Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Машинное обучение (МО)

Отчет о лабораторной работе №4

«Нейронные сети»

|  |
| --- |
| **Выполнил:** |
| **Студент гр. 858341** |
| **Немкович А. В.** |

|  |
| --- |
| **Проверил:** |
| **Стержанов М. В.** |

**Минск 2019**

Содержание

[Постановка задачи 3](#_Toc25275643)

[Ход выполнения 4](#_Toc25275644)

[Выводы 12](#_Toc25275645)

# Постановка задачи

Набор данных **ex4data1.mat** (такой же, как в лабораторной работе №2) представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит 5000 изображений 20x20 в оттенках серого. Каждый пиксель представляет собой значение яркости (вещественное число). Каждое изображение сохранено в виде вектора из 400 элементов. В результате загрузки набора данных должна быть получена матрица 5000x400. Далее расположены метки классов изображений от 1 до 9 (соответствуют цифрам от 1 до 9), а также 10 (соответствует цифре 0).

**Задание.**

1. Загрузите данные **ex4data1.mat** из файла.
2. Загрузите веса нейронной сети из файла **ex4weights.mat**, который содержит две матрицы Θ(1) (25, 401) и Θ(2) (10, 26). Какова структура полученной нейронной сети?
3. Реализуйте функцию прямого распространения с сигмоидом в качестве функции активации.
4. Вычислите процент правильных классификаций на обучающей выборке. Сравните полученный результат с логистической регрессией.
5. Перекодируйте исходные метки классов по схеме one-hot.
6. Реализуйте функцию стоимости для данной нейронной сети.
7. Добавьте L2-регуляризацию в функцию стоимости.
8. Реализуйте функцию вычисления производной для функции активации.
9. Инициализируйте веса небольшими случайными числами.
10. Реализуйте алгоритм обратного распространения ошибки для данной конфигурации сети.
11. Для того, чтобы удостоверится в правильности вычисленных значений градиентов используйте метод проверки градиента с параметром ε = 10-4.
12. Добавьте L2-регуляризацию в процесс вычисления градиентов.
13. Проверьте полученные значения градиента.
14. Обучите нейронную сеть с использованием градиентного спуска или других более эффективных методов оптимизации.
15. Вычислите процент правильных классификаций на обучающей выборке.
16. Визуализируйте скрытый слой обученной сети.
17. Подберите параметр регуляризации. Как меняются изображения на скрытом слое в зависимости от данного параметра?
18. Ответы на вопросы представьте в виде отчета.

# Ход выполнения

**Задания 1-2**: Загрузка данных ex4data1.mat из файла и весов нейронной сети из файла ex4weights.mat, который содержит две матрицы Θ(1) (25, 401) и Θ(2) (10, 26) .

Получения нейронная сеть имеет 3 слоя. На входном слое размещается 400 нейронов, на скрытом – 25, и на выходном – 10.

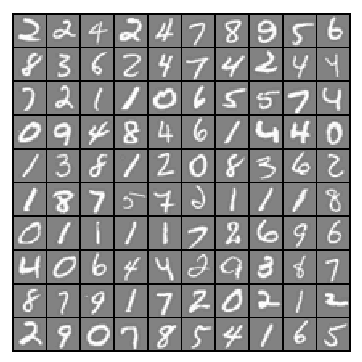


Рис 1. – визуализация данных из тренировочного набора (ex4data1.mat)

