

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И
СПОРТА УКРАИНЫ
ДОНБАССКАЯ ГОСУДАРСТВЕННАЯ МАШИНОСТРОИТЕЛЬНАЯ
АКАДЕМИЯ

КАФЕДРА
КОМПЬЮТЕРНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе по дисциплине
“ Методы и системы искусственного интеллекта ”

Выполнил:

Студент группы ИТ 12-1

Ефремов М.А. _____
(подпись)

Руководитель:

Богданова Л.М. _____
(подпись)

| Дата защиты работы | Оценка | Подпись преподавателя |
|--------------------|--------|-----------------------|
| | | |

Краматорск, 2016

СОДЕРЖАНИЕ

| | |
|--|-----|
| Введение..... | 5 |
| 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ «АВТОАССОЦИАЦИЯ ОБРАЗОВ». | 6 |
| 1.1 Описание предметной области "Автоассоциация образов" | 6 |
| 1.2 Проблема обучения распознаванию образов | 7 |
| 1.3 Анализ аналогов..... | 9 |
| 2 РАЗРАБОТКА НЕЙРОННОЙ СЕТИ | 12 |
| 2.1 Диаграмма прецедентов | 122 |
| 2.2 Выбор языка программирования..... | 12 |
| 3 СОЗДАНИЕ ПРОЕКТА НЕЙРОННОЙ СЕТИ | 13 |
| ВЫВОД..... | 18 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 19 |
| ПРИЛОЖЕНИЕ А. ПРИМЕР «СЫРЫХ» ДАННЫХ..... | 20 |
| ПРИЛОЖЕНИЕ Б. КРАТКИЙ ЛИСТИНГ ПРОГРАММЫ..... | 21 |

| | | | | | | | | |
|-----------|-------|---------------|---------|------|---|--------------|------|--------|
| | | | | | КИТ 082.21.КР | | | |
| Сме | Лист. | № докум. | Подпись | Дата | | | | |
| Разраб. | | Ефремов А.О. | | | Разработка нейронной сети для автоассоциации образов | Лит. | Лист | Листов |
| Пров. | | Богданова Л.М | | | | | 2 | 22 |
| Реценз. | | | | | | 2 ИТ-12-1 | | |
| Н. Контр. | | | | | | | | |
| Утверд. | | | | | | | | |

З А Д А Н И Е

на курсовую работу по дисциплине

« Методы и системы искусственного интеллекта »

Группа ИТ 12-1

Ефремов Михаил Александрович

Тема работы: «Разработка нейронной сети для автоассоциации образов»

- исходные данные для тестовых вариантов назначить самостоятельно;
- работу необходимо выполнить в соответствии с графиком и требованиями к выполнению курсовой работы по методам и системам искусственного интеллекта;
- изменение и уточнение темы с согласия руководителя возможно только на первом этапе работы (1-2 неделя);
- готовая работа должна быть сдана на проверку не позже, чем за два дня до защиты.

Задание получил: _____ (М.А. Ефремов).
(подпись)

Задание выдал: _____ (Л.М. Богданова)
(подпись)

Дата выдачи

РЕФЕРАТ

Курсовая работа по дисциплине “Методы и системы искусственного интеллекта” студента группы ИТ-12-1 Ефремова М.А. содержит проектирование нейронной сети для анализа и автоассоциации образов.

Курсовая работа содержит 22 страницы машинописного текста, 14 рисунков.

Цель разработки топологии – создать такую нейросеть, которая будет эффективно выполнять анализ данных и поставленную задачу.

ОБРАЗЕЦ, НЕЙРОННАЯ СЕТЬ, КАЧЕСТВО, ПРОГРАММНЫЙ ПРОДУКТ,
C#, СЕТЬ ХЭММИНГА, АВТОАССОЦИАЦИЯ

ВВЕДЕНИЕ

Сеть Хемминга — это одна из наиболее многообещающих распознающих и классифицирующих нейронных сетей. В этой сети черно-белые изображения представляются в виде m -мерных биполярных векторов. Свое название она получила от расстояния Хемминга, которое используется в сети в мере сходства R изображений входного и эталонных, хранимых с помощью весов связей сети.

Нейронная сеть Хэмминга состоит из входного, скрытого и выходного слоев нейронов. Скрытый и выходной слои содержат по K нейронов, где K - число эталонов. Нейроны скрытого слоя n синапсами соединены с выходами нейронов входного слоя сети. Выходы нейронов выходного слоя связаны со входами остальных нейронов этого слоя отрицательными обратными (ингибиторными) связями. Единственная положительная обратная связь подается с выхода для каждого нейрона выходного слоя на его же вход.

