

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего  
профессионального образования  
Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»  
Кафедра «Автоматизированные системы обработки информации и управления»



Тоноян С.А., Емельянов Д.С.

Лабораторная работа      № 4

**по курсу**

**“Оперативный анализ и визуализация данных”**

Методическое указание  
по выполнению лабораторной работы

Москва - 2014 год

## Оглавление

1.Введение.....	3
2.Цель лабораторной работы .....	4
3.Краткая характеристика объекта изучения, исследования.....	5
4.Схема и описание лабораторной .....	6
5.Задачи и порядок выполнения работы.....	15
6.Содержание отчета по лабораторной работе .....	23
7.Контрольные вопросы .....	24
8.Литература .....	27

## 1. Введение.

Одной из важных частных задач машинного обучения является задача распознавания объектов на изображениях. Задача состоит в определении, является ли предъявленное изображение изображением интересующего нас объекта. К задаче распознавания сводятся задачи обнаружения и локализации объектов: обнаружение исследуемых клеток на снимках с микроскопа, задача поиска человеческих лиц на изображениях. Последние исследования показали эффективность обучаемых методов для решения данной задачи. В лабораторной работе рассматриваются различные методы классификации с обучением по прецедентам, применяющиеся к задаче распознавания объектов на изображениях. Проводится сравнение, и анализ причин тех или иных показателей работы рассмотренных методов.

**Машинное обучение** [machine learning] изучает методы построения моделей, способных обучаться, и алгоритмы для их построения и обучения (пример алгоритмов – логистическая регрессия [logistic regression], нейронные сети [neural networks], машины опорных векторов [SVM – support vector machines], деревья принятия решений [decisions trees]) и т.д.

Классификация, регрессия являются одними из важнейших задач анализа данных. Действительно, как классификационная, так и регрессионная модель находят закономерности между входными и выходными переменными. Но если входные и выходные переменные модели непрерывные — перед нами задача регрессии. Если выходная переменная одна и она является дискретной (метка класса), то речь идет о задаче классификации.

## 2. Цель лабораторной работы

### ***2.1 Получить теоретические знания изучив:***

- линейная регрессия;
- логистическая регрессия;
- байесовская классификация;.
- деревья решений;
- решающие правила;
- нейронные сети;
- метод  $k$  ближайших соседей и др.

### ***2.2. Получение практических навыков:***

- исследование и оценка распознавания образов с применением модели линейной регрессии;
- исследование и оценка распознавания образов с применением модели логистической регрессии;

### 3.Краткая характеристика объекта изучения, исследования

Так как курс начального уровня, то в нем будут изучены базовые для некоторых алгоритмов математические понятия и приемы, в частности, линейная и логистическая регрессии. Будут применены некоторые методы оптимизации (градиентный спуск – подробно [gradient descent] , метод Бroyдена – Флетчера – Гольдфарба – Шанно, более известный как BFGS, метод сходящихся градиентов [conjugate gradient – CG] ) и их влияние на скорость и качество обучения. Будет исследовано влияние различных параметров на качество и скорость обучения (скорость обучения, размер выборки для обучения, размер критерия остановки обучения, влияние L2 регуляризации, нормализации ).

## **4. Схема и описание лабораторной работы**

Выполнение лабораторной работы предполагает получить теоретических знания исследуемых методов анализа данных для их последующего применения.

### **4.1 Теоретическая часть.**

В настоящее время для анализа данные различных природы и объемов, разработано большое количество методов и алгоритмов на базе аналитической платформы, которая базируется на следующие математические методы:

- линейная регрессия;
- логистическая регрессия;
- байесовская классификация;.
- деревья решений;
- решающие правила;
- нейронные сети;
- метод  $k$  ближайших соседей и др.

К базовому набору методов, которые входят в состав большинства аналитических платформ, относятся множественные регрессионные модели, деревья решений и искусственные нейронные сети.

#### **4.1.1 Методы с обучением по прецедентам.**

##### ***I. Основные понятия:***

Это наиболее общий подход. Задача распознавания объектов на изображение сводится к задаче классификации и для неё применяется хорошо разработанный математический аппарат построения модели обучения по прецедентам. Модель строиться автоматически по заранее собранному набору прецедентов - изображений, для которых известно, являются ли они изображениями объекта или нет. Наблюдением, в данном случае, является некоторый «вектор признаков», полученный из исходного изображения некоторым преобразованием, отображающим изображения в

пространство действительных векторов. Гипотеза, подлежащая проверке - принадлежность изображения к классу изображений искомого объекта.

Таким образом, система распадается на два модуля:

- модуль преобразования изображения в вектор признаков
- модуль классификации.

Задачей модуля преобразования является наиболее полное и информативное представление изображения в виде числового вектора.

Задачей модуля классификации является проверка гипотезы принадлежности изображения классу изображений объекта на основании наблюдения, которым является вектор признаков.

Модуль преобразования и модуль классификации тесно связаны. Главная цель модуля преобразования – представить изображение в форме наиболее удобной для модуля классификации. Основные требования, предъявляемые к модулю преобразования: скорость, наиболее полное и информативное представление данных, масштабируемость (преобразование корректно работает с изображением разных размеров, размерность вектора признаков не меняется).

Также является желательной инвариантность модуля преобразования к деформациям и пространственным искажениям объекта. Несмотря на важность метода преобразования, его подробное рассмотрение выходит за рамки этой работы.

Пусть имеются множество объектов  $X$ , множество ответов  $Y$ , и существует целевая функция (target function)  $y^* : X \rightarrow Y$ , значения которой

$y_i = y^*(x_i)$ , известны только на конечном подмножестве объектов

$\{x_1, \dots, x_t\} \subset X$ . Пары «объект–ответ»  $(x_i, y_i)$  называются прецедентами.

Совокупность пар  $X^t = (x_i, y_i)_{i=1}^t$  называется обучающей выборкой (training sample).

Задача обучения по прецедентам заключается в том, чтобы восстановить функциональную зависимость между объектами и ответами, то

есть построить отображение  $\mathbf{a}: \mathbf{X} \rightarrow \mathbf{Y}$ , удовлетворяющее следующей совокупности требований:

- Отображение  $\mathbf{a}$  должно допускать эффективную компьютерную реализацию – в виде алгоритма.
- Алгоритм  $\mathbf{a}(\mathbf{x})$  должен воспроизводить на объектах выборки заданные ответы:  $\mathbf{a}(\mathbf{x}_i) = \mathbf{y}_i, i = 1, \dots, \ell$ . Равенство здесь может пониматься как точное или как приближённое, в зависимости от конкретной задачи.
- На алгоритм  $\mathbf{a}(\mathbf{x})$  могут накладываться разного рода априорные ограничения, например, требования непрерывности, гладкости, монотонности, и т. д., или сочетание нескольких требований. В некоторых случаях может задаваться модель алгоритма – функциональный вид отображения  $\mathbf{a}(\mathbf{x})$ , определённый с точностью до параметров.
- Алгоритм  $\mathbf{a}$  должен обладать обобщающей способностью, то есть достаточно точно приближать целевую функцию  $\mathbf{y}^*(\mathbf{x})$  не только на объектах обучающей выборки, но и на всём множестве  $\mathbf{X}$ .

