Нижегородский государственный университет им. Н. И. Лобачевского
Институт информационных технологий, математики и механики

Направление подготовки «Прикладная математика и информатика»

Магистерская программа Вычислительные методы и суперкомпьютерные технологии

Образовательный курс «Методы глубокого обучения для решения задач компьютерного зрения»

# Отчёт

по лабораторной работе № 1

**«Реализация метода обратного распространения ошибки для двуслойной полностью связанной нейронной сети»**

***Выполнила:***
студентка гр. 381703м3
Подчищаева М. В.

Нижний Новгород
2018

# 1. Постановка задачи

В ходе лабораторной работы предполагается решение ***следующих задач***:

- Изучение общей схемы метода обратного распространения ошибки.

- Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.

- Проектирование и разработка программной реализации сети, решающей задачу классификации рукописных символов.

- Тестирование разработанной программной реализации.

- Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

Для обучения и тестирования сети предполагается использовать набор данных MNIST.

## 2. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов и общая схема метода обратного распространения ошибки

Пусть нейронная сеть имеет: N - входных нейронов, K - нейронов на скрытом слое и M - выходных.

В целом метод обратного распространения ошибки сводится к задаче минимизации функции ошибки относительно синаптических весов. В качестве целевой функции будем использовать кросс энтропию.

$$E(w) = -\frac{1}{L}\sum_{k=1}^{L}\sum_{j=1}^{M} y_j^k \ln\left(u_j^k\right),$$

где $y^k = \left(y_j^k\right),\ j = \overline{1,M}, k = \overline{1,L}$ – $k$-й пример из обучающей выборки, $u^k = \left(u_j^k\right), j = \overline{1,M}, k = \overline{1,L}$– выход нейронной сети на входе $x^k = \left(x_i^k\right),\ i = \overline{1,\mathrm{N}}, k = \overline{1,L}$. Также учтем, что $x_0 = 1$ – новый нейрон.

Для вывода формул и реализации будем использовать предположение, что режим обучения последовательный. Корректировка весов выполняется для каждого элемента обучающей выборки, а также $\sum_{j=1}^{M} y_j = 1$ для задач классификации.

Выведем сначала формулы для одного конкретного элемента выборки. Пусть: $x^* = (x_i^*),\ i = \overline{1,\mathrm{N}}$, $y^* = \left(y_j^*\right),\ j = \overline{1,M}, u^* = \left(u_j^*\right), j = \overline{1,M}$, тогда целевая функция для конкретного примера имеет вид:

$$E(w) = -\sum_{j=1}^{M} y_j \ln\left(u_j\right)$$

Обозначим синоптические веса от входного слоя к скрытому - $\mathrm{w}_{\mathrm{is}}^{(1)}$, от скрытого к выходному - $\mathrm{w}_{\mathrm{sj}}^{(2)}$. Также обозначим $h\left(g_j\right) = \frac{e^{g_j}}{\sum_{i=1}^{M} e^{g_i}}$ – функция активации на выходном слое (softmax), $\varphi(f_s) = \tanh(g_j)$ – функция активации на скрытом слое (гиперболический тангенс). В нашем случае все формулы для прямого прохода от входа к выходу выглядят следующим образом:

$$f_s = \sum_{i=0}^{N} w_{is}^{(1)} x_i\,, s = \overline{0,\mathrm{K}}$$

$$v_s = \varphi(f_s), s = \overline{0,\mathrm{K}}$$

$$g_j = \sum_{s=0}^{K} w_{sj}^{(2)} v_s, j = \overline{1,\mathrm{M}}$$

$$u_j = h\left(g_j\right), j = \overline{1,\mathrm{M}}$$

Итак, с учетом всех вышеуказанных формул и предположений:

$$E(w) = -\sum_{j=1}^{M} y_j \ln\left(\frac{e^{g_j}}{\sum_{i=1}^{M} e^{g_i}}\right) = -\sum_{j=1}^{M} y_j \left(g_j - \ln\left(\sum_{i=1}^{M} e^{g_i}\right)\right),$$

$$g_j = \sum_{s=0}^{K} w_{sj}^{(2)} \varphi\left(\sum_{i=0}^{N} w_{is}^{(1)} x_i\right)$$

Поскольку уже говорилось, что задача сводится к минимизации функции ошибки, в частности градиентными методами, то корректировка весов должна выполнятся по следующим формулам:

$$\mathrm{w}_{\mathrm{is}}^{(1)(\mathrm{r}+1)} = \mathrm{w}_{\mathrm{is}}^{(1)(\mathrm{r})} - \eta \frac{\partial \mathrm{E(w)}}{\partial \mathrm{w}_{\mathrm{is}}^{(1)}}$$

$$\mathrm{w}_{sj}^{(2)(\mathrm{r}+1)} = \mathrm{w}_{\mathrm{sj}}^{(2)(\mathrm{r})} - \eta \frac{\partial \mathrm{E(w)}}{\partial \mathrm{w}_{\mathrm{sj}}^{(2)}},$$

где $\eta$ – скорость обучения ($0 < \eta < 1$).

Получим градиент для корректировки весов.

