МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Кафедра информационных технологий

КУРСОВАЯ РАБОТА

ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ ФУНКЦИИ ПОСРЕДСТВОМ НЕЙРОННЫХ СЕТЕЙ

Работу выполнил_______________________________________ А. С. Горенко
(подпись, дата)

Факультет компьютерных технологий и прикладной математики курс 3

Специальность 020303 - «Математическое обеспечение и администрирование информационных систем»

Научный руководитель
доц., канд. физ-мат. наук. ______________________________В. В. Подколзин
(подпись, дата)
Нормоконтроллер
ст. преп. ____ А. В. Харченко
(подпись, дата)

Краснодар 2017

РЕФЕРАТ

В курсовой работе 36 страниц, 3 части, 24 рисунка, 5 источников.
АППРОКСИМАЦИЯ ФУНКЦИЙ, НЕЙРОННЫЕ СЕТИ, МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ.

Объект курсовой работы – полиномиальная аппроксимация функций.

Цель курсовой работы — построение модели нейронной сети для аппроксимации одноместных функций.

В рамках работы были получены следующие результаты:

- построена модель нейронной сети для аппроксимации одноместных функций.
- для автоматизации процесса аппроксимации функций разработано пользовательское приложение NN_Approximation с использованием языка программирования C++ и графической библиотеки SFML.
- проведён сравнительный анализ двух моделей нейронных сетей прямого распространения.

СОДЕРЖАНИЕ

Введение..4
1 Нейронные сети...5
1.1 Определение ИНС...7
1.2 Архитектура нейронной сети..9
1.2.1 Многослойный персептрон (MLP)..10
2 Обучение нейронных сетей...11
2.1 Алгоритм обратного распространения ошибки..............................16
2.2 Переобучение и обобщение..18
3 Аппроксимация функций...22
3.1 Аппроксимация функций нейронной сетью...................................23
3.2 Разработка программы...27
3.3 Схема программы..28
3.4 Описание интерфейса программы...29
3.5 Анализ точности аппроксимации относительно конфигурации нейронной сети 31

Заключение...37
Список использованных источников...36

ВВЕДЕНИЕ

Исследования в области нейронных сетей начались в 40-е годы XX века. Первое систематическое изучение искусственных нейронных сетей было предпринято Маккалокком и Питтсом в 1943 г. Позднее они исследовали сетевые парадигмы для распознавания изображений, подвергаемых сдвигам и поворотам.

Основные задачи, которые ставятся перед нейронными сетями, относятся к задачам распознавания образов. Они заключаются в том, чтобы классифицировать входной образ, то есть отнести его к какому-либо известному сети классу. Изначально сети даются эталонные образы – такие образы, принадлежность которых к определенному классу известна. Затем на вход сети подается некоторый неизвестный образ, и сеть пытается по определенному алгоритму соотнести его с каким-либо эталонный образом. Можно сказать, что нейросети проводят кластеризацию образов. Так как кластерный анализ применяется исследователями рынка ценных бумаг, то нейронные сети могут быть использованы и для прогнозирования стоимости акций, что является актуальной задачей, к тому же строго неразрешимой на данный момент.

Так же нейронный сети могут применяться в криминалистике (анализ отпечатков пальцев) или же для облегчения работы правоохранительных органов в поимке преступников, потому что сейчас уже созданы такие программы, которые распознают лица.

В данной работе рассматриваются вопросы аппроксимации одноместных функций полиномиальными нейронными сетями на неоднородных и случайных обучающих выборках.

1 Нейронные сети

Нейронные сети возникли из исследований в области искусственного интеллекта, а именно, из попыток воспроизвести способность биологических нервных систем обучаться и исправлять ошибки, моделируя низкоуровневую структуру мозга (Patterson, 1996). Основной областью исследований по искусственному интеллекту в 60-е - 80-е годы были экспертные системы. Такие системы основывались на высокоуровневом моделировании процесса мышления (в частности, на представлении, что процесс нашего мышления построен на манипуляциях с символами). Скоро стало ясно, что подобные системы, хотя и могут принести пользу в некоторых областях, не ухватывают некоторые ключевые аспекты человеческого интеллекта. Согласно одной из точек зрения, причина этого состоит в том, что они не в состоянии воспроизвести структуру мозга. Чтобы создать искусственных интеллект, необходимо построить систему с похожей архитектурой.

Мозг состоит из очень большого числа (приблизительно 10,000,000,000) нейронов, соединенных многочисленными связями (в среднем несколько тысяч связей на один нейрон, однако это число может сильно колебаться). Нейроны - это специальная клетки, способные распространять электрохимические сигналы. Нейрон имеет разветвленную структуру ввода информации (дендриты), ядро и разветвляющийся выход (аксон). Аксоны клетки соединяются с дендритами других клеток с помощью синапсов. При активации нейрон посылает электрохимический сигнал по своему аксону. Через синапсы этот сигнал достигает других нейронов, которые могут в свою очередь активироваться. Нейрон активируется тогда, когда суммарный уровень сигналов, пришедших в его ядро из дендритов, превысит определенный уровень (порог активации).

Интенсивность сигнала, получаемого нейроном (а следовательно и возможность его активации), сильно зависит от активности синапсов. Каждый синапс имеет протяженность, и специальные химические вещества передают сигнал вдоль него. Один из самых авторитетных исследователей нейросистем, Дональд Хебб, высказал постулат, что обучение заключается в первую очередь в изменениях "силы" синаптических связей. Например, в классическом опыте Павлова, каждый раз непосредственно перед кормлением собаки звонил колокольчик, и собака быстро научилась связывать звонок колокольчика с пищей. Синаптические связи между участками коры головного мозга, ответственными за слух, и слюнными железами усилились, и при возбуждении коры звуком колокольчика у собаки начиналось слюноотделение.

Таким образом, будучи построен из очень большого числа совсем простых элементов (каждый из которых берет взвешенную сумму входных сигналов и в случае, если суммарный вход превышает определенный уровень, передает дальше двоичный сигнал), мозг способен решать чрезвычайно сложные задачи. Разумеется, мы не затронули здесь многих сложных аспектов устройства мозга, однако интересно то, что искусственные нейронные сети способны достичь замечательных результатов, используя модель, которая ненамного сложнее, чем описанная выше.

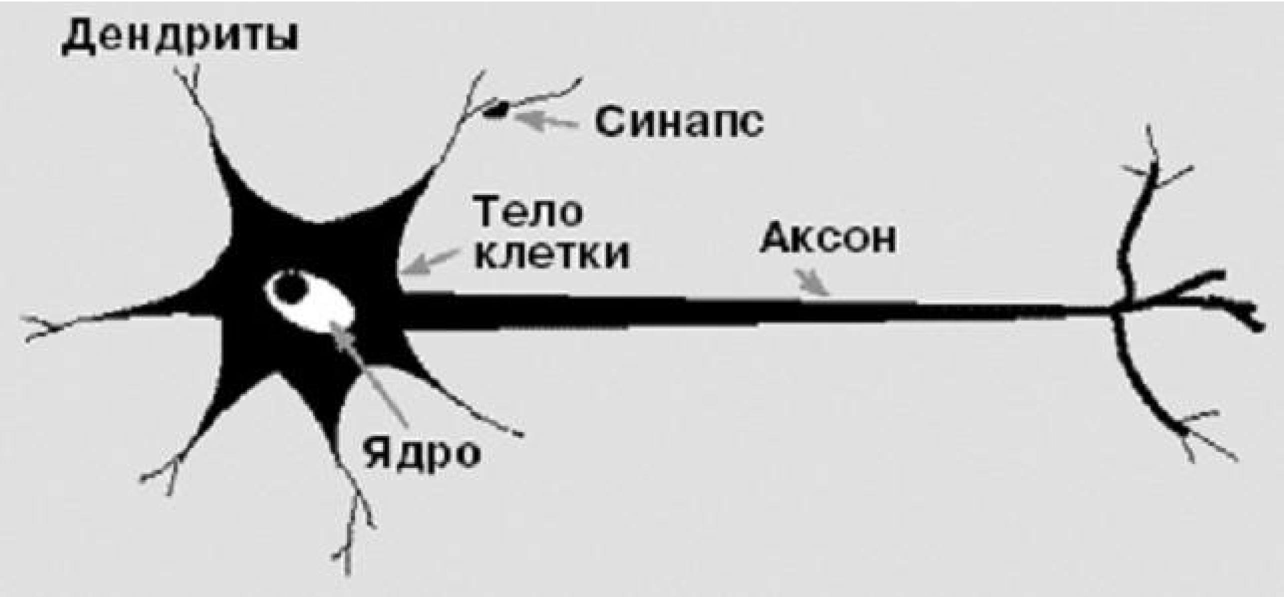


Рис. 1- Нейрон

1.1 Определение ИНС

Искусственная нейронная сеть — математическая модель, а также программная или аппаратныая реализация, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети Маккалока и Питтса. Впоследствии, после разработки алгоритмов обучения, получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

Основу каждой искусственной нейронной сети составляют относительно простые, в большинстве случаев - однотипные, элементы, имитирующие работу нейронов мозга (далее под нейроном мы будем подразумевать искусственный нейрон, ячейку искусственной нейронной сети).