## ***II. Разновидности задач обучения по прецедентам***

В зависимости от природы множества  $\mathbf{Y}$  задачи, обучения по прецедентам делятся на следующие типы:

- $\mathbf{Y} = \{1, \dots, M\}$  – задача классификации (classification) на  $M$  непересекающихся классов  $\mathbf{K}_y = \{\mathbf{x} \in \mathbf{X}: \mathbf{y}^*(\mathbf{x}) = \mathbf{y}\}$ . В некоторых приложениях классы называют образами и говорят о задаче распознавания образов (pattern recognition).
- $\mathbf{Y} = \{0, 1\}^M$  – задача классификации на  $M$  пересекающихся классов. В простейшем случае эта задача сводится к решению  $M$  независимых задач классификации с двумя непересекающимися классами.
- $\mathbf{Y} = \mathbf{R}$  – задача восстановления регрессии (regression estimation).
- Задача прогнозирования (forecasting) является частным случаем классификации или восстановления регрессии, когда  $\mathbf{X}$  – описание



прошлого поведения объекта,  $\mathbf{Y}$  - описание некоторых характеристик его будущего поведения.

#### 4.1.2 Линейная регрессия

Задача линейной регрессии заключается в нахождении коэффициентов *уравнения линейной регрессии*, которое имеет вид:

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad (8.1) \quad (1)$$

где  $y$  — выходная (зависимая) переменная модели;

$x_1, x_2 \dots x_n$  — входные (независимые) переменные;

$b_i$  — коэффициенты линейной регрессии, называемые также параметрами модели ( $b_0$  — свободный член).

Задача линейной регрессии заключается в подборе коэффициентов  $b_i$ , уравнения (1) таким образом, чтобы на заданный входной вектор  $\mathbf{X} = (x_1, x_2 \dots x_n)$  регрессионная модель формировала желаемое выходное значение  $y$ .

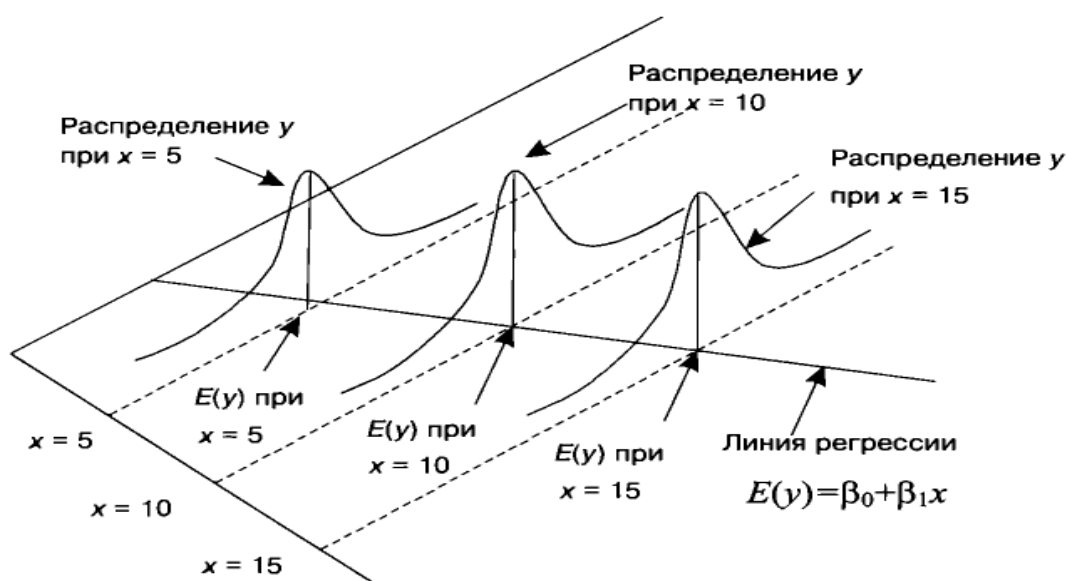


Рис 1.1 Для каждого значения  $x$  значение  $y$  есть нормально распределенная случайная величина

Статистическая модель простой линейной регрессии предполагает, что для каждого значения входной переменной  $x$  наблюдаемое значение выходной переменной  $y$  является нормально распределенной случайной величиной со средним  $E(y) = b_0 + b_1x$  и постоянной дисперсией  $\sigma^2$ . Данное

предположение иллюстрируется на рис. 1.1 для случаев  $x = 5$ ,  $x = 10$  и  $x = 15$ . Видно, что все кривые нормального распределения имеют одну и ту же форму, из чего следует, что дисперсия постоянна для всех  $x$ .

Одним из наиболее востребованных приложений линейной регрессии является *прогнозирование*. В этом случае входными переменными модели  $x$ -, являются наблюдения из прошлого (предикторы), а  $y$  — прогнозируемое значение. Несмотря на свою универсальность, линейная регрессионная модель не всегда пригодна для качественного предсказания зависимой переменной. Когда для решения задачи строят модель линейной регрессии, на значения зависимой переменной обычно не налагают никаких ограничений. Но на практике такие ограничения могут быть весьма существенными. Например, выходная переменная может быть категориальной или бинарной. В таких случаях приходится использовать различные специальные модификации регрессии, одной из которых является *логистическая регрессия*.

#### 4.1.3 Логистическая регрессия

*Логистическая регрессия* предназначена для предсказания зависимой переменной, принимающей значения в интервале от 0 до 1. Такая ситуация характерна для задач оценки вероятности некоторого события на основе значений независимых переменных. Кроме того, логистическая регрессия используется для решения задач бинарной классификации, в которых выходная переменная может принимать только два значения — 0 или 1, «Да» или «Нет» и т. д.

Таким образом, логистическая регрессия служит не для предсказания значений зависимой переменной, а скорее для оценки вероятности того, что зависимая переменная примет заданное значение.

Предположим, что выходная переменная  $y$  может принимать два возможных значения — 0 и 1. Основываясь на доступных данных, можно вычислить вероятности их появления:  $P(y = 0) = 1 - p$ ;  $P(y = 1) = p$ . Иными

словами, вероятность появления одного значения равна 1 минус вероятность появления другого, поскольку одно из них появится обязательно и их общая вероятность равна 1. Для определения этих вероятностей используется логистическая регрессия:

$$\log(p/(1-p)) = \beta_0 + (\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n) \quad (2)$$

Правая часть формулы (2) эквивалентна обычному уравнению линейной регрессии (1). Однако вместо непрерывной выходной переменной  $y$  в левой **части** отношения вероятностей двух взаимоисключающих событий (в нашем приоре — вероятность появления 0 и вероятность появления 1).

функция вида  $\log(p/(1-p))$  называется *логит - преобразованием* и обозначается  $\text{logit}(p)$ . Использование *логит* - преобразования позволяет ограничить диапазон изменения выходной переменной в пределах  $[0; 1]$ .

#### 4.1.4 Байесовская классификация

Байесовский подход объединяет группу алгоритмов классификации, основанных на принципе максимума апостериорной (условной) вероятности: для объекта с помощью формулы Байеса определяется апостериорная вероятность принадлежности к каждому классу и выбирается тот класс, для которого она максимальна.

Особое место в данной области занимает простая байесовская классификация, в основе которой лежит предположение о независимости признаков, описывающих классифицируемые объекты. Это предположение значительно упрощает задачу, поскольку вместо сложной процедуры оценки многомерной плотности вероятности требуется оценка нескольких одномерных.

