

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования

**«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)**

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе №1:
«Реализация метода обратного распространения ошибки для
двуслойной полностью связанной нейронной сети»**

Выполнил: студент группы 381603м4

Герасимов Дмитрий Михайлович

Проверила:

к.т.н., старший преподаватель
кафедры МОСТ
Кустикова В. Д.

Нижний Новгород

2017

Содержание

Постановка задачи.....	3
Вывод формул для метода обратного распространения ошибки.....	4
Алгоритм метода обратного распространения ошибки.....	8
Программная реализация.....	10
Результаты экспериментов.....	11
Литература.....	12

Постановка задачи

Выполнение лабораторной работы предполагает решение следующих задач:

1. Изучить общую схему метода обратного распространения ошибки;
2. Вывести формулы для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов;
3. Разработать программную реализацию;
4. Получить результаты для разработанной двухслойной полносвязной нейронной сети, которые соотносятся с результатами на официальной странице набора данных MNIST.

Вывод формул для метода обратного распространения ошибки

Пусть введены следующие обозначения:

N – количество входных нейронов;

M – количество выходных нейронов;

K – количество нейронов на скрытом слое;

L – количество обучающих примеров.

Функция кросс-энтропии рассматривается в качестве функции ошибки:

$$E(w) = -\frac{1}{L} \sum_{k=1}^L \sum_{j=1}^M y_j^k * \ln(u_j^k), \quad y_j^k = 1 \Leftrightarrow x_k \in j \text{ классу.}$$

Здесь $y^k = (y_j^k) \in Y, (j=1, \dots, M)$. В свою очередь Y – это множество обучающих примеров. $u^k = (u_j^k)$ – выход нейронной сети полученный для $x^k = (x_j^k) \in X, (i=1, \dots, N)$.

В последовательном режиме (также его еще называют стохастический) корректировка весов выполняется после каждого примера обучающей выборки. Пусть $x = (x_1, x_2, \dots, x_N)$, $y = (y_1, y_2, \dots, y_M)$ и $u = (u_1, u_2, \dots, u_M)$ – конкретный обучающий пример. Тогда получаем, что $E(w) = -\sum_{j=1}^M y_j * \ln(y_j)$.

Если обозначить веса синаптических связей от входных нейронов к нейронам скрытого слоя как $w_{is}^{(1)}$, а от нейронов скрытого слоя к выходным нейронам как $w_{sj}^{(2)}$, то f_s записывается в виде:

$$f_s = \sum_{i=0}^N w_{is}^{(1)} * x_i, \quad s=0, \dots, k \text{ – взвешенная сумма входных сигналов.}$$

Как функцию активации на последнем слое берут функцию *softmax* :

$$u_j = \sum_{m=1}^M \frac{e^{g_j}}{e^{g_m}} , \text{ где } g_j = \sum_{s=0}^K w_{sj}^{(2)} * v_s \quad (j=1, \dots, M) .$$

Таким образом $E(w)$ можно представить как:

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

где g_j вычисляется по формуле:

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

Задача оптимизации функции ошибки по всем синаптическим весам есть задача обучения нейронной сети:

$$E(w) \rightarrow \min_w .$$

В методе обратного распространения ошибки параметры сети выбираются с использованием градиентных методов оптимизации. Стоит заметить, что если функция активация на скрытом слое – это $\tanh(x)$, т.е. $\phi(f_s) = \tanh(f_s)$. Тогда градиент имеет следующую структуру:

$$\frac{\delta E(w)}{\delta w_{sj}^{(2)}} = \delta_j^{(2)} * v_s , \quad \frac{\delta E(w)}{\delta w_{is}^{(1)}} = \delta_s^{(1)} * x_i .$$

На каждом шаге $r+1$ обучения сети производится коррекция весов по формулам (η - скорость обучения):

$$w_{is}^{(1)(r+1)} = w_{is}^{(1)(r)} - \eta \frac{\delta E(w)}{\delta w_{is}^{(1)}} , \quad w_{sj}^{(2)(r+1)} = w_{sj}^{(2)(r)} - \eta \frac{\delta E(w)}{\delta w_{sj}^{(2)}} .$$

Производную целевой функции по параметрам скрытого слоя $w_{is}^{(1)}$ можно вычислить по формуле:

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

$$\frac{\delta E}{\delta f_s} = \sum_{j=1}^M \frac{\delta E}{\delta g_j} * \frac{\delta g_j}{\delta v_s} * \frac{\delta \phi}{\delta f_s} = \frac{\delta \phi}{\delta f_s} \sum_{j=1}^M \frac{\delta E}{\delta g_j} * \frac{\delta g_j}{\delta v_s} = \frac{\delta \phi}{\delta f_s} \sum_{j=1}^M \delta_j^{(2)} * w_{sj}^2 ,$$

то есть:

$$\frac{\delta E(w)}{\delta w_{is}^{(1)}} = \frac{\delta \phi}{\delta f_s} \left(\sum_{j=1}^M \delta_j^{(2)} * w_{sj}^2 \right) * x_i .$$