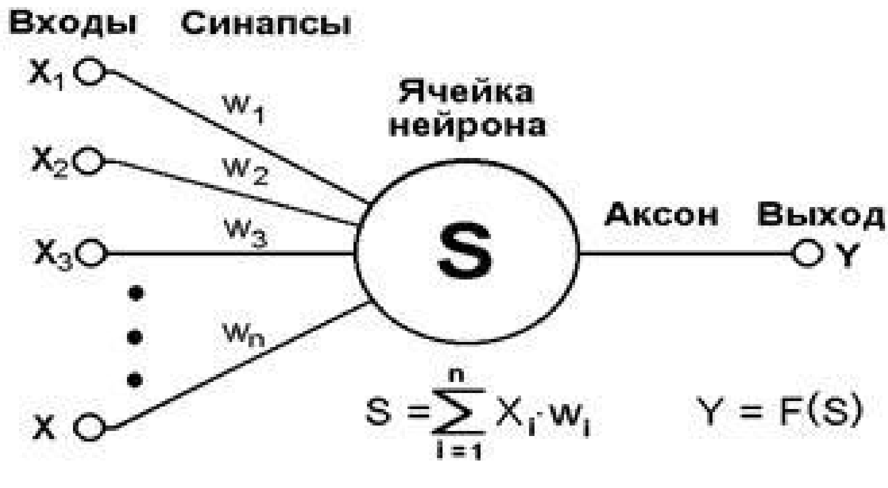


Рис. 2 - Искусственный нейрон

Нейрон обладает группой синапсов - однонаправленных входных связей, соединенных с выходами других нейронов. Каждый синапс характеризуется величиной синоптической связи или ее весом w_i.

Каждый нейрон имеет текущее состояние, которое обычно определяется, как взвешенная сумма его входов:

$$S=\sum_{i=1}^{n} x_i \cdot w_i$$

Нейрон имеет аксон - выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Выход нейрона есть функция его состояния:

y = f(s)

Нелинейная функция f называется активационной и может иметь различный вид (Рис. 3) .

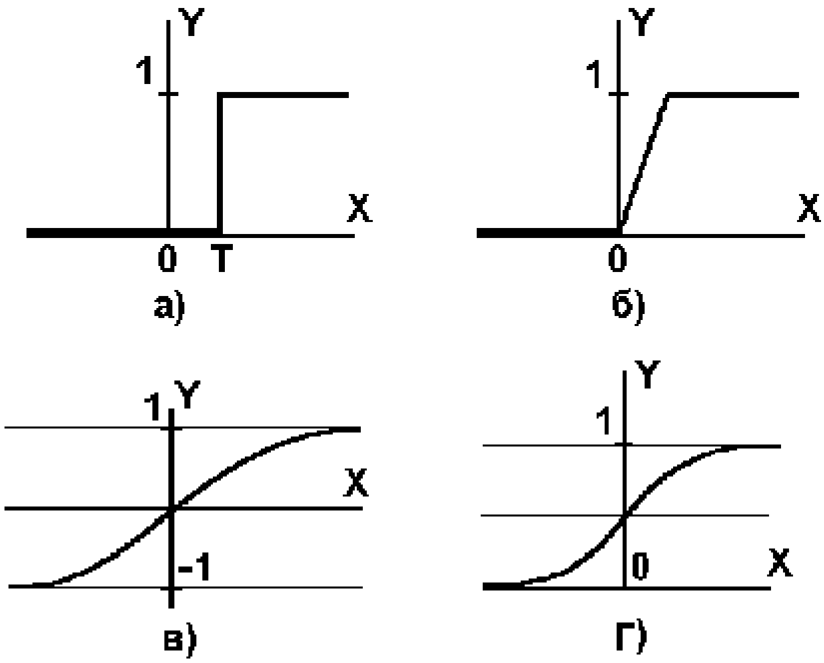


Рис. 3 — Виды функций активации

- пороговый (рис. 3.а),
- кусочно-линейный (рис. 3.б),
- сигмоид (рис. 3.в, 3.г).

Обработка сигналов происходит параллельно, это достигается путем объединения большого числа нейронов в так называемые слои и соединения определенным образом нейронов различных слоев, а также, в некоторых конфигурациях, и нейронов одного слоя между собой, причем обработка взаимодействия всех нейронов ведется послойно.

Выбор структуры нейронной сети осуществляется в соответствии с особенностями и сложностью задачи. Для некоторых классов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных классов, разработчику приходится решать задачу синтеза новой конфигурации. Проблема синтеза искусственной нейронной сети сильно зависит от задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант искусственной нейронной сети получается опытным путем.

1.2 Архитектура нейронной сети

ИНС может рассматриваться как направленный граф со взвешенными связями, в котором искусственные нейроны являются узлами. По архитектуре связей ИНС могут быть сгруппированы в два класса (рис. 4): сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями. Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети. Рекуррентные сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети.

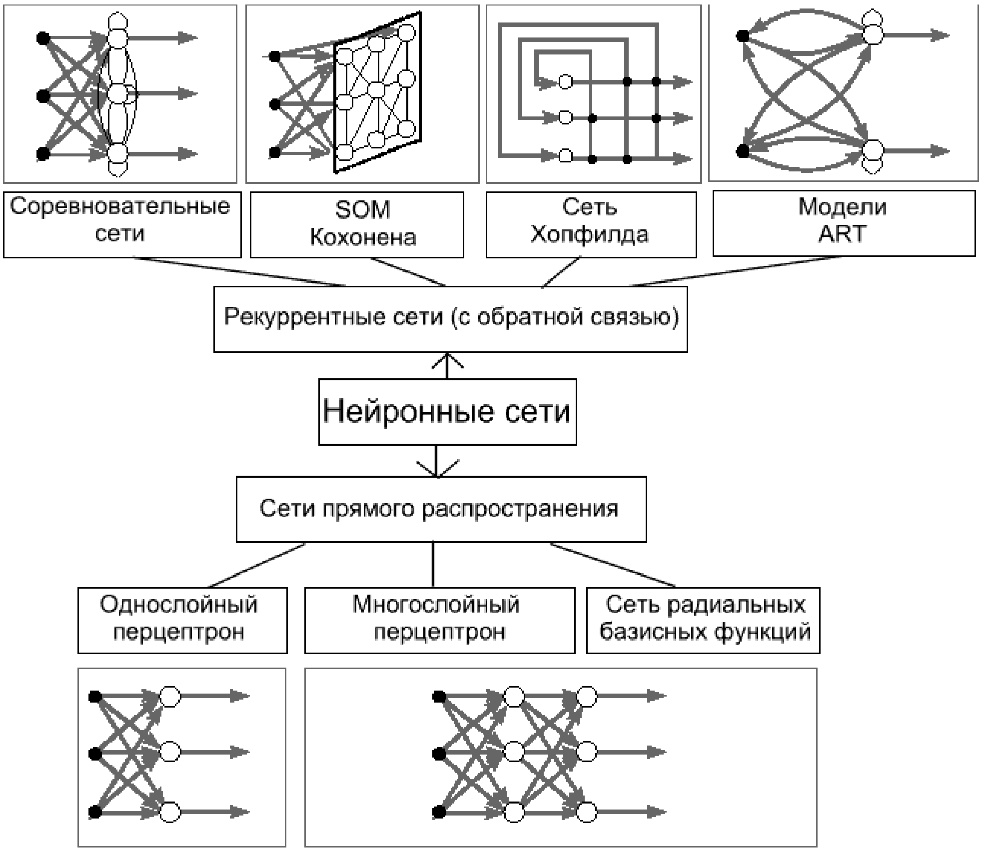


Рис. 4 - архитектура связей ИНС

На рис. 4 представлены типовые сети каждого класса.

1.2.1 Многослойный персептрон (MLP)

Наиболее распространенным классом сетей являются сети прямого распространения, называемые многослойным персептроном (рис. 5).

Эта архитектура сети была предложена в работе Rumelhart, McClelland (1986) и подробно обсуждается почти во всех учебниках по нейронным сетям. Каждый элемент сети строит взвешенную сумму своих входов с поправкой в виде слагаемого и затем пропускает эту величину активации через передаточную функцию, и таким образом получается выходное значение этого элемента. Элементы организованы в послойную топологию с прямой передачей сигнала. Такую сеть легко можно интерпретировать как модель вход-выход, в которой веса и пороговые значения (смещения) являются свободными параметрами модели. Такая сеть может моделировать функцию практически любой степени сложности, причем число слоев и число элементов в каждом слое определяют сложность функции.

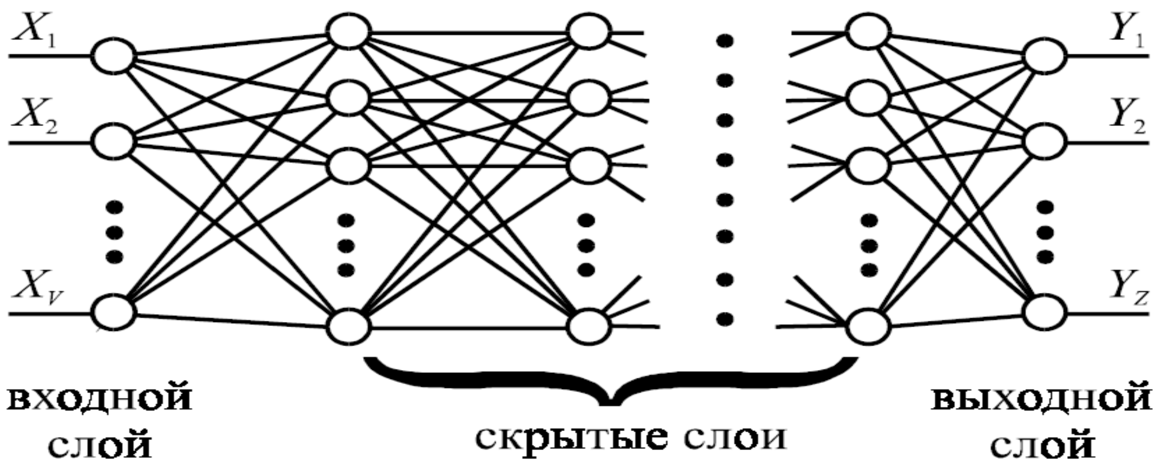


Рис. 5 — многослойный персептрон

2 Обучение нейронных сетей

В контексте ИНС процесс обучения может рассматриваться как настройка архитектуры сети и весов связей для эффективного выполнения специальной задачи. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей выборке. Функционирование сети улучшается по мере итеративной настройки весовых коэффициентов. Свойство сети обучаться на примерах делает их более привлекательными по сравнению с системами, которые следуют определенной системе правил функционирования, сформулированной экспертами.

