|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | | |
| Факультет | Компьютерных сетей и систем | | |
| Кафедра | Информатики  Дисциплина: Конструирование те технологии электронных вычислительных средств | | |
|  |  | | |
| **лабораторная работа №4**  по курсу Машинное обучение  на тему  **Нейронные сети** | | | |
| Магистрант:  гр. 858341  Кудин Н.И. | |  | Проверил:  Стержанов М.В. |
| Минск, 2019 | | | |

**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).

**Ход работы**

1. Загружаем данные из файлов ex4data1.mat, ex4weights.mat и определение структуры нейронной сети в память

На основании исходных данных можно сделать вывод, что нейронная сеть должна иметь 3 слоя:

* входной слой с 400 входных нейронов (один на каждый входной пиксель);
* скрытый слой с 25 нейронами (исходя из входной матрицы весов);
* выходной слой с 10 нейронами (один на каждый возможный результат, 0-9).

2. One Hot Encoding предполагает создание 10 признаков, все из которых равны нулю за исключением одного.

После кодирования мы получим матрицу матрицу:

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

1. Выполним проверку градиента (без регуляризации). Для заданных весов выполним алгоритм обратного распространения ошибки. Результаты работы алгоритма передадим в алгоритм проверки. Проверка идёт на случайных результатах (случайный вес, [строка, столбец]):

gradient\_check([theta\_1, theta\_2], [D1, D2], x\_train, y\_train):

|  |  |  |
| --- | --- | --- |
| [theta\_1, theta\_2] | check gradient | BP gradient |
| [8, 350] | 0.000062 | 0.000062 |
| [18, 112] | -0.000064 | -0.000064 |
| [9, 95] | 0.000165 | 0.000165 |
| [23, 309] | -0.000205 | -0.000205 |
| [9, 23] | 0.000056 | 0.000056 |
| [8, 16] | -0.000279 | -0.000279 |
| [2, 23] | 0.000845 | 0.000845 |

*Так как градиенты одинаковые, значит алгоритм работает корректно.*

1. Теперь выполним проверку градиента с регуляризацией.

gradient\_check(np.array([theta\_1, theta\_2]), [D1, D2], x\_train, y\_train, **0.5**):

|  |  |  |
| --- | --- | --- |
| [theta\_1, theta\_2] | check gradient | BP gradient |
| [5, 20] | -0.000045 | -0.000045 |
| [6, 9] | -0.000228 | -0.000228 |
| [4, 22] | 0.001833 | 0.001833 |
| [9, 23] | 0.000025 | 0.000025 |
| [12, 392] | 0.000004 | 0.000004 |
| [21, 385] | -0.000036 | -0.000036 |
| [14, 328] | 0.000061 | 0.000061 |

*Так как градиенты также сходятся (с регуляризацией в 0.5), алгоритм работает корректно.*

1. Обучение нейронной сети и подсчёт верных классификаций

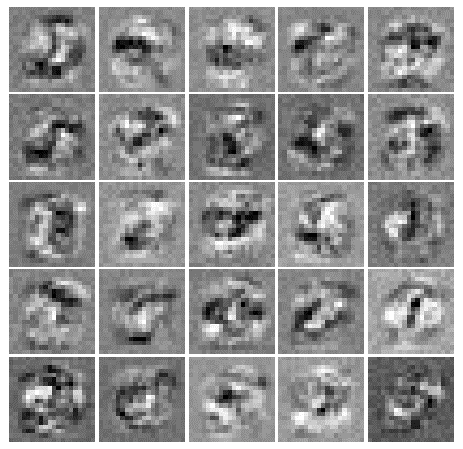
Обучим нейронную сеть на предоставленных данных предварительно сгенерировав случайные веса:

*learned\_thetas = train(x\_train, y\_train, 0., 50)*

Затем определим процент верных классификаций:

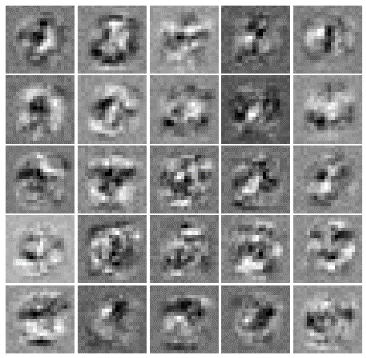
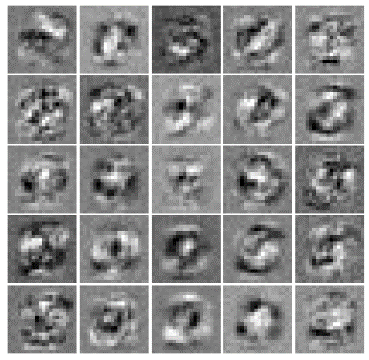
*compute\_accuracy(learned\_thetas, x\_train, y\_train): 97.325%*