#### 4.1.5 Деревья решений

*Деревья решений* (деревья классификаций) — популярная классификационная методика, в которой решающие правила извлекаются непосредственно из исходных данных в процессе обучения. Дерево решений — это древовидная иерархическая модель, где в каждом узле производится

проверка определенного атрибута (признака) с помощью правила. По результатам проверки формируются два или более дочерних узла, в которые попадают объекты, для которых значения данного атрибута удовлетворяют (или не удовлетворяют) правилу в родительском узле. Каждый конечный узел дерева (лист) содержит объекты, относящиеся к одному классу. Пример простого дерева решений приведен на рис. 1.2

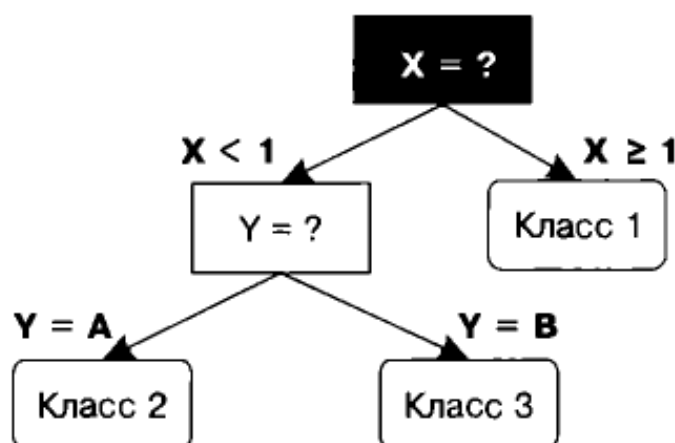


Рис. 1.2 Пример дерева решений

Классический алгоритм построения деревьев решений использует стратегию «разделяй и властвуй». Начиная с корневого узла, где присутствуют все обучающие примеры, происходит их разделение на два или более подмножества на основе значений атрибута, выбранных в соответствии с критерием (правилом) разделения. Для каждого из полученных подмножеств создается дочерний узел. Затем процесс ветвления повторяется для каждого дочернего узла до тех пор, пока не будет выполнено одно из условий остановки алгоритма.

В настоящее время разработано большое количество алгоритмов построения деревьев решений. Они отличаются способом отбора атрибутов для разбиения в каждом узле, условиями остановки и методикой упрощения построенного дерева. Упрощение дерева заключается в том, что после его построения удаляются те узлы, правила в которых имеют низкую ценность, поскольку относятся к небольшому числу примеров. Упрощение позволяет сделать дерево решений компактней

#### 4.1.6 Искусственные нейронные сети

**Нейронные сети, или искусственные нейронные сети**, представляют собой модели, которые в процессе функционирования имитируют работу головного мозга. Нейронная сеть состоит из простейших вычислительных элементов — искусственных нейронов, связанных между собой. Каждый нейрон имеет несколько входных и одну выходную связь. Каждая входная связь обладает весом, на который умножается сигнал, поступающий по ней с выхода другого нейрона. Каждый нейрон выполняет простейшее преобразование — взвешенное суммирование своих входов (рис. 1.3).

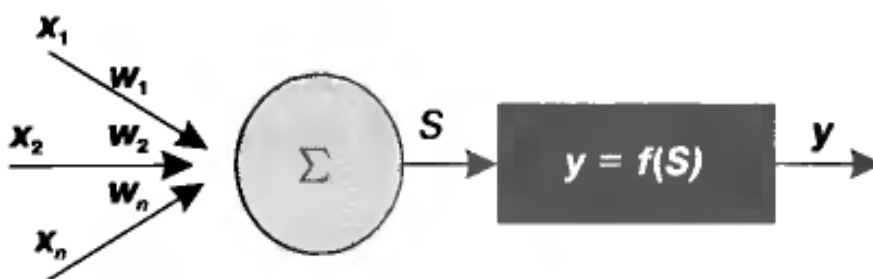


Рис. 1.3. Искусственный нейрон.

В нейронных сетях нейроны объединяются в слои, при этом выходы нейронов предыдущего слоя являются входами нейронов следующего слоя. В каждом слое нейроны выполняют параллельную обработку данных. Пример нейронной сети представлен на рис. 1.4.

Первый слой называется *входным*, его нейроны обеспечивают ввод в сеть входного вектора  $\mathbf{X} = (x_1, x_2, x_3)$  и распределяют его по нейронам следующего слоя. Нейроны последнего слоя, обеспечивающие вывод результатов, называются *выходными* и образуют *выходной* слой. Между входным и выходным нейронами расположены один или несколько промежуточных слоев, называемых *скрытыми*. Именно в скрытых слоях производится основная обработка данных.

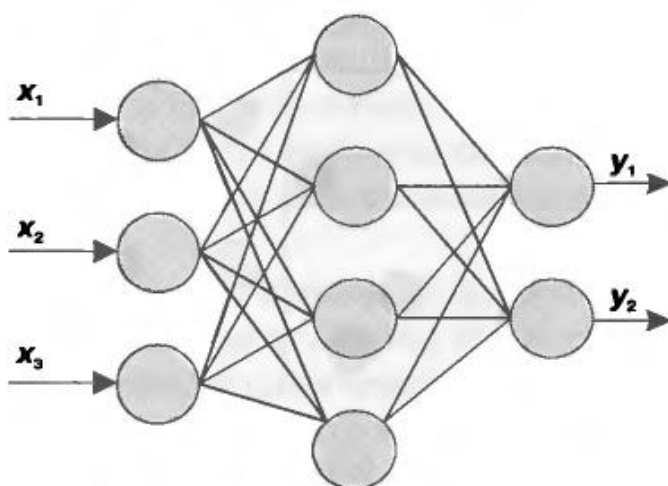


Рис. 1.4. Пример нейронной сети

В процессе работы нейронной сети значения входных переменных  $x$ , передаются по межнейронным связям и умножаются на весовые коэффициенты  $\omega_i$ , полученные значения суммируются в нейроне. Также в каждом нейроне выполняется простое преобразование с помощью активационной функции  $f(S)$ , обычно нелинейной. В результате преобразования значений входного вектора всеми нейронами сети на ее выходе формируется вектор результата (выходной вектор)  $Y = (y_1, y_2)$ .

#### 4.1.7 Метод $k$ ближайших соседей

Этот алгоритм относит любой новый пример к классу, которому принадлежит большинство его ближайших соседей в обучающем множестве. Под соседством в данном случае подразумевается не расположение объектов или наблюдений рядом в обучающем множестве, а то, что их многомерные векторы в пространстве признаков близки друг к другу. Сосед классифицируемого объекта считается ближайшим в пространстве признаков, если евклидово расстояние между ними является наименьшим. Если  $k = 1$ , то новый объект относится к тому же классу, что и его ближайший сосед.

## 4.2 Практическая реализация.

В лабораторной работе проводится исследование один из важных задач машинного обучения - задача распознавания объектов на изображениях. Задача состоит в определении, является ли предъявленное изображение изображением интересующего нас объекта.

В лабораторной работе используются изображения из базы данных MNIST [3] . Это 70 тысяч черно-белых изображений рукописных “арабских” цифр размером 28\*28 пикселей.

В ходе выполнения лабораторной работы проводится исследование и оценка распознанных изображений с применением разных методов и комбинации параметров.