Для конструирования процесса обучения, прежде всего, необходимо иметь модель внешней среды, в которой функционирует нейронная сеть - знать доступную для сети информацию. Эта модель определяет парадигму обучения. Во-вторых, необходимо понять, как модифицировать весовые параметры сети - какие правила обучения управляют процессом настройки. Алгоритм обучения означает процедуру, в которой используются правила обучения для настройки весов.

Существуют три парадигмы обучения: "с учителем", "без учителя" (самообучение) и смешанная. В первом случае нейронная сеть располагает правильными ответами (выходами сети) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам. Усиленный вариант обучения с учителем предполагает, что известна только критическая оценка правильности выхода нейронной сети, но не сами правильные значения выхода. Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или Корреляции между образцами в системе данных, что позволяет распределить образцы по категориям. При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения [3].

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: емкость, сложность образцов и вычислительная сложность. Под емкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать "переобученность" сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо - на тестовых примерах, подчиненных тому же статистическому распределению. Известны 4 основных типа правил обучения: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования.

- Правило коррекции по ошибке. При обучении с учителем для каждого входного примера задан желаемый выход d. Реальный выход сети y может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала (d-y) для модификации весов, обеспечивающей постепенное уменьшение ошибки. Обучение имеет место только в случае, когда перцептрон ошибается. Известны различные модификации этого алгоритма обучения. Обучение, основанное на коррекции ошибок, более подробно описывается в пункте 2.1 для многослойной сети прямого распространения.

- Обучение Больцмана. Представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термодинамических принципов. Нейронная сеть, созданная на основе обучения Больцмана, получила название машины Больцмана.

В машине Больцмана все нейроны представляются рекуррентными структурами, работающими с бинарными сигналами. Это значит, что они могут находиться во включенном (соответствующем значению + 1) или ыключенном (соответствующем значению 1) состоянии. Такая машина характеризуется функцией энергии E, значение которой определяется конкретными состояниями отдельных нейронов, составляющих эту машину. Это можно описать следующим выражением:

$$E = -\frac{1}{2}\sum_{j}\sum_{k(j\neq k)} w_{kj}x_k x_j, \quad ,$$

где x_j — состояние нейрона j; w_{kj} — синаптический вес связи нейронов j и k. Условие j=k подчёркивает тот факт, что в этой сети нейроны не имеют обратных связей с самими собой [1].

Нейроны машины Больцмана можно разбить на две функциональные группы: видимые и скрытые. Видимые нейроны реализуют интерфейс между сетью и средой ее функционирования, а скрытые работают зависимо от внешней среды.

Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение Корреляций состояний в двух режимах .

- Обучение Хебба. Постулат обучения Хебба является самым старым и самым известным среди всех правил обучения. В основе данного метода лежат два правила:

1. Если два нейрона по обе стороны синапса (соединения) активизируются одновременно (т.е. синхронно), то прочность этого соединения возрастает.

2. Если два нейрона по обе стороны синапса активизируются асинхронно, то такой синапс ослабляется или вообще отмирает.

Функционирующий таким образом синапс называется синапсом Хебба. Синапс Хебба характеризуют следующие четыре свойства:

1. Зависимость от времени. Синапс Хебба зависит от точного времени возникновения редсинаптического и постсинаптического сигналов.

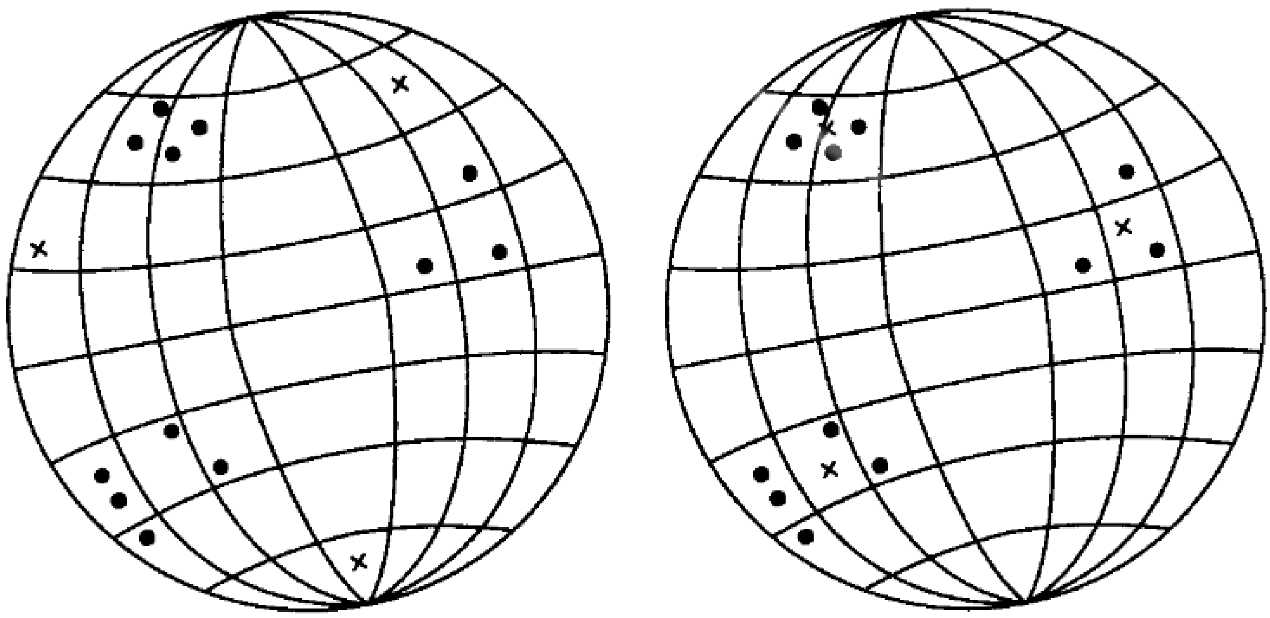
2. Локальность. По своей природе синапс является узлом передачи данных, в котором информационные сигналы (представляющие текущую активность предсинаптических и постсинаптических элементов) находятся в пространственно-временной близости. Эта локальная информация используется синапсом Хебба для выполнения локальных синаптических модификаций, характерных для данного входного сигнала.

3. Интерактивность. Изменения в синапсе Хебба определяются сигналами на обоих его концах. Это значит, что форма обучения Хебба зависит от степени взаимодействия предсинаптического и постсинаптического сигналов (предсказание нельзя построить на основе только одного из этих сигналов).

4. Корреляция. Одна из интерпретаций постулата обучения Хебба состоит в том, что условием изменения эффективности синаптической связи является зависимость между предсинаптическим и постсинаптическим сигналами [1].

- Обучение методом соревнования. В отличие от обучения Хебба, в котором множество выходных нейронов могут возбуждаться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. Это явление известно как правило "победитель берет все". Подобное обучение имеет место в биологических нейронных сетях. Обучение посредством соревнования позволяет кластеризовать входные данные: подобные примеры группируются сетью в соответствии с корреляциями и представляются одним элементом [3].

При обучении модифицируются только веса "победившего" нейрона. Эффект этого правила достигается за счет такого изменения сохраненного в сети образца (вектора весов связей победившего нейрона), при котором он становится чуть ближе ко входному примеру. На рис. 3 дана геометрическая иллюстрация обучения методом соревнования. Входные векторы нормализованы и представлены точками на поверхности сферы. Векторы весов для трех нейронов инициализированы случайными значениями. Их начальные и конечные значения после обучения отмечены X на рис. 6а и 6б соответственно. Каждая из трех групп примеров обнаружена одним из выходных нейронов, чей весовой вектор настроился на центр тяжести обнаруженной группы.



а) б)

Рис. 6 - геометрическая интерпретация процесса конкурентного обучения

Можно заметить, что сеть никогда не перестанет обучаться, если параметр скорости обучения не равен 0. Некоторый входной образец может активизировать другой выходной нейрон на последующих итерациях в процессе обучения. Это ставит вопрос об устойчивости обучающей системы. Система считается устойчивой, если ни один из примеров обучающей выборки не изменяет своей принадлежности к категории после конечного числа итераций обучающего процесса. Один из способов достижения стабильности состоит в постепенном уменьшении до 0 параметра скорости обучения. Однако это искусственное торможение обучения вызывает другую проблему, называемую пластичностью и связанную со способностью к адаптации к новым данным. Эти особенности обучения методом соревнования известны под названием дилеммы стабильности-пластичности.

2.1 Алгоритм обратного распространения

Алгоритм обратного распространения ошибки (back propagation) является одним из методов обучения многослойных нейронных сетей прямого распространения.

Обучение алгоритмом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе входной вектор подается на входной слой нейронной сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синаптические веса сети фиксированы. Во время обратного прохода все синаптические веса обновляются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синаптических связей. Отсюда и название – алгоритм обратного распространения ошибки. Синаптические веса настраиваются с целью максимального приближения выходного сигнала сети к желаемому [3].

В алгоритме обратного распространения вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из данной точки, поэтому если мы "немного" продвинемся по нему, ошибка уменьшится. Последовательность таких шагов (замедляющаяся по мере приближения к дну) в конце концов приведет к минимуму того или иного типа. Определенную трудность здесь представляет вопрос о том, какую нужно брать длину шагов.

