

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 26.07.2022 10:13:58

Уникальный идентификатор:

0b817ca911e6668abb13a5d426d39e5f1c11eabb573e943df4a4851fda56d089

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА «СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ
И ПОИСКОВОГО КОНСТРУИРОВАНИЯ»

РАЗРАБОТКА НЕЙРОННОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ AForge.NET FRAMEWORK

Методические указания к лабораторной работе



Волгоград
2017

УДК 004.032.26 (075)

Рецензент

доцент кафедры «ЭВМ и системы»
канд. техн. наук *А. Е. Андреев*

Печатается по решению редакционно-издательского совета
Волгоградского государственного технического университета

Разработка нейронной сети с использованием AForge.NET Framework : метод. указания к лабораторной работе / сост.: Н. П. Садовникова, Д. С. Парыгин ; ВолгГТУ. – Волгоград, 2016. – 20 с.

Излагаются основные принципы использования нейронных сетей для реализации задач машинного обучения. Описываются этапы построения и обучения нейронных сетей и примеры решения практических задач с использованием библиотеки AForge.NET Framework. Приводятся варианты заданий

Предназначены для студентов очной формы обучения направления 09.03.04 «Программная инженерия» и 09.03.01 «Информатика и вычислительная техника».

© Волгоградский государственный
технический университет, 2017

Оглавление

1. Теоретическая часть.....	4
2. Практическая часть	15
3. Индивидуальное задание	18
4. Контрольные вопросы	19
Рекомендуемая литература.....	20

Цель работы: ознакомиться с основами моделирования с использованием нейронных сетей.

1. Теоретическая часть

Нейронные сети (Neural Networks) - это модели биологических нейронных сетей мозга, в которых нейроны имитируются относительно простыми, часто однотипными, элементами (искусственными нейронами).

Искусственный нейрон (формальный нейрон) - элемент искусственных нейронных сетей, моделирующий некоторые функции биологического нейрона.

Главная функция искусственного нейрона - формировать выходной сигнал в зависимости от сигналов, поступающих на его входы.

В самой распространенной конфигурации входные сигналы обрабатываются адаптивным сумматором, затем выходной сигнал сумматора поступает в нелинейный преобразователь, где преобразуется функцией активации, и результат подается на выход (в точку ветвления).

Общий вид искусственного нейрона приведен на рисунке ниже.

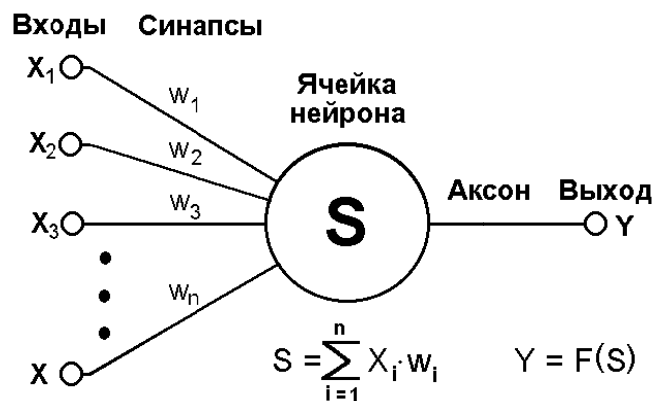


Рис 1. Искусственный нейрон

Нейрон характеризуется текущим состоянием и обладает группой синапсов - однонаправленных входных связей, соединенных с выходами других нейронов.

Нейрон имеет аксон - выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов.

Каждый синапс характеризуется величиной синаптической связи (ее весом w_i).

Текущее состояние нейрона определяется как взвешенная сумма его входов:

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

Выход нейрона есть функция его состояния:

$$y = f(s).$$

Активационная функция, которую также называют характеристической функцией, - это нелинейная функция, вычисляющая выходной сигнал формального нейрона.

Нелинейный преобразователь - это элемент искусственного нейрона, преобразующий текущее состояние нейрона (выходной сигнал адаптивного сумматора) в выходной сигнал нейрона по некоторому нелинейному закону (активационной функции).

Нейронная сеть может быть представлена направленным графом с взвешенными связями, в котором искусственные нейроны являются вершинами, а синаптические связи - дугами.

1.2 Модели нейронных сетей

Рассмотрим наиболее простые модели нейронных сетей: однослойный и многослойный персептрон.