Сеть выбирает эталон с минимальным хэмминговым расстоянием от предъявленного входного вектора путем активизации только одного выхода сети (нейрона выходного слоя), соответствующего этому эталону.

Хэммингово расстояние представляет собой пример меры сходства или, вернее, различия, первоначально введенной для бинарных функций в диадном пространстве. Она применима для сравнения любых упорядоченных наборов, принимающих дискретные значения и, вероятно, является наилучшей из известных мер сходства между цифровыми кодами.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ «АВТОАССОЦИАЦИЯ ОБРАЗОВ»

1.1 Описание предметной области "Автоассоциация образов"

Образное восприятие мира — одно из загадочных свойств живого мозга, позволяющее разобраться в бесконечном потоке воспринимаемой информации и сохранять ориентацию в океане разрозненных данных о внешнем мире. Воспринимая внешний мир, мы всегда производим классификацию воспринимаемых ощущений, т. е. разбиваем их на группы похожих, но не тождественных явлений. Например, несмотря на существенное различие, к одной группе относятся все буквы А, написанные различными почерками, или все звуки, соответствующие одной и той же ноте, взятой в любой октаве и на любом инструменте, а оператор, управляющий техническим объектом, на целое множество состояний объекта реагирует одной и той же реакцией. Характерно, что для составления понятия о группе восприятий определенного класса достаточно ознакомиться с незначительным количеством ее представителей. Ребенку можно показать всего один раз какую-либо букву, чтобы он смог найти эту букву в тексте, написанном различными шрифтами, или узнать ее, даже если она написана в умышленно искаженном виде. Это свойство мозга позволяет сформулировать такое понятие, как образ.

Образы обладают характерным свойством, проявляющимся в том, что ознакомление с конечным числом явлений из одного и того же множества дает возможность узнавать сколь угодно большое число его представителей. Примерами образов могут быть: река, море, жидкость, музыка Чайковского, стихи Маяковского и т. д. В качестве образа можно рассматривать и некоторую совокупность состояний объекта управления, причем вся эта совокупность состояний характеризуется тем, что для достижения заданной цели требуется одинаковое воздействие на объект. Образы обладают характерными объективными свойствами в том смысле, что разные люди, обучающиеся на различном материале наблюдений, большей частью одинаково и независимо друг от друга классифицируют одни и те же объекты.

Способность восприятия внешнего мира в форме образов позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов позволяет моделировать процесс их распознавания. Будучи отражением объективной реальности, понятие образа столь же объективно, как и сама реальность, а поэтому это понятие может быть само по себе объектом специального исследования.

1.2 Проблема обучения распознаванию образов

Одним из самых интересных свойств человеческого мозга является способность отвечать на бесконечное множество состояний внешней среды конечным числом реакций. Может быть, именно это свойство позволило человеку достигнуть высшей формы существования живой материи, выражающейся в способности к мышлению, т. е. активному отражению объективного мира в виде образов, понятий, суждений и т. д. Поэтому проблема ОРО возникла при изучении физиологических свойств мозга.

В целом проблема распознавания образов состоит из двух частей: обучения и распознавания. Обучение осуществляется путем показа отдельных объектов с указанием их принадлежности тому или другому образу. В результате обучения распознающая система должна приобрести способность реагировать одинаковыми реакциями на все объекты одного образа и различными — на все объекты различных образов. Очень важно, что процесс обучения должен завершиться только путем показов конечного числа объектов без каких-либо других подсказок. В качестве объектов обучения могут быть либо картинки, либо другие визуальные изображения (буквы), либо различные явления внешнего мира, например, звуки, состояния организма при медицинском диагнозе, состояние технического объекта в системах управления и др. Важно, что в процессе обучения указываются только сами объекты и их принадлежность образу. За обучением следует процесс распознавания новых объектов, который характеризует действия уже обученной системы. Автоматизация этих процедур и составляет проблему

обучения распознаванию образов. В том случае, когда человек сам разгадывает или придумывает, а затем навязывает машине правило классификации, проблема распознавания решается частично, так как основную и главную часть проблемы (обучение) человек берет на себя.

Круг задач, которые могут решаться с помощью распознающих систем, чрезвычайно широк. Сюда относятся не только задачи распознавания зрительных и слуховых образов, но и задачи распознавания сложных процессов и явлений, возникающих, например, при выборе целесообразных действий руководителем предприятия или выборе оптимального управления технологическими, экономическими, транспортными или военными операциями. В каждой из таких задач анализируются некоторые явления, процессы, состояния внешнего мира, всюду далее называемые объектами наблюдения. Прежде чем начать анализ какого-либо объекта, нужно получить о нем определенную, каким-либо способом упорядоченную информацию. Такая информация представляет собой характеристику объектов, их отображение на множестве воспринимающих органов распознающей системы.