### 5. Задачи и порядок выполнения работы

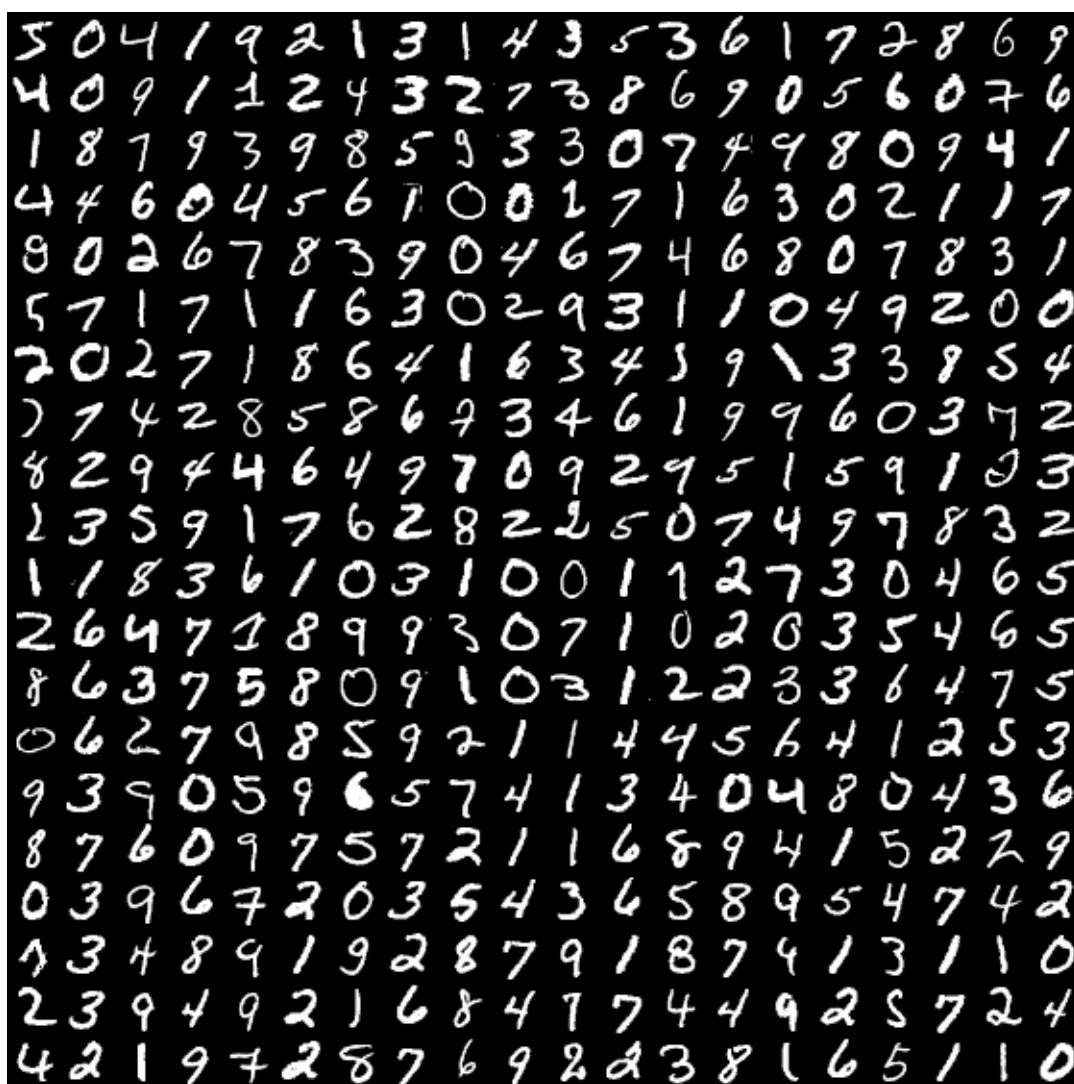
#### 5.1 Задачи

исследование и оценка распознавания образов с применением модели линейной регрессии с множеством параметров;

исследование и оценка распознавания образов с применением модели логистической регрессии с множеством параметров.

На рис.1.5 представлена снимок изображений из базы MNIST, которые являются объектами исследования в лабораторной работе, где проводится оценка полученных результатов по распознаванию образов.

База данных разделена на 50 тысяч изображений. Для обучения модели используется 10 000 экземпляров и 10 тысяч для тестирования - подтверждения результатов. Для курса был разработан специальный стенд для тестирования качества модели при различных параметрах.



▪ Рис. 1.5 Пример изображений из базы MNIST.

В процессе выполнения лабораторных работ студентам необходимо построить несколько зависимостей (например, зависимость качества обучения от скорости обучения). Инструмент для этого предлагается студентам найти самостоятельно.

## 5.2. Порядок выполнения работы

Лабораторные работы выполняются на программном стенде, экранная форма которого представлен на рис.1.6 Исследованные проводимых в данной лабораторной работе следующие

- подобрать параметры модели так, чтобы удовлетворить требования по точности и скорости обучения;
- построить графики зависимости скорости, точности обучения от параметров модели для каждого метода.



Также можно проверить результаты вычислений в третьей вкладке стенда. Для этого матрицу параметров нужно сохранить, нажав на кнопку во всплывающем окне с результатами. Будет сформирован файл, который затем нужно открыть во вкладке. Кнопками можно перемещаться по массиву с тестовой выборкой и проверить, что машина действительно может распознать изображения. Также в этой вкладке можно проверить, насколько хорошо модель обучилась и при необходимости подправить параметры. Порядок элементов в массиве тестовых изображений не меняется

### **Лабораторная работа 1. *Линейная регрессия в задаче распознавания символов***

Исследования проводить по параметрам и условиям представленным в файле Excel «4. *Исследования Линейная регрессия Таблицы.xls*»

**Задание 1.** Подобрать параметры таким образом, чтобы точность распознавания составила как минимум 50%. Время обучения модели ограничено 5-ю минутами.

Внешний вид окна для лабораторной работы рис .

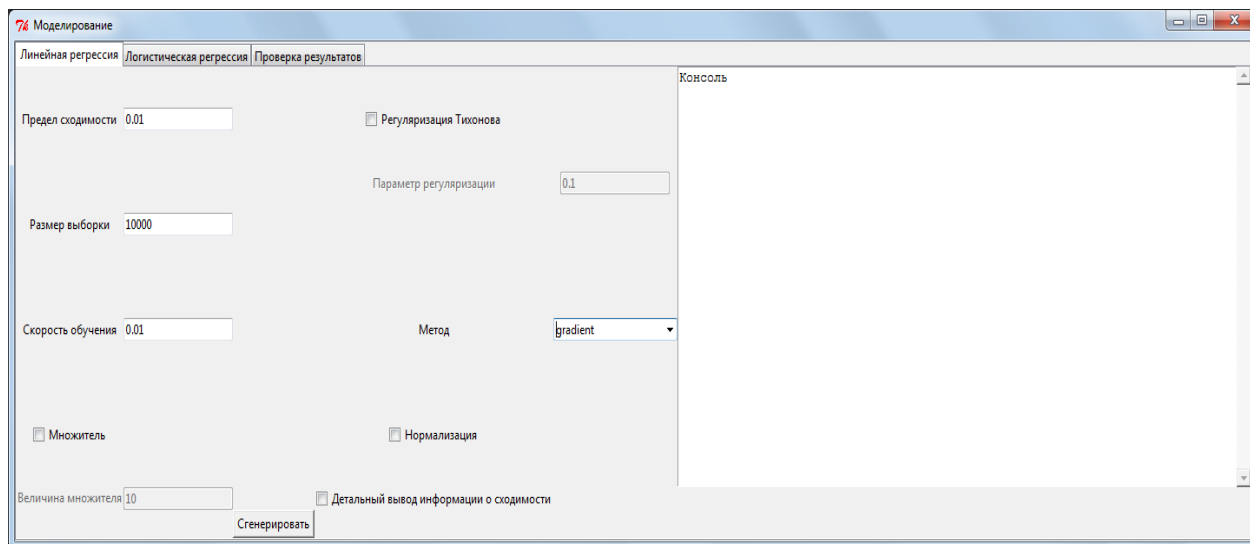


Рис.1.6 Интерфейс пользователя

*Рекомендуемые интервалы для параметров:*

- *предел сходимости (эпсилон из теоретической части) – [0.001 - 10]*
- *размер выборки (количество изображений в обучающей выборке) – [10 – 50000]*