При большой длине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или (если поверхность ошибок имеет особо вычурную форму) уйти в неправильном направлении [1]. Классическим примером такого явления при обучении нейронной сети является ситуация, когда алгоритм очень медленно продвигается по узкому оврагу с крутыми склонами, прыгая с одной его стороны на другую. Напротив, при маленьком шаге, вероятно, будет схвачено верное направление, однако при этом потребуется очень много итераций. На практике величина шага берется пропорциональной крутизне склона (так что алгоритм замедляет ход вблизи минимума) с некоторой константой, которая называется скоростью обучения. Правильный выбор скорости обучения зависит от конкретной задачи и обычно осуществляется опытным путем; эта константа может также зависеть от времени, уменьшаясь по мере продвижения алгоритма.

Функция ошибки вычисляется по формуле:

$$E=\frac{1}{2}\sum_{i=1}^{p}\left(Y_i-y_i\right)^2 ,$$

где Y_i - желаемый выход сети, y_i — фактический выход сети

$i=1\ldots p$, p — количество входных значений.

Вектор градиента поверхности ошибок вычисляется по следующей формуле:

$$\nabla E = \left(\frac{\partial E}{\partial E w_1}, \frac{\partial E}{\partial E w_2}, \ldots, \frac{\partial E}{\partial E w_n} \right) ,$$

где n — число весов в сети.

Каждый вес сети обновляется на значение равное:

$$\Delta w_i = -\mu \frac{\partial E}{\partial w_i} ,$$

где μ – скорость обучения нейронной сети.

2.2 Переобучение и обобщение

Одна из наиболее серьезных трудностей изложенного подхода заключается в том, что таким образом мы минимизируем не ту ошибку, которую на самом деле нужно минимизировать, - ошибку, которую можно ожидать от сети, когда ей будут подаваться совершенно новые наблюдения. Иначе говоря, мы хотели бы, чтобы нейронная сеть обладала способностью обобщать результат на новые наблюдения. В действительности сеть обучается минимизировать ошибку на обучающем множестве, и в отсутствие идеального и бесконечно большого обучающего множества это совсем не то же самое, что минимизировать "настоящую" ошибку на поверхности ошибок в заранее неизвестной модели явления [5].

Сильнее всего это различие проявляется в проблеме переобучения, или слишком близкой подгонки. Это явление проще будет продемонстрировать не для нейронной сети, а на примере аппроксимации посредством полиномов, - при этом суть явления абсолютно та же. Полином (или многочлен) - это выражение, содержащее только константы и целые степени независимой переменной. Графики полиномов могут иметь различную форму, причем чем выше степень многочлена (и, тем самым, чем больше членов в него входит), тем более сложной может быть эта форма. Если у нас есть некоторые данные, мы можем поставить цель подогнать к ним полиномиальную кривую (модель) и получить таким образом объяснение для имеющейся зависимости. Наши данные могут быть зашумлены, поэтому нельзя считать, что самая лучшая модель задается кривой, которая в точности проходит через все имеющиеся точки. Полином низкого порядка может быть недостаточно гибким средством для аппроксимации данных, в то время как полином высокого порядка может оказаться чересчур гибким, и будет точно следовать данным, принимая при этом замысловатую форму, не имеющую никакого отношения к форме настоящей зависимости .

Нейронная сеть сталкивается с точно такой же трудностью. Сети с большим числом весов моделируют более сложные функции и, следовательно, склонны к переобучению. Сеть же с небольшим числом весов может оказаться недостаточно гибкой, чтобы смоделировать имеющуюся зависимость. Например, сеть без промежуточных слоев на самом деле моделирует обычную линейную функцию.

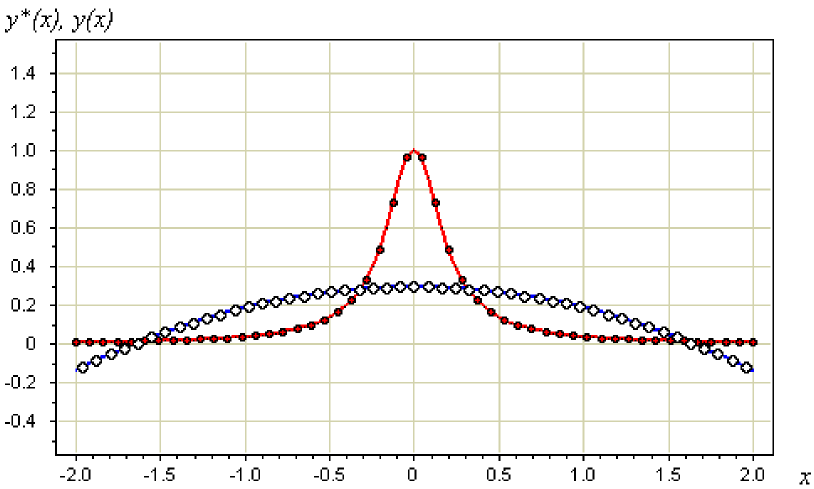


Рис. 7 — недостаточное количество нейронов. Сеть недообучена

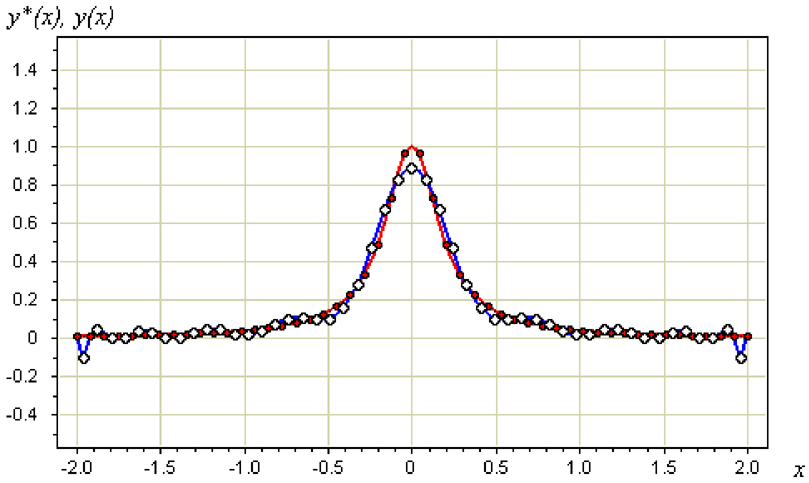


Рис. 8 — оптимальная сложность сети

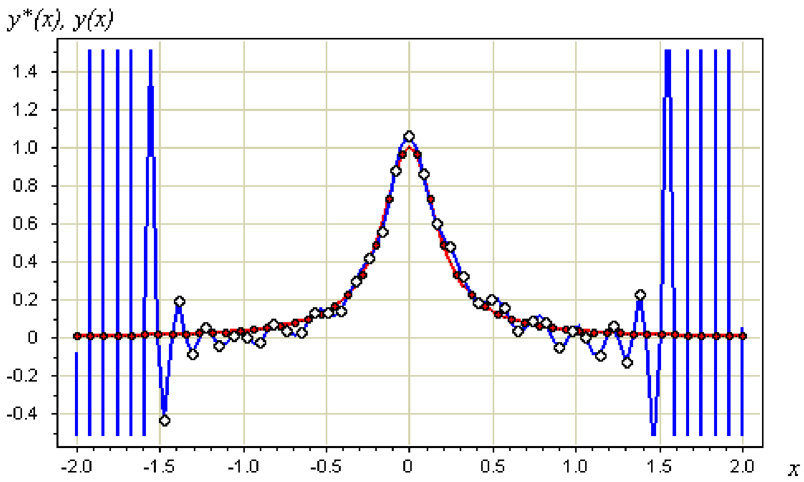


Рис. 9 — избыточное количество нейронов. Сеть переобучена и неустойчива

Почти всегда более сложная сеть дает меньшую ошибку, но это может свидетельствовать не о хорошем качестве модели, а о переобучении.

Для того, чтобы выбрать правильную сложность для сети необходимо использовать механизм контрольной кросс-проверки. Мы резервируем часть обучающих наблюдений и не используем их в обучении по алгоритму обратного распространения. Вместо этого, по мере работы алгоритма, они используются для независимого контроля результата. В самом начале работы ошибка сети на обучающем и контрольном множестве будет одинаковой (если они существенно отличаются, то, вероятно, разбиение всех наблюдений на два множества было неоднородно). По мере того, как сеть обучается, ошибка обучения, естественно, убывает, и, пока обучение уменьшает действительную функцию ошибок, ошибка на контрольном множестве также будет убывать. Если же контрольная ошибка перестала убывать или даже стала расти, это указывает на то, что сеть начала слишком близко аппроксимировать данные и обучение следует остановить. Это явление чересчур точной аппроксимации в процессе обучения и называется переобучением. Если такое случилось, то обычно советуют уменьшить число скрытых элементов и/или слоев, ибо сеть является слишком мощной для данной задачи. Если же сеть, наоборот, была взята недостаточно богатой для того, чтобы моделировать имеющуюся зависимость, то переобучения, скорее всего, не произойдет, и обе ошибки - обучения и проверки - не достигнут достаточного уровня малости.

Описанные проблемы с локальными минимумами и выбором размера сети приводят к тому, что при практической работе с нейронными сетями, как правило, приходится экспериментировать с большим числом различных сетей, порой обучая каждую из них по нескольку раз (чтобы не быть введенным в заблуждение локальными минимумами) и сравнивая полученные результаты. Главным показателем качества результата является здесь контрольная ошибка. При этом, в соответствии с общенаучным принципом, согласно которому при прочих равных следует предпочесть более простую модель, из двух сетей с приблизительно равными ошибками контроля имеет смысл выбрать ту, которая меньше.

