

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Инфокоммуникационных технологий»

РАБОТА ПРОВЕРЕНА  
Рецензент

\_\_\_\_\_ 201\_ г.  
“ \_\_\_ ” \_\_\_\_\_

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ИКТ

\_\_\_\_\_ С.Н. Даровских  
“ \_\_\_ ” \_\_\_\_\_ 201\_ г.

**Исследование нейросетевого экстраполятора радиотехнического сигнала**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ  
КВАЛИФИКАЦИОННОЙ РАБОТЕ (ВКР)  
ЮУрГУ – Д.11.05.01.2018.313.00 ПЗ

Консультанты:

По экономике

\_\_\_\_\_ Н.С. Дзензелюк  
“ \_\_\_ ” \_\_\_\_\_ 2017 г.

Руководитель работы:

\_\_\_\_\_ А.Н. Рагозин  
“ \_\_\_ ” \_\_\_\_\_ 201\_ г.

По БЖД

\_\_\_\_\_ И.С.Окраинская  
“ \_\_\_ ” \_\_\_\_\_ 2017 г.

Автор работы:

студент группы КЭ-609  
\_\_\_\_\_ Н.О. Лёвочкин  
“ \_\_\_ ” \_\_\_\_\_ 201\_ г.

Нормоконтролер:

\_\_\_\_\_ В.Д. Спицына  
“ \_\_\_ ” \_\_\_\_\_ 2018 г.

Челябинск  
2018

## РЕФЕРАТ

Лёвочкин Н.О. Исследование нейросетевого экстраполятора радиотехнического сигнала. – Челябинск: ЮУрГУ, ВШЭКН; 2018, 83с. 24ил., библиогр. список – 9 наим., 1 прил., 4 листа чертежей ф. А1.

Объектом исследования является искусственная нейронная сеть с использованием генетического алгоритма при обучении. Рассматривается задача предсказания будущих координат сопровождаемого динамического объекта по ранее полученным отметкам.

Целью работы является повышение точности краткосрочного прогнозирования сигналов в зависимости от типа прогнозируемой системы.

Для достижения цели работы были решены следующие задачи:

- освоить теоретические основы построения искусственных нейронных сетей;
- освоить аппарат обучения нейронных сетей генетическим алгоритмом;
- освоить программный пакет Neural Network Toolbox.

Работа была реализована с помощью программных продуктов: MathWorks Matlab R2014; Microsoft Word; Microsoft Excel.

Область применения – использование искусственной нейронной сети с целью предсказания может быть использована в задачах управления различными объектами.

					ЮУрГУ-Д.11.05.01.2018.313.00 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.	Лёвочкин Н.О.				Исследование нейросетевого экстраполятора радиотехнического сигнала	Лит.	Лист	Листов
Проверил	Рагозин А.Н.					Д	3	83
Н. Контр.	Спицына В.Д.					ЮУрГУ Кафедра ИКТ		
Утв.	Даровских С.Н.							

## ОГЛАВЛЕНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	6
ВВЕДЕНИЕ.....	7
1 Генетический алгоритм.....	9
1.1 Основа теории о генетическом алгоритме.....	9
1.2 Результаты практического исследования.....	17
2 Искусственная нейронная сеть.....	28
2.1 Основа теории нейронной сети.....	28
2.1.1 Модель нейрона.....	32
2.1.2 Типы функций активаций.....	34
2.1.3 Архитектура сетей.....	35
2.1.4 Алгоритмы обучения.....	38
2.2 Результаты практического исследования.....	43
3 Нейросетевой экстраполятор с использование генетического алгоритма при обучении.....	48
4 Организационно-экономический раздел.....	52
4.1 Составление и расчет параметров сетевого графика.....	53
4.1.1 Перечень, параметры и вероятностные характеристики работ ..	55
4.1.2 Расчет параметров событий сетевого графика.....	58
4.1.3 Расчёт параметров работ сетевого графика.....	59
4.1.4 Расчёт параметров сетевого графика в целом.....	60
4.2 Расчёт стоимостных параметров сетевого графика.....	61
4.2.1 Расчёт материальных затрат.....	61
4.2.2 Расчёт стоимости контрагентских услуг.....	61
4.2.3 Расчет стоимости покупных изделий.....	62
4.2.4 Расчет стоимости заработной платы.....	62
4.2.5 Расчет стоимости накладных расходов.....	62
4.2.6 Расчет стоимости полной сметы затрат.....	63
4.3 Анализ технико-экономической эффективности.....	63

5 Безопасность жизнедеятельности.....	65
5.1 Требования к помещениям для работы с ПЭВМ.....	65
5.2 Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе.....	66
5.3 Требования к уровням шума и вибрации.....	68
5.4 Требования к освещению на рабочих местах.....	69
5.5 Требования к уровням электромагнитных полей.....	71
5.6 Общие требования к организации рабочих мест пользователей...	71
5.7 Анализ собственного рабочего места.....	73
ЗАКЛЮЧЕНИЕ.....	75
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	76
ПРИЛОЖЕНИЯ	
ПРИЛОЖЕНИЕ А. Листинг программы.....	77

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

СКО – Среднее квадратическое отклонение,

MATLAB – Matrix Laboratory,

VLSI – Very Large Scale Integration.

## ВВЕДЕНИЕ

Процесс прогнозирования актуален в современном мире, поскольку в процессе своей деятельности при решении задач различной степени сложности каждая организация или предприятие рано или поздно сталкивается с потребностью общематематического обоснования принимаемых решений. Такие решения в большинстве своем направлены на повышение эффективности производственной деятельности.

Сфера применения методов прогнозирования охватывает области технических, военных, биологических и социально-экономических дисциплин. Прогнозирование, являющееся процессом определения эвентуального предположения о предстоящем состоянии объекта прогнозирования на основе динамики его изменения, позволяет избежать неверных, ранних или поздних решений. В большинстве случаев, прогнозы носят ориентировочный характер.

На сегодняшний момент в технических системах применяются в основном экстраполяционные методы прогнозирования, предоставляющие возможность делать ориентировочные выводы относительно разных процессов, явлений, реакций, операций. Как бы то ни было, применение нейросетевых моделей прогнозирования, позволяющих работать со смешанной информацией и применить в качестве первичной фактографическую и экспертную информацию, позволяет повысить достоверность и точность краткосрочных и долгосрочных прогнозов.

На настоящий момент существуют целый ряд прогнозных моделей и программных комплексов отечественных и зарубежных разработчиков, которые позволяют выполнять прогнозирование электропотребления с требуемой для пользователей точностью, но в ряде случаев, в частности, при

диспетчерском управлении в энергосистемах результаты по точности краткосрочного и оперативного прогнозирования не всегда удовлетворяют возрастающим требованиям.

# 1 Генетический алгоритм

## 1.1 Основа теории о генетическом алгоритме

Для решения задач оптимизации единственным возможным методом является метод случайного поиска, который при очень большом числе возможных вариантов, конечно, не гарантирует оптимальности решения, но зато находит решение, значительно лучшее, чем просто случайное.

Суть метода случайного поиска, в его наиболее прямолинейной реализации заключается в следующем. В области определения оптимизируемого вектора параметров, каждый из которых задает некоторую точку в  $n$ -мерном пространстве, генерируется некоторое, обычно достаточно большое (100... 1000), число случайных точек. Все эти точки равномерно в стохастическом смысле покрывают заданную изначально область определения. Затем из всех этих точек выбирается некоторая доля наилучших решений, т.е. тех точек, в которых значение целевой функции максимально (минимально) при решении задачи максимизации (минимизации). По выбранным наилучшим точкам строится наименьшая область определения параметров, содержащая все эти точки. Размер этой области будет хотя бы немного меньше размера исходной области. Процесс повторяется многократно, при этом каждый раз размер области наилучших решений сужается. В конце концов, этот размер становится настолько мал, что достаточно хорошо описывает искомое решение. На самом деле процесс оптимизации в виде непосредственного случайного поиска работает приемлемо только в случае, если оптимизируемая функция не имеет двух или более оптимумов с близкими значениями и размерность пространства параметров не очень велика. В противном случае, сужение области поиска будет незначительным в течение длительного времени, пока все наилучшие решения не окажутся в непосредственной близости от главного оптимума.



Поэтому для поиска оптимума в практически интересных случаях, когда целевая функция имеет несколько максимумов, необходимо привлекать более мощные методы оптимизации. [1, с.127]

В этих случаях целесообразно использовать нетрадиционные методы. Одним из таких многообещающих методов является генетический алгоритм оптимизации, точнее целое семейство таких методов, или эволюционное программирование. Методы генетической оптимизации стали применяться в широком масштабе для оптимизации весовых связей в нейронных сетях и до сих пор в этой области остаются одними из самых широко применяемых методов обучения.

Толчком для этих методов послужила эволюционная теория Дарвина и открытие генетических механизмов эволюции. Генетические алгоритмы реализуют процедуры поиска, основанные на естественном отборе и генетике. В наиболее общем виде генетические алгоритмы работают с конечным набором бинарных строк фиксированной длины.

Отбор или селекция имитируют выживание пригодных элементов в пределах популяции, обеспечивая предпочтение лучшим элементам. Конечно, для отбора необходим некоторый критерий качества. Он естественно вводится заданной целевой функцией. Таким образом, над элементами вводится частичное упорядочивание.

Здесь прослеживается явная аналогия с известными стохастическими методами оптимизации, упрощенной реализацией которых является случайный поиск. Единственное отличие заключается в том, что одновременно рассматривается множество вариантов, что эквивалентно процедуре случайного поиска, которая проводится из нескольких точек при допустимости произвольных переходов между точками.

Первичным оператором получения новых структур для исследования в большинстве генетических алгоритмов является оператор перекреста генотипов. Простейший оператор перекреста состоит из трех после-

довательных шагов. Сначала, используя заданный селективный оператор, выбираются два элемента из общей популяции для воспроизводства потомства, и затем случайным образом с равномерной плотностью выбирается точка перекреста, в ходе которого генные последовательности выбранных родителей обмениваются между собой участками генома.

Процесс продолжается, пока новая популяция не будет создана полностью, заполнившись потомками, которые построены из частей генов хороших (отселектированных) родителей. Существует значительное число других вариантов алгоритма перекреста генотипов, но общий смысл их остается одинаковым.

Для дополнительного улучшения сходимости применяют оператор мутаций. В двоично-кодированном генетическом алгоритме, мутация - это случайное изменение (с низкой вероятностью) положения единицы, и при других основаниях кодов могут применяться разнообразные методы возмущений исходных последовательностей.

Мутации описывают хаотические случайные блуждания в пространстве, образованном последовательностями. Используемые совместно с отбором, они как бы формируют параллельный, шумонезависимый алгоритм восхождения на холм. При использовании одновременно с отбором и перекрестом, мутации дают дополнительные гарантии против потери необходимого разнообразия.

Генетические алгоритмы могут решать трудные проблемы быстро и с требуемой точностью, легко расширяются, связываются с уже существующими моделями, объединяются с другими методами для повышения эффективности.

Помимо всего прочего, физическая наглядность позволяет достаточно просто модифицировать алгоритм для наилучшей работы при заданной проблеме. Например, часто в сложных многомерных задачах, возникают многомодальные целевые функции. В этих условиях целесообразно вести

одновременный поиск по всем перспективным направлениям. Реализация этого даже в самых простых генетических алгоритмах не вызывает трудностей и происходит автоматически. Когда интуитивный подход не приводит к должному успеху, возможно использование специальных проблемно-ориентированных приемов. Например, часто помимо мутаций и отбора используют и другие эвристики, например объединение глобального генетического поиска оптимума с локальной проблемно-ориентированной техникой. Информация по решаемой задаче может быть использована и в операциях кодирования и других.

Задача оптимизации сложных многобазовых радиолокационных сигналов совершенно естественно проецируется на базис генетических алгоритмов. Действительно, если для других задач одной из наиболее трудно формализуемых проблем является выбор варианта кодирования параметров системы генной последовательностью, для сигналов саму их структуру и можно рассматривать как генотип и именно к последовательности импульсов из нулей и  $\pm 1$  применять вышеописанные операторы генетических алгоритмов.

Популяция в такой формулировке представляет собой множество сигналов с различной структурой. Ее объем целесообразно выбирать не менее 100.

В работе использовался наиболее распространенный вариант ГА (рисунок 1.1), который состоит из трех операторов: отбора, перекреста хромосом (crossover), и мутаций. Для синтеза сигналов используется три оператора мутаций: добавление отдельного импульса, исключение импульса или сдвиг.

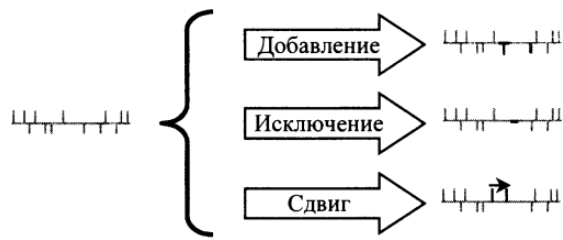


Рисунок 1.1 – Операторы мутации

Оператор мутации сводится к замене единственного импульса в сигнале, а, следовательно, и одного элемента в сигнальной последовательности на один из двух других возможных или сдвигу такого импульса. При этом с большим успехом может использоваться вышеописанный метод уточнения автокорреляционной функции для ускорения проведения расчетов. Кроме того, так как наихудшая часть популяции в дальнейшем будет отсеяна, можно на каждом шаге указать пороговое значение, при достижении которого целевой функцией, вычисления могут быть прекращены ввиду очевидной непригодности рассматриваемого экземпляра. Такой сокращенный вариант вычислений может уменьшить время на один-два порядка, особенно при приближении к оптимуму, когда улучшение достигается редко. Последовательность вычисления автокорреляционной функции при этом очень важна. Поэтому при расчетах необходимо сначала оценивать значения целевой функции по главным осям, где наиболее часто появляются наихудшие значения. [1, с.158]

Оператор перекреста генотипов может быть реализован в двух вариантах. Первый практически полностью совпадает с описанным выше (рисунок 1.2).

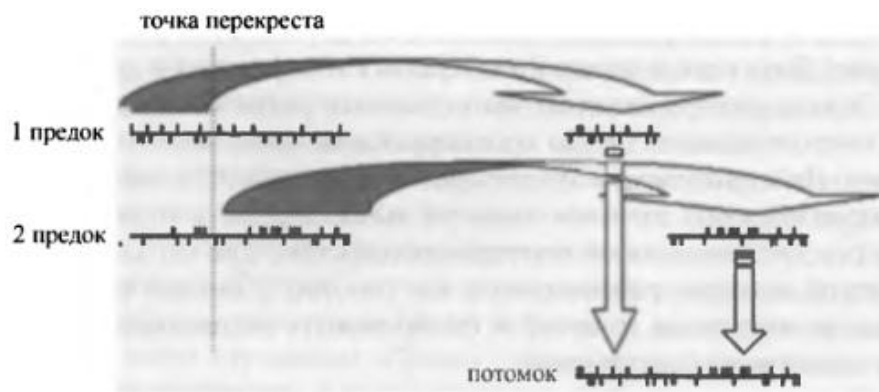


Рисунок 1.2 – Операция перекреста (crossover)

Второй опирается на тот факт, что отдельные импульсы в сигнале между собой не связаны и перенос в сигналы потомки достаточно связных кусков родительских сигналов не должен иметь существенного значения. Наоборот, это может привести к потере перспективных направлений поиска в параметрическом пространстве. Поэтому обмен элементами родительских сигналов при воспроизводстве сигнала потомка для каждого отдельного импульса осуществляется случайным образом с заданной вероятностью. Причем, процесс происходит независимо для всех импульсов. [2, с.6]

В воспроизведенной популяции присутствуют потомки, полученные обоими способами и с учетом мутаций. Оператор мутации применяется к потомкам особей, получаемых после операции скрещивания.

Эффективность генетических алгоритмов по сравнению с другими методами случайного поиска достаточно очевидна из-за целевой направленной обусловленности процесса поиска.

Для полноценного использования преимуществ целесообразно использовать гибридную технику «генетический алгоритм / восхождение на холм». Так как генетические алгоритмы весьма медленны при получении точных результатов, данный метод способен быстро определить сосредотачивается ли алгоритм на наиболее многообещающих областях.