- *скорость обучения (лямбда из теоретической части) – [0.001 - 1]*
- *параметр регуляризации (альфа из теоретической части) – [0.1 - 10]*
- *величина множителя – [0.1 - 255]*

**Задание 2.** Для всех методов: построить графики зависимостей:

- *времени обучения модели от предела сходимости*
- *качества обучения от предела сходимости*
- *времени обучения модели от размера выборки*

**Для метода градиентного спуска:**

- *качество обучения от скорости обучения*
- *время обучения от скорости обучения*

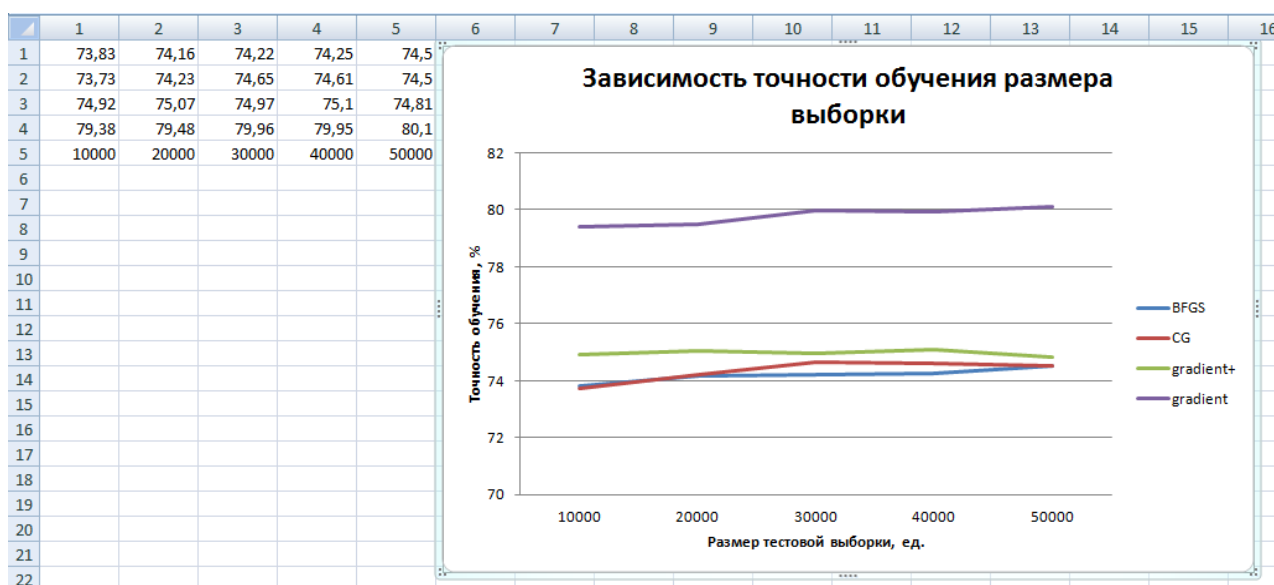


Рис. 1.7 Зависимости точности обучения от размера выборки

График построен средствами MS Excel. Для метода градиентного спуска использовалась скорость обучения = 0.01, для метода наискорейшего спуска = 0.001. Для всех методов предел сходимости = 0.01, без регуляризации, нормализации и множителя. Из графика видно, что для используемой модели точность не меняется от изменения размера тестовой выборки. Цель построения подобных графиков – выявить остальные зависимости или их отсутствие.

Также для подтверждения предыдущего тезиса был построен дополнительный график с логарифмической шкалой на оси X **рис 1.8.**

**Термины, значение которых изучается самостоятельно:**

- переобучение, недообучение;
- робастность, робастное оценивание, методы оценивания качества обучения;
- методы линейной оптимизации (например, метод золотого сечения)
- Якобиан, матрица Якоби
- Гессиан функции, матрица Гессе

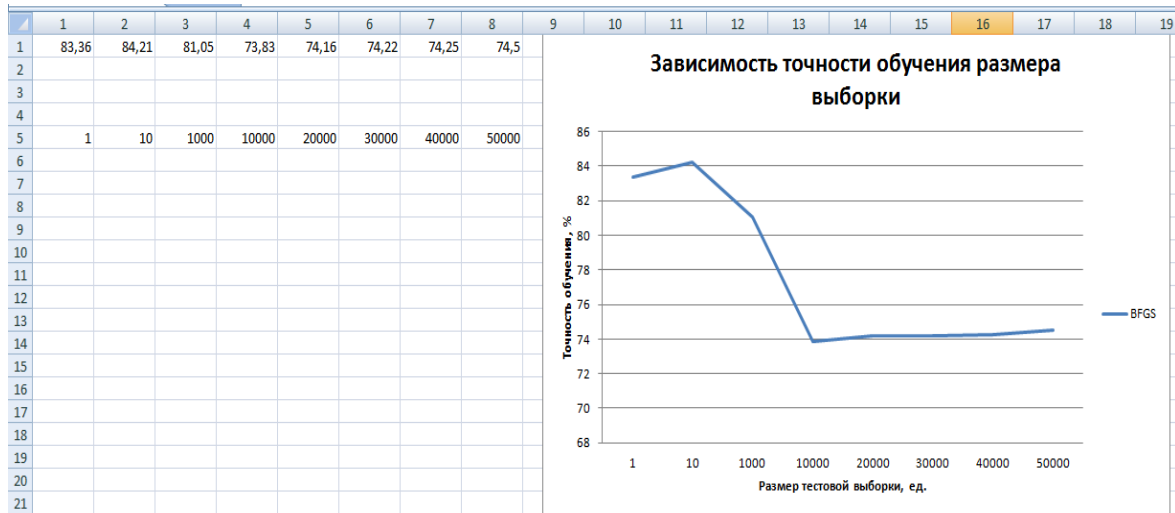


Рис. 1.8 Зависимости точности обучения от размера выборки

## Лабораторная работа 2. Логистическая регрессия в задаче распознавания символов

Лабораторная работа проводится на второй вкладке стенда. Параметры для модели те же самые, что и для линейной регрессии, но методы значительно различаются.

Исследования проводить по параметрам и условиям представленные в файле Excel «4. **Исследования Линейная регрессия Таблицы.xls**»

**Задание 1.** Подобрать параметры таким образом, чтобы для метода градиентного спуска процент ошибок распознавания на тестовой выборке составил менее 8%. Время обучения менее 5 минут.