Необходимость многократных экспериментов ведет к тому, что контрольное множество начинает играть ключевую роль в выборе модели, то есть становится частью процесса обучения. Тем самым ослабляется его роль как независимого критерия качества модели - при большом числе экспериментов есть риск выбрать "удачную" сеть, дающую хороший результат на контрольном множестве. Для того, чтобы придать окончательной модели должную надежность, часто (по крайней мере, когда объем обучающих данных это позволяет) поступают так: резервируют еще одно - тестовое множество наблюдений. Итоговая модель тестируется на данных из этого множества, чтобы убедиться, что результаты, достигнутые на обучающем и контрольном множествах реальны, а не являются артефактами процесса обучения. Разумеется, для того чтобы хорошо играть свою роль, тестовое множество должно быть использовано только один раз: если его использовать повторно для корректировки процесса обучения, то оно фактически превратится в контрольное множество.

3 Аппроксимация функции

Аппроксимация — это способ нахождения функции, которая наиболее соответствует таблице значений. При аппроксимации выбирается вид функции и определяются параметры этой функции, таким образом, что значения аппроксимирующей функции наиболее приближены к табличным значениям.

На практике информация, как правило, представляется в виде массивов числовых данных, которые являются дискретным представлением функциональных зависимостей, характеризующих исследуемые объекты и процессы разной природы. Работа с такими массивами связана с рядом серьёзных трудностей, возникающих при их использовании, например, в задачах математического моделирования; при восстановлении значений дискретно заданной функции на «неосвещенных» замерами участках, а также при хранении и скоростной передаче по каналам связи больших и сверхбольших по объему массивов. Для преодоления указанных трудностей применяется математическая обработка массивов числовых данных с использованием аппарата приближения (аппроксимации) функций с целью сжатия этих массивов путём замены дискретного представления функциональных зависимостей аналитическими выражениями (аппроксимантами) с небольшим числом параметров коэффициентов.

3.1 Аппроксимация функций нейронной сетью

Из теоремы Стоуна-Вейерштрасса следует, что любая непрерывная функция многих переменных может быть апроксимирована с любой требуемой точностью с помощью многослойного перцептрона с одним скрытым слоем, при достаточном числе нейронов в этом слое [3]. Для аппроксимации многочленов высокой степени количество нейронов на одном скрытом слое может быть очень велико. Рассмотрим модель нейронной сети которая позволяет искать аппроксимирующий полином произвольной степени при относительно небольшом размере сети.

Для аппроксимации полинома аппаратом нейронных сетей будем использовать алгоритм обучения с обратным распространением ошибки, используя произвольные начальные значения весовых коэффициентов. В качестве ошибки используется среднеквадратичное отклонение для пар *(x, Y)* обучающей выборки.

Рассмотрим функциональную сеть полиномов, каждый узел которой соответствует члену $f_i=\alpha_i x^i$ некоторого полинома n-ой степени $g(x)=\alpha_n x^n + \alpha_{n-1}x^{n-1} +...+ \alpha_2 x^2 +\alpha_1 x+\alpha_0$. Значение полинома определяется в узле, суммирующем значения выходов $\alpha_i x^i$ (рис. 10).

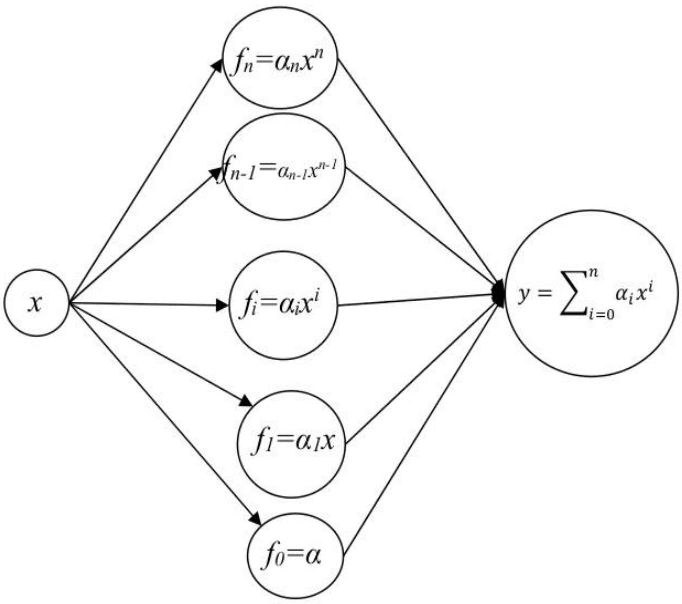


Рис. 10 — структура функциональной сети полинома

Произвольная инициализация весовых коэффициентов влечет за собой существенно разные результаты обучения сети. Действительно, так как каждое слагаемое полинома g(x) линейно, изменение любого из них может привести к изменению ошибки, в частности к ее уменьшению. Данный факт может привести к постоянному изменению весовых коэффициентов одного из узлов сети в ущерб других.

В целях минимизации вероятности лавинного эффекта предлагается использовать два узла с противоположными по знаку коэффициентами для определения слагаемого полинома заданной степени, т.е. узел сети представляющий $f_i = \alpha_i x^i$ заменяется на два совместных узла $f_i' = \alpha_i' x_i$ и $f_i'' = \alpha_i'' x_i$ ($\alpha_i' > 0$, $\alpha_i'' < 0$). Такое разделение позволит организовать «конкуренцию» между парами узлов сети и изменять тот из них, для которого ошибка называется более существенной. В силу линейности коэффициентов можно ожидать резкое увеличение корректирующих значений, что может повлечь «лавинный» эффект изменения искомых параметров. Для ее решения в рамках данной работы предлагается заменить положительные значения α_i на функцию $\alpha_i/(1+e^{-\alpha i})$ (рис. 11), а отрицательные – на $\alpha_i/(1+e^{\alpha i})$ (рис. 12).

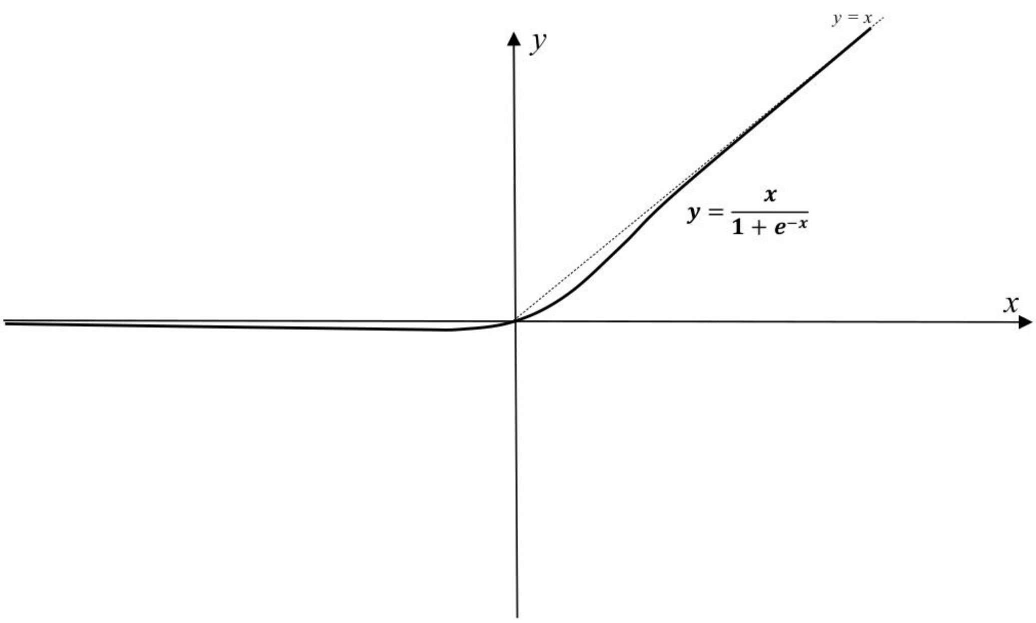


Рис. 11 — График функции положительных коэффициентов.

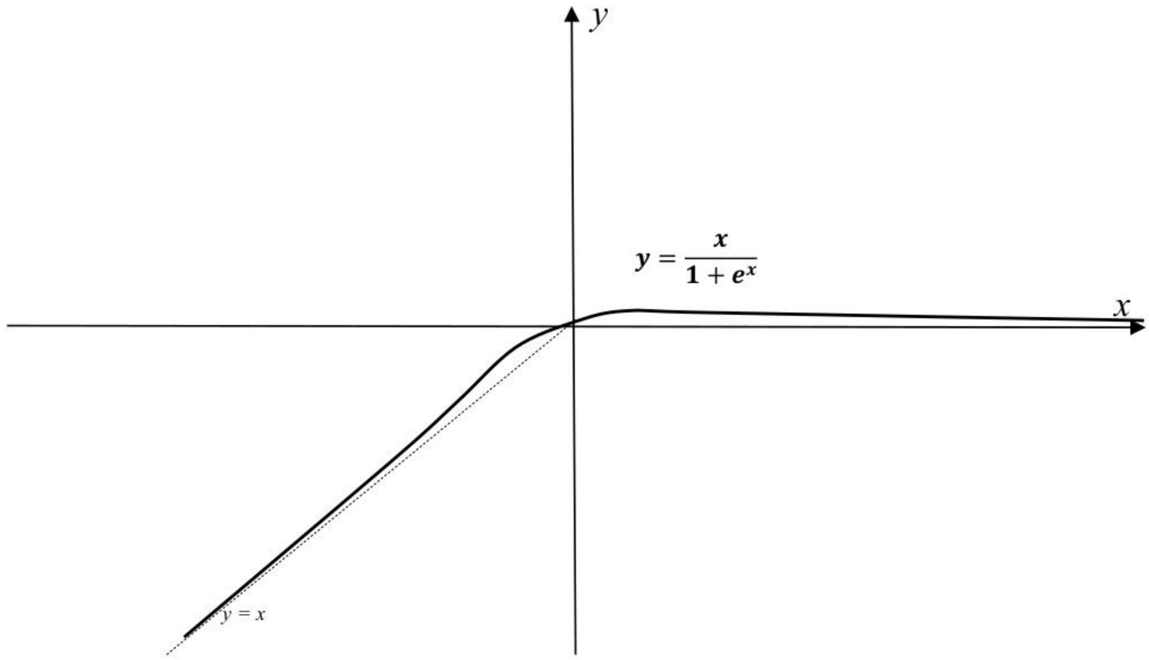


Рис. 12 — График функции отрицательных коэффициентов.