Персептрон

Однослойный персептрон (персептрон Розенблатта) – однослойная нейронная сеть, все нейроны которой имеют жесткую пороговую функцию активации.

Однослойный персептрон имеет простой алгоритм обучения и способен решать лишь самые простые задачи. Эта модель вызвала к себе большой интерес в начале 1960-х годов и стала толчком к развитию искусственных нейронных сетей.

Классический пример такой нейронной сети - однослойный трехнейронный персептрон - представлен на рисунке [2].

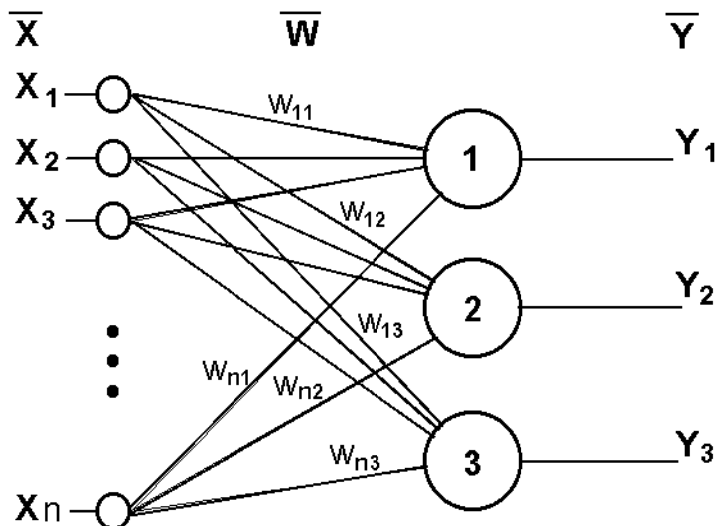


Рис 2. Однослойный трехнейронный персептрон

Сеть, изображенная на рисунке, имеет n входов, на которые поступают сигналы, идущие по синапсам на 3 нейрона. Эти три нейрона образуют единственный слой данной сети и выдают три выходных сигнала.

Многослойный перцептрон

Многослойный перцептрон (MLP) - нейронная сеть прямого распространения сигнала (без обратных связей), в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько слоев.

Первый из таких слоев называют входным, последний - выходным. Эти слои содержат так называемые вырожденные нейроны и иногда в количестве слоев не учитываются. Кроме входного и выходного слоев, в многослойном перцептроне есть один или несколько промежуточных слоев, которые называют скрытыми.

В этой модели перцептрона должен быть хотя бы один скрытый слой. Присутствие нескольких таких слоев оправдано лишь в случае использования нелинейных функций активации.

Пример двухслойного перцептрона представлен на рисунке.

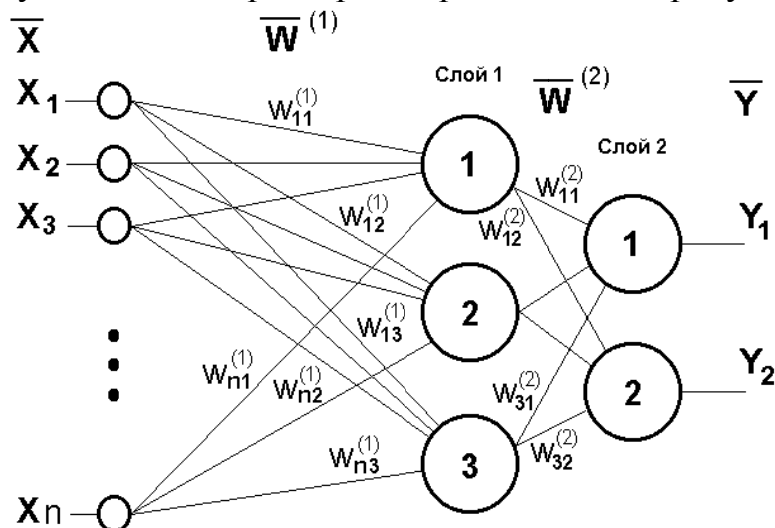


Рис 3. Двухслойный перцептрон

Сеть, изображенная на рисунке, имеет n входов. На них поступают сигналы, идущие далее по синапсам на 3 нейрона, которые образуют первый слой. Выходные сигналы первого слоя передаются двум нейронам второго слоя. Последние, в свою очередь, выдают два выходных сигнала.