При оптимизации радиолокационных сигналов целевая функция и параметры оптимизации не являются непрерывными величинами, что затрудняет

использование обычной техники восхождения на холм. Для ее применения можно использовать «локальный» генетический алгоритм тонкой настройки структуры организма. Используя небольшую популяцию организмов, подобных последнему полученному результату и локализуя операторы мутации и наследственности таким образом, чтобы отдавать предпочтение небольшим изменениям, этот локальный генетический алгоритм аналогичен алгоритму динамического восхождения на холм.

На рисунке 1.3 изображена общая схема процесса проведения генетической оптимизации многобазовых радиолокационных сигналов.

Задача оптимизации радиолокационных сигналов включает ряд ограничений (примерный уровень скважности для возможности приема сигналов в просветах между импульсами, отсутствие какой бы то ни было регулярности для повышения скрытности излучения и т.п.). На первый взгляд может показаться, что в генетических алгоритмах нужно просто не рассматривать решения, в которых не выполняются заданные ограничения. Такие «жесткие» ограничения полезны в традиционной технике оптимизации, но в генетических алгоритмах разнообразие составляет суть процесса оптимизации.

Если организм исключается из рассмотрения только потому, что он не удовлетворяет некоторым ограничениям, можно потерять важную информацию, которая, в конечном счете, могла бы привести к оптимальному решению. Кроме того, при «жестких» ограничениях исключаются решения, в которых лишь немного нарушается единственное ограничение и одновременно достигается очень хорошее значение целевой функции. [2, с.8]

Поэтому считается общепринятым, что замена «жестких» ограничений на хорошо подобранные «мягкие» ограничения позволяет улучшить работу генетических алгоритмов.

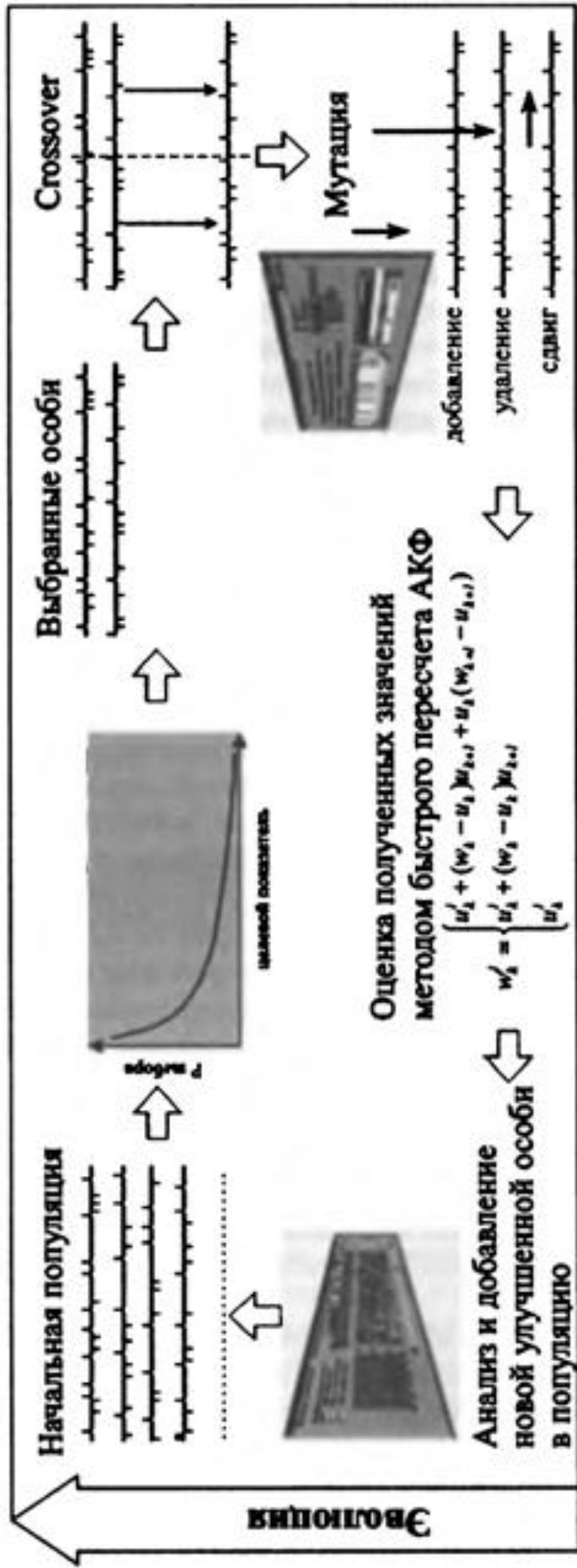


Рисунок 1.3 – Общая схема генетического алгоритма

В большинство генетических алгоритмов используется лишь небольшое число изменяемых параметров: размер популяции, уровень перекреста генотипов родителей и уровень мутаций. Путем точной настройки значений этих параметров в некоторых случаях можно удвоить скорость оптимизации.

Автоматическую настройку параметров можно проводить путем отображения наилучших организмов на графике после каждой попытки. С помощью полученной кривой определяется уровень улучшения, и уточняются параметры перекреста генотипов и мутаций соответственно. Размер популяций также должен меняться в течение эволюционного процесса для того, чтобы сбалансировать скорость развития с разнообразием генов. Адаптивный генетический алгоритм автоматически наращивает популяцию (и увеличивает разнообразие генов), когда процесс оптимизации замедляется и, наоборот, сокращает гетерогенные популяции.

## 1.2 Результаты практического исследования

В среде Matlab был проведен цикл численных экспериментов по оптимизации сложных сигналов. При этом задавались различные переменные. [3, 108 с.]

В зависимости от того, какие принимаются параметры, вида ограничений задачи и желаемых выходных данных используется варианты вызова функции `ga` генетического алгоритма. Мы использовали следующий вариант вызова:

$$x = \text{ga}(\text{fitnessfcn}, \text{nvars}, \text{A}, \text{b}, \text{Aeq}, \text{beq}, \text{LB}, \text{UB}, \text{nonlcon}, \text{options}) \quad (1)$$

где `fitnessfcn` – имя целевой функции,

`nvars` – число переменных,

`A` – матрица для линейных ограничений неравенства вида:  $A \cdot x \leq b$ ,

`b` – вектор для линейных ограничений неравенства вида:  $A \cdot x \leq b$ ,



$A_{eq}$  – матрица для линейных ограничений равенства вида:  $A_{eq} \cdot x = b_{eq}$ ,  
 $b_{eq}$  – вектор для линейных ограничений равенства вида:  $A_{eq} \cdot x = b_{eq}$ ,  
 $LB$  – вектор нижних границ.  $ga$  заставляет итерации оставаться над  $LB$ ,  
 $UB$  – вектор верхних границ.  $ga$  заставляет итерации оставаться ниже  $UB$ ,

$Nonlcon$  – дескриптор функции, возвращающий два выхода:  $[C, seq] = nonlcon(x)$ ,

$options$  – структура, содержащая параметры оптимизации.

Генетический алгоритм реализован в функции  $ga$ , а для задания параметров алгоритма используется функция  $gaoptimset$ . Если ввести в командную строку

$options = gaoptimset(@ga)$ ,

то будет создана структура параметров, имеющих значения по умолчанию. Для изменения параметров каждый параметр указывается в виде пары «имя», значение» как аргументы функции  $gaoptimset$ . Все параметры генетического алгоритма, их смысл и значения по умолчанию приведены в таблице 1.1.

Таблица 1.1 – Параметры Genetic Algorithm

Параметр	Описание	Значение по умолчанию
CreationFcn	Задаёт функцию, создающую начальную популяцию: $@gacreationuniform$ , $@gacreationlinearfeasible$	$@gacreationuniform$
CrossoverFcn	Задаёт функцию кроссовера: $@crossoverheuristic$ , $@crossoversscattered$ , $@crossoverintermediate$ , $@crossoverssinglepoint$ , $@crossoverstwopoint$ , $@crossoverarithmic$	$@crossoversscattered$
CrossoverFraction	Доля популяции следующего поколения, создаваемая кроссовером	0.8
Display	Уровень вывода: 'off', 'iter', 'diagnose', 'final'	'final'
DistanceMeasureFcn	Задаёт функцию, которая вычисляет меру расстояния между особями в пространстве переменных (генотип) или функции (фенотип): $\{@distancecrowding, 'phenotype'\}$	[]
EliteCount	Число элитных особей, Не используется в $gamultiobj$	2

Продолжение таблицы 1.1

Параметр	Описание	Значение по умолчанию
FitnessLimit	Значение фитнес-функции, при достижении которого останавливается алгоритм	-Inf
FitnessScalingFcn	Задаёт функцию, масштабирующую фитнес-функцию: @fitscalingshiftlinear, @fitscalingprop, @fitscalingtop, @fitscalingrank	@fitscalingrank
Generations	Определяет максимальное число итераций	100
HybridFcn	Задаёт функцию, которая продолжает оптимизацию после окончания ga: собственная, @fminsearch, @patternsearch, @fminunc, @fmincon или {@solver, hybridoptions} (при типе данных double)	[]
InitialPenalty	Начальное значение параметра штрафа	10
InitialPopulation	Задаваемая пользователем матрица начальной популяции или ее части	[]
InitialScores	Задаваемый пользователем вектор-столбец начальных меток фитнес, возможно частичный	[]
MigrationDirection	Направление миграции (перемещения): 'both', 'forward'	'forward'
MigrationFraction	Доля особей в каждой подпопуляции, перемещаемая в другую подпопуляцию	0.2
MigrationInterval	Число поколений между миграциями особей между подпопуляциями	20
MutationFcn	Указывает функцию получения потомков мутацией: @mutationuniform, @mutationadaptfeasible, @mutationgaussian	@mutation gaussian
OutputFcns	Указывает функцию, вызываемую на каждой итерации: @gaoutputgen	[]
ParetoFraction	Доля особей, удерживаемых на первом фронте Парето, в то время как решатель выбирает особи из более высоких фронтов	0.35
PenaltyFactor	Коэффициент обновления штрафа	100
PlotFcns	Задаёт функции графики: @gaplotbestf, @gaplotbestindiv, @gaplotdistance, @gaplotexpectation, @gaplotgeneology, @gaplotselection, @gaplotrange, @gaplotscorediversity, @gaplotscores, gaplotstopping	[]
PlotInterval	Число поколений между вызовами функций графики	1
PopInitRange	Матрица или вектор диапазона особей в начальной популяции	[0;1]
PopulationSize	Размер популяции, целое число или вектор	20
PopulationType	Задаёт тип данных популяции: 'bitstring', 'custom' (оба при отсутствии ограничений), 'doubleVector'	'doubleVector'

### Окончание таблицы 1.1

Параметр	Описание	Значение по умолчанию
SelectionFcn	Определяет функцию выбора родителей: @selectionremainder, @selectionuniform, @selectionstochunif, @selectionroulette, @selectiontournament	@selectionstochunif
StallGenLimit	Число последовательных поколений, не дающее улучшения целевой функции, после которого останавливается алгоритм	50
StallTimeLimit	Время, за которое, если нет улучшения целевой функции, алгоритм останавливается, с	Inf
TimeLimit	Максимально допустимое время работы ga, с	Inf
TolCon	Приемлемая величина нарушения нелинейных ограничений	1e-6
TolFun	Если изменение целевой функции за StallGenLimit поколений меньше этой величины, алгоритм останавливается	1e-6
UseParallel	Определяет, вычислять ли фитнес-функции популяции в параллель: 'always','never'	'never'
Vectorized	Определяет вычисление фитнес-функции как векторной величины: 'on','off'	'off'

Важным параметром алгоритма является размер популяции (PopulationSize), его значением может быть целое положительное число или целочисленный вектор. Во втором случае будет создано столько подпопуляций, какова размерность вектора, а размер подпопуляций будет равен значениям компонент вектора.

Рассмотрим работу генетического алгоритма на примере функции:

$$f = 3(1 - x_1)^2 \exp(-x_1^2 - (x_2^2 + 1)^2) - 10 \left( \frac{x_1}{5} - x_1^3 - x_2^5 \right) \exp(-x_1^2 - x_1^2) - \frac{1}{3} \exp(-(x_1 + 1)^2 - x_2^2) \quad (2)$$

при условиях  $-3 \leq x_j \leq 3, j = 1, 2$ .

Примем размер популяции 20 и составим и исполним следующую программу:

```
options=gaoptimset('OutputFcn',@outputfun1,'PlotFcns',{ @gaplotbestf,...
    @plotfun2,@plotfun3},'PopulationSize',20);
```

```
[x,fval,exitFlag,output]=ga(@myfun,2,[],[],[],[],[-3 -3],...  
[3 3],[],options)
```

Optimization terminated: average change in the fitness value less than options.TolFun.

```
x = 0.2283 -1.6255
```

```
fval = -6.5511
```

```
exitFlag = 1
```

```
output =
```

```
problemtype: 'boundconstraints'
```

```
rngstate: [1x1 struct]
```

```
generations: 77
```

```
funccount: 1560
```

```
message: 'Optimization terminated: average change in the fitness value  
less than o...'
```

```
maxconstraint: 0
```

Полученные результаты показывают, что за 77 итераций, в течение которых целевая функция вычислялась 1560 раз, найден минимум, оказавшийся глобальным.

На рисунке 1.4 показано изменения значения целевой функции в ходе процесса оптимизации.

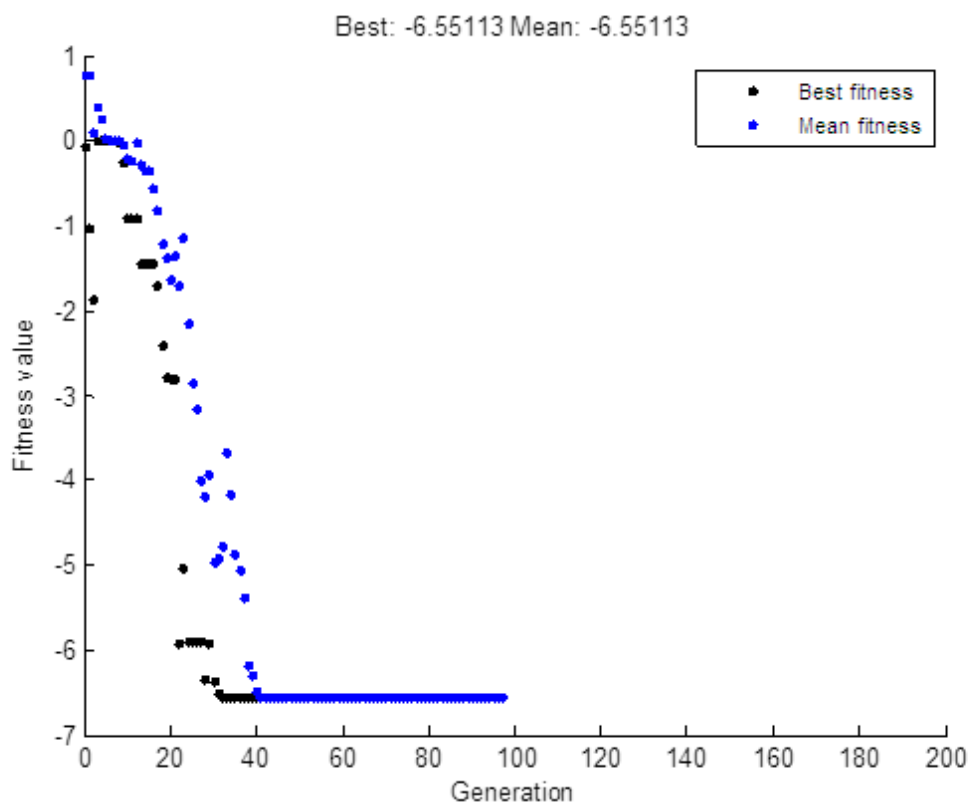


Рисунок 1.4 – Оптимизация целевой функции

Полученные результаты показывают, что значительного уменьшения уровня удастся добиться уже на первых шагах эволюции сигналов. Это связано с тем, что на каждой итерации проводится многократный процесс дополнительной оптимизации за счет мутаций, при котором выбираются наилучшие коды, получаемые из потомка путем изменения состояния одного из импульсов. Причем, проводится почти полный перебор возможных однократных перестановок импульсов.

