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**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

Направление подготовки: «Фундаментальная информатика

и информационные технологии»

Магистерская программа: «Вычислительные методы и суперкомпьютерные технологии»

**Отчет по лабораторной работе**

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

Выполнил:

студент группы

381603м4

Кулдаев А.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2017

Содержание

[Постановка задачи 3](#_Toc506467609)

[Математическое объяснение метода. Постановка задачи оптимизации 4](#_Toc506467610)

[Обратное распространение 6](#_Toc506467611)

[Структура проекта 8](#_Toc506467612)

[Результаты 9](#_Toc506467613)

# Постановка задачи

Необходимо изучить и реализовать метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двуслойной полностью связанной сети (один скрытый слой).

В работе рассматривается сеть, ориентированная на решение задачи классификации рукописных цифр из набора данных MNIST.

Для выполнения лабораторной работы требуется решить следующие задачи:

1. Изучить общую схему метода обратного распространения ошибки.
2. Вывести математические формулы для вычисления градиентов функции ошибки по параметрам нейронной сети и формулы коррекции весов.
3. Разработать программную реализацию.
4. Протестировать разработанную программную реализацию.

Математическое объяснение метода. Постановка задачи оптимизации

Для обозначения количества нейронов в сети будут использоваться следующие обозначения:

* N – количество входных нейронов;
* K *–* количество нейронов на скрытом слое;
* M – количество выходных нейронов;
* L – количество обучающих примеров.

Для задачи классификации в качестве функции ошибки, как правило, выбирается кросс-энтропия, которая является дифференцируемой аппроксимацией функции ошибки классификации «0-1»:

где – множество обучающих примеров, а – выход нейронной сети, полученный для входа .

Следует отметить, что зачастую также вводится нормализация указанной метрики по числу примеров обучающей выборки **.**

Для вывода, а также в программной реализации будем использовать предположение, что у нас последовательный режим обучения. В этом режиме корректировка весов выполняется после прохода каждого примера обучающей выборки.

Возьмём конкретный обучающий пример

𝑥= (𝑥1, 𝑥2, …, 𝑥𝑁),

𝑦= (𝑦1, 𝑦2, …, 𝑦𝑀),

𝑢= (𝑢1, 𝑢2, …, 𝑢𝑀).

Тогда функция ошибки примет вид .

Введём следующие обозначения: - синаптические веса от входных нейронов к нейронам скрытого слоя, – от нейронов скрытого слоя к выходным нейронам. Выходной сигнал нейрона скрытого слоя вычисляется как , где – взвешенная сумма входных сигналов.

– функция активации на скрытом слое.

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

В итоге получаем:

где вычисляется по формуле:

Задача оптимизации функции ошибки по всем синапсическим весам есть

задача обучения нейронной сети:

.

# Обратное распространение

В ходе метода обратного распространения ошибки выполняется уточнение весов нейронной сети с целью уменьшения ошибки. В начальный момент веса инициализируются случайным образом. Далее для каждого примера обучающей выборки выполняется следующая процедура:

1. Прямой проход нейронной сети, на котором вычисляются выходы каждого слоя.
2. Вычисление значений целевой функции и градиента этой функции.
3. Обратный проход нейронной сети, при котором корректируются веса.
4. Повторение этапов 1-3 до момента выполнения критериев остановки (количество проходов или достигнутая точность).

Каждый выходной нейрон получает целевые значения и вычисляет ошибку:

Используем получившиеся значения:

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

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

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

# Структура проекта

Разработан проект в MS VS2013 который содержит следующие файлы:

* read\_mnist.h – функции для работы с набором данных MNIST
* neuron\_net.h – описание класса нейронной сети
* neuron\_net.cpp – реализация методов для работы с нейросетью
* main.cpp – тестовое приложение

# Результаты

Было разработано приложение позволяющее обучать и тестировать двухслойную нейронную сеть с использование набора данных MNIST.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кол-во нейронов на скр. слое | Скорость обучения | Кол-во эпох | Точность  классификации  на  тренировочном  наборе | Точность классификации на тестовом наборе |
| 100 | 0,008 | 16 | 0.999817 | 0.979200 |
| 200 | 0,005 | 20 | 0.999717 | 0.981000 |
| 100 | 0,005 | 23 | 0.999900 | 0.977100 |
| 200 | 0,01 | 11 | 0.999683 | 0.981400 |
| **200** | **0,008** | **14** | **0.999883** | **0.982100** |