

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра радиофизики и нелинейной динамики

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

**Разработка программного интерфейса для работы с искусственными
нейронными сетями и применение его к задачам обработки сигналов**

студента 4 курса 4061 группы
направления 11.03.02 Инфокоммуникационные
технологии и системы связи
Института физики
Гайдука Кирилла Алексеевича

Научный руководитель
профессор, доктор физ.-мат. наук, доцент _____ А.В. Шабунин

Зав. кафедрой радиофизики
и нелинейной динамики
д.ф.-м.н., доцент _____ Г.И. Стрелкова

Саратов 2021 г.

Введение. С момента появления компьютера стояла задача научить его мышлению. Способность компьютера самостоятельно подбирать необходимые параметры могла бы решить эту задачу и одним из решений было изобретение искусственных нейронных сетей (ИНС).

Целью выпускной квалификационной работы является разработка программного интерфейса для работы с ИНС и применение его для решения задач обработки сигналов.

На основе поставленной цели необходимо решить следующие задачи:

- Исследование базовых структур и парадигм ИНС;
- Описание задач обработки сигналов, решаемых ИНС;
- Анализ существующих программных интерфейсов ИНС;
- Разработка пакета функций для работы с ИНС в системе Octave.

Актуальность темы подтверждается повышенным интересом к ИНС и их программным реализациям, в частности, благодаря широкому классу задач, которые решают ИНС.

Дипломная работа занимает 58 страниц, имеет 20 рисунков и 3 таблицы.

Обзор составлен по 25 информационным источникам.

Во введении рассматривается история развития ИНС, описание системы разработки, актуальность работы, устанавливается цель и выдвигаются задачи для достижения поставленной цели.

Первый раздел с главы 1 по 3 представляет собой теоретический обзор основных сведений о парадигмах ИНС.

Во втором разделе работы с главы 4 по 5 описаны ряд задач [4], стоящие перед ИНС и общие рекомендации по их решению.

В третьем разделе, в главе 6 описаны программные реализации ИНС.

В практической части (глава 7) описаны разработанные алгоритмы функционирования ИНС и полученные с помощью них результаты.

Основное содержание работы. В первой главе описывается основная модель ИНС. Описание начинается с формального нейрона, на вход которого подаётся входные сигналы, проходящие через синаптические веса, суммируются вместе с пороговым элементом и проходят через функцию активации

Формальный нейрон состоит из синапсов, которые характеризуют силу (вес) связи между нейронами, сумматора, который выполняет сложение входных сигналов, умноженные на соответствующие веса. Преобразователь реализует функцию от выхода сумматора. На математическом языке можно представить следующим образом:

$$y_k = f(S), \quad (1)$$

$$S = \sum_{i=1}^n w_i x_i + b, \quad (2)$$

где S - выход линейного сумматора. В таблице 1 описаны основные функции активации нейронов с их формулой, производной и областью значений.

Таблица 1 – Функции активации нейронов

Название	Формула	Производная	Область значений
Пороговая	$f(S) = \begin{cases} 0, S < \theta \\ n, S = 0 \\ 1, S \geq \theta \end{cases}$	$f'(S) = \begin{cases} 0, S \neq 0 \\ \text{нет}, S \geq 0 \end{cases}$	(0,1)
Линейная	$f(S) = aS$	$f'(S) = a$	$(-\infty, +\infty)$
Сигмоидная (логистическая)	$f(S) = \frac{1}{1 + e^{-aS}}$	$f'(S) = a(f(S)(1 - f(S)))$	(0,1)
Гиперболический тангенс	$f(S) = \frac{e^{aS} - e^{-aS}}{e^{aS} + e^{-aS}}$	$f'(S) = a - af(S)^2$	(-1,1)
ReLU (Выпрямитель)	$f(S) = x^+ = \max(0, S)$	$f'(S) = \begin{cases} 0, S < 0 \\ 1, S \geq 0 \end{cases}$	$(0, +\infty)$

Градиент должен быть связан с переменной, поэтому широко применяется класс функций – сигмоидные.