Частная производная целевой функции по весам от скрытого к выходному слою.

$$\frac{\partial E(w)}{\partial w_{sj}^{(2)}} = \frac{\partial E}{\partial g_j} \frac{\partial g_j}{\partial w_{sj}^{(2)}} = \delta_j^{(2)} v_s$$

$$\frac{\partial g_j}{\partial w_{sj}^{(2)}} = v_s$$

$$\delta_j^{(2)} = \frac{\partial E}{\partial g_j} = \left(\sum_{j=1}^{M} y_j\right) \frac{e^{g_j}}{\sum_{i=1}^{M} e^{g_i}} - y_j = \frac{e^{g_j}}{\sum_{i=1}^{M} e^{g_i}} - y_j = u_j - y_j$$

Частная производная целевой функции по весам от входного к скрытому слою.

$$\frac{\partial E(w)}{\partial w_{is}^{(1)}} = \frac{\partial E}{\partial f_s} \frac{\partial f_s}{\partial w_{is}^{(1)}} = \delta_s^{(1)} x_i$$

$$\delta_s^{(1)} = \frac{\partial E}{\partial f_s} = \sum_{j=1}^{M} \frac{\partial E}{\partial g_j} \frac{\partial g_j}{\partial v_s} \frac{\partial \varphi}{\partial f_s} = \frac{\partial \varphi}{\partial f_s} \sum_{j=1}^{M} \frac{\partial E}{\partial g_j} \frac{\partial g_j}{\partial v_s} = \frac{\partial \varphi}{\partial f_s} \sum_{j=1}^{M} \delta_j^{(2)} w_{sj}^{(2)}$$

$$\frac{\partial \varphi}{\partial f_s} = (1-\varphi)(1+\varphi) = (1-v_s)(1+v_s)$$

Алгоритм метода обратного распространения ошибки

1. Инициализация весов $w$ некоторыми значениями

2. $for\ epoch\ = \overline{1, maxEpochs}$

3. $for\ i\ = \overline{0, W \cdot H}$

4. Прямой проход нейронной сети

5. Обратный проход

6. Шаги 3-5 повторяются до тех пока, пока не выполнится критерий остановки. Как правило, это либо максимальное число эпох, либо достигнутая точность обучения.

<u>Прямой проход.</u>

На вход подается $x_i$. Необходимо вычислить значения выходных сигналов нейронов скрытого слоя $v_s$ , $s\ = \overline{0, K}$, где $K$- количество нейронов на скрытом слое и значение производной функции активации на скрытом слое $\frac{\partial \varphi}{\partial f_s}$ .

<u>Обратный проход:</u>

Вычислим значения градиентов целевой функции, начиная с конца:

$$\delta_j^{(2)} = u_j - y_j, \frac{\partial \mathrm{E(w)}}{w_{sj}^{(2)}} = \delta_j^{(2)} \cdot v_s\ , j\ = \overline{1, M}$$

Скрытый слой, $s\ = \overline{0, K}$ :

$$\delta_s^{(1)} = -\frac{\partial \varphi}{\partial \mathrm{f_s}}\left[\sum_{j=1}^{M} \delta_j^{(2)} \cdot w_{sj}^{(2)}\right], \frac{\partial \mathrm{E(w)}}{\partial w_{is}^{(1)}} = \delta_s^{(1)} \cdot x_i$$

По дугам:

$$w_{is}^{(1)} = w_{is}^{(1)} - \eta \frac{\partial \mathrm{E(w)}}{\partial w_{is}^{(1)}},$$

$$w_{sj}^{(2)} = w_{sj}^{(2)} - \eta \frac{\partial \mathrm{E(w)}}{w_{sj}^{(2)}}.$$

## 3. Описание программной реализации

В ходе лабораторной работы была реализована программа для обучения двуслойной нейронной сети, решающей задачу классификации, на языке программирования Python 3.6. Класс *NNetwork* решает задачу обучения нейронной сети. Программа принимает следующие параметры:

1. Количество эпох
2. Требуемое значение кросс-энтропии
3. Скорость обучения
4. Количество нейронов на скрытом слое

Если параметры не переданы, то в качестве вышеупомянутых значений будут использованы значения по умолчанию. Данные параметры передаются в конструктор класса *NNetwork*.

Для обучения созданной сети на примерах из обучающей выборки используется метод *train()*, на вход которому подаются изображения из обучающей выборки и векторы, имеющие одну единичную компоненту, номер которой соответствует цифре, изображенной на картинке. Обучение сети заканчивается либо по количеству эпох, либо по достигнутому значению точности.

Затем вызывается метод *test()*, принимающий на вход аналогичные данные для тестовой выборки и предсказывает по ним выход с помощью ранее обученной сети.

Для запуска программы необходимо клонировать репозиторий, установить пакет numpy и запустить python интерпретатор на файле Train.py.

## 4. Результаты экспериментов

Обучение сети проходило на тренировочной выборке размера 60000 изображений, тестирование - на тестовой выборке набора данных MNIST размера 10000 изображений. Требуемое значение кросс-энтропии для всех экспериментов было указано 0.001, количество эпох - 10. Результаты представлены в таблице ниже.

| Количество нейронов на скрытом слое | Скорость обучения | Точность на обучающей выборке | Точность на тестовой выборке |
|---|---|---|---|
| 500 | 0,01 | 0,95 | 0.93 |
| 500 | 0,001 | 0,92 | 0,91 |
| 800 | 0,01 | 0,95 | 0,95 |
| 800 | 0,001 | 0,93 | 0,91 |
| 1000 | 0,01 | 0,96 | 0,94 |