На следующих этапах из популяции сигналов исключаются неудачные варианты, которые не могут привести к значительным улучшениям, и дополнительно включаются сигналы, имеющие потенциальные возможности к значительным улучшениям свойств автокорреляционной функции. Из графика видно, что оптимизация достигается на 40 итерации.

Для обучения нейросети нам необходима проверка значений  $x_1$  и  $x_2$  на тестовых функциях. Подставляя значения  $x_1$  и  $x_2$  в тестовые функции мы можем наблюдать их изменения, представленные на рисунке 1.5.

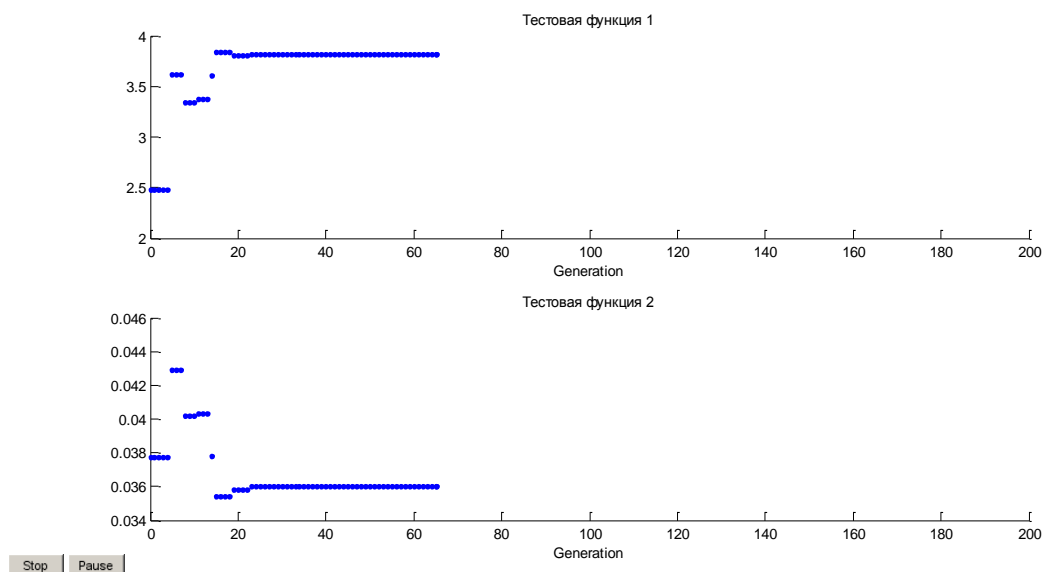


Рисунок 1.5 – Графические изменения тестовых функций

На рисунке 1.6 представлено расположение особей в четырех итерациях: черными кружками – начальная популяция, зелеными ромбиками – особи первого поколения, синими – второго поколения, лазурными – 20-го поколения и красными снежинками – финальная популяция. Видно, что уже 20-е поколение располагается близко к минимуму, а особи конечной популяции настолько близки друг к другу, что слились в одну точку. На первых итерациях хорошо видно, что по крайней мере две особи смежных поколений совпадают, что обусловлено использованием элитизма (по умолчанию две особи элитные).

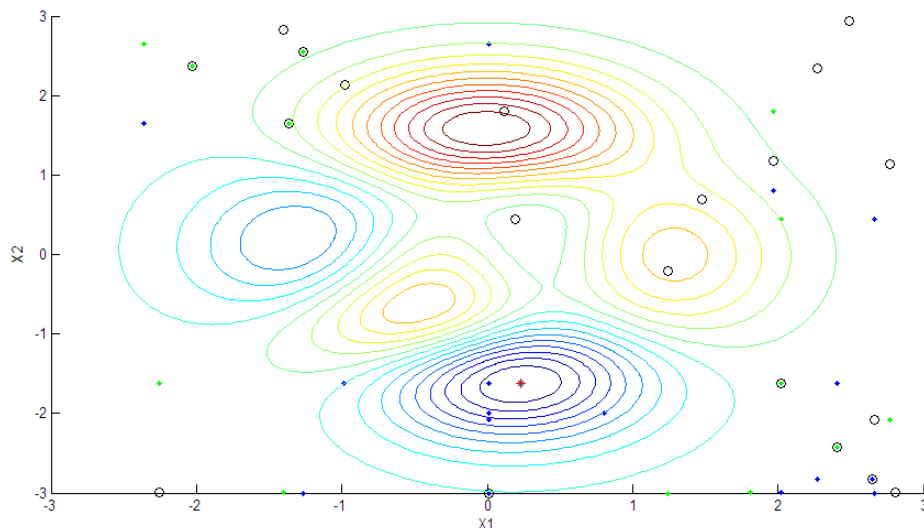


Рисунок 1.6 – Расположение особей в четырех итерациях

Для использования данных в дальнейшей работе мы должны составить матрицу данных.

Используя следующую функцию составим подпрограмму для вывода матрицы данных:

```
[state,options,optchanged] = outputfun1(options,state,flag,interval)
```

В последующем мы сможем вызывать матрицу данных из командной строки, пример указан на рисунке 1.7.

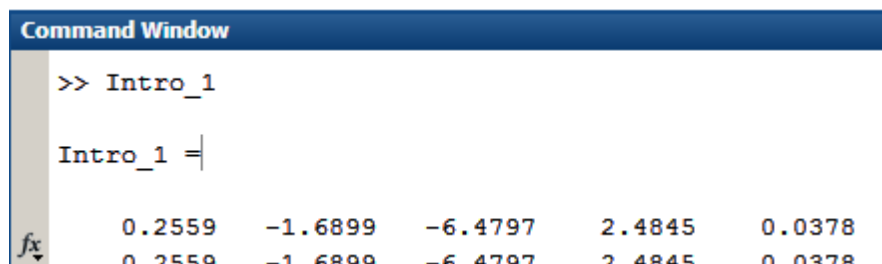


Рисунок 1.7 – Матрица данных

В конечном итоге мы получаем следующую матрицу данных:

Intro\_1 =

0,2559	-1,6899	-6,4797	2,4845	0,0378
0,2559	-1,6899	-6,4797	2,4845	0,0378
0,2559	-1,6899	-6,4797	2,4845	0,0378

0,2559	-1,6899	-6,4797	2,4845	0,0378
0,2559	-1,6899	-6,4797	2,4845	0,0378
0,2559	-1,5649	-6,4966	3,6140	0,0429
0,2559	-1,5649	-6,4966	3,6140	0,0429
0,2559	-1,5649	-6,4966	3,6140	0,0429
0,2559	-1,6274	-6,5436	3,3437	0,0402
0,2559	-1,6274	-6,5436	3,3437	0,0402
0,2559	-1,6274	-6,5436	3,3437	0,0402
0,2559	-1,6246	-6,5439	3,3770	0,0403
0,2559	-1,6246	-6,5439	3,3770	0,0403
0,2559	-1,6246	-6,5439	3,3770	0,0403
0,2403	-1,6274	-6,5496	3,6078	0,0378
0,2247	-1,6274	-6,5510	3,8428	0,0354
0,2247	-1,6274	-6,5510	3,8428	0,0354
0,2247	-1,6274	-6,5510	3,8428	0,0354
0,2247	-1,6274	-6,5510	3,8428	0,0354
0,2274	-1,6274	-6,5511	3,8053	0,0358
0,2274	-1,6274	-6,5511	3,8053	0,0358
0,2274	-1,6274	-6,5511	3,8053	0,0358
0,2274	-1,6274	-6,5511	3,8053	0,0358
0,2282	-1,6254	-6,5511	3,8161	0,0360
0,2282	-1,6254	-6,5511	3,8161	0,0360
0,2282	-1,6254	-6,5511	3,8161	0,0360
0,2282	-1,6254	-6,5511	3,8161	0,0360
0,2282	-1,6254	-6,5511	3,8161	0,0360
0,2282	-1,6254	-6,5511	3,8161	0,0360
0,2283	-1,6256	-6,5511	3,8141	0,0360
0,2283	-1,6256	-6,5511	3,8141	0,0360
0,2283	-1,6256	-6,5511	3,8141	0,0360





0,2283	-1,6255	-6,5511	3,8145	0,0360
0,2283	-1,6255	-6,5511	3,8145	0,0360
0,2283	-1,6255	-6,5511	3,8145	0,0360
0,2283	-1,6255	-6,5511	3,8145	0,0360
0,2283	-1,6255	-6,5511	3,8145	0,0360

В первом столбце указаны изменения значений  $x_1$ , во втором изменения значений  $x_2$ , в третьем изменения значений целевой функции, а в четвертом и пятом изменения значений тестовых функций.

## 2 Искусственная нейронная сеть

### 2.1 Основа теории нейронной сети

Исследования по искусственным нейронным сетям (далее — нейронные сети) связаны с тем, что способ обработки информации человеческим мозгом в корне отличается от методов, применяемых обычными цифровыми компьютерами. Мозг представляет собой чрезвычайно сложный, нелинейный, параллельный компьютер (систему обработки информации). Он обладает способностью организовывать свои структурные компоненты, называемые нейронами, так, чтобы они могли выполнять конкретные задачи (такие как распознавание образов, обработку сигналов органов чувств, моторные функции) во много раз быстрее, чем могут позволить самые быстродействующие современные компьютеры. Примером такой задачи обработки информации может служить обычное зрение. В функции зрительной системы входит создание представления окружающего мира в таком виде, который обеспечивает возможность взаимодействия с этим миром.

Более точно, мозг последовательно выполняет ряд задач распознавания (например, распознавание знакомого лица в незнакомом окружении). На это него уходит около 100...200 миллисекунд, в то время как выполнение аналогичных задач даже меньшей сложности на компьютере может занять несколько дней.

Аналогично, в искусственных нейронных сетях работа проводится с искусственными нейронами. В общем случае нейронная сеть представляет собой машину, моделирующую способ обработки мозгом конкретной задачи. Эта сеть обычно реализуется с помощью электронных компонентов или моделируется программой, выполняемой на цифровом компьютере. Предметом рассмотрения настоящей книги является важный класс нейронных сетей, осуществляющих вычисления с помощью процесса

обучения. Для того чтобы добиться высокой производительности, нейронные сети используют множество взаимосвязей между элементарными ячейками вычислений — нейронами. Таким образом, можно дать следующее определение нейронных сетей, выступающих в роли адаптивной машины.

Нейронная сеть — это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и предоставляющих их для последующей обработки. [4, с.32]

Нейронная сеть сходна с мозгом с двух точек зрения:

— знания поступают в нейронную сеть из окружающей среды и используются в процессе обучения;

— для накопления знаний применяются связи между нейронами, называемые синаптическими весами.

Процедура, используемая для процесса обучения, называется алгоритмом обучения. Эта процедура выстраивает в определенном порядке синаптические веса нейронной сети для обеспечения необходимой структуры взаимосвязей нейронов.

Изменение синаптических весов представляет собой традиционный метод настройки нейронных сетей. Этот подход очень близок к теории линейных адаптивных фильтров, которая уже давно заявила о себе и применяется в различных областях деятельности человека. Однако нейронные сети могут изменять собственную топологию. Это обусловлено тем фактом, что нейроны в человеческом мозге постоянно отмирают, а новые синаптические связи постоянно создаются.

Использование нейронных сетей обеспечивает следующие полезные свойства систем.

1. Нелинейность. Искусственные нейроны могут быть линейными и нелинейными. Нейронные сети, построенные на соединении нелинейных нейронов, сами являются нелинейными. Нелинейность является чрезвычайно

важным свойством, особенно если сам физический механизм, отвечающий за формирование входного сигнала, тоже является нелинейным (например, человеческая речь).

2. Отображение входной информации в выходную. Одной из популярных парадигм общения является обучение с учителем. Это подразумевает изменение синаптических весов на основе набора маркированных учебных примеров. Каждый пример состоит из входного сигнала и соответствующего ему желаемого отклика. Из этого множества случайным образом выбирается пример, а нейронная сеть модифицирует синаптические веса для минимизации расхождений желаемого выходного сигнала и формируемого сетью согласно выбранному статистическому критерию. При этом собственно модифицируются свободные параметры сети.

3. Адаптивность. Нейронные сети обладают способностью адаптировать свои синаптические веса к изменениям окружающей среды. В частности, нейронные сети, обученные действовать в определенной среде, могут быть легко переучены для работы в условиях незначительных колебаний параметров среды. Известно, что чем выше адаптивные способности системы, тем более устойчивой будет ее работа в нестационарной среде.

4. Очевидность ответа. В контексте задачи классификации образов можно разработать нейронную сеть, собирающую информацию не только для определения конкретного класса, но и для увеличения достоверности принимаемого решения. Впоследствии эта информация может использоваться для исключения сомнительных решений, что повысит продуктивность нейронной сети.

5. Контекстная информации. Знания представляются в самой структуре нейронной сети с помощью ее состояния активации. Каждый нейрон сети потенциально может быть подвержен влиянию всех остальных ее нейронов.

Как следствие, существование нейронной сети непосредственно связано с контекстной информацией.

6. Отказоустойчивость. Нейронные сети, облаченные в форму электроники, потенциально отказоустойчивы. Это значит, что при неблагоприятных условиях их производительность падает незначительно. Например, если поврежден какой-то нейрон или его связи, извлечение запомненной информации затрудняется. Однако, принимая в расчет распределенный характер хранения информации в нейронной сети, можно утверждать, что только серьезные повреждения структуры нейронной сети существенно повлияют на работоспособность. Чтобы гарантировать отказоустойчивость работы нейронной сети, в алгоритмы обучения нужно закладывать соответствующие поправки.

7. Масштабируемость. Параллельная структура нейронных сетей потенциально ускоряет решение некоторых задач и обеспечивает масштабируемость нейронных сетей в рамках технологии VLSI. Одним из преимуществ технологий VLSI является возможность представить достаточно сложное поведение с помощью иерархической структуры.

8. Единообразие анализа и проектирования. Нейронные сети являются универсальным механизмом обработки информации. Это означает, что одно и то же проектное решение нейронной сети может использоваться во многих предметных областях. Это свойство проявляется несколькими способами:

— нейроны в той или иной форме являются стандартными составными частями любой нейронной сети;

— эта общность позволяет использовать один и те же теории, и алгоритмы обучения в различных нейросетевых приложениях;

— модульные сети могут быть построены на основе интеграции целых модулей.

9. Аналогия с нейробиологией. Строение нейронных сетей определяется аналогией с человеческим мозгом, который является живым доказательством того, что отказоустойчивые параллельные вычисления не только физически реализуемы, но и являются быстрым и мощным инструментом решения задач.

Преимущества применения нейросетевых технологий:

- гибкость и адаптивность алгоритмов;
- возможность достижения высокого быстродействия за счет использования параллельной обработки данных;
- возможность создания системы без трудоемкого математического описания ее функционирования;
- возможность наращивания возможности сети после запуска в работу;
- способность к повторному обучению для изменения характеристик сети.

Среди недостатков применения нейросетей можно выделить следующие:

- отсутствие теории выбора конкретной модели;
- повсеместное использование эмпирических, а не строгих сведений о скорости сходимости алгоритмов.

### 1.2.1 Модель нейрона

Нейрон представляет собой единицу обработки информации в нейронной сети. На блок-схеме рисунке 2.1 показана модель нейрона, лежащего в основе искусственных нейронных сетей. В этой модели можно выделить три основных элемента. [4, с.42]

1. Набор синапсов или связей, каждый из которых характеризуется своим весом или силой. В частности, сигнал  $x_j$  на входе синапса  $j$ , связанного с нейроном  $k$ , умножается на вес  $w_{kj}$ . Важно обратить внимание на то, в

каком порядке указаны индексы синаптического веса  $w_{kj}$ . Первый индекс относится к рассматриваемому нейрону, а второй — ко входному окончанию синапса, с которым связан данный вес. Синаптический вес искусственного нейрона может иметь как положительное, так и отрицательные значения.

2. Сумматор складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как линейную комбинацию.

3. Функция активации ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией сжатия. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале  $[0, 1]$  или  $[-1, 1]$ .

