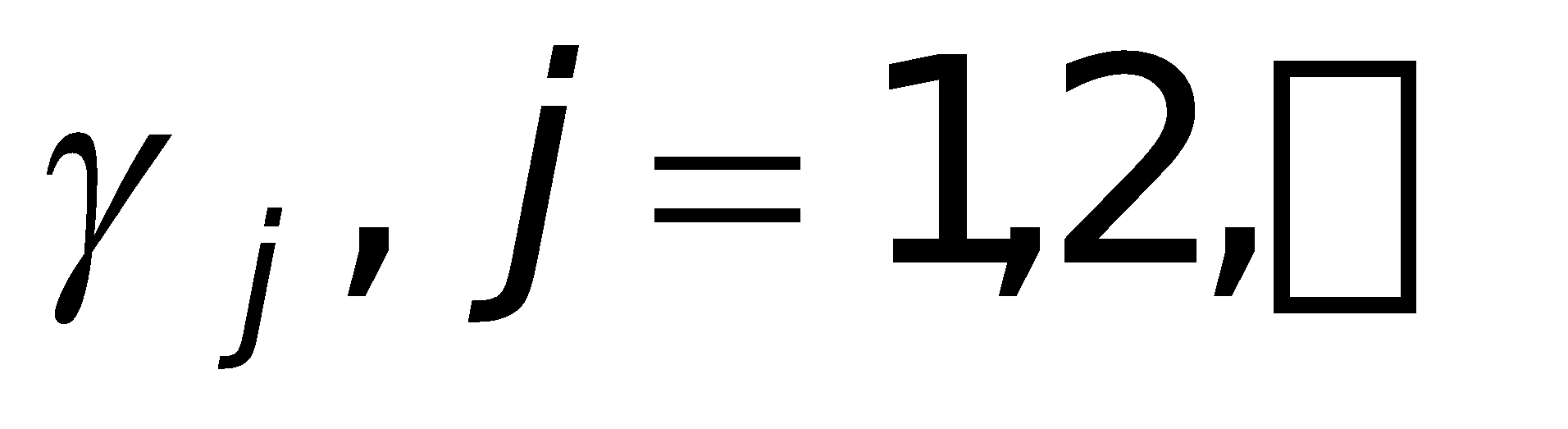
ошибка нейронной сети .

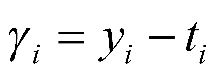
2. Случайным образом инициализируются весовые коэффициенты и пороговые значения нейронной сети.

3. Последовательно подаются образы из обучающей выборки на вход нейронной сети. При этом для каждого входного образа выполняются следующие действия:

a) производится фаза прямого распространения входного образа по нейронной сети. При этом вычисляется выходная активность всех нейронных элементов сети:

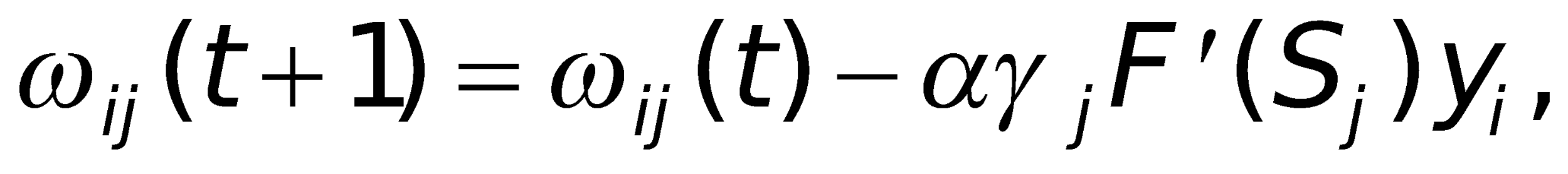
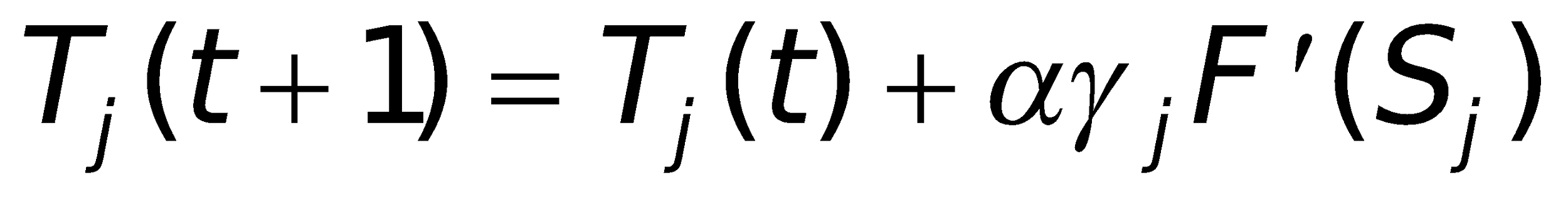
,   
где индекс j характеризует нейроны следующего слоя по отношению к слою i.