Но каждый объект наблюдения может воздействовать по-разному, в зависимости от условий восприятия. Например, какая-либо буква, даже одинаково написанная, может в принципе как угодно смещаться относительно воспринимающих органов. Кроме того, объекты одного и того же образа могут достаточно сильно отличаться друг от друга и, естественно, по-разному воздействовать на воспринимающие органы.

Каждое отображение какого-либо объекта на воспринимающие органы распознающей системы, независимо от его положения относительно этих органов, принято называть изображением объекта, а множества таких изображений, объединенные какими-либо общими свойствами, представляют собой образы.

Выбор исходного описания объектов является одной из центральных задач проблемы ОРО. При удачном выборе исходного описания (пространства признаков) задача распознавания может оказаться тривиальной и, наоборот, неудачно выбранное исходное описание может привести либо к очень сложной дальнейшей переработке информации, либо вообще к отсутствию

решения. Например, если решается задача распознавания объектов, отличающихся по цвету, а в качестве исходного описания выбраны сигналы, получаемые от датчиков веса, то задача распознавания в принципе не может быть решена.

1.3 Анализ аналогов

В качестве наиболее близких аналогов мы выбрали программный продукт, который мы использовали в лабораторной работе №7 курса «Методы и Системы Искусственного Интеллекта».

| | | | | | | | | | | |
|---|--------------|-----|----|----|-------|-------|---|----|---|--|
| 1 | LETTERS.C.35 | | | | | | | | | |
| 2 | 11111 | 1 | 11 | 11 | 1 | 11111 | | | | |
| 3 | 11 | 1 | 1 | 1 | 1 | 11111 | | | | |
| 4 | 11 | 111 | 1 | 1 | 111 | 11 | | | | |
| 5 | 1 | 1 | 1 | 1 | 11111 | 1 | 1 | 1 | 1 | |
| 6 | 11111111 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | | |

Рис. 1 – хранение данных в продукте-аналоге

[illegible]