В модель нейрона, показанную на рисунке 2.1, включен пороговый элемент, который обозначен символом  $b_k$ . Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации.

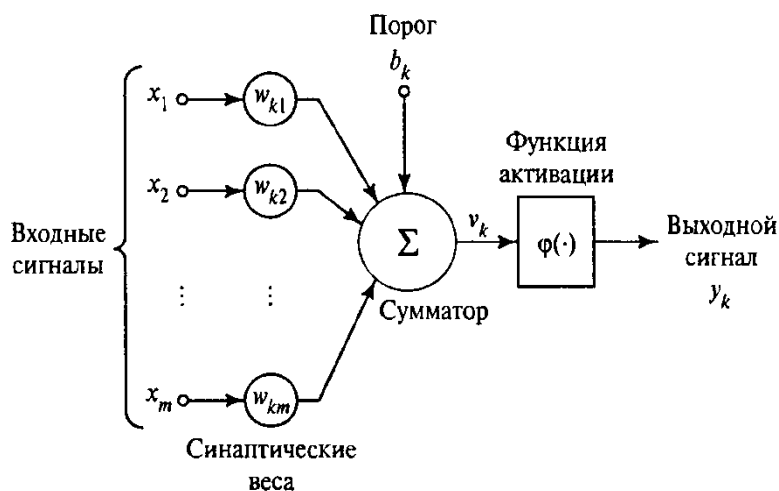


Рисунок 2.1 – Нелинейная модель нейрона



## 1.2.2 Типы функций активаций

Функции активации, определяют выходной сигнал нейрона в зависимости от индуцированного локального поля  $v$ . Можно выделить три основных типа функций активации.

1. Функция единичного скачка, или пороговая функция. Этот тип функции показан на рисунке 2.2, и описывается следующим образом:

$$\varphi(v) = \begin{cases} 1, & \text{если } v \geq 0; \\ 0, & \text{если } v < 0; \end{cases} \quad (3)$$

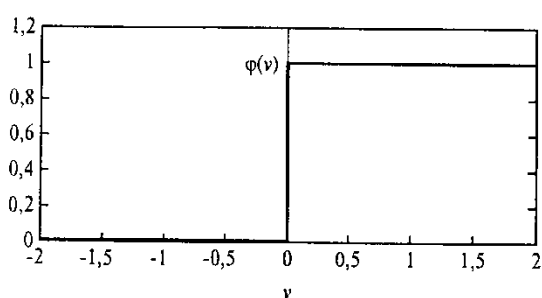


Рисунок 2.2 – Пороговая функция активации

2. Кусочно-линейная функция. Кусочно-линейная функция, показанная на рисунке 2.3, описывается следующим выражением:

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2}; \\ |v|, & +\frac{1}{2} > v > -\frac{1}{2}; \\ 0, & v \leq -\frac{1}{2}, \end{cases} \quad (4)$$

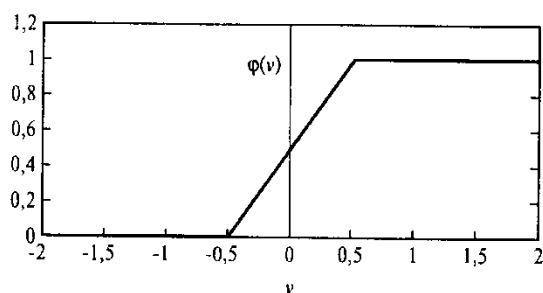


Рисунок 2.3 – Кусочно-линейная функция активации

3. Сигмоидальная функция. Сигмоидальная функция, представленная на рисунке 2.4, график которой напоминает букву S, является, пожалуй, самой распространенной функцией, используемой для создания искусственных нейронных сетей. Это быстро возрастающая функция, которая поддерживает

баланс между линейным и нелинейным поведением. Примером сигмоидальной функции может служить логистическая функция, задаваемая следующим выражением:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}, \quad (5)$$

где  $a$  – параметр наклона сигмоидальной функции.

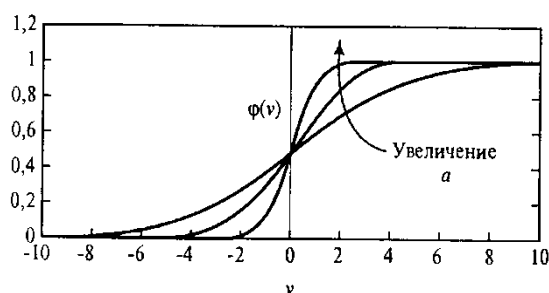


Рисунок 2.4 – Сигмоидальная функция активации

### 1.2.3 Архитектура сетей

В общем случае можно выделить три фундаментальных класса нейросетевых архитектур.

#### **Однослойные сети прямого распространения**

В многослойной нейронной сети нейроны располагаются по слоям. В простейшем случае в такой сети существует входной слой узлов источника, информация от которого передается на выходной слой нейронов, но не наоборот. Такая сеть называется сетью прямого распространения или ациклической сетью. [6, с.31]

На рисунке 2.5 показана структура такой сети для случая четырех узлов в каждом из слоев. Такая нейронная сеть называется однослойной, при этом под единственным слоем подразумевается слой вычислительных элементов (нейронов). При подсчете числа слоев мы не принимаем во внимание узлы источника, так как они не выполняют никаких вычислений.

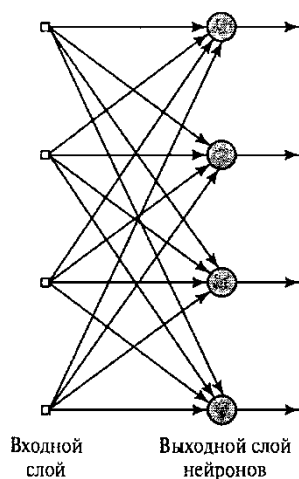


Рисунок 2.5 – Однослойная сеть прямого распространения

### **Многослойные сети прямого распространения**

Другой класс нейронных сетей прямого распространения характеризуется наличием одного или нескольких скрытых слоев, узлы которых называются скрытыми нейронами, или скрытыми элементами. Функция последних заключается в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько скрытых слоев, мы можем выделить статистики высокого порядка. Такая сеть позволяет выделять глобальные свойства данных с помощью локальных соединений за счет наличия дополнительных синаптических связей и повышения уровня взаимодействия нейронов. Способность скрытых нейронов выделять статистические зависимости высокого порядка особенно существенна, когда размер входного слоя достаточно велик. На рисунке 2.6 представлена структура многослойной сети.

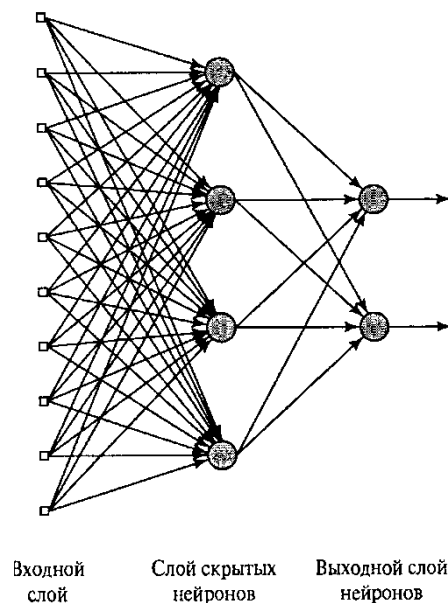


Рисунок 2.6– Многослойная сеть прямого распространения

### Рекуррентные сети

Рекуррентная нейронная сеть отличается от сети прямого распространения наличием по крайней мере одной обратной связи. Например, рекуррентная сеть, представленная на рисунке 2.7, может состоять из единственного слоя нейронов, каждый из которых направляет свой выходной сигнал на входы всех остальных нейронов слоя.

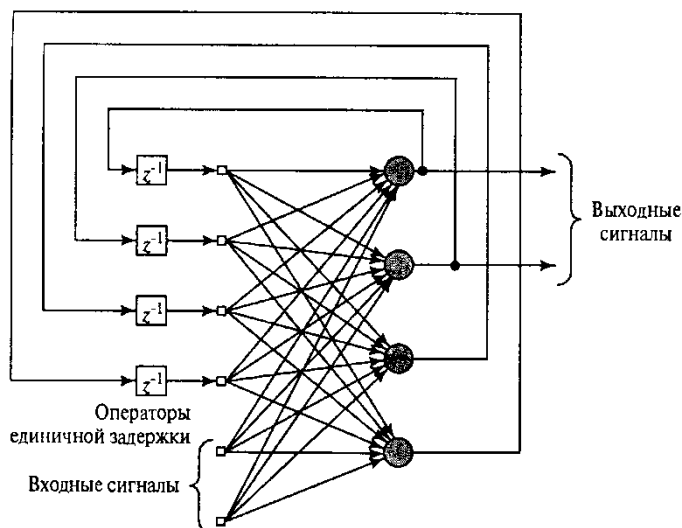


Рисунок 2.7 – Рекуррентная сеть со скрытыми нейронами

## 1.2.4 Алгоритмы обучения

### Алгоритм обратного распространения ошибки

Только в середине 80-х годов прошлого столетия впервые было введено понятие «обратного распространения ошибки» алгоритма обучения многослойного персептрона на основе градиентного метода спуска. Структурная схема многослойного персептрона представлена рисунке 2.8.

Суть алгоритма основывается на коррекции ошибок, и его можно считать обобщением метода минимизации СКО для случая линейного нейрона. Обучение персептрона происходит в два прохода: прямого и обратного.

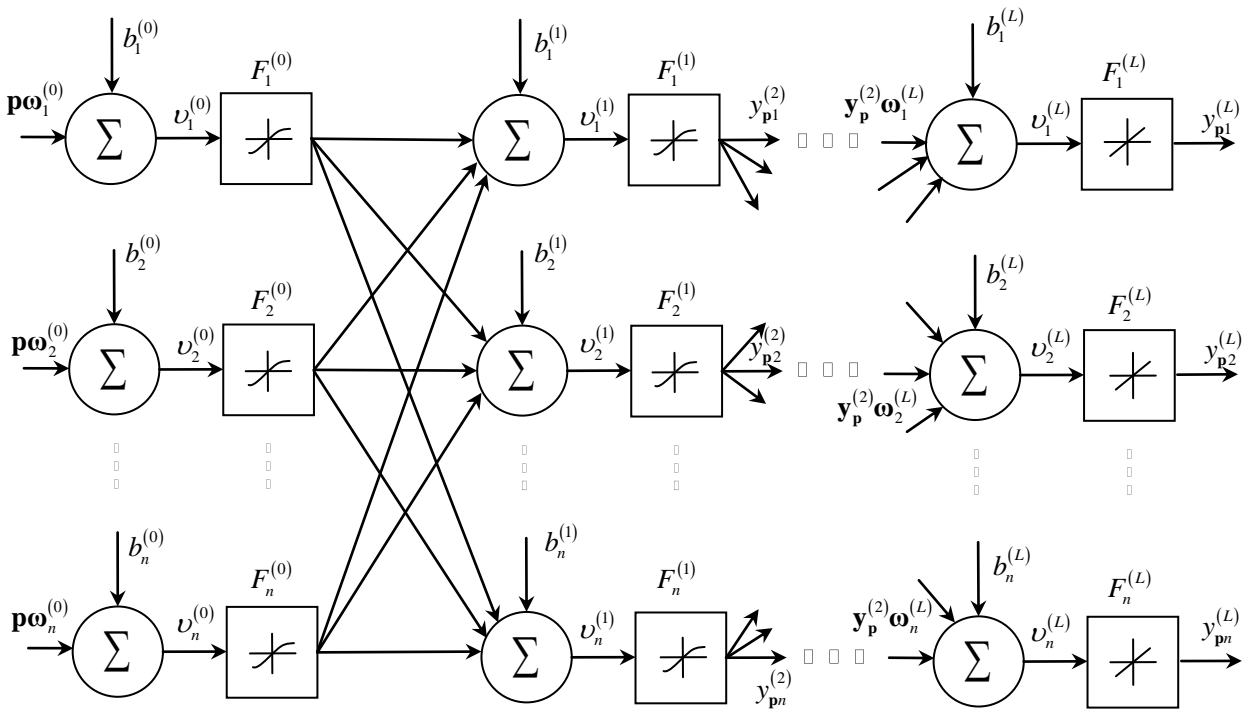


Рисунок 2.8 – Структурная схема многослойного персептрона

При прямом проходе сигнал, подаваемый на вход ИНС распространяется к выходам сети через все слои. При этом синаптические веса и смещения нейронов имеют фиксированные значения. Во время обратного прохода вычисляется сигнал ошибки, равный разности фактического и целевого (желаемого) выхода сети. Сигнал ошибки распространяется в обратном направлении, а синаптические веса и смещения корректируются.

Опишем алгоритм обратного распространения ошибки для обучения многослойной сети прямого распространения.

а) Прямой проход

Вычисляем индуцированные локальные поля и функциональные сигналы нейронов, проходя по сети послойно.

Взвешенная сумма  $j$ -го нейрона слоя  $l$  вычисляется по формуле:

$$v_j^{(l)}(k) = \sum_{i=0}^m \omega_{ji}^{(l)}(k) \cdot y_i^{(l-1)}(k), \quad (6)$$

где  $y_i^{(l-1)}(k)$  – выходной сигнал нейрона  $i$ , расположенного в предыдущем слое  $(l-1)$  на итерации  $k$ ;

$\omega_{ji}^{(l)}(k)$  – синапс нейрона  $j$  слоя  $l$  с нейроном  $i$  слоя  $(l-1)$ ;

$m$  – размер  $(l-1)$ -го слоя.

Выходной сигнал  $j$ -го нейрона с функцией активации  $F_j$ :

$$y_j^{(l-1)}(k) = F_j(v_j^{(l-1)}(k)). \quad (7)$$

При этом, если нейрон  $j$  находится в первом скрытом слое ( $l = 1$ ), то:

$$y_j^{(1)}(k) = x_j(k), \quad (8)$$

где  $x_j(k)$  –  $j$ -й элемент входного вектора  $\mathbf{x}(k)$ .

Если же, нейрон находится в выходном слое ( $l = L$ ,  $L$  – глубина сети), то выход нейрона равен  $j$ -му выходу сети. Тогда ошибка сети для  $j$ -го выхода равна:

$$e_j(k) = d_j(k) - y_j(k), \quad (9)$$

где  $d_j(k)$  – желаемый отклик  $j$ -го выхода ИНС.

б) Обратный проход

Вычисляем локальные градиенты узлов. Для нейрона  $j$  выходного слоя  $L$  имеем:

$$\delta_j^{(L)}(k) = e_j^{(L)}(k) F_j'(v_j^{(L)}(k)), \quad (10)$$

и для нейрона  $j$  скрытого слоя  $l$ :

$$\delta_j^{(l)}(k) = F_j'(v_j^{(l)}(k)) \sum_t \delta_t^{(l+1)}(k) \delta_{ij}^{(l+1)}(k), \quad (11)$$

где  $F_j'$  – производная функции  $F_j$ .

Модификация весов будет проводиться по обобщенному дельта-правилу:

$$\omega_{ij}^l(k+1) = \omega_{ij}^l(k) + \alpha [\omega_{ij}^l(k-1)] + \eta \cdot \delta_j^{(l)}(n) y_j^{(l-1)}(k), \quad (12)$$

где  $\eta$  – параметр скорости обучения;

$\alpha$  – константа момента. После выполнения этих расчетов по всем нейронам  $L$  слоев возвращаемся к п. а).

## Алгоритм обучения Левенберга-Марквардта

Алгоритм Левенберга-Марквардта, используемый для минимизации функционала ошибки описывается следующим выражением:

$$\mathbf{a}_{k+1} = \mathbf{a}_k - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e}, \quad (13)$$

где  $\mathbf{a}_{k+1}, \mathbf{a}_k$  – матрицы весов и смещений нейронов на текущем и предыдущем шаге итерации соответственно;

$\mathbf{J}$  – матрица Якоби;

$\mathbf{I}$  – единичная матрица;

$\mathbf{e}$  – вектор среднеквадратичных ошибок выходных нейронов;

$\mu$  – параметр скорости изменения градиента.