1.3 Классификация нейронных сетей

Можно провести следующую классификацию нейронных сетей:



Рис 4. Классификация нейронных сетей

Характер обучения

Классификация нейронных сетей по характеру обучения делит их на:

- нейронные сети, использующие обучение с учителем;
- нейронные сети, использующие обучение без учителя.

Настройка весов

- сети с фиксированными связями – весовые коэффициенты нейронной сети выбираются сразу, исходя из условий задачи;
- сети с динамическими связями – для них в процессе обучения происходит настройка синаптических весов.

Тип входной информации

- аналоговая – входная информация представлена в форме действительных чисел;
- двоичная – вся входная информация в таких сетях представляется в виде нулей и единиц.

Применяемая модель нейронной сети

- Сети прямого распространения – все связи направлены строго от входных нейронов к выходным. К таким сетям относятся, например:

простейший персептрон (разработанный Розенблаттом) и многослойный персептрон.

– Рекуррентные нейронные сети – сигнал с выходных нейронов или нейронов скрытого слоя частично передается обратно на входы нейронов входного слоя.

– Радиально базисные функции – вид нейронной сети, имеющий скрытый слой из радиальных элементов и выходной слой из линейных элементов. Сети этого типа довольно компактны и быстро обучаются. Предложены в работах Broomhead and Lowe (1988) и Moody and Darkin (1989). Радиально базисная сеть обладает следующими особенностями: один скрытый слой, только нейроны скрытого слоя имеют нелинейную активационную функцию и синаптические веса входного и скрытого слоев равны единицы.

– Самоорганизующиеся карты или Сети Кохонена – такой класс сетей, как правило, обучается без учителя и успешно применяется в задачах распознавания. Сети такого класса способны выявлять новизну во входных данных: если после обучения сеть встретится с набором данных, непохожим ни на один из известных образцов, то она не сможет классифицировать такой набор и тем самым выявит его новизну. Сеть Кохонена имеет всего два слоя: входной и выходной, составленный из радиальных элементов.

Обучение нейронной сети

Алгоритм обучения нейронной сети – это набор математических действий, который позволяет по вектору ошибки вычислить такие поправки для весов нейронной сети, чтобы суммарная ошибка (для контроля процесса обучения обычно используют сумму квадратов ошибок по всем выходам) уменьшилась.

Существуют два концептуальных подхода к обучению нейронных сетей: обучение с учителем и обучение без учителя.

В общем виде алгоритм обучения с учителем будет выглядеть следующим образом:

1. Инициализировать синаптические веса маленькими случайными значениями.
2. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
3. Вычислить выход сети.
4. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
5. Подкорректировать веса сети для минимизации ошибки.
6. Повторять шаги с 2 по 5 для каждой пары обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Обучение нейронной сети без учителя является намного более правдоподобной моделью обучения с точки зрения биологических корней искусственных нейронных сетей. Обучающее множество состоит лишь из входных векторов. Алгоритм обучения нейронной сети подстраивает веса сети так, чтобы получались согласованные выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы.

Функции активации в нейронных сетях

Функция активации (активационная функция, функция возбуждения) – функция, вычисляющая выходной сигнал искусственного нейрона. В качестве аргумента принимает сигнал Y , получаемый на выходе входного сумматора Σ . Наиболее часто используются следующие функции активации.

1. Жесткая пороговая функция

Простая кусочно-линейная функция (рис.5). Если входное значение меньше порогового, то значение функции активации равно минимально допустимому, иначе – максимально допустимому.

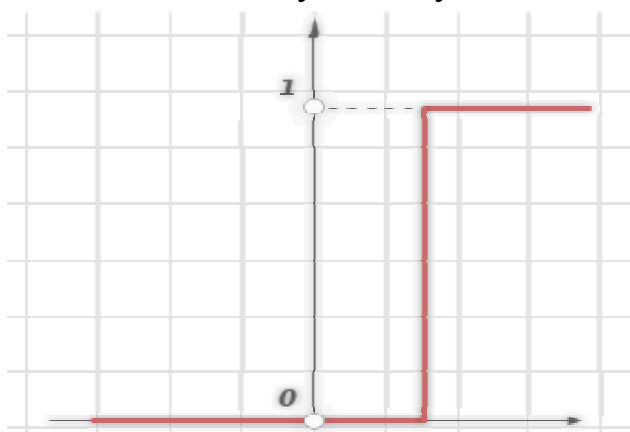


Рис 5. Простая кусочно-линейная функция.