Рис. 2 – хранение данных в нашем продукте

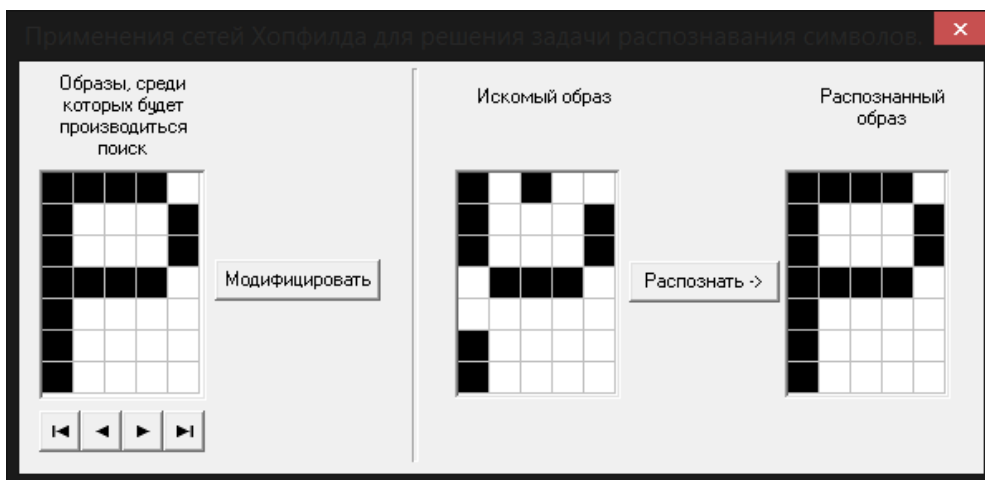


Рис. 3 – процесс распознавания в аналоге

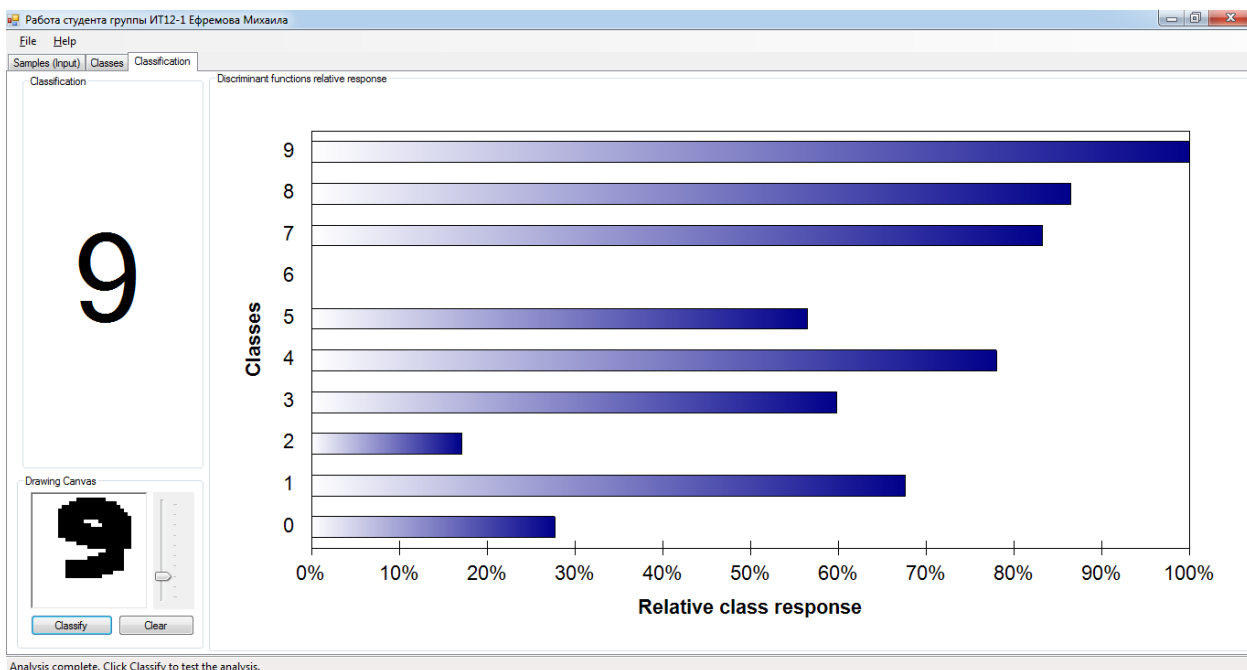


Рис.4 – процесс распознавания в нашем продукте

Минусы аналога:

- Программный продукт реализован на языке DELPHI, который на данный момент является уже морально устарелым.
- В качестве образов используются области (массивы) 5x7, что в значительной степени сокращает пространство для работы (в реализованном нами продукте области 32x32).
- В программном продукте нет предустановленной тестовой и тестируемой выборки, при каждом запуске программы её необходимо определять заново.

- Неспособность системы применяться на реальных задачах из-за маленькой области анализа.
- Строго ограниченная по количеству обучающая и тестируемая выборка (в нашем случае – никаких пределов нет, что открывает практически безграничное пространство для работы).
- Формат хранения данных не нагляден (1 и пробел, против 1 и 0 в нашем продукте).
- Продукт не позволяет определить вероятности того или иного варианта.

Плюсы аналога:

- Из-за отсутствия предустановленной выборки можно быстро выполнять анализ для нужных данных.
- Сравнительно быстрый способ добавления тестируемых данных.
- Опять же из-за отсутствия предустановленных выборок продукт позволяет использовать литеры.

2 РАЗРАБОТКА НЕЙРОННОЙ СЕТИ

2.1 Диаграмма прецедентов

Для заданной предметной области основной целью является получение наилучшего окружения для обучения сети. UML-диаграмма представлена на рисунке 5.