При больших значениях  $\mu$  вектор поправки  $\mathbf{a}_{k+1}$  находится ближе к антиградиенту минимизируемой функции. Наоборот, при малых значениях  $\mu$  поправка близка к вектору поправки метода Ньютона-Гаусса. Таким образом, задав на начальном этапе достаточно большое значение  $\mu$  и постепенно его уменьшая, мы получаем все преимущества градиентного метода и метода Ньютона-Гаусса (градиент быстро сходится к минимуму, когда вектор поправки лежит на достаточном большом расстоянии от минимизирующего вектора, напротив алгоритм Ньютона-Гаусса хорошо работает, когда текущее приближение лежит в непосредственной близости от оценки минимума).

Матрица Якоби рассчитывается с помощью алгоритма обратного распространения ошибки. [5, с.65]



## Алгоритм обучения Байдена-Флэтчера-Гольдфарба-Шанно

Квазиньютоновские методы обучения базируются на Ньютоновском методе аппроксимации функций, но не требуют вычисления вторых производных, что обычно вызывает большие ресурсные затраты. По аналогии с (13) получим:

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \mathbf{H}_k^{-1} \cdot \boldsymbol{\delta}_k, \quad (14)$$

где  $\boldsymbol{\delta}_k$  – вектор градиента;

$\mathbf{H}_k$  матрица вторых частных производных (матрица Гессе).

Идея квазиньютоновских методов заключается в возможности аппроксимации кривизны нелинейной оптимизируемой функции без явного формирования ее матрицы Гессе. Сама матрица при этом не хранится, а ее действие аппроксимируется скалярными произведениями специально подобранных векторов. Наиболее удачным методом из этой группы является метод Брайден-Флетчера-Гольдфарба-Шанно, согласно которому:

$$s_{k+1} = -\boldsymbol{\delta}_{k+1} + \frac{(\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k)^T \boldsymbol{\delta}_{k+1}}{(\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k)^T s_k} s_k, \quad (15)$$

где  $s_k$  – направление вдоль которого проводится оптимизация на  $k$ -м шаге;

$\boldsymbol{\delta}_{k+1}$  – вектор градиента на  $(k+1)$ -м шаге.

Для квазиньютоновских методов одним из лучших алгоритмов одномерного поиска является метод перебора с возвратами (от англ. «*BackTracking*»).

Принцип работы алгоритма следующий. Выбор шага поиска в этом направлении регулируется параметром  $\lambda$ , который первоначально

устанавливается равным 1. На первой итерации алгоритм использует значение целевой функции и ее производной, чтобы построить квадратичную аппроксимацию целевой функции вдоль направления поиска. Минимум этой аппроксимирующей функции выбирается в качестве приближения к оптимальной точке и затем оценивается функционал ошибки в этой точке. Если значение функционала недостаточно мало, то строится кубическая интерполяция и ее минимум выбирается в качестве новой оптимальной точки. Этот процесс продолжается до тех пор, пока не будет достигнуто существенное уменьшение функционала ошибки.

## 2.2 Результаты практического исследования

Neural Network Toolbox – это пакет, входящий в MATLAB. Он обеспечивает всестороннюю поддержку проектирования, обучения и моделирования множества известных сетевых парадигм, от базовых моделей перцептрона до самых современных ассоциативных и самоорганизующихся сетей. Пакет может быть использован для исследования и применения нейронных сетей к таким задачам, как обработка сигналов, нелинейное управление и финансовое моделирование.

Для создания в MATLAB нейронной сети с произвольной архитектурой нужно использовать функцию «network».

```
net = network()
```

Задать основные параметры нейронной сети можно как при вызове функции, так и после вызова.

Описание аргументов функции представлены в таблице 2.1:

Таблица 2.1 – Аргументы функции

Параметр	Описание параметра
numInputs	Количество входов
numLayers	Количество слоев
biasConnect	Логический вектор смещений numLayers-на-1
inputConnect	Логическая матрица соединений входов numLayers-на-numInputs
layerConnect	Логическая матрица соединений слоев numLayers-на-numLayers

Продолжение таблицы 2.1

Параметр	Описание параметра
outputConnect	Логический вектор соединений выходов 1-на-numLayers

После создания структуры нейронной сети, необходимо ее настроить. Функция возвращает объект нейронной сети с заданными параметрами.

Свойства, описывающие архитектуру нейронной сети представлены в таблице 2.2:

Таблица 2.2 – Свойства архитектуры сети

Параметр	Значение параметра	Описание параметра
net.numInputs	0 или положительное целое число	Количество входов.
net.numLayers	0 или положительное целое число	Количество слоев.
net.biasConnect	Логический вектор numLayer-на-1	Если net.biasConnect(i) равен 1, то у слоя под индексом i есть смещением, и net.biases{i} является структурой, описывающей это смещение.
net.inputConnect	Логический вектор numLayer-на-numInputs	Если net.inputConnect(i,j) равен 1, то слой под индексом i и вход под индексом j имеют связь с весом, структура которого описана в net.inputWeights{i,j}.
net.layerConnect	Логический вектор numLayer-на-numLayers	Если net.layerConnect(i,j) равен 1, то слой под индексом i и слой под индексом j имеют связь с весом, структура которого описана в net.layerWeights{i,j}.
net.outputConnect	Логический вектор 1-на-numLayers	Если net.outputConnect(i) равен 1, то у сети есть выход от слоя под индексом i, net.outputs{i} – структура, описывающая этот выход.
net.numOutputs	0 или положительное число (только для чтения)	Количество выходов, согласно net.outputConnect.
net.numInputDelays	0 или положительное число (только для чтения)	Максимальная входная задержка, согласно net.inputWeights{i,j}.delays.
net.numLayerDelays	0 или положительное число (только для чтения)	Максимальная задержка между слоями, согласно net.layerWeights{i,j}.delays.

Свойства, описывающие структуру нейронной сети представлены в таблице 2.3:

Таблица 2.3 – Свойства структуры сети

Параметр	Значение параметра	Описание параметра
net.inputs	Массив numInputs-на-1	net.inputs{i} есть структура, описывающая вход под индексом i.
net.layers	Массив numLayers-на-1	net.layers{i} есть структура, описывающая слой под индексом i.
net.biases	Массив numLayers-на-1	Если net.biasConnect(i) равен 1, то net.biases{i} есть структура, описывающая смещение слоя под индексом i.
net.inputWeights	Массив numLayers-на-numInputs	Если net.inputConnect(i,j) равен 1, то net.inputWeights{i,j} есть структура, описывающая весовой коэффициент связи слоя i со входом j.
net.layerWeights	Массив numLayers-by-numLayers	Если net.layerConnect(i,j) равен 1, то net.layerWeights{i,j} есть структура, описывающая весовой коэффициент связи слоя i со слоем j.
net.outputs	Массив 1-на-numLayers	Если net.outputConnect(i) равен 1, то net.outputs{i} есть структура, описывающая выход под индексом i.

Функции, использующие в нейросети представлены в таблице 2.4.

Таблица 2.4 – Функции сети

Параметр	Описание параметра
net.adaptFcn	Название функции адаптации сети или пустая строка
net.initFcn	Название функции инициализации сети или пустая строка
net.performFcn	Название функции производительности сети или пустая строка
net.trainFcn	Название функции тренировки сети или пустая строка

Параметры, использующие в нейросети представлены в таблице 2.5.

Таблица 2.5 – Параметры сети

Параметр	Описание параметра
net.adaptParam	Параметры адаптации сети
net.initParam	Параметры инициализации сети
net.performParam	Параметры производительности сети
net.trainParam	Параметры тренировки сети

Параметры весов и смещений представлены в таблице 2.6

Таблица 2.6 – Весы и смещения

Параметр	Описание параметра
net.IW	Массив numLayers-на-numInputs весовых коэффициентов входных связей
net.LW	Массив numLayers-на-numLayers весовых коэффициентов связей между слоями
net.b	Массив numLayers-на-1 смещений

На рисунке 2.9 представлена архитектура нейросети.

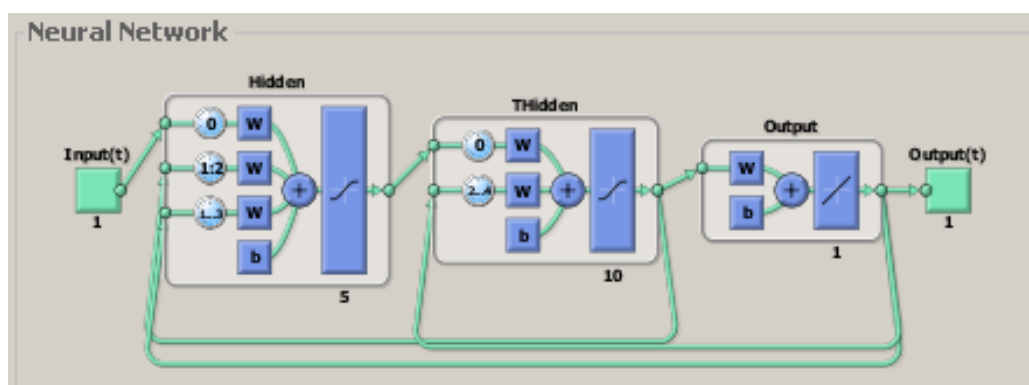


Рисунок 2.9 – Архитектура нейросети

После создания сети идет подготовка данных для обучения.

Обучение будет происходить по 100 выборок.

Перед тем, как приступить к обучению, сигналы были про нормированы.

PRESTD преобразует обучающий набор данных посредством нормирования входов и эталонов таким образом, чтобы их среднее значение было равно нулю, а стандартное отклонение равно один.

Обучение нейронной сети запускается с помощью функции train.

В рассматриваемом примере обучение производится методом Левенберга-Маркара.

```
[net, tr] = train(net, pn, tn);
```

На рисунке 2.10 представлен процесс обучения.

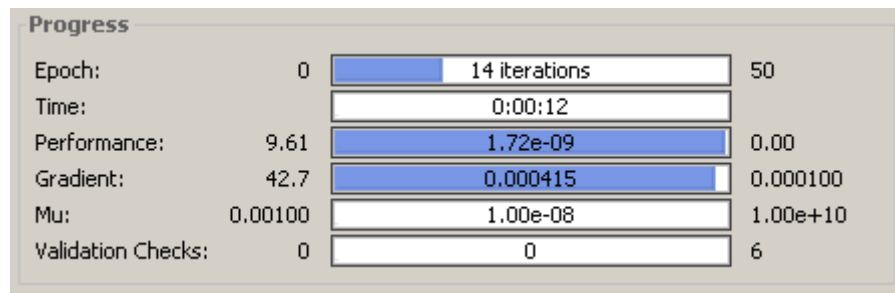


Рисунок 2.10 – Обучение нейросети

Для моделирования нейронной сети в MATLAB используется функция `sim`.

Так как при обучении использовались нормированные значения, то после моделирования полученные значения необходимо восстановить. Восстановление значений происходит с помощью функции `POSTSTD`.

Результат моделирования представлен на рисунке 2.11. Тонкой линией показан реальный сигнал, толстой – предсказанный.

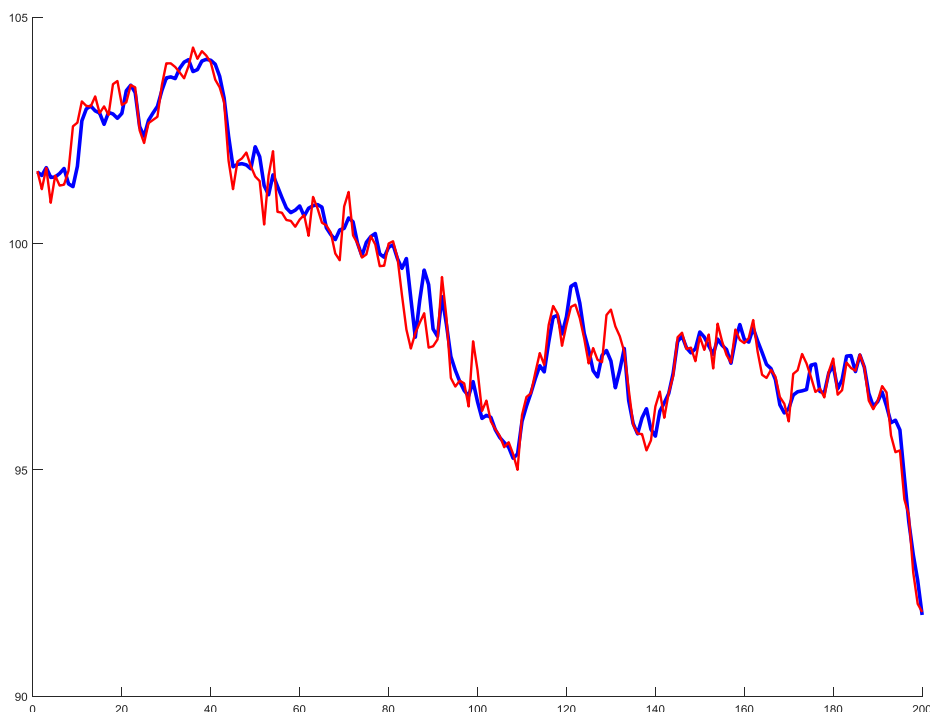


Рисунок 2.11 – Результат моделирования

### 3 Нейросетевой экстраполятор с использование генетического алгоритма при обучении

Для моделирования возьмём сигнал с 1000 отсчетов. График сигнала представлен на рисунке 3.1.

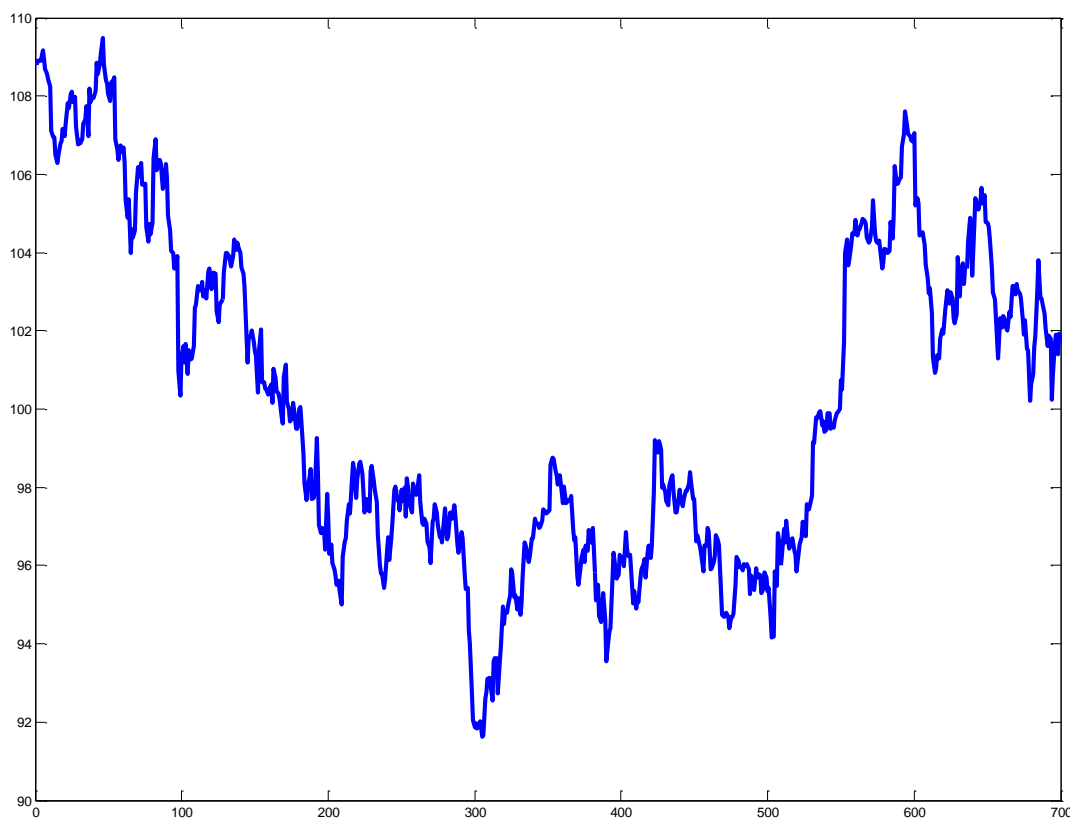


Рисунок 3.1 – Сигнал для исследования системы

Разделим сигнал на две составляющие:

- обучающая выборка. В обучающую выборку будут входить первые 700 отсчетов.
- тестовая выборка. В тестовую выборку будут входить 300 следующих отсчетов.