b) производится фаза обратного распространения сигнала, в результате которой определяется ошибка нейронных элементов для всех слоев сети. При этом соответственно для выходного и скрытого слоев:

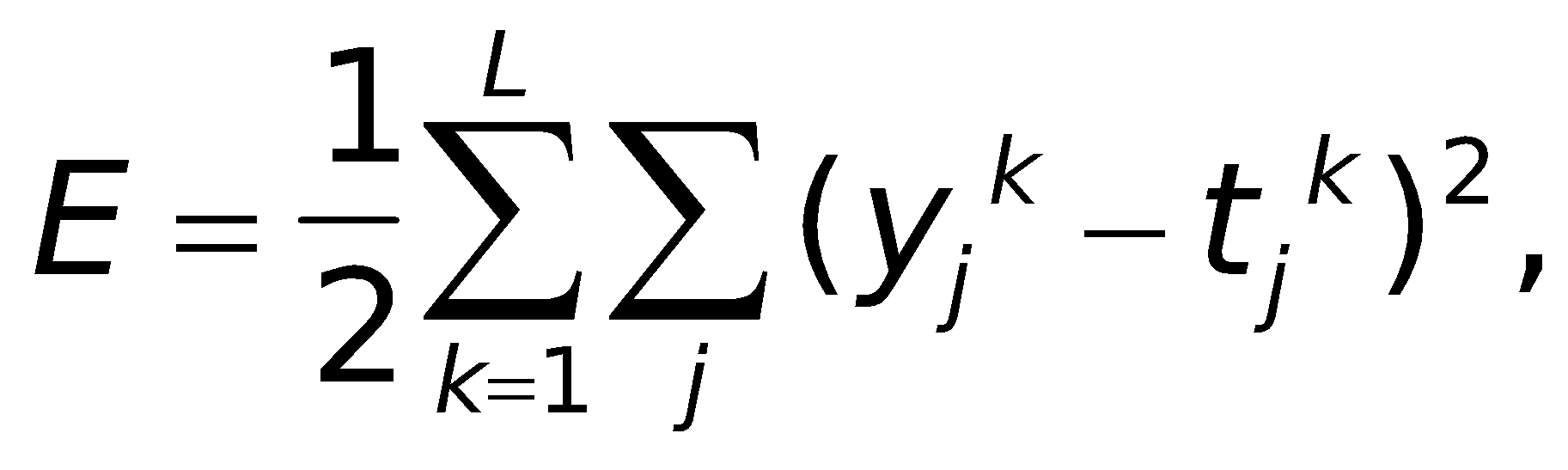


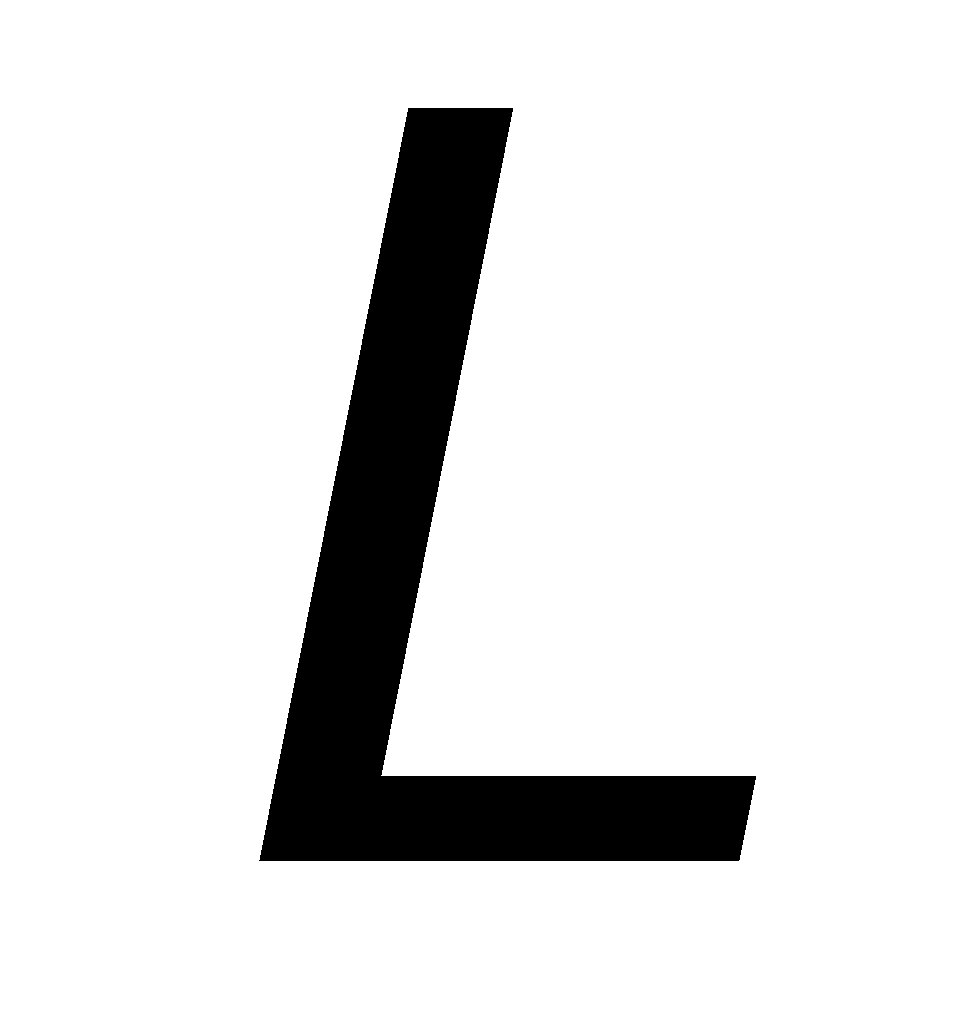
В последнем выражении индекс i характеризует нейронные элементы  
следующего слоя по отношению к слою j.

