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**

Направление подготовки: «Прикладная математика и информатика»

Магистерская программа: «Вычислительные методы и суперкомпьютерные технологии»

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

**ОТЧЕТ**

по лабораторной работе №1

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

**Выполнила:**

студентка группы 381603м4

Морозова Юлия

Нижний Новгород

2018

Содержание

[Цели и задачи 3](#_Toc506456674)

[Метод обратного распространения ошибки 4](#_Toc506456675)

[Описание программной реализации 7](#_Toc506456676)

[Результаты 8](#_Toc506456677)

Цели и задачи

В данной лабораторной работе необходимо изучить метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двуслойной полностью связанной сети (один скрытый слой).

Основными задачами данной лабораторной работы являются:

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 на наборе данных MNIST.

Метод обратного распространения ошибки

Идея метода состоит в распространении сигналов ошибки от выходов к входам сети.

Пусть есть множество входов .

Обозначим:

- вес *i*-го и *j*-го ребра;

- выход j-го нейрона;

- ожидаемый результат.

Чтобы узнать, как текущий результат отличается от ожидаемого, будем использовать функцию ошибки кросс-энтропию:

Метод обратного распространения ошибки состоит из 2х частей. Перед началом обучения необходимо инициализировать веса и смещения малыми значениями. Далее начинается обучение сети.

***Прямой проход:***

1. Подаем на вход вектор входных сигналов .
2. Каждый нейрон входного слоя отправляет полученный сигнал всем нейронам скрытого слоя.
3. Каждый скрытый нейрон суммирует взвешенные входящие сигналы , а затем применяет активационную функцию *sigmoid*:
4. Каждый выходной нейрон суммирует взвешенные входящие сигналы , а затем применяет активационную функцию *softmax*:

***Обратный проход:***

1. Каждый выходной нейрон получает целевые значения и вычисляет ошибку:

Используем получившиеся значения:

1. Каждый скрытый нейрон суммирует входящие ошибки (от нейронов в последующем слое) и вычисляет величину ошибки, умножая полученное значение на производную активационной функции:

Где – поправка, вычисленная для к-го узла.

Таким образом, получаем следующую ошибку для нейронов скрытого слоя:

И ошибку для нейронов выходного слоя равную

1. Каждый выходной и скрытый нейрон будет изменять веса своих скрытых связей по формуле:

где , a – скорость обучения нейронной сети, – ошибка, – выход предыдущего слоя.

Описание программной реализации

Разработанная программа содержит следующие файлы:

* NeuralNetwork.h – заголовочный файл с описанием класса нейронной сети;
* NeuralNetwork.cpp – реализация методов для работы с нейронной сетью;
* DataReader.h – файл с методами для чтения данных;
* main.cpp – приложение для запуска сети. В нем происходит загрузка данных MNIST, создание нейронной сети и подача данных в созданную сеть для обучения.

Результаты

Требуемое значение кросс-энтропии для всех экспериментов было указано 0.005, скорость обучения 0.01, количество эпох 15. Результаты представлены в таблице ниже.

|  |  |  |
| --- | --- | --- |
| Число нейронов скрытого слоя | Точность на тестовом наборе | Точность на обучающем наборе |
| 50 | 0.9331 | 0.9637 |
| 100 | 0.9612 | 0.9845 |
| 300 | **0.9725** | 0.9952 |