Данное преобразование позволит определить гладкую функцию при корректировке коэффициентов.

Суперпозиция двух полиномов степени n и m определяет полином степени n + m. Основываясь на этом факте, в данной работе предлагается вместо одного слоя для полинома степени n + m использовать два слоя, соответствующие полиномам степени n и m.

Функциональная сеть аппроксимации полинома имеет входной слой, один выход и один или несколько скрытых слоев. В качестве начального объекта исследования определяется сеть с двумя скрытыми слоями. Количество элементов скрытого слоя является параметром сети и определяется степенью аппроксимирующего полинома. Для полинома степени не выше n + m в первом слое определяется n+1 узлов соответствующих одному из элемента полинома. Узлы первого слоя предназначены для определения значений x_i . Каждый узел $N_{i(1)} = x_i$ первого скрытого слоя функциональной сети связан с узлами $N_{j(2)}$ второго слоя (не обязательно со всеми). Входное значение для $N_{j(2)}$ определяется соотношением:

$$z_j = \sum_{i=0}^{n} x^i \left(\frac{\alpha_{ij(1)}}{1+e^{-\alpha_{ij(1)}}} + \frac{\beta_{ij(1)}}{1+e^{\beta_{ij(1)}}}\right)$$

На этапе инициализации сети задаются произвольные значения весовых коэффициентов $w_{ij}(1) > 0$, $w_j(2) > 0$, $v_{ij}(1) < 0$, $v_j(2) < 0$, i=1..n, j=1..m. Поиск параметров осуществляется минимизацией среднеквадратичной ошибки для известных (x, Y) на основе метода градиентного спуска согласно формул:

$$E = \frac{1}{2}(Y-y)^2 = \frac{1}{2}(\Delta y)^2 \to min \tag{1}$$

$$\Delta\alpha_{j(2)} = -\eta \frac{dE}{d\alpha_{j(2)}} = \eta\, \Delta y\ \ {z_j}^j \left(1 + \alpha_{j(2)} - \frac{\alpha_{j(2)}}{1+e^{-\alpha_{j(2)}}}\right)\frac{1}{1+e^{-\alpha_{j(2)}}} \tag{2}$$

$$\Delta\beta_{j(2)} = -\eta \frac{dE}{d\beta_{j(2)}} = \eta\, \Delta y\ \ {z_j}^j \left(1 - \beta_{j(2)} + \frac{\beta_{j(2)}}{1+e^{\beta_{j(2)}}}\right)\frac{1}{1+e^{\beta_{j(2)}}} \tag{3}$$

$$\Delta z_j = -\eta \frac{dE}{dz_j} = \eta\, \Delta y\, j\, {z_j}^{j-1} \left(\frac{\alpha_{j(2)}}{1+e^{-\alpha_{j(2)}}} + \frac{\beta_{j(2)}}{1+e^{\beta_{j(2)}}}\right) \tag{4}$$

$$\begin{aligned}\Delta\alpha_{ij(1)} &= -\eta \frac{dE}{dz_j}\frac{dz_j}{d\alpha_{ij(1)}} \\ &= \Delta z_j x^i \left(1 + \alpha_{ij(1)} - \frac{\alpha_{ij(1)}}{1+e^{-\alpha_{ij(1)}}}\right)\frac{1}{1+e^{-\alpha_{ij(1)}}}\end{aligned} \tag{5}$$

$$\begin{aligned}\Delta\beta_{ij(1)} &= -\eta \frac{dE}{dz_j}\frac{dz_j}{d\beta_{ij(1)}} \\ &= \Delta z_j x^i \left(1 - \beta_{ij(1)} + \frac{\beta_{ij(1)}}{1+e^{\beta_{ij(1)}}}\right)\frac{1}{1+e^{\beta_{ij(1)}}}\end{aligned} \tag{6}$$

Последовательно применяя формулы (1) — (6) для изменения весовых коэффициентов связей между нейронами, необходимо добиться уменьшения ошибки. Результатом работы сети является полином вида:

$$y = \sum_{i=0}^{m}\ {z_i}^i \left(\frac{\alpha_{i(2)}}{1+e^{-\alpha_{i(2)}}} + \frac{\beta_{i(2)}}{1+e^{\beta_{i(2)}}}\right)$$

аппроксимирующий характеристичекую функцию.

После определения аппроксимирующего полинома коэффициенты $a_{j(2)}$, $b_{j(2)}$,

$a_{ij(2)}$, $b_{ij(2)}$ фиксируются как параметры узла, и используются при работе сети.

Предложенная модель нейронной сети позволяет искать аппроксимирующий полином произвольной степени при небольшом размере сети. Степень искомого полинома не превышает $\prod_{t=1}^{k} n_t$, где t – количество скрытых слоев сети, а n_t – количество нейронов каждого скрытого слоя. Степень аппроксимирующего полинома и его свойства, прежде всего, зависят от топологии сети. Максимальная степень достигается в полносвязанных сетях. Например, для поиска полинома 26 степени достаточно иметь два слоя с шестью и четырьмя нейронами в каждом [2].

3.2 Разработка программы

Для реализации модели, предложенной в пункте 3.1 было написано приложение NN_Aproximation с использованием языка программирования C++ и граффической библиотеки SFML.

C++ — мощный кроссплатформенный язык программирования, содержащий средства создания эффективных программ практически любого назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения. Благодаря тому, что C++ унаследовал эффективность языка C, программы написанные на C++ обладают высокой вычислительной производительностью. Вследствие чего выбор и пал на язык программирования C++.

SFML(англ. Simple and Fast Multimedia Library) — свободная кроссплатформенная мультимедийная библиотека. Написана на C++, но доступна также для C, D, Java, Python, Ruby, OCaml, .Net и Go. Представляет собой объектно-ориентированный аналог SDL [4].

3.3 Схема программы

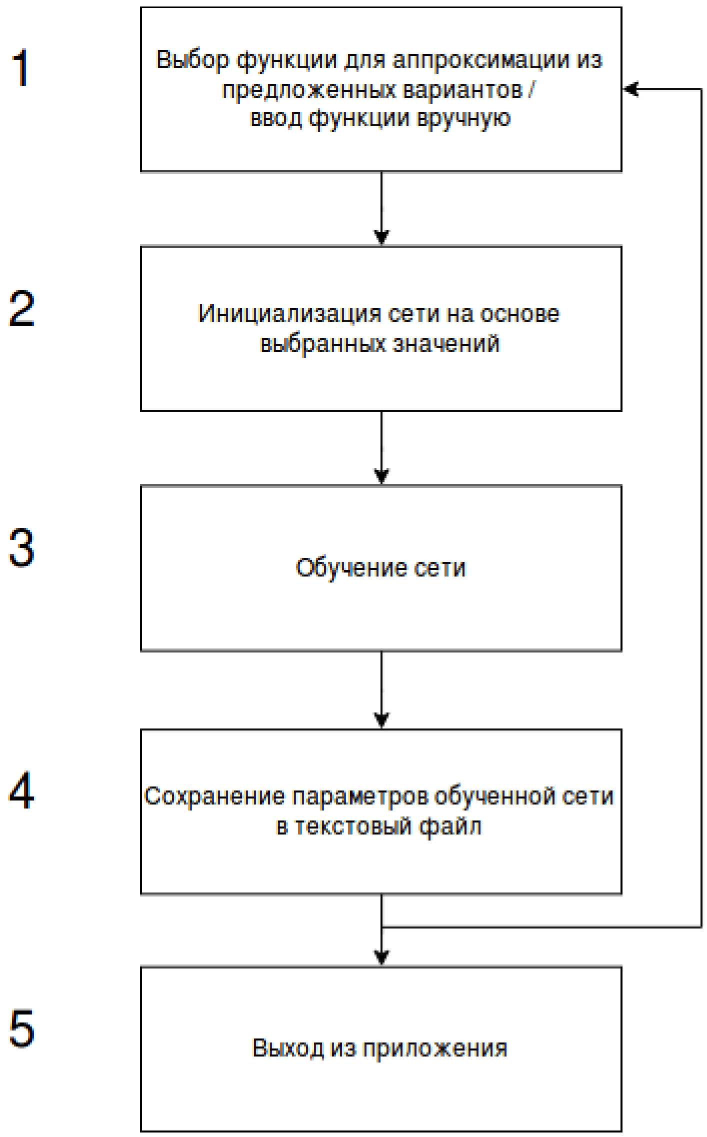


Рис. 13 –Схема программы полиномиальной аппроксимации функций.

Опишем блоки 1-5 схемы программы, представленной на рис. 13:

1. С помощью графического интерфейса пользователь выбирает способ задания функции для аппроксимации.

2. На основании выбора пользователя осуществляется инициализация сети начальными значениями, с помощью метода класса NeuralNet - void init(string expression). Метод принимает на вход строку, содержащую функцию для аппроксимации. На основани функции аппроксимации метод init строит обучающую выборку для нейронной сети и инициализирует веса сети случайными значениями.