Это класс гладких монотонных функций, имеющих форму буквы «S». Примерами таких функций являются логистическая функция и гиперболический тангенс. Такие функции имеют гладкий градиент и производную, которая зависит от переменной, в отличие от линейной функции активации. В работе описана функция активации ReLu и её модификации, которая также является нелинейной.

Существуют другие модели нейрона, но в рамках работы используют описанную выше детерминированную модель.

Нейроны объединяют в слои, в которых нейроны находятся параллельно, затем слои располагают последовательно так, что выходы одних слоёв являются входами для последующих. Последовательность слоёв и будет представлять собой нейронную сеть. Нейроны делят на входные нейроны, которые являются входными параметрами сети, скрытые нейроны, не имеющие прямых связей с входными сигналами, и выходные, которые представляют результирующий выход сети. В зависимости от межнейронных связей сети делятся на два класса: сети прямого распространения, в которых сигнал распространяется только от входных нейронов к выходным, и рекуррентные сети, в которых существуют обратная связь. В рамках работы будут применяться только сети прямого распространения.

Многослойные ИНС чаще всего объединяются так, что нейроны одного слоя имеют одинаковые входные сигналы. Помимо входных и выходных слоёв по аналогии с нейронами рассматривают скрытые слои. Многослойные сети имеют преимущества за счёт скрытых нейронов, которые обеспечивают более мощную и точную сеть.

Во второй главе обучение определяется как итеративный процесс настройки межнейронных связей. Выделяют два подхода к обучению: обучение без учителя и с учителем. В первом случае обучающее множество состоит из входных и целевых значений. Без учителя подстройка происходит в результате конкуренции между нейронами или с учётом корреляции выходных сигналов нейронов, между которыми существует связь.

В рамках работы рассматривается обучение с учителем, которое можно описать алгоритмом: подготовка обучающей выборки, выбор структуры и параметров, обучение по определённому методу. Обучение с учителем также можно представить в виде решения оптимизационной задачи.

Одной из главной трудностей обучения является переобучение, которое представляет собой излишне точным соответствием конкретному набору данных, при котором сеть теряет способность к обобщению. Одно из решений данной проблемы – деление исходное множество на обучающее, на котором сеть обучается, тестовое, на котором проводится проверка модели, и множество валидации, которое применяется для регуляризации по ранней остановке.

Базовым методом обучения с учителем, является алгоритм обратного распространения ошибок, которая минимизирует так называемую функцию стоимости $E(n)$ за счёт приближения выходного сигнала $y_k(n)$ к целевому $d_k(n)$

$$E(n) = \sum_k \frac{1}{2} e_k^2(n), \quad (3)$$

Этот алгоритм основан на дельта-правиле [9], которое после проверки отклика инициализированной сети определяет ошибку как:

$$\delta_k = \varepsilon_k f'(S) = f'(S)(y_k - d_k), \quad (4)$$

Здесь y_k - реальный выход сети, а d_k - ожидаемый отклик, $f'(S)$ - производная функции активации от выхода линейного сумматора. Само дельта-правило определяет подстройку весового коэффициента как:

$$\Delta w_{jk} = -\eta \delta_k x_j, \quad (5)$$

Где η – скорость обучения, которая является параметром обучения и задаётся вручную.

Само обратное распространение заключается в вычислении подстройки по дельта правилу на последнем слое, а ошибка на предыдущем слое будет являться произведением матрицы весов и ошибки на последнем. Таким же образом подстройка весов вычисляется на каждом слое.

Если ошибка больше заданной, то начинается новая эпоха обучения и алгоритм повторяется, если меньше, то обучение заканчивается.