Рекомендуемые интервалы для параметров:

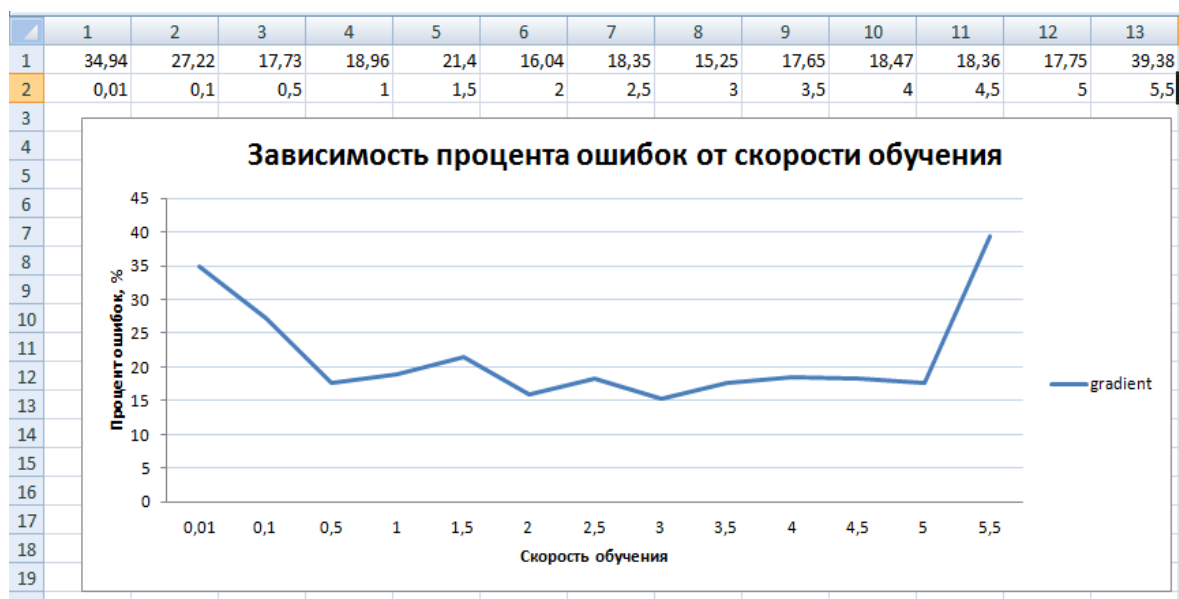
- предел сходимости (эпсилон из теоретической части) – [0.001 - 10]
- размер выборки (количество изображений в обучающей выборке) – [10 – 50000]
- скорость обучения (лямбда из теоретической части) – [0.1 - 10]

- параметр регуляризации (альфа из теоретической части) – [0.1 - 10]
- величина множителя – [0.1 - 255]

**Задание 2. Построить графики зависимостей:**

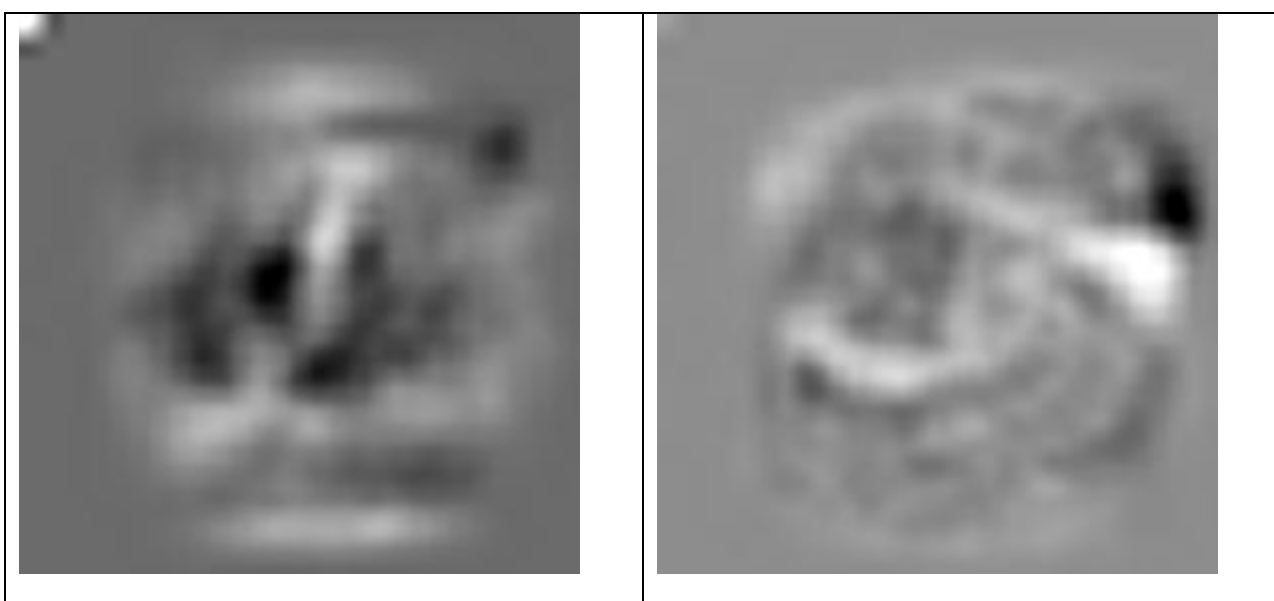
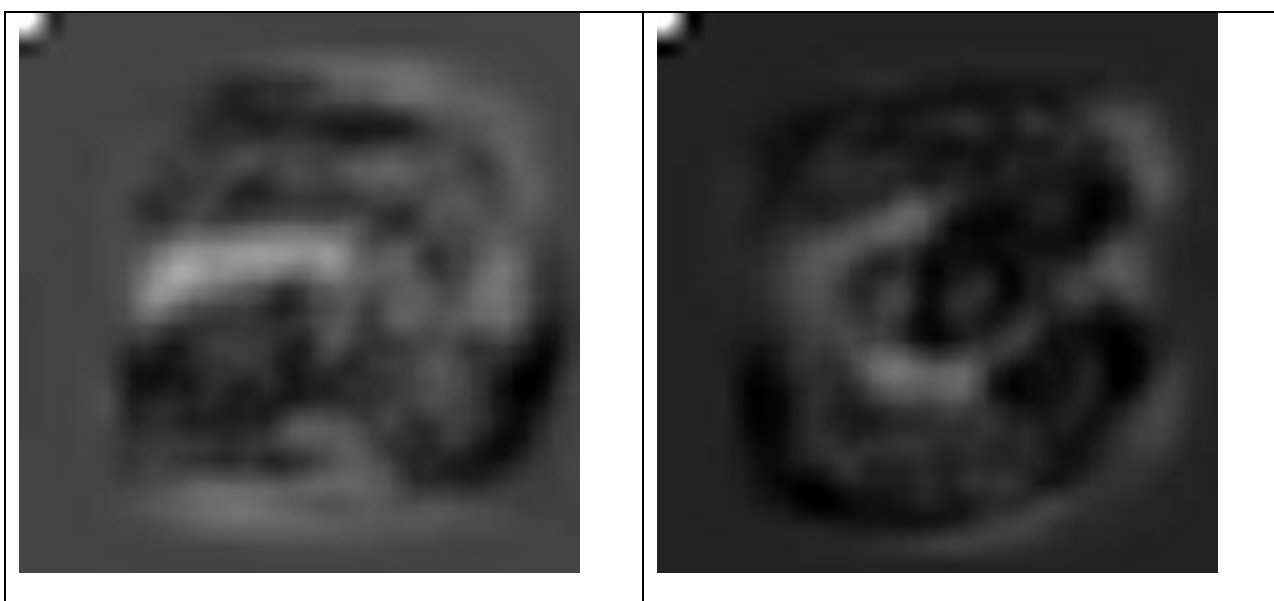
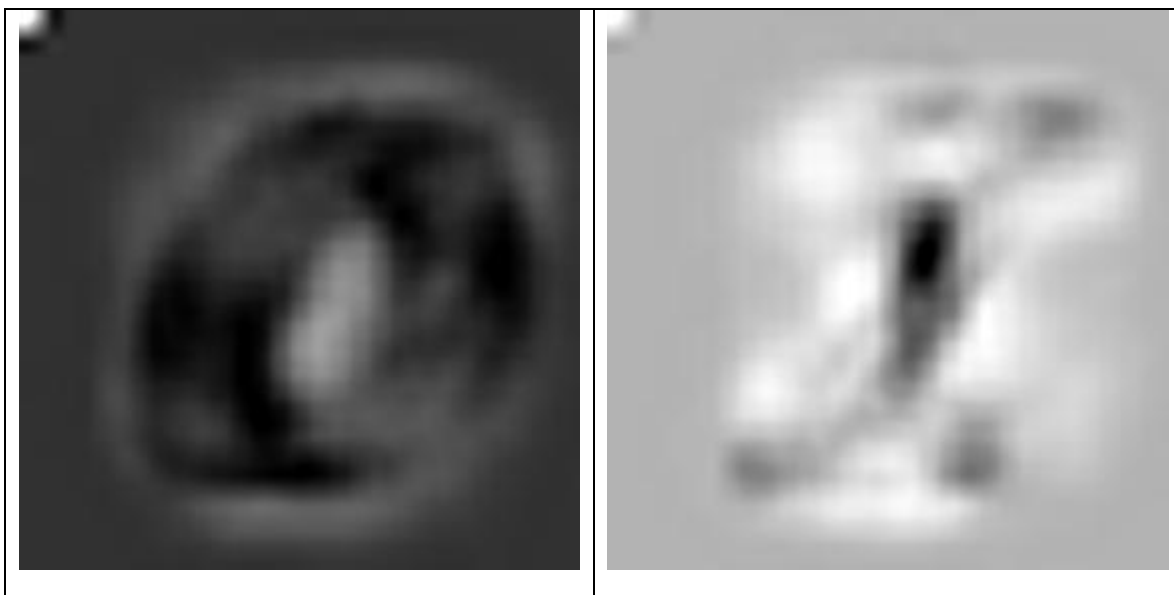
- качества обучения от размера выборки
- качества обучения от величины предела сходимости
- влияние регуляризации на качество и скорость обучения
- для градиентного спуска влияние скорости обучения на качество и время обучения