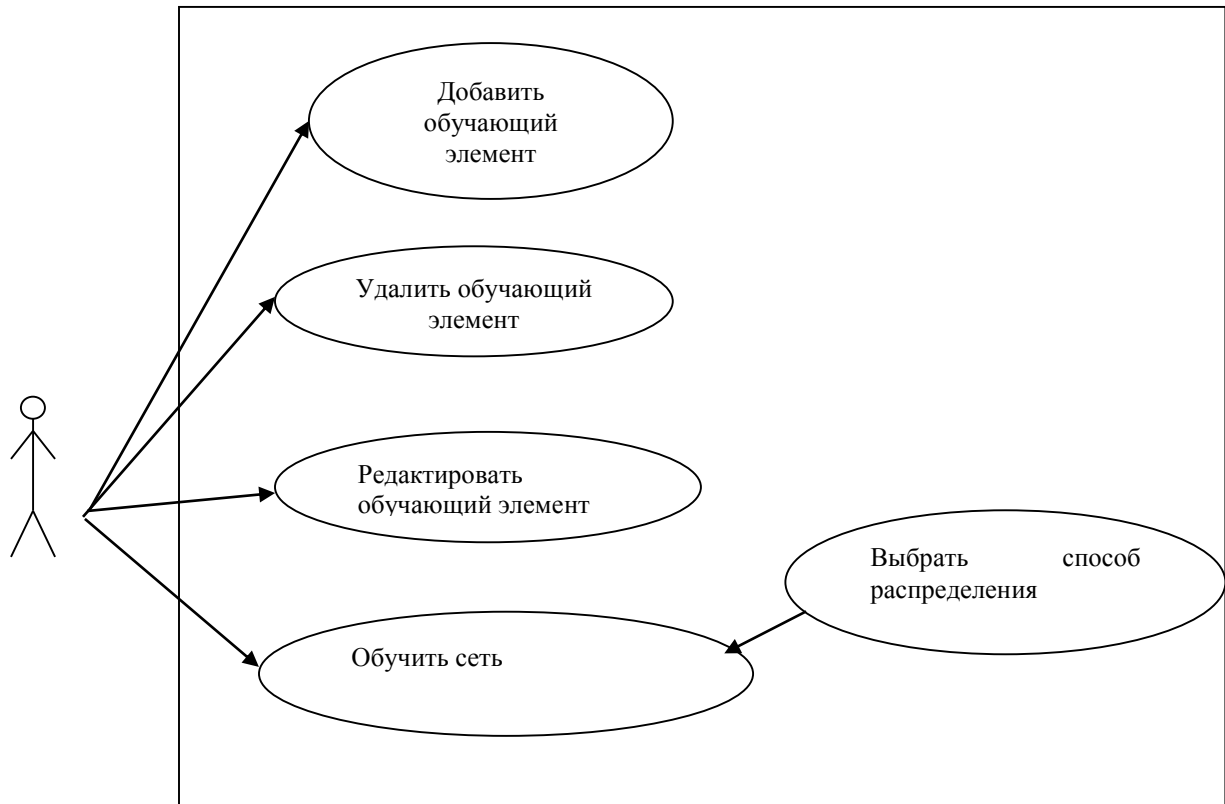


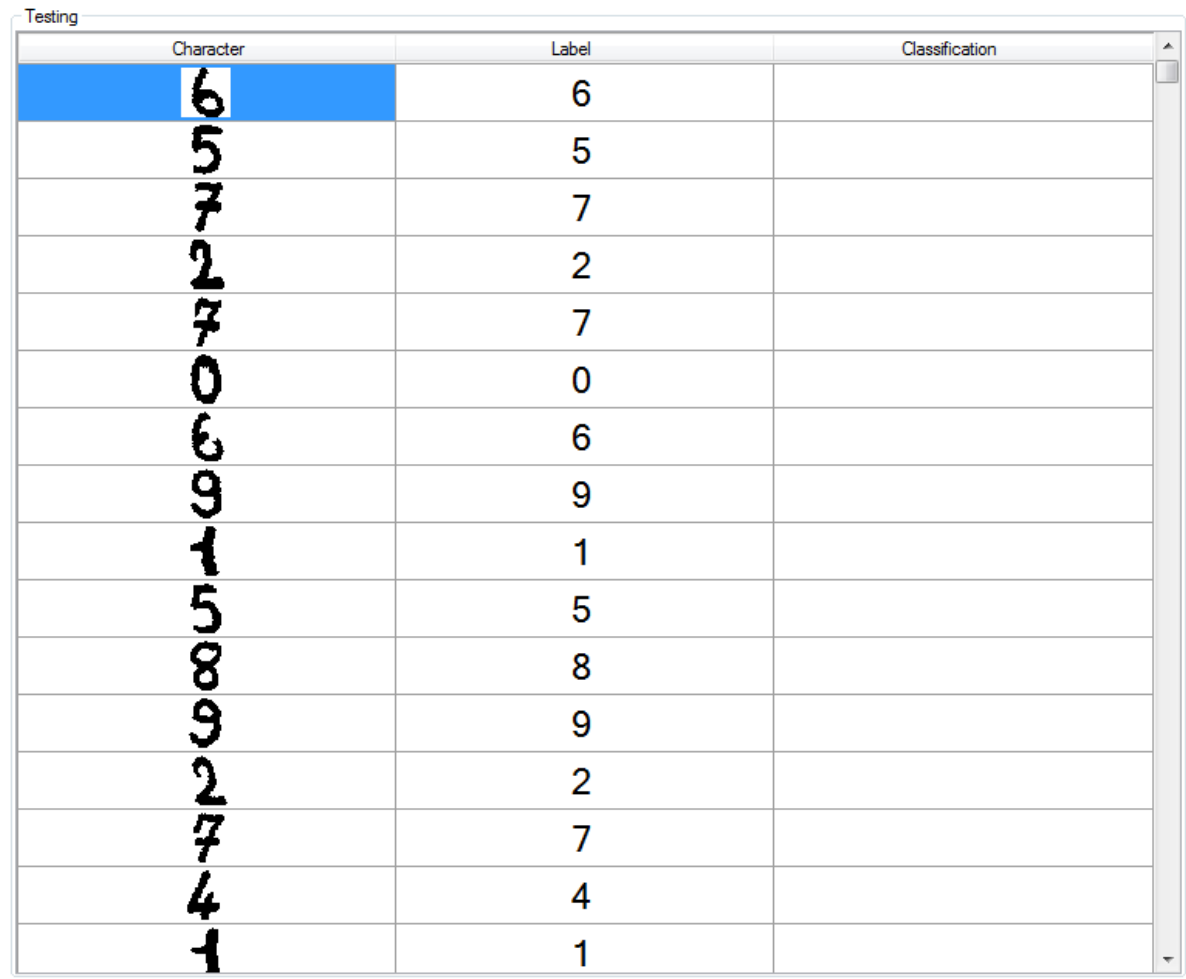
Рис. 5 – UML-диаграмма для получения наивыгоднейшего качества образца