Скрытый слой для выученных весов:

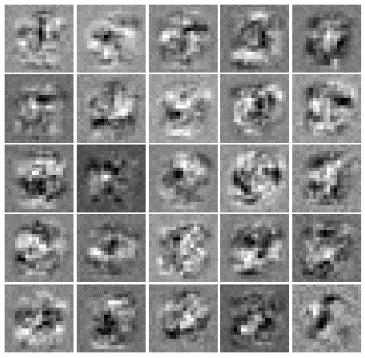
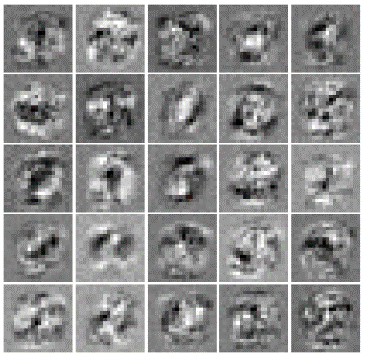


1. Подберем параметр регуляризации и визуализация скрытого слоя

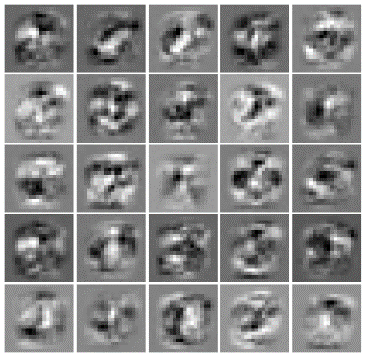
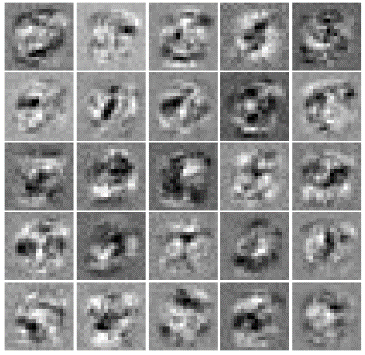
Обучим нейронную сеть с следующими параметрами регуляризации: [0, 0.001, 0.01, 0.1, 1, 10, 30]. Визуализации скрытых слоёв показаны на рисунках 2-5 (значения идут слева направо, сверху вниз).



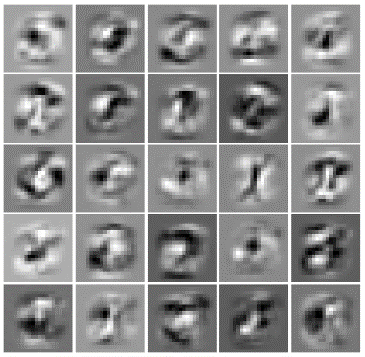
Регуляризация 0 / 0.001 с точностью 96.825% / 97.375%



Регуляризация 0.01 / 0.1 с точностью 97.825% / 97.6%

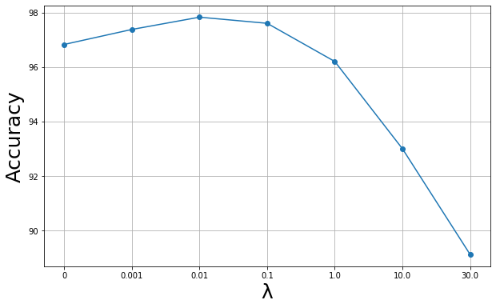


Регуляризация 1 / 10 c точностью 96.2% / 93.0%



Регуляризация 30 c точностью 89.125%

Из визуализаций видно, что чем больше значение параметра регуляризации (после определённой точки), тем детализция более низкая и процент верхных предсказаний падает. График зависимости:



**Заключение**

Выполнив лабораторную работу, мы получили необходимы навыки для развертывания нейронной сети, реализации функции прямого распространения (с сигмоидом в качестве функции активации), вычислили процент правильных классификаций в выборке, а также перекодировали метки классов по схеме one-hot.

Основное преимущество нейронных сетей перед другими методами машинного обучения состоит в том, что они могут распознавать более глубокие, иногда неожиданные закономерности в данных. В процессе обучения нейроны способны реагировать на полученную информацию в соответствии с принципами генерализации, тем самым решая поставленную перед ними задачу. Нейронные сети могут использоваться практически для чего угодно в силу своей универсальности. Однако для того, чтобы они работали должным образом необходимо много предварительной работы.