Для устранения ряда недостатков скорость обучения берётся меньше единицы и постепенно уменьшается в процессе обучения. Но зачастую этот метод не обеспечивает быстрой и точной сходимости, поэтому применяют его модификации, как например обратное распространение с учётом момента инерции, когда подстройка в одной эпохе учитывает подстройку на предыдущей:

$$\Delta w_k = m\Delta w_{k-1} + (1 - m)\eta\delta_k, \quad (6)$$

где вводится параметр инерции m . Этот алгоритм позволяет не останавливаться в локальных минимумах поверхности ошибок и ускоряет обучение нейронной сети в 4-5 раз.

Так как обучение итеративно существует несколько методов предъявления обучающего множества. Последовательный режим обучения называют интерактивным, так как в данном режиме корректировка весов происходит после каждого примера, т.е. на каждой обучающей паре. Пакетный режим называют «оффлайн» режимом, так как обучение на одном шаге происходит после предъявления всех примеров из обучающего множества. Пакетный режим чаще применяется в задачах распознавания образов и классификации, так как обеспечивает сходимость к локальному минимуму ошибок при довольно широком разбросе параметров обучения. Последовательный режим применяется в задачах аппроксимации и распознавании систем, но сходимость обучения к локальному минимуму требует тщательного выбора параметров.

В третьей главе описана одна из основных моделей ИНС- персептрон, который изначально предназначался для классификации линейно-разделимых сигналов.

Многослойный персептрон – это сеть прямого распространения сигнала без обратных связей, в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько слоев. Многослойные персептроны используют алгоритм обратного распространения ошибки, описанный выше.

В ряде публикаций описаны эксперименты, в которых многослойный персептрон решал задачу классификации радиосигналов с высокой эффективностью.

Также многослойные перцептроны способны успешно решать задачи управления, обобщения, аппроксимации, распознавания систем, что говорит о применении данной модели к задачам обработки сигналов. Именно модель многослойного перцептрона была выбрана для решения поставленных в начале работы задач, подробная архитектура описана в практической части.

В 4 главе описан ряд практических задач, которые способны решать ИНС. Задача распознавания сигналов является одной из самых популярных и классических задач. ИНС способны распознавать и классифицировать практически любые сигналы. Для начала распознаваемые сигналы следует дискретизировать с необходимой точностью и задать входные вектора с соответствующими целевыми векторами на выходе. Входная размерность – это количество выборок исходного сигнала на количество сигналов, а выходная размерность – число классов. Сигнал принадлежит классу I , которому соответствует признак в i позиции вектора. Чаще всего за признак принадлежности к классу принимают 1 на i выходе. Тем большее количество скрытых слоёв и нейронов будет в сети, тем сложнее обучаться сети, но вместе с этим точность распознавания растёт. Это правило применяется и в ряде других задач. После успешного обучения сеть проверяется на исходных и зашумленных сигналах. Сигнал в таком случае считается распознанным и достоверно относится к классу I если на соответствующем выходе i значение максимально и больше порогового (обычно порог выбирают около 0.5).

Более сложная задача - аппроксимация сигналов, которые реализуют нелинейные функции $y=F(x)$. Согласно теореме Цыбенко (универсальная теорема аппроксимации) ИНС даже с одним скрытым слоем может аппроксимировать любую непрерывную функцию многих переменных с любой точностью. Важными условиями являются достаточное количество нейронов скрытого слоя и правильный подбор весов т.е. качественное обучение.

Размерности входных и целевых значения будут определяться размерностями областей определения и значений аппроксимируемой функции соответственно, целевой вектор представляет собой значение функции от

элементов входного вектора, которые выступают в качестве аргумента. Архитектура сети может представлять собой радиально-базисные сети, вероятностные сети, но в данной работе данная задача решается с помощью многослойного персептрона благодаря его простой топологии.

Задача распознавания систем схожа с задачей аппроксимации и является её частным случаем, когда в практических задачах поведение системы неизвестно. Задача заключается в определении поведения системы, которая по умолчанию представляет для нас «чёрный ящик». Например, если система реализует какое-либо отображение и задавая начальные условия и параметры мы можем получить отклик этой системы.

