

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет»
(национальный исследовательский университет)
Институт естественных и точных наук
Кафедра прикладной математики и программирования

Отчет по лабораторной работе №1
по дисциплине «Современные нейросетевые технологии»

Автор работы
Студент группы ЕТ-122
_____/Вилисов Е. А.
« ____ » _____ 2024 г.

Руководитель работы,
_____/Кириллов Е.В.
« ____ » _____ 2024 г.

Челябинск 2024

**Реализация метода обратного распространения ошибки для
двухслойной полностью связанной нейронной сети**

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

Задачи:

Выполнение практической 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.
5. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

В процессе выполнения лабораторной работы предполагается, что сеть ориентирована на решение задачи классификации одноканальных изображений. Типичным примером такой задачи является задача классификации рукописных цифр. Именно ее предлагается использовать в качестве тестовой задачи на примере набора данных MNIST [1].

Метод обратного распространения ошибки разрабатывается, исходя из следующих предположений:

1. На входе сети имеется $w \times h$ нейронов, что соответствует разрешению изображения.
 2. На выходе сети имеется k нейронов, что соответствует количеству классов изображений.
 3. Скрытый слой содержит s нейронов.
 4. В качестве функции активации на втором слое используется функция softmax
- .

Выполнение работы

- 1) Были изучены схема и математические формулы описывающие метод обратного распространения ошибки, функции активации, потерь и точности
- 2) Для реализации в качестве функции активации была выбрана сигмоида:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \Rightarrow e^{-x} = \frac{1 - \sigma(x)}{\sigma(x)}$$
$$\sigma'(x) = -\frac{(1 + e^{-x})'}{(1 + e^{-x})^2} = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{\frac{1 - \sigma(x)}{\sigma(x)}}{\frac{1}{\sigma^2(x)}} =$$
$$= \sigma^2(x) \frac{1 - \sigma(x)}{\sigma(x)} = \sigma(x)(1 - \sigma(x))$$

В качестве функции потерь была выбрана кросс энтропия

$$\text{logloss}(a, y) = -y \log a - (1 - y) \log(1 - a)$$

Где y – настоящие значения, a – предсказанные

Linear:

Pred = Weights * Input + Bias

Output = Sigmoid(Pred)

Primes = Sigmoid'(Pred)

LinearBackprop:

OutputDerivative = F' * Primes'

WeightsDerivative = OutputDerivative * Input^T

$$\text{BiasDerivative} = \begin{pmatrix} \sum_i \text{OutputDerivative}_{oi} \\ \sum_i \text{OutputDerivative}_{1i} \\ \dots \\ \sum_i \text{OutputDerivative}_{mi} \end{pmatrix}$$

$$\text{Error} = \text{Weights}^T * \text{OutputDerivative}$$

Где Weight – веса, Input – входные значения, Bias – смещение

F' – производная функции (cross entropy, linear)

m – первая размерность матрицы OutputDerivative

- 3) Была разработана программная реализация на языке python использованием библиотек numpy, pytorch, matplotlib и оформлен в виде блокнота ipynb, представленного в репозитории
- 4) Программная реализация была протестирована, получены следующие графики функции точности и потерь

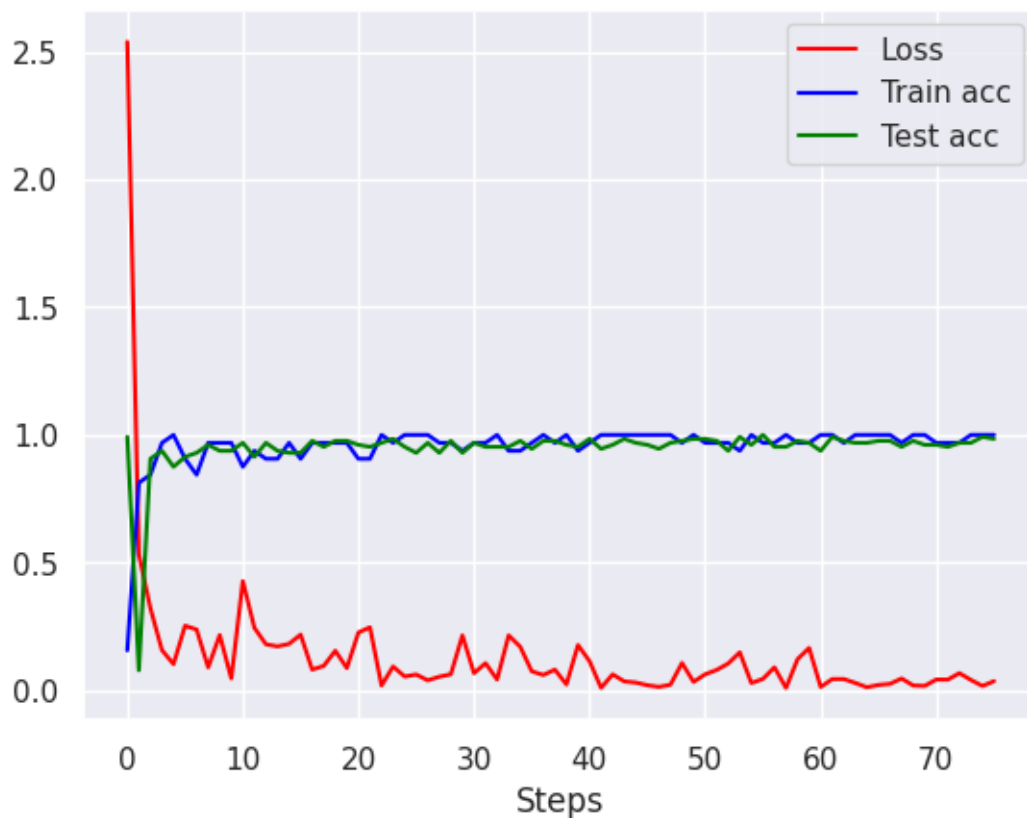


Рисунок 1 – графики loss и accuracy

Итоговая точность составила 0.982

- 5) По проведенной работе был составлен отчет