2. Линейный порог или гистерезис

Данный вид функции активации имеет два линейных участка, где функция активации тождественно равна минимально допустимому и максимально допустимому значению и есть участок, на котором функция строго монотонно возрастает (рис.6).

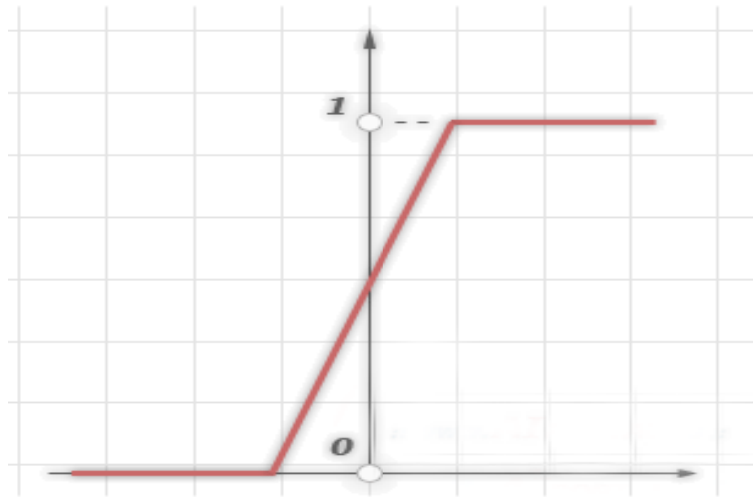


Рис 6. Линейный порог или гистерезис

3. Сигмоидальная функция или сигмоид

Монотонно возрастающая всюду дифференцируемая S-образная нелинейная функция с насыщением (рис.7). Сигмоид позволяет усиливать слабые сигналы и не насыщаться от сильных сигналов. Гроссберг (1973 год) обнаружил, что подобная нелинейная функция активации решает поставленную им дилемму шумового насыщения.

Слабые сигналы нуждаются в большом сетевом усилении, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления могут привести к насыщению выхода шумами усилителей, которые присутствуют в любой физически реализованной сети. Сильные входные сигналы в свою очередь также будут приводить к насыщению усилительных каскадов, исключая возможность полезного использования выхода. Примером сигмоидальной функции активации может служить логистическая функция, задаваемая следующим выражением:

$$OUT = \frac{1}{1 + \exp(-\alpha Y)}$$

, где α – параметр наклона сигмоидальной функции активации. Изменяя этот параметр, можно построить функции с различной крутизной.

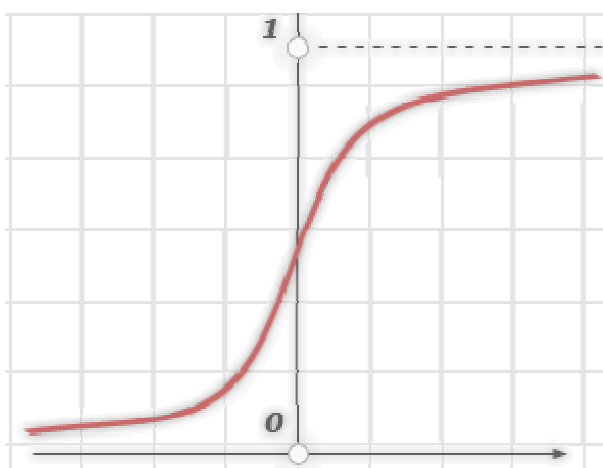


Рис 7. Сигмоидальная функция

Еще одним примером сигмоидальной функции активации является гиперболический тангенс (рис.8), задаваемая следующим выражением:

$$OUT = th\left(\frac{Y}{\alpha}\right)$$

, где α – это также параметр, влияющий на наклон сигмоидальной функции.