Структура искусственной нейронной сети представлена на рисунке 3.2.

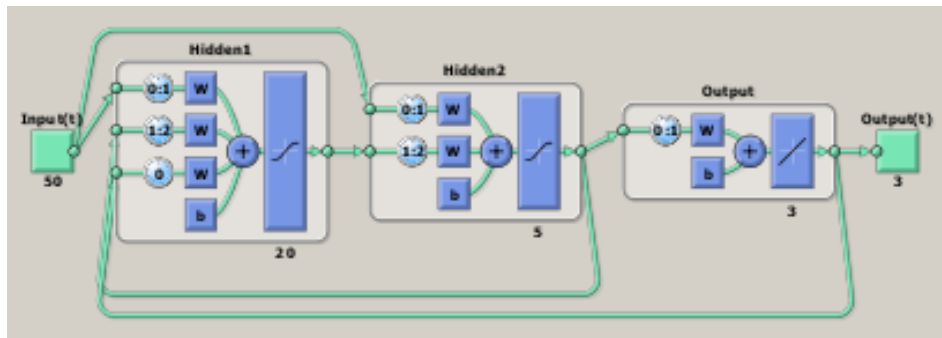


Рисунок 3.2 – Структура искусственной нейронной сети

Искусственная нейронная сеть состоит из двух внутренних слоёв и одного выходного слоя. На вход подаются 50 выборок из сигнала, на выходе получает три предсказанные выборки. Предсказанные выборки получаются путём умножения входных выборок на весовые коэффициенты, суммирования всех выборок и прогон через функцию активации.

Обучение искусственной нейронной сети происходит при помощи генетического алгоритма. Генетический алгоритм задаёт начальные условия весовых коэффициентов искусственной нейронной сети, вычисляет значение выборок на выходе искусственной нейронной сети. Заданная фитнес-функция сравнивает истинное значение выборок с вычисленным и на выходе генетического алгоритма мы получаем ошибку.

Задача генетического алгоритма состоит в нахождении минимальной ошибки и значений весовых коэффициентов соответствующих данной ошибке. Процесс обучения представлен на рисунке 3.3.



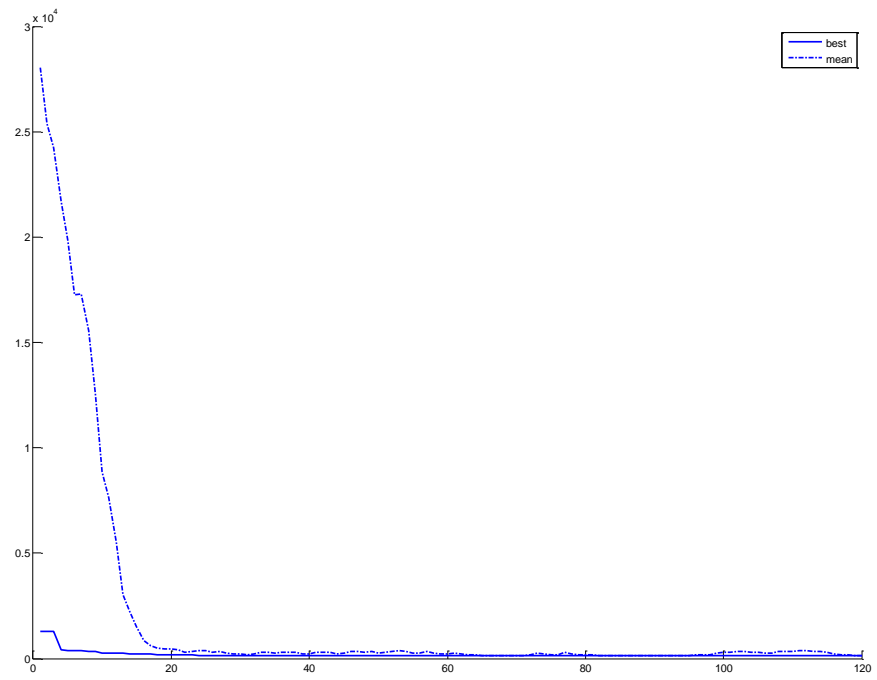


Рисунок 3.3 – Оптимизация весовых коэффициентов

Подадим на вход обученной искусственной нейронной сети тестовые 300 отсчетов. Результаты моделирование отображены на рисунке 3.4.

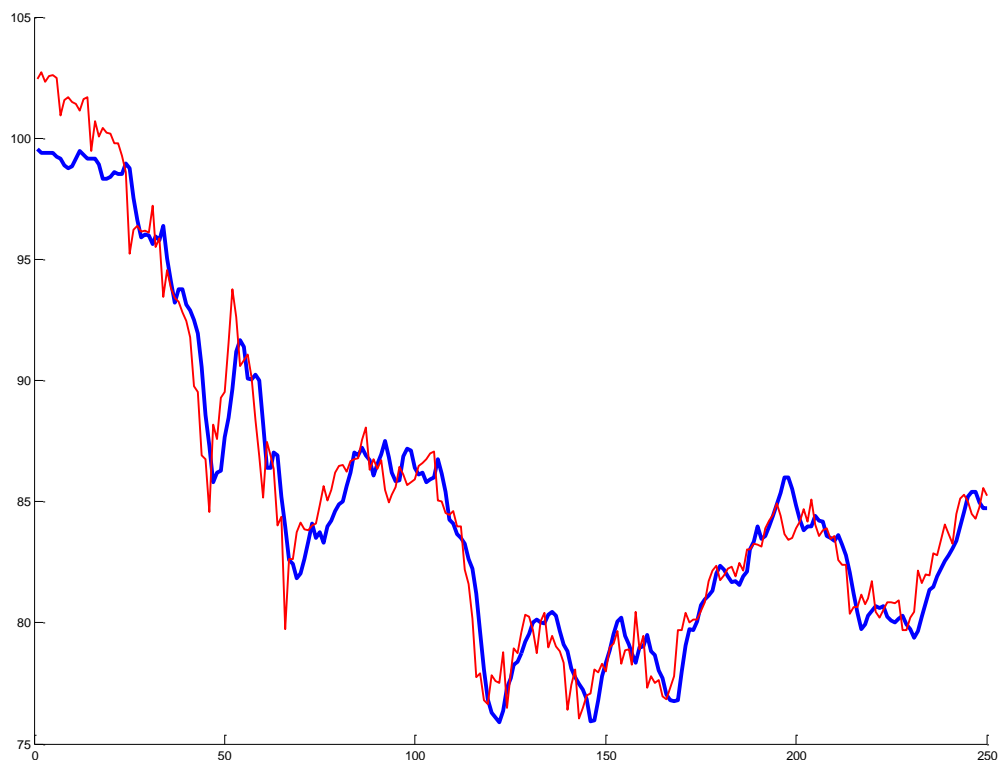


Рисунок 3.4 – Результаты тестового моделирования

Среднеквадратичная ошибка между рассчитанным и истинным сигналом равна

$$AbError = 8,2358.$$

Анализ показывает, что ни один из существующих методов не может дать достаточной точности прогнозов на большое время. Применяемый в прогнозировании метод экстраполяции не дает точных результатов на долгосрочный прогноз, потому что данный метод исходит из прошлого и настоящего, и тем самым погрешность накапливается. Этот метод дает положительные результаты на короткосрочную перспективу прогнозирования тех или иных объектов.

#### 4 Организационно-экономический раздел

**Сетевое планирование** – метод управления, основанный на использовании математического аппарата теории графов и системного подхода для отображения и алгоритмизации комплексов взаимосвязанных работ, действий или мероприятий для достижения четко поставленной цели.

Разработаны в начале 50-х г. XX в.

Наиболее известны практически одновременно и независимо разработанные метод критического пути и метод оценки и пересмотра планов.

Применяются для оптимизации планирования и управления сложными разветвленными комплексами работ, требующими участия большого числа исполнителей и затрат ограниченных ресурсов.

Основная цель сетевого планирования - сокращение до минимума продолжительности проекта.

Задача сетевого планирования состоит в том, чтобы графически, наглядно и системно, отобразить и оптимизировать последовательность и взаимозависимость работ, действий или мероприятий, обеспечивающих своевременное и планомерное достижение конечных целей. Для отображения и алгоритмизации тех или иных действий или ситуаций используются экономико-математические модели, которые принято называть сетевыми моделями, простейшие из них - сетевые графики. С помощью сетевой модели руководитель работ или операции имеет возможность системно и масштабно представлять весь ход работ или оперативных мероприятий, управлять процессом их осуществления, а также маневрировать ресурсами.

Наиболее распространенными направлениями применения сетевого планирования являются:

- целевые научно-исследовательские и проектно-конструкторские разработки сложных объектов, машин и установок, в создании которых принимают участие многие предприятия и организации;
- планирование и управление основной деятельностью разрабатывающих организаций;
- планирование комплекса работ по подготовке и освоению производства новых видов промышленной продукции;
- строительство и монтаж объектов промышленного, культурно-бытового и жилищного назначения;
- реконструкция и ремонт действующих промышленных и других объектов;
- планирование подготовки и переподготовки кадров, проверка исполнения принятых решений, организация комплексной проверки деятельности предприятий, объединений, строительного-монтажных организаций и учреждений.

Использование методов сетевого планирования способствует сокращению сроков создания новых объектов на 15-20%, обеспечению рационального использования трудовых ресурсов и техники.

#### 4.1 Составление и расчет параметров сетевого графика

Комплекс работ СГ включает перечень работ.

1. Получение технического задания.
2. Анализ технического задания.
3. Подбор литературы по генетическому алгоритму.
4. Подбор литературы по нейросетям.
5. Анализ литературы по генетическому алгоритму.

6. Анализ литературы по нейросетям.
7. Выявление аналогичных разработок программ.
8. Анализ и выбор аналога.
9. Ознакомление со структурной схемой аналога.
10. Разработка структурной схемы.
11. Подбор литературы по БЖД.
12. Анализ литературы по БЖД.
13. Разработка мероприятий по БЖД.
14. Написание кода генетического алгоритма.
15. Проверка и анализ работы кода генетического алгоритма.
16. Написание кода нейросети.
17. Проверка и анализ работы кода нейросети.
18. Согласование с преподавателем работы кода генетического алгоритма по результатам анализа.
19. Доработка кода генетического алгоритма в соответствии с предложениями преподавателя.
20. Согласование с преподавателем работы кода нейросети по результатам анализа.
21. Доработка кода нейросети в соответствии с предложениями преподавателя.
22. Объединение кода нейросети и генетического алгоритма.
23. Оптимизация кода нейросетевого экстраполятора.
24. Разработка сетевого графика.
25. Анализ работы нейросетевого экстраполятора.
26. Оформление ПЗ.
27. Защита работы.

Коэффициент сложности СГ должен быть не меньше 1,2.

Нумерация событий ведётся по возрастанию номера слева направо и сверху вниз; исходное событие имеет нулевой номер, а завершающее - наибольший.

Код работы составляется из номеров начального и конечного событий этой работы, отделяемых запятой (i, j), причём для всех работ номер начального события i должен быть меньше номера конечного события j.

Рассчитаем коэффициент сложности.

$$k=27/22=1,2$$

Построенный сетевой график показан на рисунке 4.1.

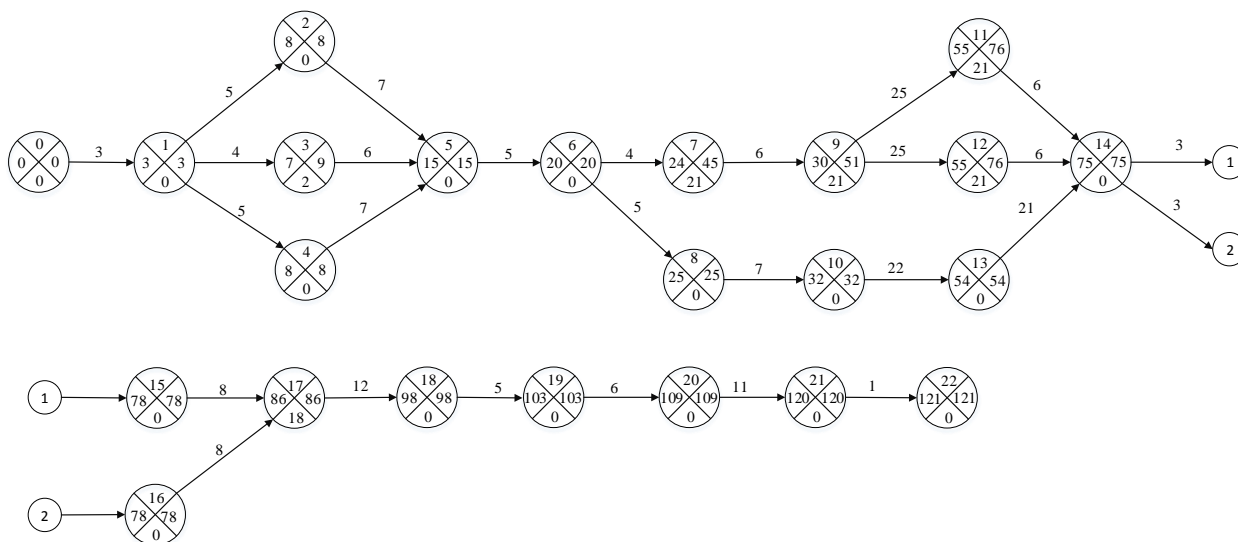


Рисунок 4.1 – Сетевой график

#### 4.1.1 Перечень, параметры и вероятностные характеристики работ СГ

Ожидаемая продолжительность каждой работы определяется по формуле

$$t_{ож} = 0,6 t_{мин} + 0,4 t_{макс}.$$

Среднеквадратическое отклонение  $\delta$  продолжительности в двухоченочной методике рассчитывается по формуле

$$\delta = 0,2 (t_{макс} - t_{мин}).$$

Дисперсия определяется по формуле

$$D = 0,04 (t_{макс} - t_{мин})^2.$$

Перечень, параметры и вероятностные характеристики работ сетевого графика представлены в таблице 4.1.

Таблица 4.1 – Перечень, параметры и вероятностные характеристики работ СГ

Код работы	Наименование работы	Продолжительность, дн			Исполнители, чел.			Среднеквадратическое отклонение, дн	Дисперсия, дн <sup>2</sup>
		мин.	макс.	ожид.	рук.	инж.	лаб.		
0,1	Получение технического задания	2	4	3	1	2	0	0,4	0,16
1,2	Подбор литературы по генетическому алгоритму	3	7	5	0	0	1	0,8	0,64
1,3	Анализ технического задания	3	6	4	0	2	0	0,6	0,36
1,4	Подбор литературы по нейросетям	3	7	5	0	0	1	0,8	0,64
2,5	Анализ литературы по генетическому алгоритму	5	10	7	0	1	0	1,0	1,00
3,5	Выявление аналогичных разработок программ	4	8	6	0	0	1	0,8	0,64
4,5	Анализ литературы по нейросетям	5	10	7	0	1	0	1,0	1,00
5,6	Анализ и выбор аналога	3	7	5	0	1	0	0,8	0,64
6,7	Ознакомление со структурной схемой аналога	3	6	4	0	1	0	0,6	0,36
6,8	Подбор литературы по БЖД	3	7	5	0	0	1	0,8	0,64
7,9	Разработка структурной схемы	5	8	6	0	1	0	0,6	0,36
8,10	Анализ литературы по БЖД	5	10	7	0	1	0	1,0	1,00
9,11	Написание кода нейросети	21	30	25	0	1	0	1,8	3,24

Продолжение таблицы 4.1

Код работы	Наименование работы	Продолжительность, дн			Исполнители, чел.			Среднеквадратическое отклонение, дн	Дисперсия, дн <sup>2</sup>
		мин	макс	ожд	рук	инж	лаб		
9,12	Написание кода генетического алгоритма	21	30	25	0	1	0	1,8	3,24
10,13	Разработка мероприятий по БЖД	18	29	22	0	2	1	2,2	4,84
11,14	Проверка и анализ работы кода нейросети	4	10	6	0	1	0	1,2	1,44
12,14	Проверка и анализ работы кода генетического алгоритма	4	10	6	0	1	0	1,2	1,44
13,14	Разработка сетевого графика	16	28	21	0	1	1	2,4	5,76
14,15	Согласование с преподавателем работы кода генетического алгоритма по результатам анализа	2	5	3	1	1	0	0,6	0,36
14,16	Согласование с преподавателем работы кода нейросети по результатам анализа	2	5	3	1	1	0	0,6	0,36
15,17	Доработка кода генетического алгоритма в соответствии с предложениями преподавателя	6	11	8	0	1	0	1,0	1,00
16,17	Доработка кода нейросети в соответствии с предложениями преподавателя	6	11	8	0	1	0	1,0	1,00
17,18	Объединение кода нейросети и генетического	10	16	12	0	2	0	1,2	1,44



	алгоритма							
--	-----------	--	--	--	--	--	--	--

#### Окончание таблицы 4.1

Код работы	Наименование работы	Продолжительность, дн			Исполнители, чел.			Среднеквадратическое отклонение, дн	Дисперсия, дн <sup>2</sup>
		мин	макс	ожид	рук	инж	лаб		
18,19	Оптимизация кода нейросетевого экстраполятора	4	6	5	0	2	0	0,4	0,16
19,20	Анализ работы нейросетевого экстраполятора	5	8	6	0	2	0	0,6	0,36
20,21	Оформление ПЗ	8	15	11	1	2	0	1,4	1,96
21,22	Защита работы	1	1	1	1	2	0	0	0

#### 4.1.2 Расчет параметров событий сетевого графика

Параметры событий сетевого графика представлены в таблице 4.2.