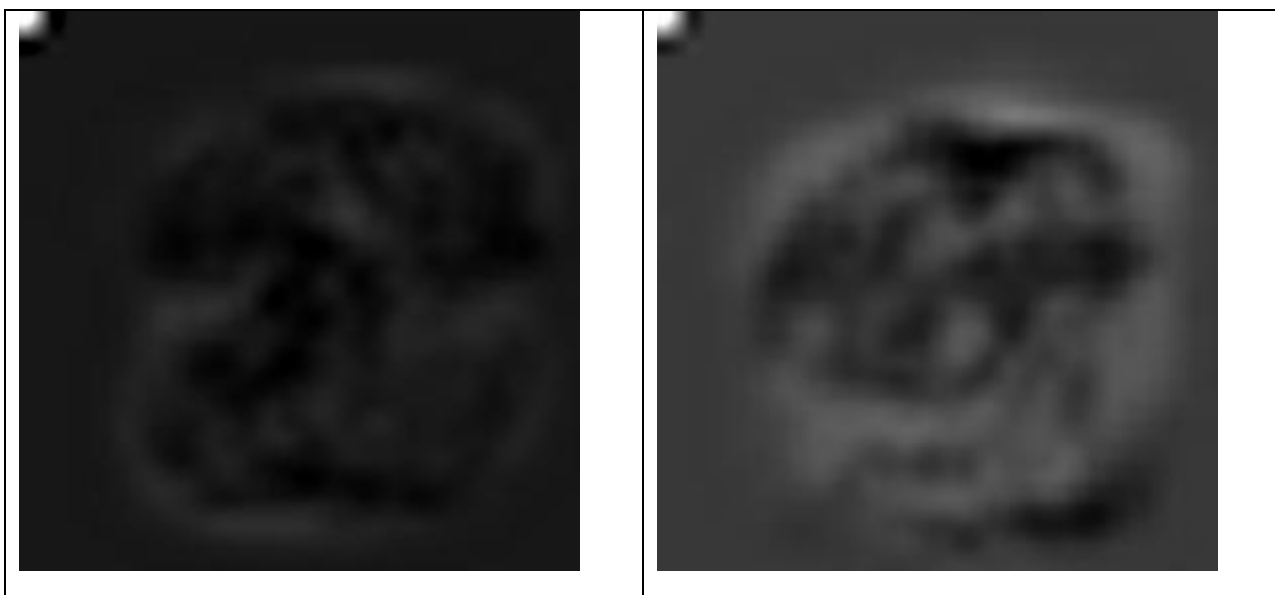
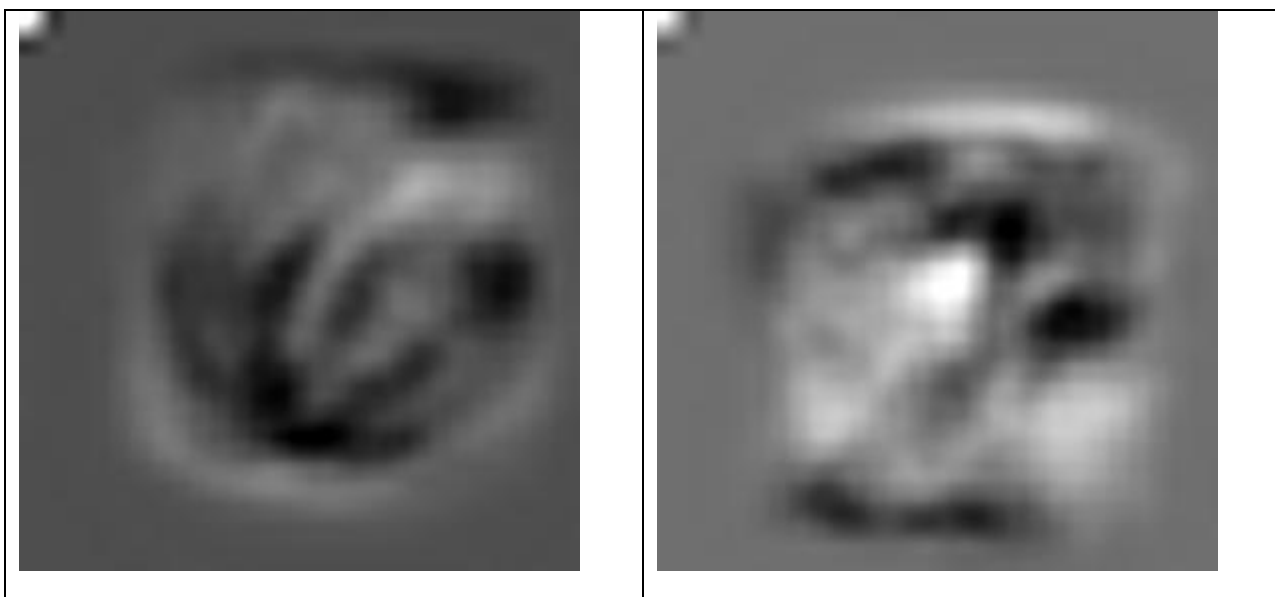
Пример зависимости качества обучения от скорости обучения для градиентного спуска. Использовалась выборка из 50000 изображений, предел сходимости = 0.01, без регуляризации, нормализации и множителя.