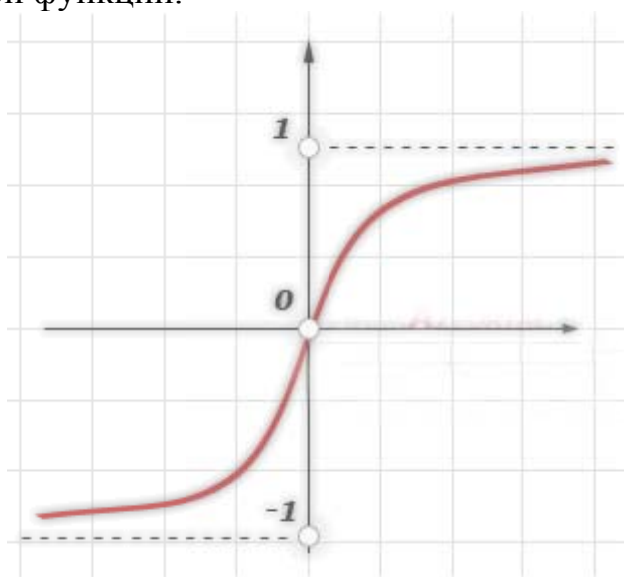


Рис 8. Гиперболический тангенс

Применение нейронных сетей для задач классификации

Задача классификации представляет собой задачу отнесения образца к одному из нескольких попарно не пересекающихся множеств. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется *выборкой*. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный *классифицировать* произвольный объект из исходного множества.

Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать в решении задач классификации.

Подготовка исходных данных

Для построения классификатора необходимо определить, какие параметры влияют на принятие решения о том, к какому классу принадлежит образец. При этом могут возникнуть две проблемы. Во-первых, если количество параметров мало, то может возникнуть ситуация, при которой один и тот же набор исходных данных соответствует примерам, находящимся в разных классах. Тогда невозможно обучить нейронную сеть, и система не будет корректно работать (невозможно найти минимум, который соответствует такому набору исходных данных). Исходные данные обязательно должны быть непротиворечивы. Для решения этой проблемы необходимо увеличить размерность пространства признаков (количество компонент входного вектора, соответствующего образцу). Но при увеличении размерности пространства признаков может возникнуть ситуация, когда число примеров может стать недостаточным для обучения сети, и она вместо обобщения просто запомнит примеры из обучающей выборки и не сможет корректно функционировать. Таким образом, при определении признаков необходимо найти компромисс с их количеством.

Далее необходимо определить способ представления входных данных для нейронной сети, т.е. определить способ нормирования. Нормировка необходима, поскольку нейронные сети работают с данными, представленными числами в диапазоне 0..1, а исходные данные могут иметь произвольный диапазон или вообще быть нечисловыми данными. При этом возможны различные способы, начиная от простого линейного преобразования в требуемый диапазон и заканчивая многомерным анализом параметров и нелинейной нормировкой в зависимости от влияния параметров друг на друга.

Кодирование выходных значений

Задача классификации при наличии двух классов может быть решена на сети с одним нейроном в выходном слое, который может принимать одно из двух значений 0 или 1, в зависимости от того, к какому классу принадлежит образец. При наличии нескольких классов возникает проблема, связанная с представлением этих данных для выхода сети. Наиболее простым способом представления выходных данных в таком случае является вектор, компоненты которого соответствуют различным номерам классов. При этом i -я компонента вектора соответствует i -му классу. Все остальные компоненты при этом устанавливаются в 0. Тогда, например, второму классу будет соответствовать 1 на 2 выходе сети и 0 на

остальных. При интерпретации результата обычно считается, что номер класса определяется номером выхода сети, на котором появилось максимальное значение.

Выбор объема сети

Правильный выбор объема сети имеет большое значение. Построить небольшую и качественную модель часто бывает просто невозможно, а большая модель будет просто запоминать примеры из обучающей выборки и не производить аппроксимацию, что, естественно, приведет к некорректной работе классификатора. Существуют два основных подхода к построению сети – конструктивный и деструктивный. При первом из них вначале берется сеть минимального размера, и постепенно увеличивают ее до достижения требуемой точности. При этом на каждом шаге ее заново обучают. Также существует так называемый метод каскадной корреляции, при котором после окончания эпохи происходит корректировка архитектуры сети с целью минимизации ошибки. При деструктивном подходе вначале берется сеть завышенного объема, и затем из нее удаляются узлы и связи, мало влияющие на решение. При этом полезно помнить следующее правило: число примеров в обучающем множестве должно быть больше числа настраиваемых весов. Иначе вместо обобщения сеть просто запомнит данные и утратит способность к классификации – результат будет неопределенным для примеров, которые не вошли в обучающую выборку.

Выбор архитектуры сети

При выборе архитектуры сети обычно опробуется несколько конфигураций с различным количеством элементов. При этом основным показателем является объем обучающего множества и обобщающая способность сети. Обычно используется алгоритм обучения Back Propagation (обратного распространения) с подтверждающим множеством.