c) для каждого слоя нейронной сети происходит изменение весовых коэффициентов и порогов нейронных элементов:

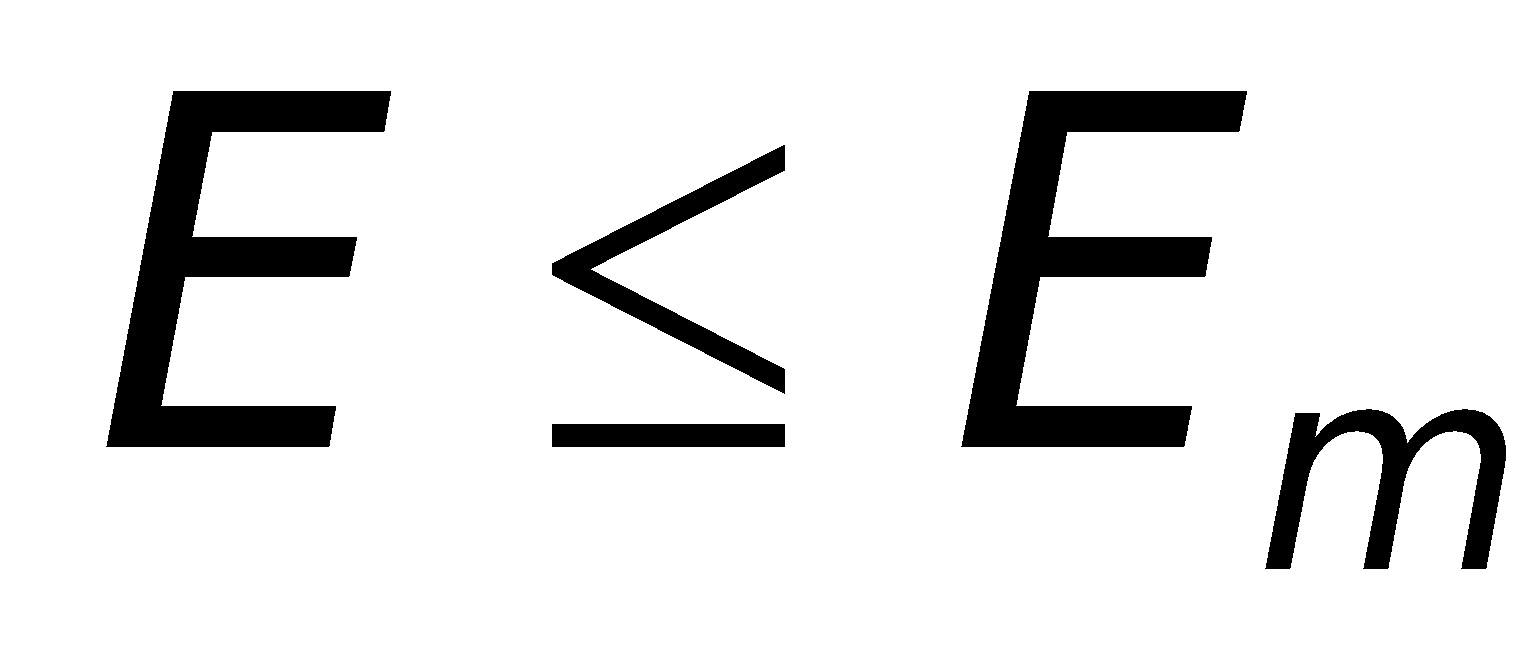
 .