2.2 Выбор языка программирования

Для выполнения курсовой работы я выбрал среду разработки Visual Studio с фреймворком .Net версии 4.6.1 и язык C# 6.0.

3 СОЗДАНИЕ ПРОЕКТА НЕЙРОННОЙ СЕТИ

На рисунке 6 приведена часть таблицы, содержащей данные для анализа. На рисунке 7 – таблица в исходном формате, понятном для анализа программы.



The image shows a screenshot of a software window titled "Testing". Inside the window is a table with three columns: "Character", "Label", and "Classification". The "Character" column contains handwritten digits from 0 to 9, repeated twice. The "Label" column contains the corresponding numerical labels for these digits. The "Classification" column is currently empty. The first row, where the character is '6', is highlighted in blue.

| Character | Label | Classification |
|-----------|-------|----------------|
| 6 | 6 | |
| 5 | 5 | |
| 7 | 7 | |
| 2 | 2 | |
| 7 | 7 | |
| 0 | 0 | |
| 6 | 6 | |
| 9 | 9 | |
| 1 | 1 | |
| 5 | 5 | |
| 8 | 8 | |
| 9 | 9 | |
| 2 | 2 | |
| 7 | 7 | |
| 4 | 4 | |
| 1 | 1 | |

Рис. 6 – часть таблицы с анализируемыми данными

Рис.7 – исходный формат для тестируемых данных

| Training | |
|-----------|-------|
| Character | Label |
| 0 | 0 |
| 0 | 0 |
| 7 | 7 |
| 4 | 4 |
| 6 | 6 |
| 2 | 2 |
| 5 | 5 |
| 5 | 5 |
| 0 | 0 |
| 8 | 8 |
| 7 | 7 |
| 1 | 1 |
| 9 | 9 |
| 5 | 5 |
| 3 | 3 |
| 0 | 0 |

Рисунок 8 – Данные для обучения

Рисунок 9 – «сырые» данные для обучения

Сама инициализация данных для обучения и для тестирования выполняется с помощью опций File -> Open. Данные находятся в прикрепленном файле формата .txt, поэтому пользователь не должен их искать.

После удачной инициализации данных системе необходимо провести обучение, которое выполняется с помощью кнопки «Run Analysis» (рисунок 10). Программный продукт инициализирует классы для каждого уникального символа, анализирует концентрацию данных и обучает нейронную сеть.

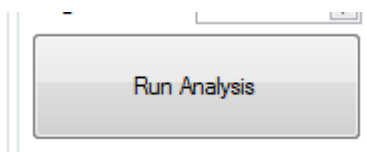


Рис. 10 – кнопка, начинающая процесс обучения сети данными

Выполнив обучение пользователь может переключиться на пункт «Classes» (рисунок 11), в котором система описала все входящие символы и проанализировала их концентрацию на фоне всех данных.

| Samples (Input) Classes Classification | | | |
|--|----------------------|-------|-------|
| Classes | | | |
| Number | Prevalence | Count | Index |
| 0 | 0,089719626168224292 | 48 | 0 |
| 1 | 0,102803738317757 | 55 | 1 |
| 2 | 0,10093457943925234 | 54 | 2 |
| 3 | 0,095327102803738323 | 51 | 3 |
| 4 | 0,11588785046728972 | 62 | 4 |
| 5 | 0,099065420560747658 | 53 | 5 |
| 6 | 0,085981308411214957 | 46 | 6 |
| 7 | 0,10467289719626169 | 56 | 7 |
| 8 | 0,09719626168224299 | 52 | 8 |
| 9 | 0,10841121495327102 | 58 | 9 |

Рис.11 – Анализ данных, после обучения сети

Кроме того, в этом же меню можно визуально просмотреть данные для каждого из классов (рисунок 12).

| Sample Subset | | | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис.12 – Все элементы класса «0» обучающей выборки

После визуального анализа обучающей выборки можно приступить к анализу тестируемой выборки с помощью кнопки «Classify». Полученный результат можно увидеть на рисунке 13.

