

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«Нижегородский государственный университет им. Н.И. Лобачевского»
Институт информационных технологий, математики и механики

Отчет по лабораторной работе

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

Выполнила Баestraкова Ксения
Группа 381603м4

Нижний Новгород, 2017

Содержание

Постановка задачи	3
Вывод математических формул	3
Программная реализация	4
Результаты экспериментов	5
Заключение.....	6
Список литературы.....	6

Постановка задачи

Необходимо изучить и реализовать метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двуслойной полносвязной сети (один скрытый слой). В качестве данных для обучения и тестирования сети необходимо использовать набор MNIST.

В работе решаются следующие задачи:

1. Изучение общей схемы метода обратного распространения ошибки.
2. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
3. Разработка программной реализации.
4. Тестирование разработанной программной реализации на наборе данных MNIST.

Вывод математических формул

Рассматривается двухслойная полносвязная нейронная сеть с N нейронами входного слоя, K нейронами скрытого слоя и M нейронами выходного слоя [1].

Обозначим w набор весов на всех уровнях.

В качестве функции ошибки используется кросс энтропия [1]

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

где $y^k = (y_j^k)$, $j = \overline{1, M}$, $k = \overline{1, L}$ – k -й пример из обучающей выборки, $u^k = (u_j^k)$, $j = \overline{1, M}$, $k = \overline{1, L}$ – выход нейронной сети на соответствующем входе.

Для вывода, а также в программной реализации будем использовать предположение, что используется последовательный режим обучения. В этом режиме корректировка весов выполняется после обработки каждого элемента обучающей выборки.

Рассмотрим некоторый обучающий пример:

$$\begin{aligned}x^k &= (x_j^k), \\y^k &= (y_j^k), \\u^k &= (u_j^k),\end{aligned}$$

где x – вектор входных параметров, y – вектор результатов сети, u – вектор правильных результатов.

В этом случае функция ошибки примет следующий вид:

$$E(w) = \sum_{j=1}^M y_i \ln(u_j).$$

Обозначим $w_{js}^{(1)}$ веса синаптических связей от входных нейронов к нейронам скрытого слоя, $w_{sj}^{(2)}$ – веса от нейронов скрытого слоя к выходным нейронам нашей сети.

В качестве функции активации на скрытом слое – гиперболический тангенс:

$$\varphi(f_s) = \text{th}(f_s).$$

Выходной сигнал нейрона скрытого слоя вычисляется как $v_s = \varphi(f_s)$. При этом f_s определяется как

$$f_s = \sum_{i=1}^N w_{is}^{(1)} x_i.$$

На последнем слое используется **SoftMax**-функция активации:

$$h(u_j) = \frac{e^{u_j}}{\sum_{j=1}^M e^{u_j}}$$

Сигнал выходного нейрона определяется как $u_j = h(g_j)$.

Запишем соответствующее значение кросс – энтропии:

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

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

Опишем метод обратного распространения ошибки: с помощью него определяются параметры сети w .

Запишем частную производную по каждому из весов:

$$\frac{\partial E(w)}{\partial w_{sj}^{(2)}} = \frac{\partial E}{\partial e^{u_j}} \frac{\partial e^{u_j}}{\partial w_{sj}^{(2)}}, \quad \frac{\partial e^{u_j}}{\partial w_{sj}^{(2)}} = v_s$$

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

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

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

На шаге $r + 1$ обучения сети производится коррекция весов согласно направлению антиградиента по следующим формулам:

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

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

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

Для гиперболического тангенса выражение принимает вид:

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

Программная реализация

Были реализованы следующие структуры и алгоритмы:

- **Neuron** – данный класс хранит набор синапсов и значение смещения. Инициализируется начальными весами.

- **NeuralNetwork** – класс имплементирующий нейронную сеть. Хранит множество нейронов по слоям.
- **LearningAlgorithm**– предназначен для тренировки, а так же анализа точности нейронной сети.

Сборка проекта осуществляется с помощью Visual Studio 2015.