4. Вычисляется суммарная среднеквадратичная ошибка нейронной сети:



где — размерность обучающей выборки.

5. Если E. > . то происходит переход к шагу 3 алгоритма. В противном случае алгоритм обратного распространения ошибки заканчивается.

Таким образом, данный алгоритм функционирует до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной, т. е. .

# 5. ТЕСТИРОВАНИЕ И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Целью этого этапа курсового проекта является определение тех параметров нейронных сетей, которые не были определены на этапе проектирования. Для сети MLP 3/p/3 и 6/p/3 необходимо определить число нейронных элементов в скрытом слое.

Главным критерием, на основании которого будем определять эти параметры, является ошибка аппроксимации и прогнозирования функции (1.1), причем нейронная сеть должна обладать как достаточной точностью аппроксимации функции, так и приемлемой точностью прогнозирования. Точность аппроксимации и прогнозирования функции (1.1) нейронной сетью должна определяться как визуально по степени сходства оригинального графика функции и графика, построенного по выходным значениям нейронной сети, так и аналитически на основе расчета суммарной квадратичной ошибки. Также на данном этапе необходимо провести сравнительный анализ полученных результатов, на основании чего определить нейронную сеть, которая наилучшим образом подходит для решения поставленной задачи. Ниже приведен пример результатов визуальной аппроксимации и прогнозирования функции (1.1):

Параметры обучения для архитектуры 3/p/3:

Сигмоидная функция активации на скрытом слое,

Линейная функция активации на внешнем слое;

Число нейронных элементов в скрытом слое p = 7;

Обучающая выборка L = 500;

Желаемая ошибка сети Em = 0,1;

Шаг обучения α = 0,00035.

Нейронная сеть была обучена 15 минут.

Результаты аппроксимации функции:

Рис. 4.1. Зависимость X(t)

Рис. 4.2. Зависимость Y(t)

Рис. 4.3. Зависимость Z(t)

Рис. 4.4. Зависимость XY

Рис. 4.5. Зависимость XZ

Рис. 4.6. Зависимость YZ

Результаты прогнозирования функции:

Рис. 4.7. Зависимость X(t)

Рис. 4.8. Зависимость Y(t)

Рис. 4.9. Зависимость Z(t)

Рис. 4.10. Зависимость XY

Рис. 4.11. Зависимость XZ

Рис. 4.12. Зависимость YZ

Параметры обучения для архитектуры 6/p/3:

Сигмоидная функция активации на скрытом слое,

Линейная функция активации на внешнем слое;

Число нейронных элементов в скрытом слое p = 15;

Обучающая выборка L = 500;

Желаемая ошибка сети Em = 0,1;

Шаг обучения α = 0,00035.

Нейронная сеть была обучена 7 минут.

Результаты аппроксимации функции:

Рис. 5.1. Зависимость X(t)

Рис. 5.2. Зависимость Y(t)

Рис. 5.3. Зависимость Z(t)

Рис. 5.4. Зависимость XY

Рис. 5.5. Зависимость XZ

Рис. 5.6. Зависимость YZ

Результаты прогнозирования функции:

Рис. 5.7. Зависимость X(t)

Рис. 5.8. Зависимость Y(t)

Рис. 5.9. Зависимость Z(t)

Рис. 5.10. Зависимость XY

Рис. 5.11. Зависимость XZ

Рис. 5.12. Зависимость YZ

# ЗАКЛЮЧЕНИЕ

Был проведён анализ нейронной сети путём изменения её параметров и архитектуры. Нейронная сеть справляется с поставленной для неё задачей с разной скоростью обучения. Процесс её обучения и получения необходимого результата можно ускорить путём подбора наиболее подходящих параметров. Шаг обучения сети играет роль в скорости её обучения и вероятность попадания в локальные минимумы. Программная реализация нейронной сети сделана на языке программирования C# в среде разработки Visual Studia для быстрого расчета главных параметров сети и легкой замены ее параметров. При больших обучающих выборках программе нужно больше времени для привидения к нужной ошибке сети, но это приводит к большей точности прогнозирования, особенно, кода сохраняется период в прогнозирующей функции и этот период был в обучающей выборке.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1). Головко, В.А. Нейронные сети: обучение, организация и применение: учеб. пособие для вузов / В.А. Головко; общая ред. А.И. Галушкина. – М.: ИПРЖР, 2001. – Кн. 4. – 256 с.

2). ГОСТ 7.1-2003. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – Минск, 2004. – 48 с.

3). ГОСТ 19.504 – 79. ЕСПД. Руководство программиста. Требования к содержанию и оформлению.

4). ГОСТ 2.105-95. ЕСКД. Общие требования к текстовым документам.