В 5 главе описаны общие рекомендации к обучению и проектированию ИНС. Опишем те рекомендации, которые применяются в разработанном интерфейсе. Целевые значения должны принадлежать области значений функции активации и, если необходимо, нормализованы путём центрирования, ограничения по величине и декорреляции. Размер обучающего множества должен зависеть от общего числа настраиваемых весов и допустимой вероятности ошибки (точности), как

$$K \sim \frac{N_w}{P_e}, \quad (7)$$

где N_w - число настраиваемых синаптических весов, P_e - вероятность ошибки. Для избегания переобучения примеры из обучающего множества должны быть перемешаны или рандомизированы. Структура сети должна быть максимально простой, т.е. содержать минимальное количество слоёв и нейронов в ней, при котором сеть будет работать с необходимой точностью. Синаптические веса сети должны выбираться случайным образом и иметь равномерное распределение. Веса также следует нормировать на величину, обратную корню из размерности входного слоя.

В процессе обучения скорость также можно изменять в зависимости от количества эпох для большей сходимости (формула 8)

$$\eta(n) = \frac{\eta_0}{1+n/\tau}, \quad (8)$$

где η – переменная скорость обучения, η_0 – начальная скорость обучения, выбранная из соображений, описанных выше, n - номер итерации, τ –параметр плавности уменьшение скорости обучения.

Количество нейронов в скрытых слоёв можно определить по формуле, являющейся следствием из теорем Арнольда-Колмогорова-Хехт-Ниельсена:

$$\frac{N_y Q}{1+\log_2(Q)} \leq N_w \leq N_y \left(\frac{Q}{N_x} + 1 \right) (N_x + N_y + 1) + N_y, \quad (9)$$

где N_w – число синаптических связей, N_x - размерность входного сигнала, N_y – размерность выходного сигнала,

Алгоритм обучения должен иметь механизм завершения обучения. Глобальным критерием остановки является достаточная малость целевой функции ошибок, причём важно иметь возможность задавать это малое значение. Более локально обучение должно завершаться, когда величина изменения целевой функции, то есть синаптических весов, достигла достаточно малой величины.

В 6 главе описаны реализации ИНС в виде программных интерфейсов. В этом случае реализации ИНС не отличаются от другого проекта, связанного с разработкой программного обеспечения. Рассмотрены интерфейсы в виде дополнения для пакетов прикладных вычислений: PyBrian, StatisticaNeuralNetworks, Neural Excel, NNTool. Такие продукты как Neural10, MemBrain, QwikNet32, NeuroSolutions, анализ которых приведён в [22] имеют собственный функционал и не интересны с точки зрения разработки на основе прикладных систем. Из этого анализа за основу парадигмы разработки программного интерфейса был выбран пакет Neural Network Toolbox для Matlab, который представляет собой набор прикладных функций. Таким образом за основу разработки была взята Octave - бесплатный аналог системы Matlab.

В практической части (глава 7) на основе анализа существующих интерфейсов описан подход к разработке программного интерфейса. Поставлена задача написания набора функций работы нейронного слоя и нейронной сети.

Были разработки функции инициализации, проверки отклика и обучения нейронной сети и нейронного слоя. Также были разработанные функции итеративного обучения в последовательном и пакетном режимах и их модификации с использованием момента инерции. Весь набор функций представлен в виде приложения к работе.