Таблица 4.2 - Параметры событий сетевого графика

Номер события	Ранний срок свершения	Поздний срок свершения	Резерв времени
0	0	0	0
1	3	3	0
2	8	8	0
3	7	9	2
4	8	8	0
5	15	15	0
6	20	20	0
7	24	45	21
8	25	25	0
9	30	51	21
10	32	32	0
11	55	76	21
12	55	76	21
13	54	54	0
14	75	75	0
15	78	78	0
16	78	78	0
17	86	86	0
18	98	98	0

19	103	103	0
----	-----	-----	---

Продолжение таблицы 4.2

Номер события	Ранний срок свершения	Поздний срок свершения	Резерв времени
20	109	109	0
21	120	120	0
22	121	121	0

#### 4.1.3 Расчёт параметров работ сетевого графика

Работы, не лежащие на критическом пути, обладают резервами времени.

Полный резерв времени работы  $R_{pij}$  образуется вычитанием из позднего срока свершения её конечного события раннего срока свершения её начального события и её ожидаемой продолжительности.

Частный резерв времени первого рода  $R^1_{pij}$  равен разности поздних сроков свершения её конечного и начального событий за вычетом её ожидаемой продолжительности.

Частный резерв времени второго рода  $R^2_{pij}$  равен разности ранних сроков свершения её конечного и начального событий за вычетом её ожидаемой продолжительности.

Свободный (независимый) резерв времени работы  $R_{cij}$  образуется вычитанием из раннего срока свершения её конечного события позднего срока свершения её начального события и её ожидаемой продолжительности. Свободный резерв времени может быть отрицательным.

Для работ, лежащих на критическом пути никаких резервов времени нет и, следовательно, коэффициент напряжённости  $K_{nij}$  таких работ равен единице. Если работа не лежит на критическом пути, она располагает резервами времени и её коэффициент напряжённости меньше единицы. Его величина подсчитывается как отношение суммы продолжительностей отрезков максимального пути, проходящего через данную работу, не совпадающих с критическим путём  $t_{\max ij}$  к сумме продолжительностей отрезков критического пути, не совпадающих с максимальным путём,

проходящим через эту работу ткрїј. Параметры работ сетевого графика представлены в таблице 4.3.

Таблица 4.3 - Параметры работ сетевого графика

Код работы	Ожидаемая продолжительность	Сроки начала		Сроки окончания		Резервы времени				Коэффициент напряжённости
		ранний	поздний	ранний	поздний	полный	частный 1 рода	частный 2 рода	свободный	
0,1	3	0	0	3	3	0	0	0	0	1,00
1,2	5	3	3	8	8	0	0	0	0	1,00
1,3	4	3	5	7	9	2	2	0	0	0,83
1,4	5	3	3	8	8	0	0	0	0	1,00
2,5	7	8	8	15	15	0	0	0	0	1,00
3,5	6	7	9	13	15	2	0	2	0	0,83
4,5	7	8	8	15	15	0	0	0	0	1,00
5,6	5	15	15	20	20	0	0	0	0	1,00
6,7	4	20	41	24	45	21	21	0	0	0,74
6,8	5	20	20	25	25	0	0	0	0	1,00
7,9	6	24	45	30	51	21	0	0	-21	0,74
8,1	7	25	25	32	32	0	0	0	0	1,00
9,11	25	30	51	55	76	21	0	0	-21	0,74
9,12	25	30	51	55	76	21	0	0	-21	0,74
10,13	22	32	32	54	54	0	0	0	0	1,00
11,14	6	55	69	61	75	14	0	21	0	0,74
12,14	6	55	69	61	75	14	0	21	0	0,74
13,14	21	54	54	75	75	0	0	0	0	1,00
14,15	3	75	75	78	78	0	0	0	0	1,00
14,16	3	75	75	78	78	0	0	0	0	1,00
15,17	8	78	78	86	86	0	0	0	0	1,00
16,17	8	78	78	86	86	0	0	0	0	1,00
17,18	12	86	86	98	98	0	0	0	0	1,00
18,19	5	98	98	103	103	0	0	0	0	1,00
19,2	6	103	103	109	109	0	0	0	0	1,00
20,21	11	109	109	120	120	0	0	0	0	1,00
21,22	1	120	120	121	121	0	0	0	0	1,00

#### 4.1.4 Расчёт параметров сетевого графика в целом

Среднеквадратическое отклонение продолжительности критического пути определяется по формуле:

$$\delta_{кр} = \sqrt{D_{кр}}, \quad (16)$$

где  $D_{кр}$  - дисперсия срока наступления завершающего события, равная сумме дисперсий работ критического пути. Вероятность свершения завершающего события в директивный срок представлена в таблице 4.4.

Таблица 4.4 – Вероятность свершения завершающего события в директивный срок

Наименование показателя	Значение показателя
Критический срок	121
Среднеквадратическое отклонение	4,791659
Директивный срок	122
Нормальное отклонение	0,208696
Вероятность свершения	0,582657

## 4.2 Расчёт стоимостных параметров сетевого графика

### 4.2.1 Расчёт материальных затрат

В таблице 4.5 записаны все материалы необходимые для проектирования программного продукта.

Таблица 4.5 – Материальные затраты

Наименование	Количество, шт	Цена, руб	Стоимость, руб
Erich Krause Ручка шариковая R-301 Orange синяя	3	10	30
Тетрадь 48 листов	1	29	29
Итого			59

### 4.2.2 Расчёт стоимости контрагентских услуг

Контрагентские расходы включают стоимость работ, выполняемых сторонними организациями и предприятиями. Контрагентские расходы представлены в таблице 4.6.

Таблица 4.6 – Контрагентские услуги

Наименование	Количество, шт	Цена, руб	Стоимость, руб
Печать А4	62	2	124
Печать А1	5	160	800
Итого			924

#### 4.2.3 Расчет стоимости покупных изделий

Покупные изделия включают в себя лицензии на продукты. Покупные изделия представлены в таблице 4.7.

Таблица 4.7 – Покупные изделия

	Количество, шт	Цена, руб	Стоимость, руб
MatLab	1	5 637	5 637
Symbolic Math Toolbox	1	1 726	1 726
Итого			7 363

#### 4.2.4 Расчет стоимости заработной платы

В таблице 4.8 записаны расходы потраченные на заработную плату

Таблица 4.8 – Заработная плата

Сотрудник	Занятость, дн	Оклад, руб/мес	Средняя з/п, руб
Руководитель(1)	21	50 000	37 500
Инженер(2)	273	30 000	292 500
Лаборант(1)	64	10 000	22 857
Итого			352 857
Итого + начисления на фонд оплаты труда 30%			458 714

#### 4.2.5 Расчет стоимости накладных расходов

Накладные расходы связаны с затратами на управление и хозяйственное обслуживание, на них мы выделяем 10% от суммы по предыдущим пунктам - 46 706 рублей.

#### 4.2.6 Расчет стоимости полной сметы затрат

Рассчитаем полную смету затрат, сложив все числа из предыдущих пунктов. Полная смета представлена в таблице 4.9.

Таблица 4.9 – Полная смета затрат

Наименование	Расходы, руб
Смета затрат на материалы	59
Контрагентские	924
Покупные изделия	7 363
Заработная плата	458 714
Накладные расходы	46 706
Итого	513 766

Для проектирования программного продукта нам необходимо затратить 513766 рублей.

#### 4.3 Анализ технико-экономической эффективности

Для оценки экономической эффективности проектируемой деятельности были определены следующие показатели:

- временные показатели (параметры и вероятностные характеристики работ и событий СГ);
- экономические показатели (материальные затраты, контрагентские услуги, покупные изделия, заработная плата, накладные расходы);

Расчет экономических показателей показывает, что на проектирование программного продукта необходимо затратить 513 766 рублей.

В случае, если бы мы заказывали проектирование программного продукта у сторонней фирмы, то на проектирование потребовалось бы:

$121 \cdot 8 = 968$  часов

Для расчетов примем часовую норму оплаты труда – 1500 рублей в час.

$968 \cdot 1500 = 1452000$  рублей.

Анализ этих показателей демонстрирует, что реализация запланированных мероприятий, направленных на совершенствование организации технологического процесса, способно привести к уменьшению затрат на 948234 рублей.

## 5 Безопасность жизнедеятельности

Научно-технический прогресс внес серьезные изменения в условия производственной деятельности работников умственного труда. Их труд стал более интенсивным, напряженным, требующим значительных затрат умственной, эмоциональной и физической энергии. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы человека-оператора.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

### 5.1 Требования к помещениям для работы с ПЭВМ

В помещении, естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электронно-лучевой трубки (ЭЛТ) должна составлять не менее 6 м<sup>2</sup>, в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м<sup>2</sup>.



При использовании ПВЭМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее четырёх часов в день допускается минимальная площадь 4,5 м на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

Для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно отражающие материалы с коэффициентом отражения для потолка - 0,7...0,8; для стен - 0,5...0,6; для пола - 0,3...0,5. Полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

Не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ.

## 5.2 Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе

В производственных помещениях, в которых работа с использованием ПЭВМ является вспомогательной, температура, относительная влажность и скорость движения воздуха на рабочих местах должны быть в соответствии с СанПин 2.2.4.3359-16. [8]

В производственных помещениях, в которых работа с использованием ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники и др.) и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные

параметры микроклимата для категории работ 1а и 1б в соответствии с СанПин 2.2.4.3359-16. [8]

Оптимальные параметры микроклимата представлены в таблице 5.1.

Таблица 5.1 - Оптимальные величины параметров микроклимата на рабочих местах производственных помещений

Период года	Категория работ по уровням энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с, не более
Холодный	1а (до 139)	22 ... 24	21 ... 25	60 ... 40	0,1
	1б (140 ... 174)	21 ... 23	20 ... 24	60 ... 40	0,1
Теплый	1а (до 139)	23 ... 25	22 ... 26	60 ... 40	0,1
	1б (140 ... 174)	22 ... 24	21 ... 25	60 ... 40	0,1

На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, в соответствии с СанПин 2.2.4.3359-16. [8]

В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ЭВМ.

Уровни положительных и отрицательных аэроионов в воздухе помещений, где расположены ПЭВМ, должны соответствовать СанПин 2.2.4.1294-03. [9]

Содержание вредных химических веществ в воздухе производственных помещений, в которых работа с использованием ПЭВМ является вспомогательной, не должно превышать предельно допустимых концентраций вредных веществ в воздухе рабочей зоны в соответствии ГН 2.2.5.1313-03. [10]

Содержание вредных химических веществ в производственных помещениях, в которых работа с использованием ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники и др.), не должно превышать предельно допустимых концентраций загрязняющих веществ в атмосферном воздухе

населенных мест в соответствии с действующими гигиеническими нормативами.

Содержание вредных химических веществ в воздухе помещений, предназначенных для использования ПЭВМ во всех типах образовательных учреждений, не должно превышать предельно допустимых среднесуточных концентраций для атмосферного воздуха в соответствии с действующими санитарно-эпидемиологическими нормативами.

### 5.3 Требования к уровням шума и вибрации

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать 80 дБА согласно СанПин 2.2.4.3359-16. [8]

В помещениях всех образовательных и культурно-развлекательных учреждений для детей и подростков, где расположены ПЭВМ, уровни шума не должны превышать допустимых значений, установленных для жилых и общественных зданий.

При выполнении работ с использованием ПЭВМ в производственных помещениях уровень вибрации не должен превышать допустимых значений вибрации для рабочих мест (категория 3, тип "в") в соответствии с СанПин 2.2.4.3359-16. [8]

В помещениях всех типов образовательных и культурно-развлекательных учреждений, в которых эксплуатируются ПЭВМ, уровень вибрации не должен превышать допустимых значений для жилых и общественных зданий в соответствии с действующими санитарно-эпидемиологическими нормативами.

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ.

#### 5.4 Требования к освещению на рабочих местах

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300...500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать прямую блесккость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м.

Следует ограничивать отраженную блесккость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м и яркость потолка не должна превышать 200 кд/м.

Показатель ослепленности для источников общего искусственного освещения в производственных помещениях должен быть не более 20. Показатель дискомфорта в административно-общественных помещениях не более 40, в дошкольных и учебных помещениях не более 15.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90° с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м, защитный угол светильников должен быть не менее 40°.

Светильники местного освещения должны иметь непросвечивающий отражатель с защитным углом не менее 40°.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

Допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору.

Коэффициент запаса ( $K_z$ ) для осветительных установок общего освещения должен приниматься равным 1,4.

Коэффициент пульсации не должен превышать 5%.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

## 5.5 Требования к уровням электромагнитных полей

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей, а также в помещениях образовательных, дошкольных и культурно-развлекательных учреждений, представлены таблице 5.2.

Таблица 5.2 – Допустимые уровни ЭМП

Наименование параметров		ВДУ
Напряженность электрического поля	в диапазоне частот 5 Гц ... 2 кГц	25 В/м
	в диапазоне частот 2 кГц ... 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц ... 2 кГц	250 нТл
	в диапазоне частот 2 кГц ... 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

## 5.6 Общие требования к организации рабочих мест пользователей ПЭВМ

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5...2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600...700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом

его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5...0,7.

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680...800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

- Конструкция рабочего стула должна обеспечивать:
- ширину и глубину поверхности сиденья не менее 400 мм;
  - поверхность сиденья с закругленным передним краем;
  - регулировку высоты поверхности сиденья в пределах 400...550 мм и углам наклона вперед до 15° и назад до 5°;
  - высоту опорной поверхности спинки 300±20 мм, ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм;
  - угол наклона спинки в вертикальной плоскости в пределах ±30°;
  - регулировку расстояния спинки от переднего края сиденья в пределах 260...400 мм;

– стационарные или съемные подлокотники длиной не менее 250 мм и шириной – 50...70 мм;

– регулировку подлокотников по высоте над сиденьем в пределах  $230\pm 30$  мм и внутреннего расстояния между подлокотниками в пределах 350...500 мм.

Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до  $20^\circ$ . Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100...300 мм от края, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

## 5.7 Анализ собственного рабочего места

Комната имеет площадь  $12\text{ м}^2$ , что является достаточным для размещения одного рабочего места. Окно имеет направление на юго-запад и



оборудованы занавесями. В системе электроснабжения нет заземляющего проводника, поэтому ПЭВМ не имеет защитного заземления.

Влажная уборка делается каждую неделю, проветривание каждый день по 4 раза в холодное время года.

Источниками шума являются ПЭВМ и принтер.