| Testing | | |
|-----------|-------|----------------|
| Character | Label | Classification |
| 5 | 5 | 9 |
| 7 | 7 | 7 |
| 6 | 6 | 6 |
| 1 | 1 | 1 |
| 9 | 9 | 9 |
| 3 | 3 | 3 |
| 8 | 8 | 8 |
| 9 | 9 | 9 |
| 1 | 1 | 1 |
| 6 | 6 | 6 |
| 5 | 5 | 5 |
| 0 | 0 | 0 |
| 4 | 4 | 4 |
| 6 | 6 | 6 |
| 6 | 6 | 6 |
| 9 | 9 | 9 |

Рис.13 – Результат распознавания тестируемой выборки

Можно увидеть, что качество распознавания очень велико (около 94%). Подобный результат можно объяснить хорошей работой самой нейронной сети и обучающей выборке, которая хорошо коррелирует с тестируемой, используя другие выборки результат может отличаться.

Кроме тестирования уже предустановленной выборки программный продукт поддерживает тестирование элементов, которые определил сам пользователь. Данный функционал находится в меню «Classification» (рисунок 14).

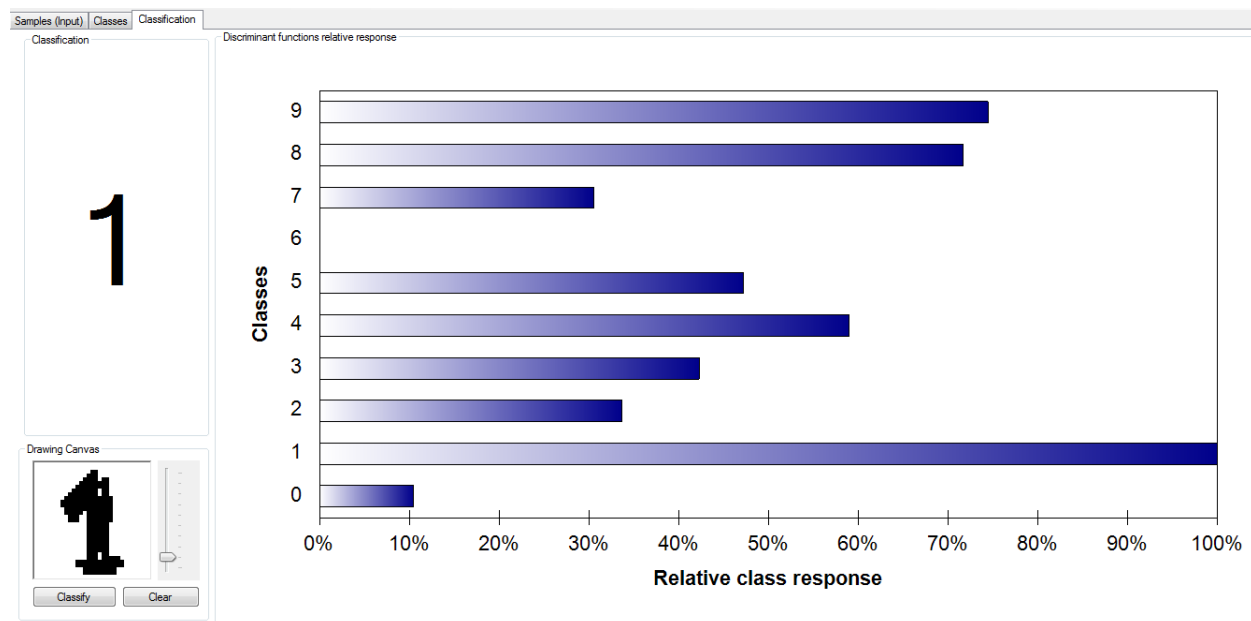


Рис. 14 – Анализ пользовательских данных

Стоит заметить, что кроме самого механизма определения меню отображает вероятности принадлежности элемента к тому или иному классу обучающей выборки, что очень наглядно и полезно.

ВЫВОД

В ходе данной курсовой работе были выработаны практические навыки в разработке нейронной сети Хэмминга.

Был реализован программный продукт, который позволяет с высокой точностью распознавать образы. В противовес аналогам данный продукт имеет кардинально более широкие возможности для анализа, начиная с более большого поля для анализа, заканчивая возможностью сохранить данные в нужное хранилище для последующего анализа.

Программный продукт был реализован с учётом всех тенденций разработки ПО, что позволяет легко интегрировать его с любой системой простым подключением. Учитывая модульность самой системы её можно без проблем внедрять в пользовательские приложения и использовать как весь функционал, так и его часть.