С помощью разработанных функций удалось решить распознавание сигналов, которые представляют собой функции Уолша. Эти функции получили широкое распространение в радиосвязи, где они осуществляют кодовое разделение сигналов. Вид этих функций, массив значений в си

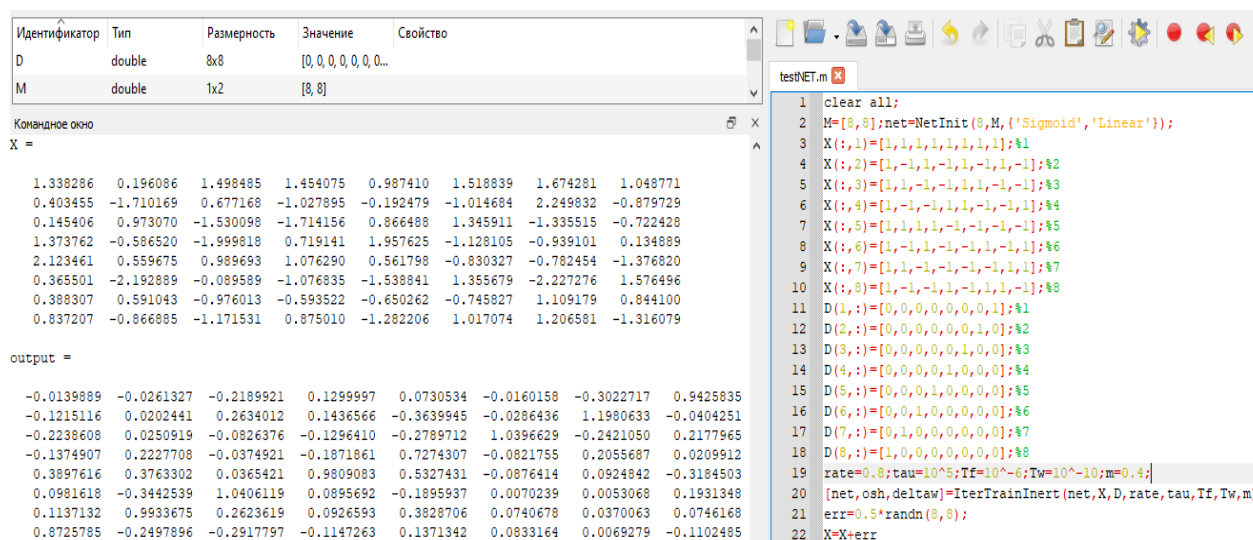


Рисунок 1 - Проверка отклика обученной сети на зашумленном входном сигнале в графическом интерфейсе системы Octave

Для решения этой задачи была выбрана структура из скрытого слоя с сигмоидной функцией активации и 8 скрытыми нейронами, выходной слой с линейной функцией активации. Было выбрано обучение в пакетном режиме с моментом инерции со следующими параметрами: скорость обучения $\alpha=0.8$, параметр инерции $m=0.4$. Из рисунка 2 мы видим, что сеть обучилась за 130 итераций до среднеквадратичной ошибки (СКО) порядка 10^{-6} , что является хорошим результатом.

Также была решена задача аппроксимации сигналов. Архитектура для решения этой задачи состоит из скрытого слоя с тангенциальной функцией и выходного слоя с линейной функцией активации.

Для удобства обозначим изменяемые параметры обучения: α - скорость обучения, β - количество скрытых нейронов, S - количество обучающих пар в эпохе и m - момент инерции в случае обучения с моментом инерции.

