

Отчет по лабораторной работе №1

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

Выполнила:
студентка группы 381603м4
Решетова Анастасия

Постановка задачи

Цель настоящей работы состоит в том, чтобы изучить метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двуслойной полностью связанной сети (один скрытый слой).

Выполнение лабораторной работы предполагает решение следующих задач:

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. Проектирование и разработка программной реализации.
4. Тестирование разработанной программной реализации.

Метод обратного распространения ошибки

Многослойная нейронная сеть содержит нейроны, которые распределены по слоям. В общем случае сеть содержит входной, выходной и множество промежуточных (скрытых) слоев. В полностью связанной нейронной сети все узлы каждого слоя соединены с узлами следующего. Графическое представление полностью связанной нейронной сети:



Входной слой содержит $m \times n$ нейронов, что соответствует разрешению изображения.

Скрытый слой содержит S нейронов (подается на вход программы)

Выходной слой содержит k нейронов, что соответствует количеству классов.

В процессе обучения подбираются веса $w_{j,i}^r$ (вес ребра, соединяющего j -й нейрон слоя r и i -й нейрон слоя $(r+1)$). Задача обучения нейронной сети сводится к задаче оптимизации функции ошибки по всем синаптическим весам:

$$\min_w E(w)$$

Обучение нейронной сети будем проводить методом обратного распространения ошибки. Общая схема метода:

1. Прямой проход (от входного сигнала через скрытые слои к выходному слою).
Вычисляются значения выходных сигналов и соответствующие значения производных функций активации на каждом слое.

2. Вычисление значения целевой функции и ее градиента

3. Обратный проход. Корректировка синаптических весов:

$$w(k+1) = w(k) + \Delta w,$$

$$\Delta w = \eta p(w), \quad \eta, 0 < \eta < 1 - \text{скорость обучения, } p(w) = -\nabla E(w) - \text{направление}$$

4. Повторение 1-3 до выполнения критерия остановки.

В данной лабораторной работе рассматривается трехслойная полносвязная нейронная сеть (один скрытый слой).

В качестве функции активации на втором слое используется softmax:

$$\varphi(t_j) = \frac{e^{t_j}}{\sum_{j=1}^M e^{t_j}}, \quad \text{где } t_j = \sum_{i=0}^S \omega_{ji} v_i$$

В качестве функции ошибки используется кросс-энтропия:

$$E(w) = - \sum_{k=1}^L (y^k \ln u^k + (1 - y^k) \ln(1 - u^k))$$

Вычисление градиента функции ошибки.

Вычисление производной функции ошибки по параметрам последнего слоя:

$$\frac{\partial E}{\partial \omega_{sj}} = \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial \omega_{sj}}$$

$$\frac{\partial E}{\partial u_j} = - \left(y_j \frac{1}{u_j} + \frac{-1}{1 - u_j} + \frac{y_j}{1 - u_j} \right) = \frac{u_j - y_j}{u_j(1 - u_j)}$$

$$\frac{\partial u_j}{\partial \omega_{sj}} = \frac{\partial u_j}{\partial t_j} \frac{\partial t_j}{\partial \omega_{sj}} = \frac{\partial \frac{e^{t_j}}{\sum_{j=1}^M e^{t_j}}}{\partial t_j} v_s = \frac{e^{t_j} \sum_j e^{t_j} - (e^{t_j})^2}{(\sum_j e^{t_j})^2} v_s = \left(\frac{e^{t_j}}{\sum_j e^{t_j}} - \frac{(e^{t_j})^2}{(\sum_j e^{t_j})^2} \right) v_s = u_j(1 - u_j) v_s$$

$$\frac{\partial E}{\partial \omega_{sj}} = (u_j - y_j) v_s = \delta_j v_s$$

Вычисление производной функции ошибки по параметрам скрытого слоя:

$$\frac{\partial E}{\partial \omega_{is}} = \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial f_s} \frac{\partial f_s}{\partial \omega_{is}}, \quad \text{где } f_s = \sum_{i=0}^N \omega_{si} x_i$$

$$\frac{\partial E}{\partial \omega_{is}} = \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial f_s} x_i = u_j (1 - u_j) \left(\sum_s \delta_s \omega_{is} \right) x_i, \quad \text{где } \delta_s = u_s - y_s$$

Корректировка весов:

$$\omega_{ij}^{n+1} = \omega_{ij}^n - \eta \frac{\partial E}{\partial \omega_{ij}}$$

$$\omega_{ij}^{n+1} = \omega_{ij}^n - \eta \delta_j v_i, \quad \text{для нейронов выходного слоя,}$$

$$\omega_{ij}^{n+1} = \omega_{ij}^n - \eta u_j (1 - u_j) \left(\sum_s \delta_s \omega_{is} \right) x_i, \quad \text{для нейронов выходного слоя}$$

Программная реализация

Программа реализована на языке Python. В качестве параметров запуска задаются количество нейронов на скрытом слое, а скорость обучения (величина шага в направлении антиградиента при корректировке весов), а так же максимальное количество эпох.

Программа реализует алгоритм обратного распространения ошибки. Критерием остановки может явиться как достижение достаточной точности, так и превышение заданного количества итераций. На выходе программа выдает достигнутую точность на обучающей и на тестовой выборках, а также количество потребовавшихся для этого эпох.

Для вычисления точности использовалась следующая формула:

$$Accuracy = \frac{N_{true}}{N},$$

где N_{true} — количество правильно классифицированных изображений,

N — общее количество изображений

Результаты исследований

<i>Number of neuron (hidden)</i>	<i>Train speed</i>	<i>Number of epoch</i>	<i>Train accuracy</i>	<i>Test accuracy</i>
100	0,05	10	0,9902	0,9712
100	0,01	15	0,9902	0,9754
150	0,03	8	0,9915	0,9759
150	0,005	25	0,9905	0,9764
200	0,05	8	0,991	0,9759
200	0,02	10	0,9916	0,9751