Алгоритм построения классификатора на основе нейронных сетей

1. Работа с данными
 - Составить базу данных из примеров, характерных для данной задачи
 - Разбить всю совокупность данных на два множества: обучающее и тестовое (возможно разбиение на 3 множества: обучающее, тестовое и подтверждающее).
2. Предварительная обработка
 - Выбрать систему признаков, характерных для данной задачи, и преобразовать данные соответствующим образом для подачи на вход сети

(нормировка, стандартизация и т.д.). В результате желательно получить линейно отделяемое пространство множества образцов.

- Выбрать систему кодирования выходных значений (классическое кодирование, 2 на 2 кодирование и т.д.)

3. Конструирование, обучение и оценка качества сети

- Выбрать топологию сети: количество слоев, число нейронов в слоях и т.д.

- Выбрать функцию активации нейронов

- Выбрать алгоритм обучения сети

- Оценить качество работы сети на основе подтверждающего множества или другому критерию, оптимизировать архитектуру (уменьшение весов, прореживание пространства признаков)

- Остановиться на варианте сети, который обеспечивает наилучшую способность к обобщению и оценить качество работы по тестовому множеству

4. Использование и диагностика

- Выяснить степень влияния различных факторов на принимаемое решение (эвристический подход).

- Убедиться, что сеть дает требуемую точность классификации (число неправильно распознанных примеров мало)

- При необходимости вернуться на этап 2, изменив способ представления образцов или изменив базу данных.

- Использовать сеть для решения поставленной задачи.

2. Практическая часть

2.1. Цель практической части лабораторной работы ознакомится с основами построения нейронных сетей и их обучения.

Существует множество инструментов, позволяющих упростить задачу создания и обучения нейронной сети. В данном случае предлагается использовать AForge.NET framework для C# (<http://www.aforgenet.com/framework/>).

AForge.NET Framework предназначен для разработчиков и исследователей в области искусственного интеллекта. Данный фреймворк позволяет реализовывать нейронные сети, генетические алгоритмы, методы машинного обучения и т.д.

AForge.NET framework поддерживает библиотеку нейронных сетей, которая содержит набор классов, предназначенных для создания различного типа искусственных нейронных сетей и их обучения с ориентацией на решение определенной задачи, например задачи распознавания, классификации, прогнозирования и т.д.

Для разработки нейронной сети нам необходимо подключить пространство имён AForge.Neuro. Пространство имен AForge.Neuro содержит интерфейсы и классы для реализации нейронных сетей, они позволяют создавать популярные нейронно-сетевые архитектуры, классы для обучения этих сетей.

Таблица 1. Описание классов AForge.Net

Класс	Описание
ActivationLayer	Активация слоя
ActivationNetwork	Активация сети
ActivationNeuron	Активация нейрона
BipolarSigmoidFunction	Биполярная сигмовидная функция активации
DistanceLayer	Слой дистанционных нейронов
DistanceNetwork	Сеть дистанционных нейронов
DistanceNeuron	Дистанционные нейроны
Layer	Класс «слой нейрона»
Network	Класс «нейронная сеть»
Neuron	Класс «нейрон»
SigmoidFunction	Сигмовидная функция активации
ThresholdFunction	Пороговая функция активации

С методами и свойствами данных классов, а так же с примерами их применения можно познакомиться в руководстве пользователя AForge.NET framework.

Рассмотрим пример обучения нейронной сети.

Приложение «Perceptron Classifier» реализовано с помощью AForge.NET framework. Приложение содержит один нейрон (персептрон) с пороговой функцией. Реализовано обучение для решения задачи - классификации данных, принадлежащих двум классам. Персептрон обучается для задачи классификации на примере таблицы для логического «И» в первом примере и логического «ИЛИ» во втором и третьем примерах. Эти примеры демонстрируют 100% обучаемость нейронной сети. На входе вектор значений $\{X_1, X_2, \dots, X_n\}$, на выходе его принадлежность к Class. На графике под таблицей с данными мы видим наглядную демонстрацию линейной разделимости классов. На примере логической таблицы получаем (рис.9-11):