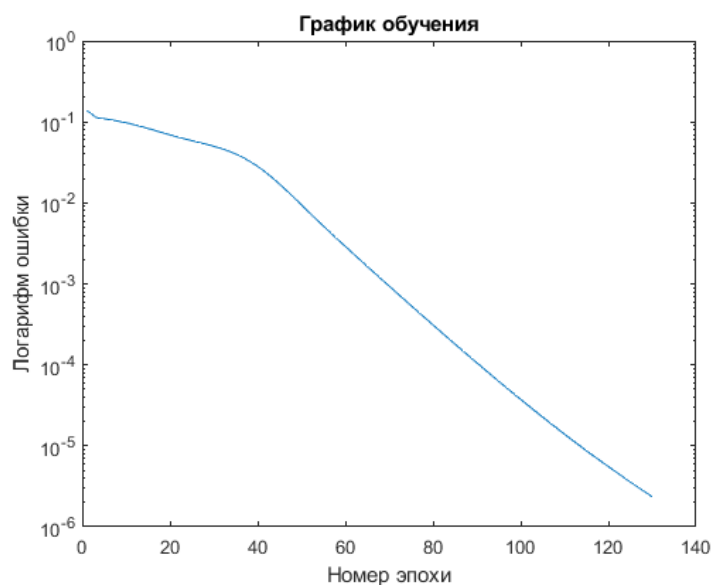


Рисунок 2 - График обучения распознавания функций Уолша

На рисунке 3 и 4 представлено сравнение аппроксимации $y=x^3$ в двух режимах обучения

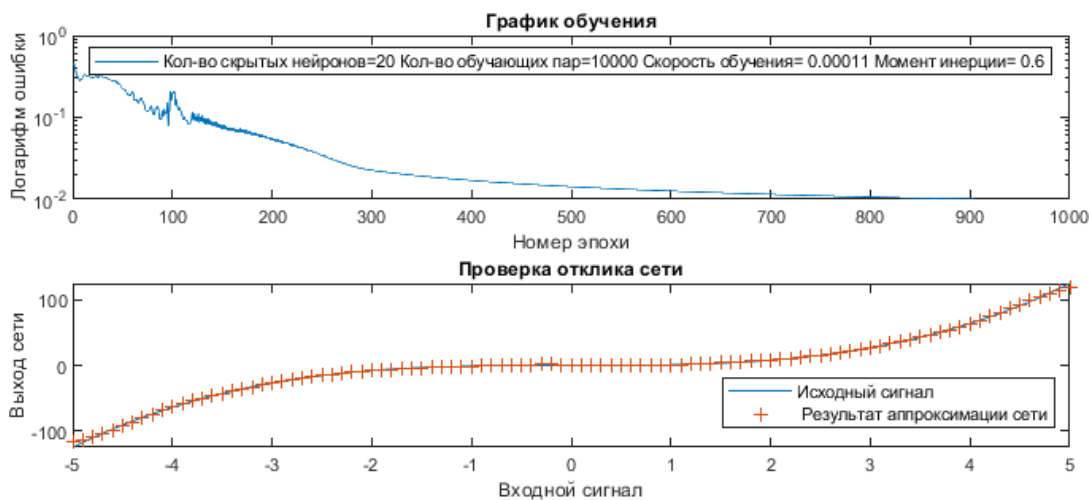


Рисунок 3 – Результат аппроксимации $y=x^3$ с моментом инерции в пакетном режиме. $\alpha=0.00011$, $\beta=20$, $S=10000$, $m=0.6$

Также были аппроксимированы некоторые другие сигналы, в том числе более сложный сигнал $y=\sin(x+2*\cos(x))$ в последовательном режиме с учётом момента инерции (рисунок 5).