**Задание 3**: Реализация функции прямого распространения с сигмоидом в качестве функции активации.

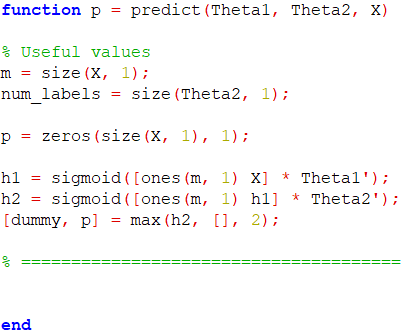


Рис 2. – Код функции прямого распространения

**Задание 4**: Вычисление процента правильных классификаций на обучающей выборке.



Рис 3. – Доля правильных классификаций, используя нейронную сеть

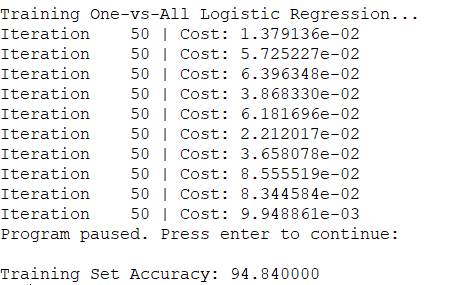


Рис 4. – Доля правильных классификаций, используя логическую регрессию

**Задание 5**: Перекодировка исходных меток классов по схеме one-hot.

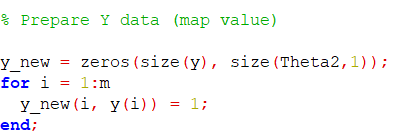
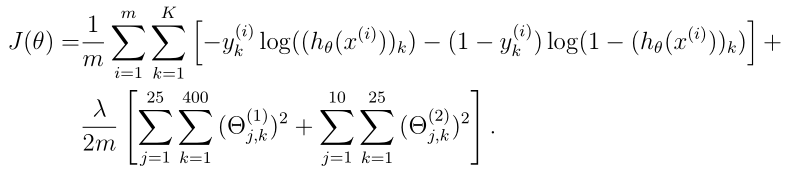


Рис 5. – Код перекодировки исходных меток классов

**Задания 6-7**: Реализация функции стоимости для данной нейронной сети. Добавление L2-регуляризации в функцию стоимости.

Формула L2-регуляризации для функции стоимости для данной нейронной сети:



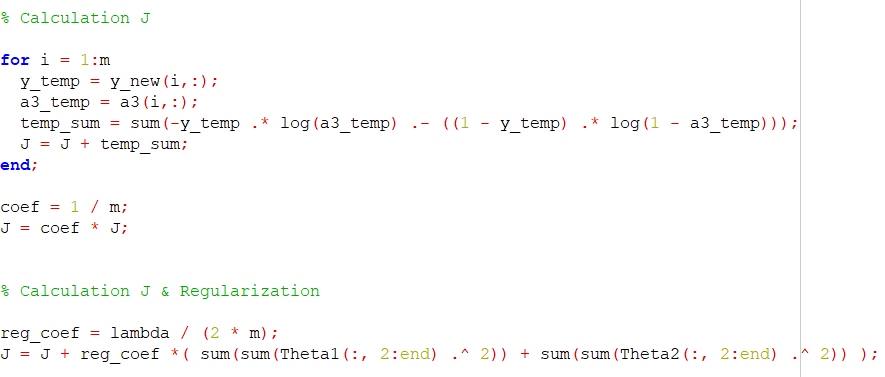
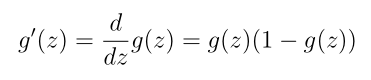


Рис 6. – код функции стоимости с L2-регуляризацией

**Задание 8**: Реализация функции вычисления производной для функции активации.

Формула производной для функции активации:



где



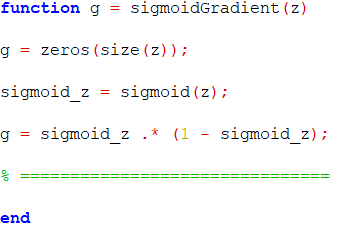


Рис 7. – код функции sigmoidGradient

**Задание 9**: Инициализация весов небольшими случайными числами.

Для инициализации весов небольшими случайными числами использовалась функция randInitializeWeights:

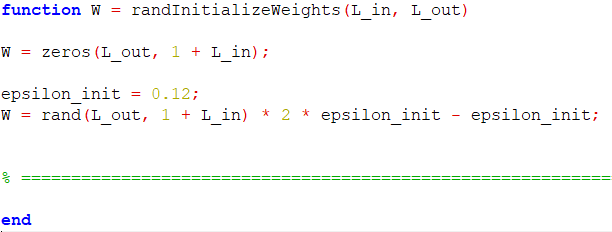


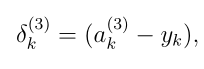
Рис 8. – Код функции randInitializeWeights