**Задание 3. В отчете объяснить влияние нормализации на качество обучения.**

**Дополнение.** В ходе работы модели должны были получиться вот такие «усредненные» изображения, по которым в дальнейшем ведется распознавание. Эти изображения и есть вектор параметров, что должен был вычислиться в процессе обучения. На каждую цифру – вектор параметров из 784 элементов, все цифры указаны по порядку и их размер увеличен для наглядности в 10 раз.







***Термины, значение которых изучается самостоятельно:***

- нейронные сети, персептрон***
- преобразование Хаффа***

## **6. Содержание отчета по лабораторной работе**

### ***6.1 Цель лабораторной работы***

### ***6.2. Теоретические основы предметной области***

### ***6.3 Задачи лабораторной работы***

### ***6.4. Результаты, полученные в лабораторной работе***

### ***6.5 Выводы.***

## 7. Контрольные вопросы

*При ответе указывайте источник. Если на ответ натолкнула конкретная формула, то вплоть до нее.*

1. Как происходит собственно процесс обучения и распознавания? Как машина может дать ответ на вопрос, какой это символ, имея только нужное изображение и матрицу параметров?
2. Как оценить, насколько уверенно модель распознает символы?
3. Как улучшить качество обучения для логит-регрессии?
4. Насколько сильно на результат влияет регуляризация и почему?
5. Насколько сильно на результат влияет нормализация и почему?
6. Насколько сильно на результат влияет множитель и почему?
7. Если посмотреть на результат обучения модели, то в левом верхнем углу заметно белое пятно. Это влияние так называемого bias\threshold нейрона. Как видно из картинок, для нашей модели использование этого приема вообще не имеет смысла и оказывает небольшое (совсем небольшое) негативное влияние. Какое именно?
8. Вводная часть из предыдущего вопроса. Вопрос: зачем используют нейрон-смещение в нейронных сетях (на примере персептрона)?
9. Для обучения нейронных сетей (и не только) используются в частности методы прямого и обратного распространения ошибки. Почему не используют градиентный спуск, например?
10. Почему слагаемое регуляризации складывают со значением функции стоимости, а не отнимают? Почему не умножают или делят?
11. Назовите 3 метода регуляризации. Опишите плюсы и минусы каждого и в каких случаях следует использовать каждый из них.
12. На примере графика зависимости процента ошибок от скорости обучения видно, что значение скорости должно лежать в интервале от 0.5 до 5, иначе результаты неудовлетворительны. Почему значение меньше 0.5 не подходит? Почему не подходит более 5? Может ли оказаться так, что есть подходящие значения скорости далеко за



- пределами данного интервала (менее 0.001 или более 1000, например), а данный интервал лишь локальный минимум, но не глобальный?
- 13.**Метод градиентного спуска иногда “зацикливается” на одних и тех же значениях при определенных значениях скорости обучения ( $=2$ ,  $=3$ ) и предела сходимости. Объясните, почему это происходит и как это предотвратить?
- 14.**Можно ли все вычисления или их часть перенести на GPU? Почему? И с какой целью это желание вообще может появиться? Какие библиотеки можно использовать для этой цели? Назовите задачи, для решения которых сегодня применяют вычисления на GPU или аналогичных (в данном контексте) устройствах.
- 15.**Альтернативы вычислениям на GPU?
- 16.**Почему нельзя остановить поток, который производит все вычисления? Почему вообще нельзя взять и остановить любой поток?
- 17.**Похож ли данный метод (логит-регрессия) на нейронные сети и чем?
- 18.**Как может помочь преобразование Хаффа улучшить качество обучения?
- 19.**Как скажется на скорости и качестве обучения увеличение размера исходных изображений в несколько раз (в 2 раза, в десять раз)? (P.S. рассмотреть два случая: масштабирование исходных изображений и новые изображения лучшего качества)
- 20.**Почему метод оптимизации BFGS или CG в общем достигает лучших результатов, нежели градиентный спуск, хотя и у одного, и у другого одна задача – найти минимум функции?
- 21.**Можно ли для градиентного спуска указать отрицательную или нулевую скорость обучения и к каким последствиям это приведет? Для всех ли исходных данных для обучения оптимальная скорость обучения одинакова и почему (допустим, мы сменим изображения цифр на котиков, нужно ли поменять параметр скорости или нет)?

- 22.**Почему в программе нельзя указать количество изображений в тестовой выборке равным нулю? Ошибка в коде программы или ограничение методов? Почему нельзя указать отрицательное число?
- 23.**Какими функциями из языка программирования R можно воспользоваться для написания регрессионных моделей? Как насчет примера кода?

## 8. Литература

1. Тоноян С.А., Балдин А.В., Елисеев Д.В. Анализ избыточности хранения темпоральных данных средствами реляционных СУБД <http://engjournal.ru/catalog/it/hidden/1273.html>. 2014г.
2. Тоноян С.А., Сараев Д.В. Темпоральные модели базы данных и их свойства. <http://engjournal.ru/catalog/it/hidden/1333.html>. 2014.
3. Тоноян С.А., Балдин А.В., Елисеев Д.В. Язык запросов к миварному представлению реляционных баз данных, содержащих архив информации из предыдущих кадровых систем. <http://engjournal.ru/catalog/it/hidden/1053.html>. 2013г.
4. Барсегян А.А., Куприянов М.С., Холод И.И. «Анализ данных». БХВ-Петербург 2009г. 512с.
5. Куприянов М.С, Холод И.И0, Барсегян А.А. «Анализ данных и процессов». БХВ-Петербург 2009г. 512с.
6. Джестон Д., Нелис И. «Управление бизнес процессами». Пер с англ. С-Петербург 208г -512с.
7. Тоноян С.А. Черненький В.М., Балдин А.В., Информационная управляющая система МГТУ им. Н.Э.Баумана «Электронный Университет». Изд. МГТУ им. Н.Э.Баумана 2009. -376 с. 304-325.
8. Тоноян С.А., Балдин А.В., Елисеев Д.В. «Методика модернизации стандартных модулей типовой конфигурации на базе технологической платформы «1С: Предприятие 8» с минимальными доработками». Наука и образование (МГТУ им. Н.Э. Баумана). № 08, август 2012 URL: <http://technomag.edu.ru/>.
9. Официальный сайт компании QlikView - <http://www.qlikview.com/ru>