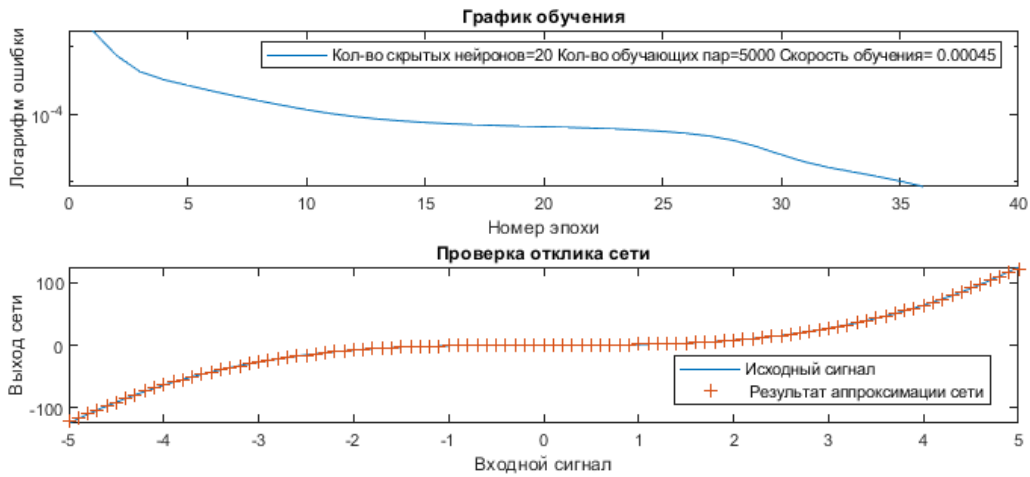


Рисунок 4 – Результат аппроксимации $y=x^3$ в последовательном режиме $\alpha=0.00045$, $\beta=20$, $S=5000$

Также были аппроксимирован более сложный сигнал $y=\sin(x+2*\cos(x))$ в последовательном режиме с учётом момента инерции (рисунок 5).

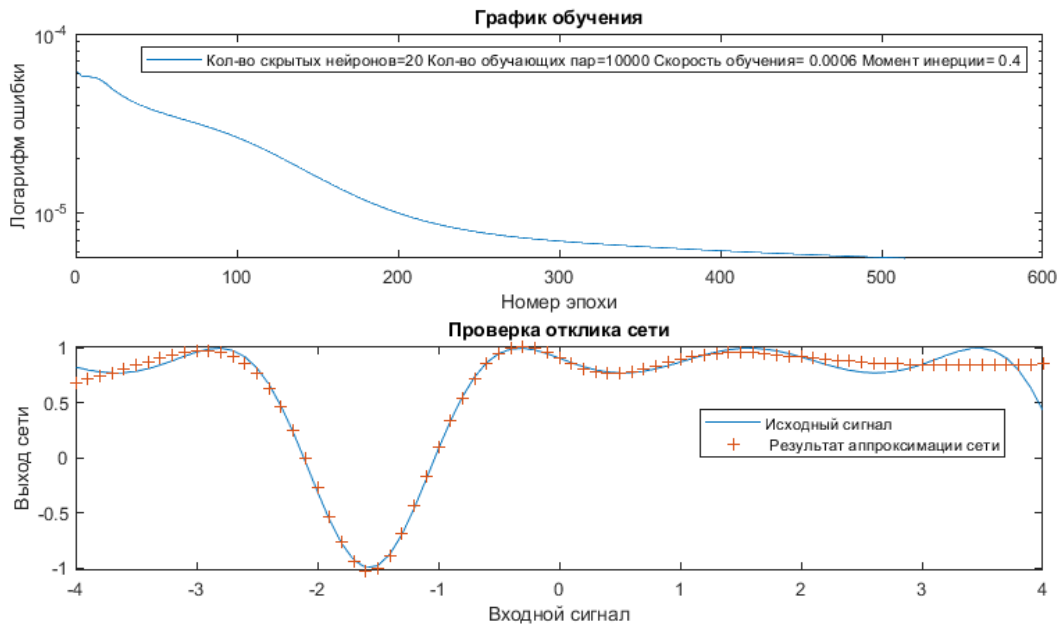


Рисунок 5 – Результат аппроксимации $y=\sin(x+2*\cos(x))$ в последовательном режиме с моментом инерции. $\alpha =0.0006$, $\beta=20$, $S=10000$, $m=0.4$

Таким образом задача аппроксимации может считаться решённой.

Также с помощью разработанного интерфейса решена задача распознавания систем на примере распознавания системы логистического отображения.

$$x_{n+1} = rx_n(1 - x_n), \quad (10)$$

где x_n принимает значения от 0 до 1, r – положительный параметр.

Зафиксируем параметр $r=3.99$, когда система демонстрирует хаотическое поведение и вычислим 1000 значений x_{n+1} , фиксируя x_n . То есть моделируется ситуация, когда у нас есть ограниченное число откликов от нашей системы и стоит задача повторить поведение этой системы с помощью нейронной сети.

