Нижегородский государственный университет им. Н. И. Лобачевского

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»

**Отчёт**

по лабораторной работе № 1

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

***Выполнила:***

студентка гр. 381703м3

Гладкова Т. А.

Нижний Новгород

2018

Оглавление

[Постановка задачи 3](#_Toc526796787)

[Описание метода обратного распространения ошибки. Вывод математических формул 4](#_Toc526796788)

[Математическое объяснение метода. Постановка задачи оптимизации. 4](#_Toc526796789)

[Обратное распространение ошибки 5](#_Toc526796790)

[Алгоритм метода обратного распространения ошибки 6](#_Toc526796791)

[Описание программной реализации 8](#_Toc526796792)

[Результаты 9](#_Toc526796793)

Постановка задачи

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.

В ходе лабораторной работы предполагается решение следующих задач:

* Изучение общей схемы метода обратного распространения ошибки.
* Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
* Проектирование и разработка программной реализации сети, решающей задачу классификации рукописных символов.
* Тестирование разработанной программной реализации.
* Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

Метод обратного распространения ошибки разрабатывается, исходя из следующих предположений:

* На входе сети имеется 𝑤 × ℎ нейронов, что соответствует разрешению изображения.
* На выходе сети имеется 𝑘 нейронов, что соответствует количеству классов изображений.
* Скрытый слой содержит 𝑠 нейронов.
* В качестве функции активации на втором слое используется функция softmax.
* В качестве функции ошибки используется кросс-энтропия.

# **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 ошибки

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

1. Инициализация весов 𝑤 некоторыми значениями
2. Прямой проход нейронной сети
3. Обратный проход
4. 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

* 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.

Обязательные параметры:

1. Path to MNIST train-images – набор данных
2. Path to MNIST train-labels – разметка (количество элементов),
3. Path to MNIST test-images – набор данных
4. Path to MNIST train-labels – разметка (количество элементов),
5. numberOfEpochs – число эпох для расчета (по умолчанию = 10)
6. crossError – точность обучения для критерия остановки (по умолчанию = 0.005)
7. learningRate – скорость обучения (по умолчанию = 0.01)
8. numberOfHiddenNeurons– число нейронов скрытого слоя.

Результаты

Общие параметры для всех экспериментов:

numberOfEpochs = 10 – количество эпох,

crossError = 0.005 – требуемое значение кросс-энтропии,

learningRate = 0.01 – скорость обучения.

Результаты представлены в таблице ниже.

|  |  |  |
| --- | --- | --- |
| Число нейронов скрытого слоя | Точность классификации на тестовом наборе | Точность классификации на обучающем наборе |
| 50 | 0.9211 | 0.9279 |
| 100 | 0.9443 | 0.9521 |
| 150 | 0.951 | 0.9607 |
| 200 | 0.9525 | 0.9636 |
| 300 | 0.9533 | 0.9674 |