**Задание 10**: Реализация алгоритма обратного распространения ошибки для данной конфигурации сети.

Алгоритм обратного распространения доя данной конфигурации сети:

1. Выполнить алгоритм прямого распространения для получения значений на нейронах выходного слоя, используя один из примеров из обучающей выборки;

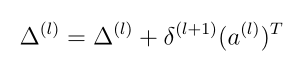
2. Для каждого нейрона выходного слоя вычислить:



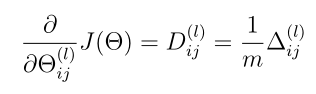
3. Для скрытого слоя вычислить:



4. Рассчитать градиент для этого примера:



5. Получить градиент для функции стоимости нейронной сети, разделив рассчитанные градиенты на 1/m.



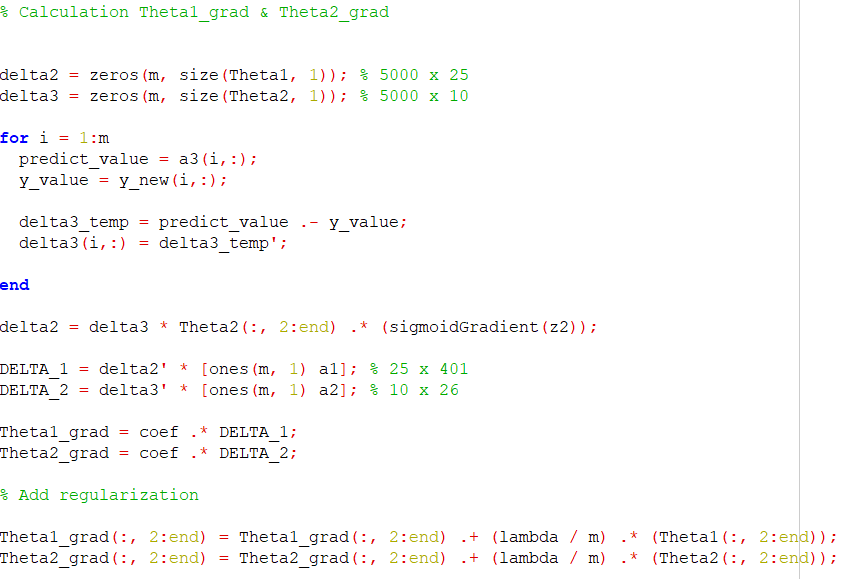


Рис 9. – Код алгоритма обратного распространения

**Задания 10-15**: Проверка градиента, чтобы подтвердить, что обратное распространение работает.

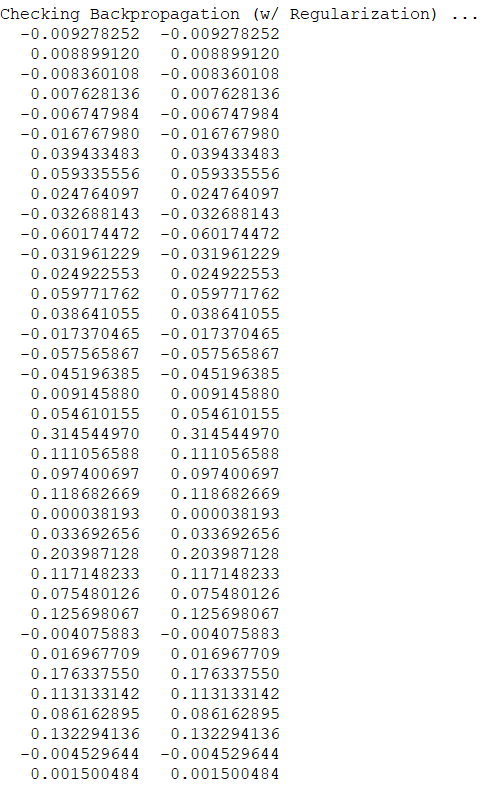


Рис 10. – Сравнение градиентов

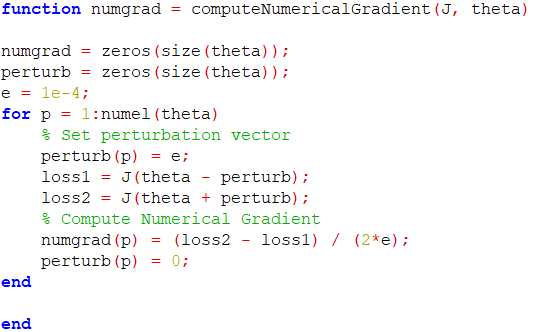


Рис 11. – Код функции проверки градиента

**Задание 16**: Визуализация скрытого слой обученной сети.

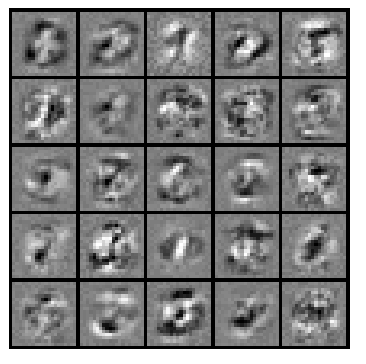


Рис 12. – скрытый слой обученной сети с параметром регуляризации = 1

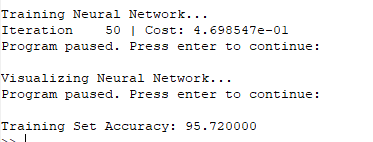


Рис 13. – процент правильной классификации на обучающей выборке с параметром регуляризации = 1