Архитектура выбрана такая же, как и в случае аппроксимации : скрытый слой с тангенциальной и выходной с линейной функцией активации. Обучение также проходило в последовательном режиме с учётом момента инерции, результат изображен на рисунке 6. Именно в этом примере решающим фактором для точности служило количество скрытых нейронов, поэтому для большей точности их количество увеличили.

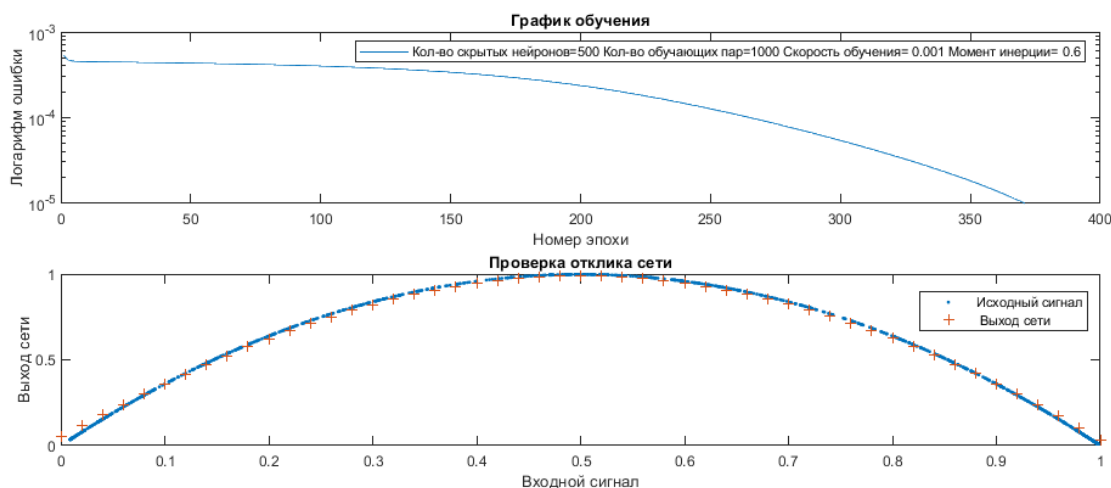


Рисунок 6 – Результат распознавания логистического отображения в последовательном режиме с моментом инерции.

$$\alpha = 0.001, \beta = 500, S = 1000, m = 0.6$$

Заключение. В данной работе была раскрыта тема явления ИНС, обзор простейших моделей и анализ структур. Было проведено исследование алгоритмов обучения ИНС и их примеры. Анализ основных задач, решаемых в данной работе и формирование рекомендаций по построению ИНС позволил чётко определить структуру и требования к программному интерфейсу. Проведён анализ реализаций программных интерфейсов в качестве пакетов для прикладных программ, на основе которых была выбрана программная среда и общая парадигма разработки

В ходе выполнения практики были получены следующие результаты:

- Были исследованы базовые структуры и парадигмы функционирования ИНС
- Описаны задачи обработки сигналов, решаемые ИНС
- Сформулированы общие рекомендации по обучению, обучающему множеству и архитектуре ИНС
- Проведён анализ существующих программных интерфейсов ИНС
- С помощью системы OCTAVE был создан набор функций, моделирующий функционирование нейронного слоя и нейронной сети;
- С помощью разработанного интерфейса были решены задачи распознавания и классификации сигналов, аппроксимации и распознавания систем.

Таким образом существующие программные интерфейсы обладают более широким, но в тоже время громоздким функционалом, зачастую не обеспечивающий необходимой наглядности. Дальнейшая разработка программного интерфейса для работы с разными парадигмами ИНС в совокупности с базовым интерфейсом позволит решить более широкий класс задач, что подтверждает актуальность данного исследования.