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 №4

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

**Выполнил:**

студентка группы 381603м4

Горбунова Наталья

Нижний Новгород

2018

СОДЕРЖАНИЕ

[1 ПОСТАНОВКА ЗАДАЧИ 2](#_Toc501496762)

[1.1 Цель работы 2](#_Toc501496763)

[1.2 Задачи работы 2](#_Toc501496764)

[2 ОПИСАНИЕ МЕТОДА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ. ВЫВОД МАТЕМАТИЧЕСКИХ ФОРМУЛ 3](#_Toc501496765)

[2.1 Математическое объяснение метода. Постановка задачи оптимизации 3](#_Toc501496766)

[2.2 Обратное распространение 4](#_Toc501496767)

[3 АЛГОРИТМ МЕТОДА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ 7](#_Toc501496768)

[4 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ 9](#_Toc501496769)

[4.1 Структура проекта 9](#_Toc501496770)

[4.2 Руководство пользователя 9](#_Toc501496771)

[5 РЕЗУЛЬТАТЫ 10](#_Toc501496772)

# ПОСТАНОВКА ЗАДАЧИ

## Цель работы

Целью лабораторной работы является изучение и реализация метода обратного распространения ошибки для обучения глубоких нейронных сетей на примере двухслойной полностью связанной сети (один скрытый слой), используя набор данных MNIST.

## Задачи работы

В соответствии с поставленной целью, необходимо решение следующих задач:

1. Изучить общую схему метода обратного распространения ошибки.
2. Вывести необходимые математические формулы для вычисления градиентов функции ошибки по параметрам нейронной сети и формулу коррекции весов.
3. Спроектировать и разработать программную реализацию метода, который позволяет работать с набором данных MNIST
4. Произвести тестирование разработанной программной реализации.
5. Нахождение оптимальных параметров функционирования.

# ОПИСАНИЕ МЕТОДА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ. ВЫВОД МАТЕМАТИЧЕСКИХ ФОРМУЛ

## Математическое объяснение метода. Постановка задачи оптимизации

Введем следующие буквенные обозначения:

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

В качестве функции ошибки целесообразно рассмотреть кросс-энтропию:

, ,

где

– множество обучающих примеров,

– выход нейронной сети, полученный для входного примера.

.

Предположим, что режим обучения является последовательным. Тогда корректировка весов должна выполняться после прохода каждого примера обучающей выборки.

Рассмотрим пример:

,

,

.

В этом случае, а функция ошибки принимает следующий вид:

.

Введем следующие обозначения веса синаптических связей:

– от входных нейронов к нейронам скрытого слоя,

– от нейронов скрытого слоя к выходным нейронам нашей сети. Выходной сигнал нейрона скрытого слоя вычисляется следующим образом:

, где 𝜑 – функция активации на скрытом слое,

- взвешенная сумма входных сигналов.

Сигнал выходного нейрона можно определить как

, где ℎ - функция активации на последнем слое,

– взвешенная сумма сигналов со скрытого слоя.

В качестве функции активации на выходном слое рассмотрим функцию *softmax*:

Таким образом,

,

.

Глядя на полученную функцию ошибки, можно сказать, что задача обучения нейронной сети сводится к задаче оптимизации функции ошибки по всем весам сети

𝐸(𝑤) → 𝑚𝑖𝑛𝑤.

## Обратное распространение

Метод обратного распространения ошибки определяет способ проведения изменения параметров сети 𝑤.

Для этого можно использовать градиентные методы оптимизации. Производная целевой функции по параметрам последнего слоя вычисляется по следующей формуле:

,

,

В рассматриваемой задаче:

.

Таким образом:

.

Производная целевой функции по параметрам скрытого слоя вычисляется по формуле:

Таким образом:

.

В случае, если на скрытом слое функция активации является гиперболическим тангенсом: 𝜑(𝑓𝑠 ) = 𝑡ℎ(𝑓𝑠), то

Градиент может быть выражен следующим образом:

,

.

Согласно градиентным методам на каждом шаге 𝑟 + 1 обучения сети необходимо производить коррекцию весов следующим образом:

,

,

где 𝜂 – скорость обучения.

# АЛГОРИТМ МЕТОДА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

Алгоритм метода обратного распространения ошибки фактически состоит из шести шагов:

1. Инициализация весов 𝑤 некоторыми значениями
3. Прямой проход нейронной сети
4. Обратный проход
5. Шаги 3-5 повторяются до тех пока, пока не выполнится критерий остановки. Как правило, это либо максимальное число эпох либо достигнутая точность обучения.

Прямой проход.

На вход подается 𝑥𝑖. Необходимо вычислить значения выходных сигналов нейронов скрытого слоя - количество нейронов на скрытом слое и значение производной функции активации на скрытом слое .

Вычислить выходные сигналы нейронов последнего слоя

– количество классов изображений.

Коротко, его можно изобразить как: 𝑥𝑖 → 𝑣𝑠 ,

Обратный проход:

Вычислим значения градиентов целевой функции, начиная с конца:

*for*

,

Скрытый слой:

*for*

По дугам:

,

,

# ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

## Структура проекта

Проект был разработан в MS Visual Studio 2015 и содержиь следующие файлы:

* Mnist\_M.h – заголовочный файл, содержащий функции для работы с набором данных MNIST
* labN.h – заголовочный файл, содержащий описание класса нейронной сети
* labN.cpp – файл кода C++, который. содержит методы для работы с нейронной сетью
* Main.cpp – главный файл проекта

## Руководство пользователя

При запуске приложения пользователю доступна подсказка с необходимыми аргументами командной строки. Всего их восемь:

1. Path to MNIST train-images – обязательный параметр
2. Path to MNIST train-labels – обязательный параметр
3. Path to MNIST test-images – обязательный параметр
4. Path to MNIST train-labels – обязательный параметр
5. number hidden neuron – число нейронов скрытого слоя (по умолчанию = 300)
6. maxEpochs – число эпох для расчета (по умолчанию = 25)
7. learnRate – скорость обучения (по умолчанию = 0.008)
8. crossError – точность обучения для критерия остановки ( по умолчанию = 0.005)

# РЕЗУЛЬТАТЫ

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

Наилучшие результаты были достигнуты при следующих параметрах:

* Число нейронов скрытого слоя – 300 нейронов
* Число эпох – 25
* Скорость обучения - 0.008
* Точность на тестовой выборке – 0.9805
* Точность на тренировочной выборке - 0.999517