Кроме обучения и классификации данных ПО предварительно анализирует поступающие данные, позволяет зрительно их анализировать и не имеет приделов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Википедия. Свободная энциклопедия: Теория распознавания образов. [http://ru.wikipedia.org/wiki/ Теория_расознавания_образов](http://ru.wikipedia.org/wiki/Теория_расознавания_образов).
2. Интернет ресурс <http://www.codenet.ru>
3. Методические указания по выполнению лабораторных и самостоятельных работ по дисциплине «Системы искусственного интеллекта» для студентов специальности 7.080402 "Информационные технологии проектирования" / сост.: В. И. Юдин, О. А. Лябик. – Краматорск: ДГМА, 2007.– 108 с.
4. www.AIportal.ru. Портал искусственного интеллекта: Многослойный персептрон. Режим доступа: <http://www.aiportal.ru/articles/neural-networks/multi-perceptron.html>.
5. Рихтер Джеффри CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#; Питер - Москва, 2013. - 896 с.

ПРИЛОЖЕНИЕ А. ПРИМЕР «СЫРЫХ» ДАННЫХ

[illegible]

6

[illegible]

5

ПРИЛОЖЕНИЕ Б. КРАТКИЙ ЛИСТИНГ ПРОГРАММЫ

```
private void btnLoad_Click(object sender, EventArgs e)
{
    lbStatus.Text = @"Loading data. This may take a while...";
    Application.DoEvents();

    // Load optdigits dataset into the DataGridView
    var trainerReader = new StringReader(Properties.Resources.optdigits_tra);
    var d = new StringReader(Properties.Resources.optdigits_tes);
    var test0Reader = new StringReader(Properties.Resources.optdigits_test0);

    dgvAnalysisSource.Rows.Clear();
    dgvAnalysisTesting.Rows.Clear();

    while (true)
    {
        var buffer = new char[(32 + 2) * 32];
        var read = trainerReader.ReadBlock(buffer, 0, buffer.Length);
        var label = trainerReader.ReadLine();

        if (read < buffer.Length || label == null) break;

        var bitmap = Extract(new string(buffer));
        var features = Extract(bitmap);
        var clabel = int.Parse(label);
        dgvAnalysisSource.Rows.Add(bitmap, clabel, features);
    }

    while (true)
    {
        var buffer = new char[(32 + 2) * 32];
        var read = test0Reader.ReadBlock(buffer, 0, buffer.Length);
        var label = test0Reader.ReadLine();

        lbStatus.Text = @"Dataset loaded. Click Run analysis to start the
analysis.";

        if (read < buffer.Length || label == null) break;

        var bitmap = Extract(new string(buffer));
        var features = Extract(bitmap);
        var clabel = Int32.Parse(label);
        dgvAnalysisTesting.Rows.Add(bitmap, clabel, null, features);
    }

    btnSampleRunAnalysis.Enabled = true;
}

private void btnCanvasClassify_Click(object sender, EventArgs e)
{
    if (kda != null)
    {
        var input = canvas.GetDigit();

        var sinput = new string[input.Length];
        for (var i = 0; i < input.Length; i++)
            sinput[i] = input[i].ToString();

        // Classify the input vector
    }
}
```

```

        double[] responses;
        var num = kda.Classify(input, out responses);

        // Set the actual classification answer
        lbCanvasClassification.Text = num.ToString();

        // Scale the responses to a [0,1] interval
        var max = responses.Max();
        var min = responses.Min();

        for (var i = 0; i < responses.Length; i++)
            responses[i] = Tools.Scale(min, max, 0, 1, responses[i]);

        // Create the bar graph to show the relative responses
        CreateBarGraph(graphClassification, responses);
    }
    // Get the input vector drawn
}

private void btnCanvasClear_Click(object sender, EventArgs e)
{
    canvas?.Clear();
}

private void canvas_MouseUp(object sender, MouseEventArgs e)
{
    btnCanvasClassify_Click(this, EventArgs.Empty);
}

private void tbPenWidth_Scroll(object sender, EventArgs e)
{
    canvas.PenSize = tbPenWidth.Value;
}

private void dgvClasses_CurrentCellChanged(object sender, EventArgs e)
{
    if (dgvClasses.CurrentRow != null)
    {
        var dclass =
        (DiscriminantAnalysisClass)dgvClasses.CurrentRow.DataBoundItem;

        var list = new ImageList();

        lvClass.Items.Clear();
        lvClass.LargeImageList = list;
        var idx = dclass.Indexes;
        for (var i = 0; i < idx.Length; i++)
        {
            {
                var bitmap =
                (Bitmap)dgvAnalysisSource.Rows[idx[i]].Cells["colTrainingImage"].Value;
                list.Images.Add(bitmap);
            }

            var item = new ListViewItem(String.Empty, i);
            lvClass.Items.Add(item);
        }
    }
}

```