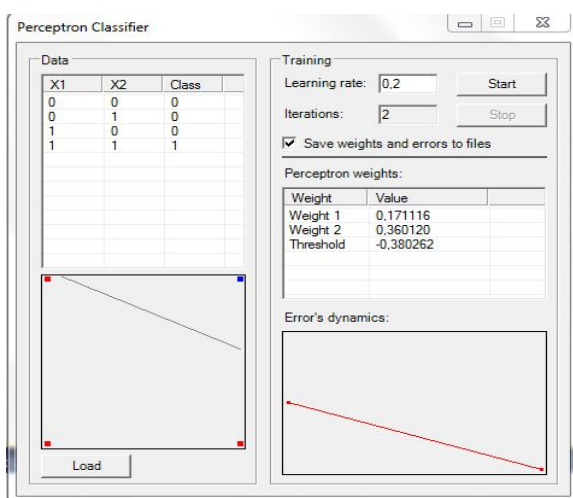


Рис. 9 Обучение персептрона функции «И»

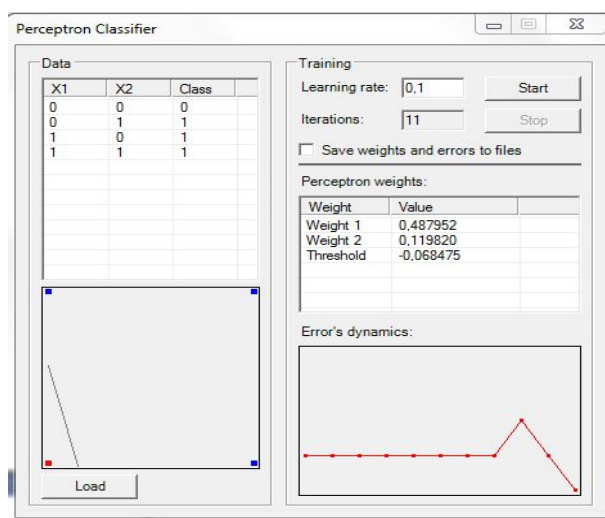


Рис. 10. Обучение персептрона функции «ИЛИ»

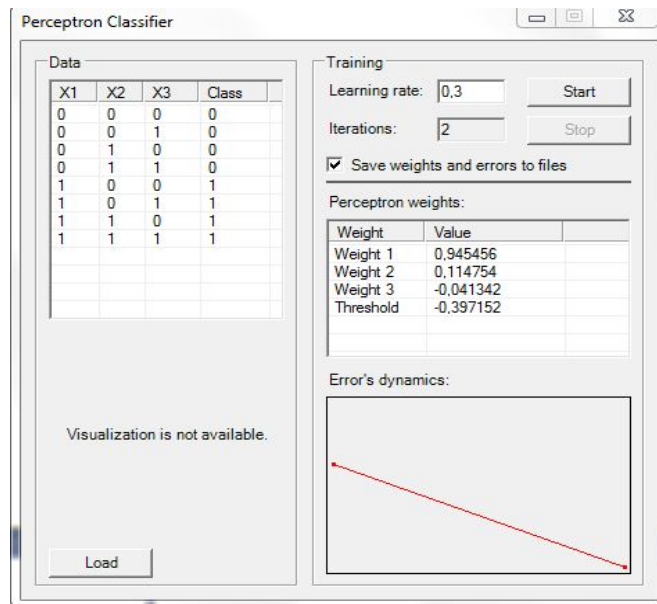


Рис. 11. Обучение персептрона функции «ИЛИ» по трём признакам

Рассмотрим подробнее листинг данной программы.

Реализация персептрона в данной программе с использованием AForge.NET framework происходит по следующему алгоритму:

1. Создаётся персептрон, для этого создаём объект класса активации нейронной сети. Объект данного класса является базой для многослойной сети с функцией активации.

```
ActivationNetwork network = new ActivationNetwork(new
ThresholdFunction( ), variables, 1 );
```

Первый параметр — это тип функции активации. В примере используется пороговая функция активации задаваемая условием « $f(x) = 1$, if $x \geq 0$, otherwise 0» и выходным значением [0,1]. Данная функция создаётся объектом ThresholdFunction();

Второй параметр – количество входных переменных;

Третий параметр – количество нейронов в первом слое, в данном примере используется один нейрон.