Производная целевой функции по параметрам последнего слоя $w_{sj}^{(2)}$ вычисляется по формуле:

$$\frac{\delta E(w)}{\delta w_{sj}^{(2)}} = \frac{\delta E}{\delta g_j} * \frac{\delta g_j}{\delta w_{sj}^{(2)}} , \text{ где } \frac{\delta g_j}{\delta w_{sj}^{(2)}} = v_s .$$

Тогда можно показать, что:

$$\begin{aligned} \delta_j^{(2)} &= \frac{\delta E}{\delta g_j} = -\frac{\delta}{\delta g_j} \left(\sum_{j=1}^M y_j * (g_j - \ln(\sum_{m=1}^M e^{g_j})) \right) = \\ &= -\left(y_1 * \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} - \dots - y_{j-1} * \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} + y_j * \left(1 - \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} \right) - y_{j+1} * \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} - \dots - y_M * \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} \right) = \\ &= \left(\sum_{j=1}^M y_j \right) * \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} - y_j . \end{aligned}$$

Воспользуемся тем фактом, что: $\sum_{j=1}^M y_j = 1$,

Тогда $\frac{\delta E}{\delta g_j}$ переписывается как:

$$\frac{\delta E}{\delta g_j} = \frac{e^{g_j}}{\sum_{m=1}^M e^{g_m}} - y_j = u_j - y_j \text{ .}$$

Таким образом исходное выражение $\frac{\delta E(w)}{\delta w_{sj}^{(2)}} = \frac{\delta E}{\delta g_j} * \frac{\delta g_j}{\delta w_{sj}^{(2)}}$ можно

записать в более упрощенном виде:

$$\frac{\delta E(w)}{\delta w_{sj}^{(2)}} = (u_j - y_j) * v_s = \delta_j^{(2)} * v_s \text{ .}$$

Алгоритм метода обратного распространения ошибки

Алгоритм обучения нейронной сети, описываемый в настоящей лабораторной работе, можно представить как последовательность следующих шагов:

1. Перед началом обучения веса нейронной сети w инициализируются случайным образом;
2. *for epoch* = 1,...,*numberOfEpochs* ;
3. *for i* = 1,..., $W * H$, где W - ширина изображения в пикселах и H - высота изображения в пикселах;
4. Следует прямой проход и обратный проход (алгоритмы описаны ниже);
5. Шаги 3-4 повторяются до тех пор пока не будет достигнута достаточная точность, то есть $E(w_{cur}) < CrossEntropyError$ или заданное количество итераций метода *numberOfEpochs* было достигнуто.

Прямой проход можно представить как:

$$x_i \rightarrow v_s, \quad \frac{\partial \phi}{\partial f_s} \rightarrow u_j$$

То есть вычисляются значения выходного сигналов нейронов скрытого слоя v_s , $s=0,\dots,K$ (где K – количество нейронов на скрытом слое) и значение производной функции активации на скрытом слое $\frac{\partial \phi}{\partial f_s}$. Затем вычисляются значения сигналов выходного поля u_j , $j=1,\dots,M$ (где M - количество классов изображений).

Обратный проход:

for $j=1,\dots,M$

$$\partial_j^{(2)} = u_j - y_j \quad ,$$

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

for $s=0,\dots,K$

$$\partial_s^{(1)} = -\delta\phi / \delta f \sum_{s,j=1}^M \delta_j^{(2)} * w_{sj}^2 \quad ,$$

$$\frac{\partial E(w)}{\partial w_{es}^{(1)}} = \delta_s^{(1)} * x_i \quad ,$$

for для каждой дуги:

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

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

Программная реализация

Исходный код программы был разработан на языке C++. Программный алгоритм метода обратного распространения ошибки был разработан в соответствии с теоретическим алгоритмом изложенным в главе “Алгоритм метода обратного распространения ошибки” настоящей лабораторной работы.

Исходный код программы можно найти по ссылке:
https://github.com/dimagerasimov/deep_learning.

Результаты экспериментов

Точность классификации объектов из набора данных MNIST была рассчитана по следующей формуле:

$$p = \frac{true}{true + false}$$

где *true* – количество правильно определённых цифр, *false* – количество неправильно определённых цифр.

Для получения сводной таблицы величина целевой функции была зафиксирована значением $7.0e-3$. Значения заданные для остальных параметров нейронной сети представлены в таблице ниже вместе с результатами экспериментов.

Количество нейронов в скрытом слое	Скорость обучения	Пройденное количество эпох до достижения заданной точности	Ошибка классификации на тренировочном наборе	Ошибка классификации на тестовом наборе
100	$5.0e-3$	21	$4.0e-4$	$2.18e-2$
100	$7.0e-3$	17	$3.33e-4$	$2.09e-2$
200	$5.0e-3$	19	$4.5e-4$	$1.91e-2$
200	$7.0e-3$	16	$5.5e-4$	$1.84e-2$
300	$4.0e-3$	22	$4.16e-4$	$1.91e-2$
300	$8.0e-3$	14	$8.8e-4$	$1.82e-2$

Литература

1. MNIST dataset [<http://yann.lecun.com/exdb/mnist>].
2. OpenCV [<http://opencv.org>].
3. Материалы Летней межвузовской школы 2016 [<https://github.com/itseez-academy/itseez-ss-2016-theory>], [<https://github.com/itseez-academy/itseez-ss-2016-practice>].