**Задание 17**: Подбор параметров регуляризации.

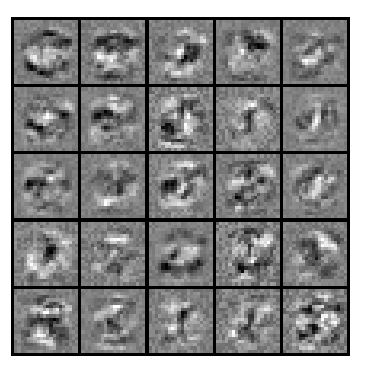


Рис 14. – скрытый слой обученной сети с параметром регуляризации = 0

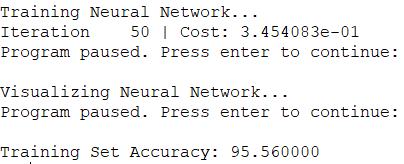


Рис 15. – процент правильной классификации на обучающей выборке с параметром регуляризации = 0

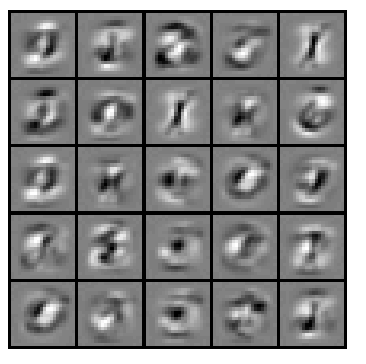


Рис 16. – скрытый слой обученной сети с параметром регуляризации = 100

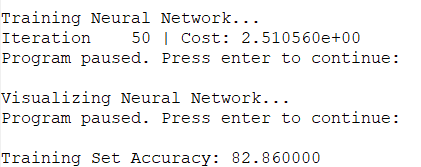


Рис 17. – процент правильной классификации на обучающей выборке с параметром регуляризации = 100

# Выводы

Мы научились использовать нейронную сеть для классификации изображений. В ходе работу изучили алгоритмы прямого и обратного распространения. Изучили влияние параметра регуляризации на процент правильной классификации.

Алгоритм обучения нейронной сети:

* Инициализация весов небольшими случайными числами;
* Реализация прямого распространение;
* Реализация функции стоимости;
* Реализация обратного распространения для вычисления частных производных;
* Проверка градиента, чтобы подтвердить, что обратное распространение работает. Затем отключение проверку градиента;
* Использование градиентного спуска или встроенной функции оптимизации, для минимизация функции стоимости с весами в тета.