3. После начальной инициализации в отдельном потоке запускается процесс обучения нейронной сети. За эту функцию отвечает метод класса NeuralNet – void study(). Работа данного метода основана на алгоритме обратного распространения ошибки. Метод заканчивает работу при достижении необходимой точности аппроксимации либо по прошествии заданного количества эпох обучения сети. Точность аппроксимации и количество эпох задаются в коде программы.

4. После завершения работы метода study значения весов обученной сети сохраняются и записываются в текстовый файл. За эту функцию отвечает метод класса NeuralNet – void save(). При завершении работы метода save отсанавливается выполнение потока, запущенного после начальной инициализации.

5. Даллее пользователь может либо выйти из приложения, либо продолжить аппроксимацию функций.

3.4 Описание интерфейса программы

При запуске программы открывается окно «NN_Approximation»(рис. 14).

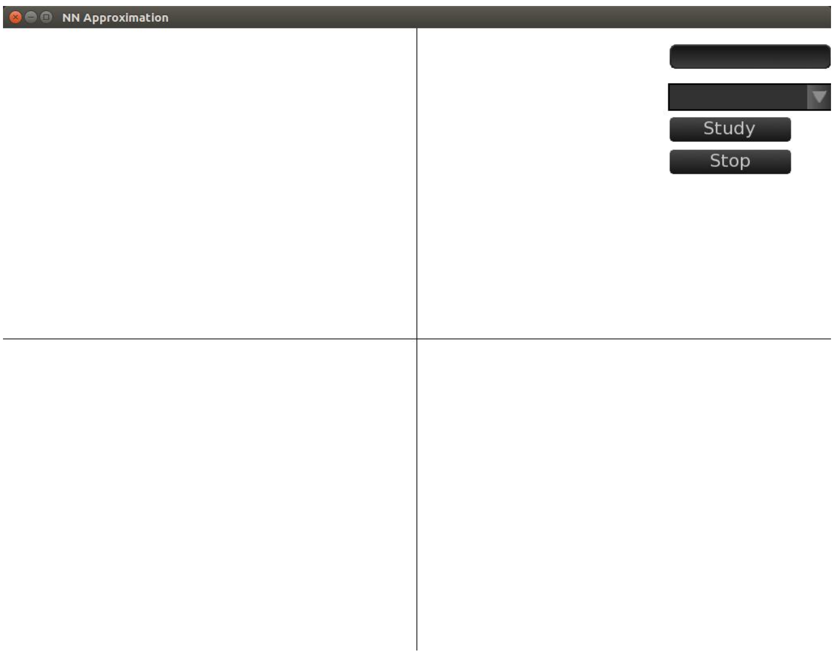


Рис. 14 – Окно «NN_Approximation».

Выбор функции для аппроксимации осуществляется двумя способами:

1. Выбор одной из функции в списке (список помечен красной стрелкой на рис. 15).

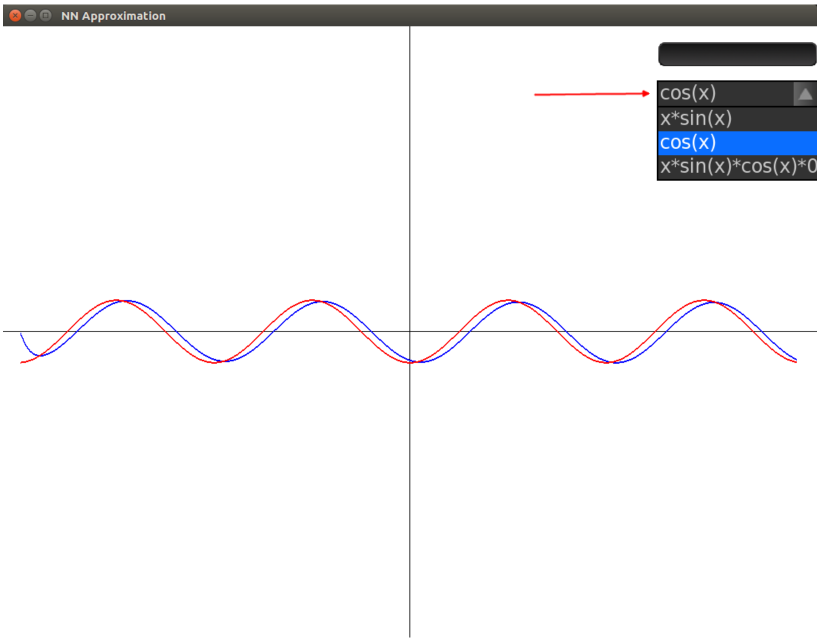


Рис. 15 – Выбор функции из списка.

2. Ввод функции вручную в текстовое поле ввода (поле ввода помечено красной стрелкой на рис. 16).

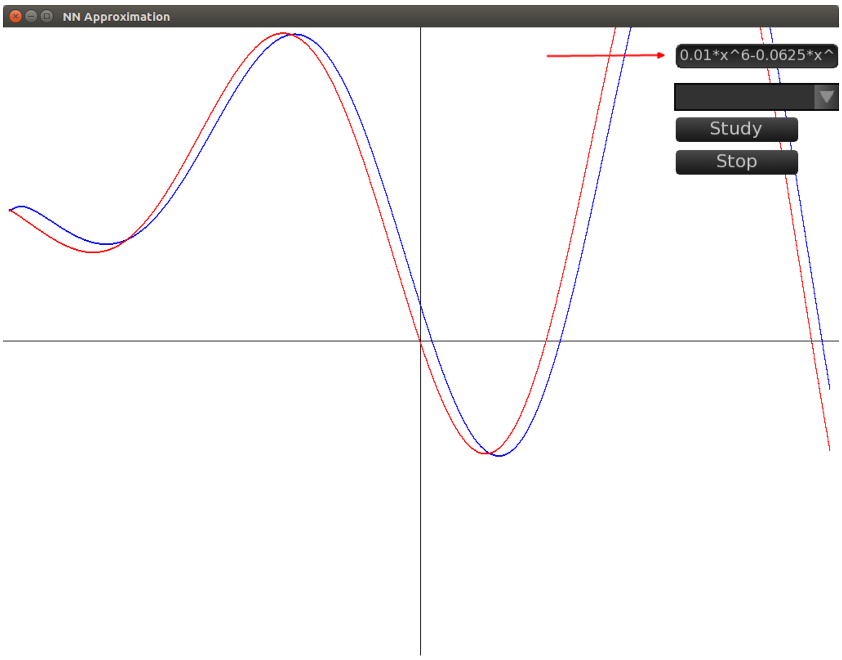


Рис. 16 – Ввод функции вручную.

При нажатии на кнопку «Study» начинается обучение нейронной сети и в окне приложения отрисовывается два граффика(рис. 15, рис. 16). График красного цвета соответствует исходной заданной функции, синего – функции восстановленной нейронной сетью. По нажатию кнопки «Stop» обучение нейронной сети завершается и параметры обученной сети сохраняются в текстовый файл.

3.5 Анализ точности аппроксимации относительно конфигурации нейронной сети

Проведём сравнительный анализ работы двух моделей нейронной сети прямого распространения, описанных ранее: полиномиальная нейронная сеть (ПНС) в пункте 3.2 и многослойный персептрон (МП) пункте 2.1, на примере аппроксимации полинома 6-го порядка: $g(x) = 0{,}01 * x^6 – 0{,}0625 * x^4 + 0{,}25 * x^2 – 4{,}5$.

Рассмотрим результаты работы ПНС и МП на обучающей выборке из 100 пар (x,y) с фикисированным шагом координатной сетки, равным 0.1:

Проведём ряд экспериментов на обучающей выборке из 100 пар (x,y) с фикисированным шагом координатной сетки, равным 0.1 и 1000 эпох обучения:

1. В качестве начальной конфигурации параметров ПНС зададим два скрытых слоя с тремя и двумя нейронами, соответственно. Для МП зададим один скрытый слой с пятью нейронами.

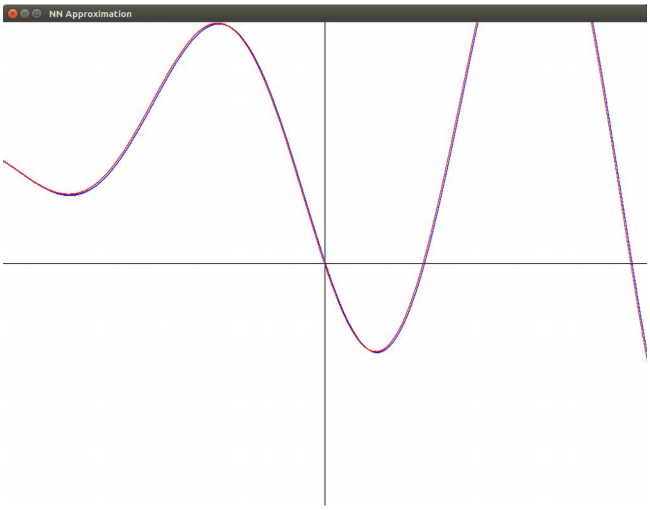


Рис. 17.1
модель сети ПНС. 3+2 нейронов в двух скрытых слоях

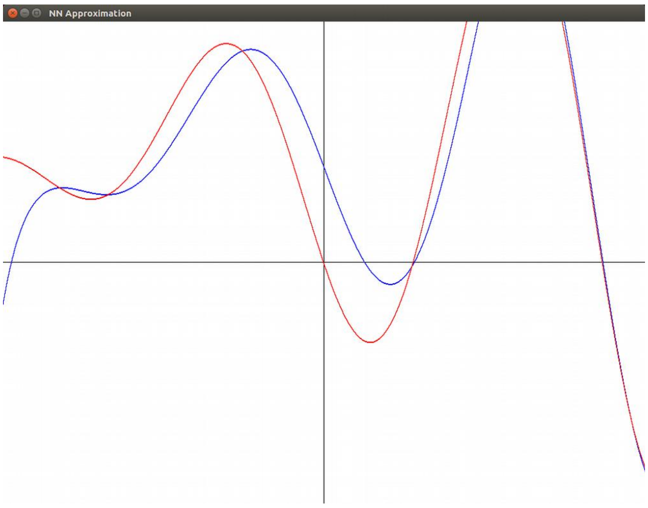


Рис. 17.2
модель сети МП. 5 нейронов на скрытом слое

Пяти нейронов на одном скрытом слое недостаточно и МП (рис. 14.2) не может обучиться. Функция, восстановленная ПНС, практически полностью совпадает с графиком исходной функции(рис. 14.1).