Запуск производится следующим образом:

```
NeuralNetwork.exe [pathlearnImg] [pathtestImg][pathLearnLabs][pathTestLabs]
[numHiddenLayers][learnRate][learnError]
```

Описание аргументов командной строки приведено ниже:

- [pathlearnImg] – путь до обучающей изображений обучающей выборки;
- [pathtestImg] – путь до изображений тестовой выборки;
- [pathLearnLabs] – путь для меток обучающей выборки;
- [pathTestLabs] – путь для меток тестовой выборки;
- [numHiddenLayers] – количество скрытых слоев (по умолчанию один);
- [learnRate] – скорость обучения (по умолчанию 0,9);
- [learnError] – значение для критерия остановки обучения (по умолчанию 0,001).

Результаты экспериментов

Тестирование проводилось на наборе данных MNIST dataset <http://yann.lecun.com/exdb/mnist> – объёмной базе данных образцов рукописного написания цифр. База данных MNIST содержит 60 000 изображений для обучения и 10 000 изображений для тестирования. В каждой эпохе обучения используется вся обучающая выборка.

Начальные веса w инициализируются случайными значениями от -1 до 1. На вход программы подается массив двумерных массивов интенсивности (градации серого) для каждого пикселя изображения, а также массив меток, определяющий, к какой цифре относится каждое изображение.

Результаты тестирования на сетке с разным количеством нейронов на скрытом слое приведены в таблице 1.

Таблица 1. Зависимость результатов от количества нейронов на скрытом слое после 50 эпох обучения, скорость обучения 0.01

<i>Количество нейронов скрытого слоя</i>	<i>Точность на обучающей выборке, %</i>	<i>Кросс-энтропия на обучающей выборке</i>	<i>Точность на тестовой выборке, %</i>	<i>Кросс-энтропия на тестовой выборке</i>
10	94,26	0,190	92,65	0,27
20	97,55	0,07	95,28	0,17
30	98,71	0,041	95,9	0,158

Таблица 2. Зависимость результатов от скорости обучения после 50 эпох обучения, количество нейронов на скрытом слое 30

<i>Скорость обучения</i>	<i>Точность на обучающей выборке, %</i>	<i>Кросс-энтропия на обучающей выборке</i>	<i>Точность на тестовой выборке, %</i>	<i>Кросс-энтропия на тестовой выборке</i>
0,005	98.42	0.052	96.16	0.135
0,01	98,71	0,041	95,9	0,158
0,02	98.57	0.041	95.74	0.199

Согласно результатам экспериментов в таблице 1 можно увидеть, что при увеличении количества нейронов скрытого слоя с 10 до 20, точность на обучающей выборке увеличилась с 94,26 до 97,55, кросс-энтропия на обучающей выборке уменьшилась с 0,190 до 0,07, а точность на тестовой выборке увеличилась с 92,65 до 95,28, кросс-энтропия на тестовой выборке уменьшилась с 0,27 до 0,17. Согласно результатам экспериментов в таблице 2. При увеличении скорости обучения с 0,01 с на 0,02 точность на обучающей выборке уменьшается с 98,71 до 98, 57, а кросс – энтропия на обучающей выборке остается неизменной. Точность на тестовой выборке уменьшается с 95,9 до 95,74, а кросс энтропия на тестовой выборке увеличивается с 0,158 до 0,199.

Заключение

Изучена общая схема метода обратного распространения ошибки. Выведены математические формулы для вычисления градиентов по параметрам нейронной сети и формул коррекции весов. Разработана программная реализация двуслойной полносвязной нейронной сети и метода обратного распространения ошибки. Программная реализация протестирована на наборе данных MNIST.

Список литературы

1. Кустикова В.Д. Образовательный курс «Методы глубокого обучения для решения задач компьютерного зрения» Лекция №4 Сверточные нейронные сети. – Нижний Новгород, 2017.
2. Хайкин С. Нейронные сети. Полный курс. – М.: Издательский дом «Вильямс». – 2006.
3. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика. – 2002.