2. Активируем нейрон. Данный объект вычисляет взвешенную сумму своих входов, добавляет пороговое значение, а затем применяет функцию активации

Активируем созданную нами сети для расчета весов:

```
ActivationNeuron neuron = network.Layers[0].Neurons[0] as
ActivationNeuron;
```

Переходим к обучению нейронной сети. Для этого создаём объект учитель. В AForge.NET framework для каждой функции активации реализован свой класс учителя. В примере используется пороговая функция, поэтому необходимо создать объект класса PerceptronLearning. Передаваемым параметром является наша обучаемая нейронная сеть.

```
PerceptronLearning teacher = new PerceptronLearning(network);  
// мы можем так же менять скорость обучения, передавая каждый  
раз новый параметр  
teacher.LearningRate = learningRate;
```

3. Индивидуальное задание

Создать свою нейронную сеть и обучить её по заданному алгоритму (алгоритмы представлены отдельными классами и описаны в руководстве пользователя фреймворка) в соответствии с номером варианта, используя для этого классы AForge.NET Framework. Решить простую задачу классификации на основе обученной нейронной сети. Предметная область задачи, данные для обучения, данные для тестирования выбираются учащимся самостоятельно. Количество входных признаков для классификации не менее 3 и не более 10. Количество классов, на которые распределяется входной вектор не менее 3.

Входные данные: файл с данными для обучения нейронной сети

Выходные данные: файл с результатом работы программы.

Содержание протокола: номер варианта, цель работы, листинг программы, описание алгоритма обучения нейронной сети, данные для обучения нейронной сети, выходной файл с примером решения задачи классификации, выходной файл со значениями весов, анализ полученных результатов, выводы.

Номер варианта соответствует порядковому номеру списка группы.

Вариант 1.

Функция активации (пороговая функция): сигмовидная функция.

Алгоритм обучения нейронной сети: BackPropagationLearning.

Вариант 2.

Функция активации (пороговая функция): сигмовидная функция.

Алгоритм обучения нейронной сети: DeltaRuleLearning.

Вариант 3.

Функция активации (пороговая функция): сигмовидная функция.

Алгоритм обучения нейронной сети: EvolutionaryLearning.

Вариант 4.

Функция активации (пороговая функция): биполярная сигмовидная функция. Алгоритм обучения нейронной сети: EvolutionaryLearning.

Вариант 5.

Функция активации (пороговая функция): биполярная сигмовидная функция. Алгоритм обучения нейронной сети: BackPropagationLearning.

Вариант 6.

Функция активации (пороговая функция): биполярная сигмовидная функция. Алгоритм обучения нейронной сети: DeltaRuleLearning.

Вариант 7.

Функция активации (пороговая функция): сигмовидная функция. Алгоритм обучения нейронной сети: ResilientBackpropagationLearning.

Вариант 8.

Функция активации (пороговая функция): биполярная сигмовидная функция. Алгоритм обучения нейронной сети: ResilientBackpropagationLearning.

4. Контрольные вопросы

1. Дайте определение понятию нейронная сеть.
2. Из каких элементов состоит искусственный нейрон? Какую функцию они выполняют?
3. Дайте определение понятию функция активации. Какие виды функций активации бывают?
4. Что такое «слой нейронной сети»? В чем особенность многослойной сети?
5. Персептрон. Определение. Особенности. Архитектура.
6. Многослойный персептрон. Определение. Особенности. Архитектура.
7. Классификация нейронных сетей.
8. Опишите процесс обучения нейронной сети.
9. Какие существуют концептуальные подходы к обучению нейронных сетей?
10. Алгоритм построения классификатора на основе нейронных сетей.

Рекомендуемая литература

1. Уотермен Ф. Нейрокомпьютерная техника: теория и практика: Пер. с англ. – М.: Радио и связь, 1992.
2. Neuroschool [Электронный ресурс] Режим доступа: <http://neuroschool.narod.ru>
3. Minsky M., and Papert S., 1969. Perceptrons. Cambridge, MA: MIT Press. (Русский перевод: Минский М.Л., Пейперт С. Перцептроны). – М. Мир. – 1971.

Учебное издание

Составители:

Наталья Петровна Садовникова
Данила Сергеевич Парыгин

РАЗРАБОТКА НЕЙРОННОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ AFORGE.NET FRAMEWORK

Методические указания к лабораторной работе

Темплан 2017 г. (учебно-методическая литература). Поз. № 29.
Подписано в печать 28.08.2017. Формат 60x84 1/16. Бумага офсетная.
Гарнитура Times. Печать офсетная. Усл. печ. л. 1,16.
Тираж 10 экз. Заказ

Волгоградский государственный технический университет.
400005, г. Волгоград, просп. В. И. Ленина, 28, корп. 1.

Отпечатано в типографии ИУНЛ ВолгГТУ.
400005, г. Волгоград, просп. В. И. Ленина, 28, корп. 7.