Искусственное освещение исполнено в виде люстры в центре помещения с шестью лампами накаливания мощностью 40 Вт. Так же на столе имеется светильник с непросвечивающим отражателем.

Рабочий стол стоит около окна. Высота стола 700 мм, глубина 800 мм.

На столе располагается монитор, размещённый левой стороной к окну. Расстояние до монитора 600 мм.

Клавиатура расположена на расстоянии 200 мм от края столешницы.

Рабочий стул имеет высоту 500 мм, подлокотником нету. Ширина и глубина поверхности сиденья около нормы.

Схема расположения вещей представлена на рисунке 5.1, где

1 – окно, 2– принтер, 3 – монитор, 4– настольная лампа, 5 – клавиатура, 6 – компьютерная мышь, 7– стул, 8 – шкаф для книг и бумаг, 9 – дверь.

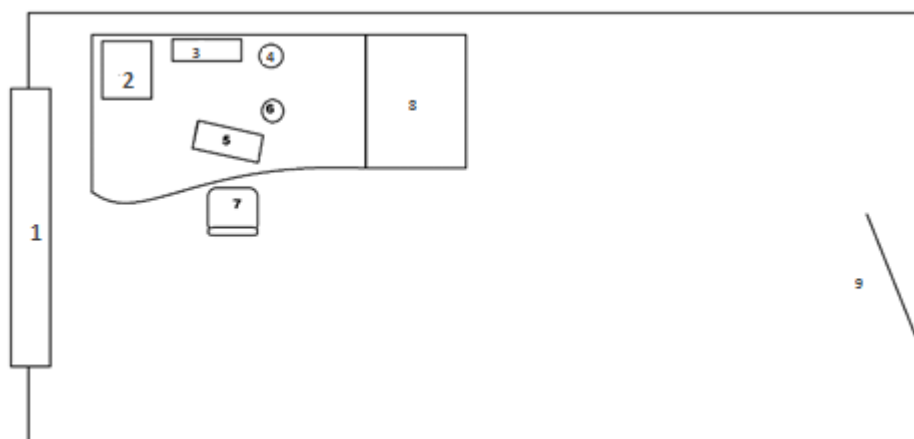


Рисунок 5.1 – Расположение вещей на рабочем месте

## ЗАКЛЮЧЕНИЕ

В ходе данной дипломной работе была смоделирована искусственная нейронная сеть в среде для программирования MatLab и исследован метод оптимизации, генетический алгоритм в целях обучения нейронной сети. Были сформированы обучающий сигнал и сигнал для тестирования системы.

Тестирование нейронной сети показало, что применяемый в прогнозировании метод экстраполяции не дает точных результатов на долгосрочный прогноз, потому что данный метод исходит из прошлого и настоящего, и тем самым погрешность накапливается. Этот метод дает положительные результаты на краткосрочную перспективу прогнозирования тех или иных объектов. Среднеквадратическая ошибка равна 8,24

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с пол. И.Д. Рудинского. – М.: Горячая линия–Телеком, 2004. – 383 с.
- 2) Мищенко В.А., Коробкин А.А. Использование генетических алгоритмов в обучении нейронных сетей // Современные проблемы науки и образования. – 2011. – № 6.;  
URL: <https://www.science-education.ru/ru/article/view?id=5138>
- 3) Гольдштейн А.Л. «Оптимизация в среде MATLAB»/ уч. пособие – Пермь: Издательство Пермского национального исследовательского политехнического университета. – 2015. - 192с.
- 4) Хайкин, Саймон Нейронные сети: полный курс 2-е издание / Саймон, Хайкин . – Москва: Изд-во «Вильямс», 2006. – 1104 с.
- 5) Осовский, С. Нейронные сети для обработки информации/ С. Осовский. – М: Изд-во Финансы и статистика, 2002. – 344 с.
- 6) Рудаков, А.С. Подходы к решению задачи прогнозирования временных рядов с помощью нейронных сетей / А.С. Рудаков // Бизнес-информатика. – 2008. – Т. 6, № 4. – С. 29–34
- 7) СанПиН 2.2.4.3359-16 «Санитарно-эпидемиологические требования к физическим факторам на рабочих местах»
- 8) СанПиН 2.2.4.1294-03 «гигиенические требования к аэроионному составу воздуха производственных и общественных помещений»
- 9) ГН 2.2.5.1313-03 «Предельно Допустимые концентрации вредных веществ в воздухе рабочей зоны»

## ПРИЛОЖЕНИЕ А

Листинг программы:

Main:

```
clc, clear all,
tic
% BlockFormINPUT;
% net = timedelaynet(Delay1:Delay2,AmountNeuro);
% net.inputs{1}.size = sizeIn;
% net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
% net.layers{2}.size = sizeOut;
% razmOb = 250;
% maxi = dl/razmOb;
% reshapeMas = zeros(sizeIn,razmOb);
% reshapeMaxIN = reshape(In2, sizeIn,razmOb,20);
% reshapeMaxTAR = reshape(tar, sizeOut,razmOb,20);
% EWW = @(x) trainn(x, net, reshapeMaxIN(:,:,1),reshapeMaxTAR(:,:,1));
% ga_opts = gaoptimset('TolFun', 1e-4,'display','iter','generations',30);
% [x_ga_opt(1,:), err_ga(1,:)] = ga(EWW,3*AmountNeuro+sizeIn+sizeOut,ga_opts);
% net = setwb(net,x_ga_opt(1,:));
% for i = 2:20
%     x = x_ga_opt(i-1,:);
%     EWW = @(x) trainn(x, net, reshapeMaxIN(:,:,i),reshapeMaxTAR(:,:,i));
%     ga_opts = gaoptimset('InitialPopulation', x_ga_opt(i-1:,:), 'InitialScores',err_ga(i-1:,:), 'TolFun',
err_ga(i-1:,:), 'display', 'iter', 'generations', 30);
%     [x_ga_opt(i,:), err_ga(i,:)] = ga(EWW,3*AmountNeuro+sizeIn+sizeOut,ga_opts);
% end;
% net = setwb(net,x_ga_opt(20,:));
% Y = net(In2);
load All_DATA
Counter=1;
if Stabiliz ~= 0, figure(Counter);
% n = max(size(Y));
plot(1:dl, tar(1:Stabiliz,1:dl),'k*-',1:dl, Y(1:Stabiliz,1:dl),'bo-'); %Ts - Целевые значения, Y -
аппроксимация
title('Аппроксимация. Стабилизаторы');
grid on;
end, Counter=Counter+1;
figure(Counter);
% n = max(size(Y));
plot(1:dl, tar(Stabiliz+1:sizeOut,1:dl),'k*-',1:dl, Y(Stabiliz+1:sizeOut,1:dl),'mo-'); %Ts - Целевые
значения, Y - аппроксимация
title('Аппроксимация. Прогноз');
grid on;
Counter=Counter+1;

sizeIn=szI; %Возвр значение
inSim = zeros(sizeIn, zxc);
tarSim = zeros(sizeOut, zxc);
```

```

sdvig=0;
dop=0;

for i=1:zxc
    for j=1:sizeIn
        inSim(j, i) = Odine(nachalo + i + j - 1 + sdvig);
    end
    for ji=1:sizeOut
        tarSim(ji, i) = Odine(nachalo + j + i + dop + ychet - Stabiliz);
        dop=dop+1;
    end
    dop=0;
end
simIN = tonndata(inSim,true,false);          %%%%%%%%%%%
simTAR = tonndata(tarSim,true,false);        %%%%%%%%%%%

if novoee==1
    sizeIn=szNov;
    rut=1;
    for j=1:zxc
        for i=3:sizeIn+1
            obmen=RMA_1(inSim(:,j),i);
            inSim2(sizeIn-i+2,rut)=obmen(length(obmen));
            clear obmen;
        end
        inSim2(sizeIn,rut)=inSim(szI,j);
        rut=rut+1;
    end
    clear simIN;
    simIN=tonndata(inSim2,true,false);        %%%%%%%%%%%
end
sizeIn=szI;

clear Xs Xi Ai Ts
[Xs,Xi,Ai,Ts] = preparets(net,simIN,simTAR); %%%%%%%%%%%
out = net(Xs,Xi,Ai);
out = cell2mat(out);
% out = mapminmax('reverse',out,minim);
Ts = cell2mat(Ts);
% Ts = mapminmax('reverse',Ts,minim);
SUMMa(1:sizeOut,:)=out;
afs(1,:)=out(sizeOut-Stabiliz,:);
afs1(1,:)=Ts(sizeOut-Stabiliz,:);
% Ts-out
%На этом графике вывожу последнюю точку из каждого прогноза, которая имеет
%наибольшую ошибку, например, если спрогнозировано 7 точек, то на график
%идет седьмая и т.д. для всех спрогнозированных выборов, также график
%целевой
%figure(Counter);
%plot(1:length(afs),afs,'bo-',...
% 1:length(afs1),afs1,'r*-', 'LineWidth',2);

```

```

%xlabel('t');ylabel('y');title('Прогноз');
%legend('Спрогнозированный сигнал','Целевой сигнал');
%grid on;
%Counter=Counter+1;

%На этом графике все точки из спрогнозированных и графики целевые
if Stabiliz ~= 0, figure(Counter);
%n = max(size(out));
plot(1:zxc, Ts(1:Stabiliz,1:zxc),'k*-',1:zxc, out(1:Stabiliz,1:zxc),'bo-');%Ts - Целевые значения, out -
Стабилизаторы
title('Прогноз. Стабилизаторы');
grid on;
end, Counter=Counter+1;
figure(Counter);
%n = max(size(out));
plot(1:zxc, Ts(Stabiliz+1:sizeOut,1:zxc),'k*-',1:zxc, out(Stabiliz+1:sizeOut,1:zxc),'mo-');%Ts -
Целевые значения, out - Спрогнозированные точки
title('Прогноз. Прогноз');
grid on;
Counter=Counter+1;

% График, самый исходный сигнал с выхода фильтра (один) + графики всех прогнозов +
графики всех стабилизаторов
figure(Counter); hold on;
plot(1:zxc, Odine(nachalo+szI+1:nachalo+szI+1+zxc-1),'ko-','LineWidth',4);
if Stabiliz ~= 0,
plot(1:zxc, Ts(1:Stabiliz,1:zxc),'k*-',1:zxc, out(1:Stabiliz,1:zxc),'bo-');%Ts - Целевые значения, out -
Стабилизаторы
%title('Прогноз. Стабилизаторы');
%grid on;
end,
plot(1:zxc, Ts(Stabiliz+1:sizeOut,1:zxc),'k*-',1:zxc, out(Stabiliz+1:sizeOut,1:zxc),'mo-');%Ts -
Целевые значения, out - Спрогнозированные точки
title('Исходный сигнал. Прогноз + Стабилизаторы');
grid on;
Counter=Counter+1;

% График, самый исходный сигнал с выхода фильтра (один) + графики всех прогнозов
figure(Counter); hold on;
plot(1:zxc, Odine(nachalo+szI+1:nachalo+szI+1+zxc-1),'ko-','LineWidth',4);
%if Stabiliz ~= 0,
%plot(1:zxc, Ts(1:Stabiliz,1:zxc),'k*-',1:zxc, out(1:Stabiliz,1:zxc),'bo-');%Ts - Целевые значения, out
- Стабилизаторы
%title('Прогноз. Стабилизаторы');
%grid on;
%end,
plot(1:zxc, Ts(Stabiliz+1:sizeOut,1:zxc),'k*-',1:zxc, out(Stabiliz+1:sizeOut,1:zxc),'mo-');%Ts -
Целевые значения, out - Спрогнозированные точки
title('Исходный сигнал. Прогноз');
grid on;

```

```

Counter=Counter+1;

% График, самый исходный сигнал с выхода фильтра (один) + графики всех прогнозов
figure(Counter); hold on;
plot(1:zxc, Odine(nachalo+szI+1:nachalo+szI+1+zxc-1),'ko-', 'LineWidth',4);
plot(1:zxc, out(Stabiliz+1:sizeOut,1:zxc),'mo-');% Ts - Целевые значения, out - Спрогнозированные
точки
title('Исходный сигнал. Прогноз');
grid on;

% График, самый исходный сигнал с выхода фильтра (один) + графики всех прогнозов +
графики всех стабилизаторов
figure(Counter);
n = max(size(out));
hold on;
plot(1:n, SUMMa((sizeOut-Stabiliz):sizeOut,:),'g-', 'LineWidth',2);
plot(1:n, SUMMa(1:(sizeOut-Stabiliz),:),'r-', 'LineWidth',2);
plot(1:n, Odine(nachalo+szI+1:nachalo+szI+1+n-1),'k-', 'LineWidth',4);
title('Прогноз');
legend('Целевой', ' ', ' ', ' ', ' ', 'Прогноз');
grid on;
Counter=Counter+1;
Точ

BlockFormINPUT:
clear all;clc;

load ( 'Fddd2' , 'ddd2' ); % импорт из mat.файла - ddd2
rr=size (ddd2), % размер импортируемого файла ddd

nmax=rr(1), % максимальное количество строк в ddd
T=30; % Период среза
Odine=RMA_1(ddd2(1:nmax,1),T); % Работа, только с одним сигналом, без SL
nachalo=4500; % С этой точки начинается прогноз + sizeIn
NumSeti=1; % Сохр нейр. сеть
Counter=1; % Счетчик (Для графиков)
dl=5000; % Размер обучающей выборки

szI=30; % Количество входов
zxc=50; % Кол. выборок для получения прогноза
sizeIn=szI; % Кол. входов
sizeOut=7; % Кол. выходов
Delay1=0; % Задержка
Delay2=0; % Задержка
AmountNeuro=8; % Кол. нейронов
maxFail=200; % Максимальное количество ошибок на контрольном массиве
Stabiliz=3; % Количество стабилизаторов(если больше, то нужно менять кол. входов
повоeee=1; % Оптимизация, преобразование сигнала из горизонтального в
вертикальный
szNov=50; % Кол. точек в вертикальном сигнале
ychet=szI-sizeIn;

```

```

% [q,minim1]=mapminmax(Odine(1:dl-51));
% [meanS,q]=mapminmax(Odine);

% clear q;

% Odine=meanS;
% minim=minim1;

in = zeros(sizeIn, dl);
tar = zeros(sizeOut, dl);
dop=0;
for i=1:dl
    for j=1:sizeIn
        in(j, i) = Odine(i + j - 1);
    end
    for ji=1:sizeOut
        tar(ji, i) = Odine(j + i + dop - Stabiliz);
        dop=dop+1;
    end
    dop=0;
end

In=tonndata(in,true,false);           %%%%%%%%%%%
Tar=tonndata(tar,true,false);        %%%%%%%%%%%

if novoe==1
    sizeIn=szNov;
    rut=1;
    for j=1:dl
        for i=3:sizeIn+1
            obmen=RMA_1(in(:,j),i);
            In2(sizeIn-i+2,rut)=obmen(length(obmen));
            clear obmen;
        end
        In2(sizeIn,rut)=in(szI,j);
        rut=rut+1;
    end
    clear In;
    In=tonndata(In2,true,false);      %%%%%%%%%%%
End

RMA_1:

function rezfltr = RMA_1(ddd,T)

if T == 0 || T == 1 , rezfltr=ddd;

```



```

else
Nk=T;
Nk=2*(Nk-ceil(Nk/2)) + 1;
Tz=floor(Nk/2); TD=fix(Tz);

for i=1:Nk ,
k1 = (i-(Nk+1)/2);
k = k1 + TD;

if k == 0 , w(i) = 1;
else
w(i) = sin(2*pi*k/T)/(2*pi*k/T);
end

if (1-(4*k/T)^2) == 0 , wa(i) = w(i)*pi/4;
else
wa(i) = w(i)*cos(2*pi*k/T)/(1-(4*k/T)^2);
end

ww(i) = wa(i)*exp(-0.5*(1.8*k1/((Nk-1)/2))^2);
end

w=ww/sum(ww);
rezfltr=filter(w,1,ddd);
end

end

train:

function EWW = trainn(x,net,inputs,targets)
net = setwb(net, x');
y = net(inputs);
EW = sum((y-targets).^2)/length(y);
EWW = sum(EW)/length(EW);
End

```