2. Увеличим число нейронов на скрытом слое МП до 10.

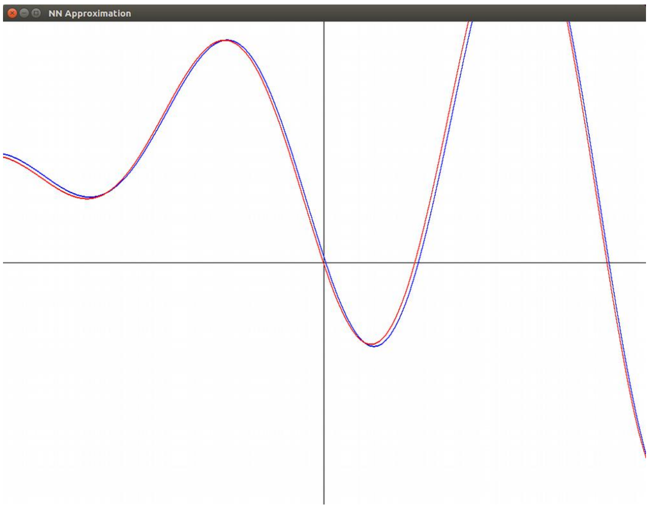


Рис. 18.1
модель сети МП. 10 нейронов на скрытом слое

Точность аппроксимации МП увеличилась(рис. 15.1), но при большем количестве нейронов, чем у ПНС.

3. Проверим работу МП при двух скрытых слоях с тремя и двумя нейронами(рис. 15.2).

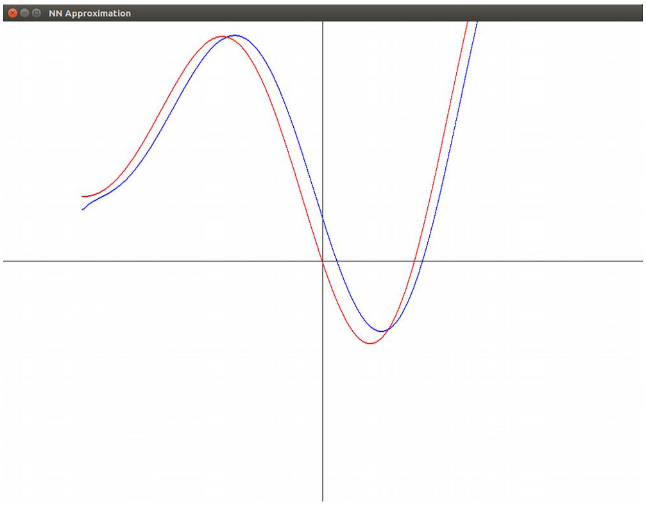


Рис. 18.2
модель сети МП. 3+2 нейронов в двух скрытых слоях

При большом обучающем множестве с небольшим, фиксированным шагом обе модели успешно справляются с аппроксимацией полинома.

Проведём ещё один ряд экспериментов на обучающей выборке из 10 пар (x,y), при этом увеличив шаг до 2:

1. ПНС и МП имеют два скрытых слоя с тремя нейронами и двумя нейронами.

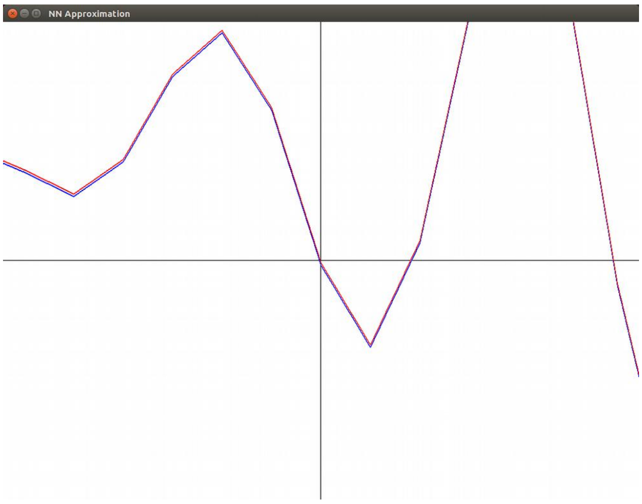


Рис. 19.1
модель сети ПНС. 3+2 нейронов в двух скрытых слоях

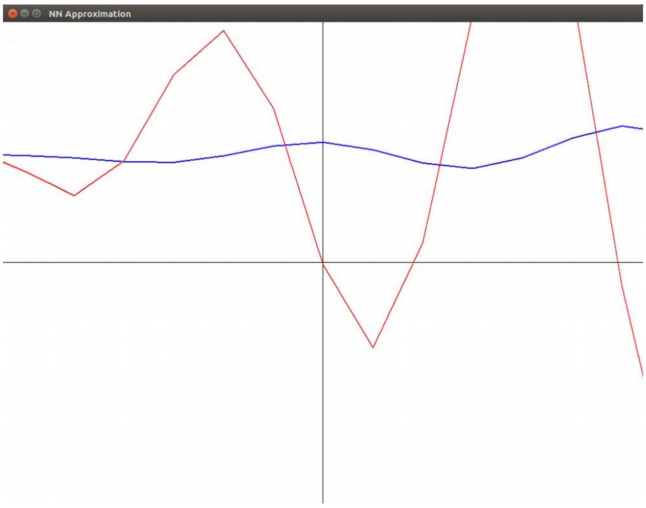


Рис. 19.2
модель сети МП. 3+2 нейронов в двух скрытых слоях

Функция, восстановленная МП вырождается в прямую линию (рис. 16.2). Функция, восстановленная ПНС, практически полностью совпадает с графиком исходной функции(рис. 16.1).

2. Шаг задётся случайным образом в диапазоне [0.1, 3].

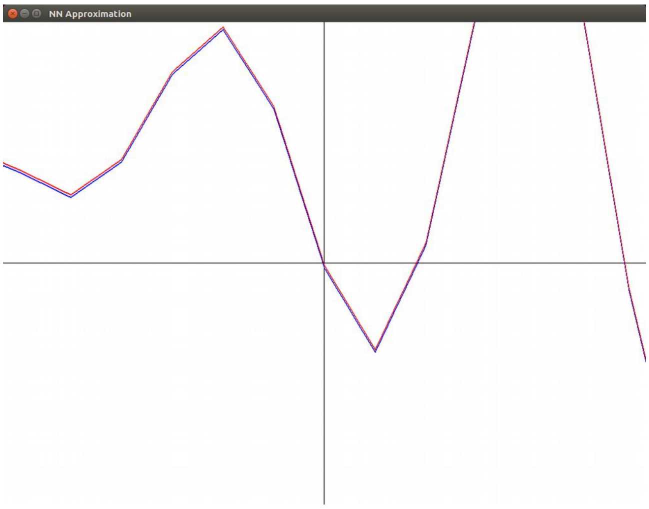


Рис. 20.1
модель сети ПНС. 3+2 нейронов в двух скрытых слоях

Рис. 20.2
модель сети МП. 3+2 нейронов в двух скрытых слоях

Как и в прошлом примере функция, восстановленная МП вырождается в прямую линию(рис. 17.2), а функция, восстановленная ПНС, практически полностью совпадает с графиком исходной функции(рис. 17.1).

Исходя из результатов работы двух рассмотренных моделей можно сказать, что МП в силу своей линейности не может с нужной точностью аппроксимировать функции на обучающей выборке, в которой часть значений зарезервирована. В то время как ПНС, в основе которой лежит функциональная сеть полиномов может подобрать коэффициенты аппроксимирующего полинома даже на случайной выборке.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы были изучены:

1 Основные понятия нейронных сетей

2 Различные архитектуры искусственных нейронных сетей

3 Обучение нейронных сетей методом обратного распространения ошибки

На основе рассмотренной модели поиска аппроксимирующего полинома произвольной степени с помощью нейронных сетей, разработана программа NN_Approximation на языке программирования C++. Программа предназначена для нахождения коэффициентов аппроксимирующего полинома посредством нейронных сетей. Для работы с программой не требуется каких–либо знаний о нейронных сетях и способах аппроксимации функций.

Проведён сравнительный анализ двух моделей нейронной сети прямого распространения(ПНС и МП). Исходя из результатов анализа показано, что модель ПНС может обучаться на неполной обучающей выборке, с высокой точностью аппроксимировать многочлены высокого порядка и применима на практике.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Хайкин С. Нейронные сети. Полный курс. М.: Вильямс, 2006. – 1104 с.

2. Уварова А.В., Подколзин В.В. Аппроксимация параметров сети на основе неполной информации о предметной области: Математические методы и информационно-технические средства: материалы XI Всероссийской научно-практической конференции, г. Краснодар Краснодарский университет МВД России, 2015. – с. 303-308.

3. Информационно-аналитический ресурс, посвященный машинному обучению. URL: http://www.machinelearning.ru/ (дата обращения 02.12.2017).

4. Документация библиотеки SFML. URL: https://www.sfml-dev.org/ (дата обращения 20.10.2017).

5. Christopher M. Bishop. Neural Networks for Pattern Recognition, 1995-482c.