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 двухслойной полностью связанной нейронной сети»**

**Выполнила:**

студентка гр.381703-3м

Крутобережская Ирина Сергеевна

Нижний Новгород

2018

Оглавление

Постановка задачи…………………………………………………………...3

Описание метода обратного распространения ошибки. Вывод математических формул.…………………………………………………………4

Обратное распространение ошибки………………………………………..6

Алгоритм метода обратного распространения ошибки…………………...8

Описание программной реализации………………………………………..9

Результаты…………………………………………………………………..10

Постановка задачи

Цель данной лабораторной работы – изучить и реализовать метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двухслойной полностью связанной сети (один скрытый слой), с использованием набора данных MNIST.

Процесс предполагает 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. Проектирование и разработка программной реализации сети.
4. Тестирование разработанной программной реализации.
5. Подготовить отчет, содержащий минимальный объем информации по каждому этапу выполнения работы.

Метод обратного распространения ошибки разрабатывается так:

1. На входе сети имеется 𝑤 × 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.
5. В качестве функции ошибки используется кросс-энтропия.

Описание метода обратного распространения ошибки. Вывод математических формул

Пусть

𝑁 – количество входных нейронов;

𝑀 – количество выходных нейронов;

𝐾 – количество нейронов на скрытом слое;

𝐿 – количество обучающих примеров.

В качестве функции ошибки рассмотрим 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:

,

,

.

Функция ошибки принимает следующий вид:

.

Введем следующие обозначения весов синоптических 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

* test\_data\_in.mat – содержит примеры для теста
* test\_data\_out.mat – ответы для примеров теста
* train\_data\_in.mat – содержит примеры для тренировки
* train\_data\_out.mat – содержит ответы для тренировки
* back\_path.m – вычисление изменения весов
* error\_norm.m – вычисляет норму ошибки
* forward\_path.m – прогон данных, вычисление ошибки
* iteration.m – форвард и бек
* main.m – главный файл
* new\_neural\_net.m – создается нейронная сеть с заданным количеством весов
* norm.m – вычисление нормы вектора
* rand\_example.m – переставляет обучающие примеры на каждой эпохе
* sigmoida.m – сигмоидальная функция
* softmax\_act.m – функция активации
* train.m – функция, где запускается тренировка сети
* work.m – функция для прямого прохода на тестовом множестве

С помощью стандартной функции пакета матлаб считали и сохранили в другом формате данные MNIST, в дальнейшем использовали четыре файла формата .mat. Это помогло ускорить загрузку данных.

Обязательные параметры:

1. epochs – число эпох для расчета (по умолчанию = 100)
2. min\_error – точность обучения для критерия остановки (по умолчанию = 0.0001)
3. speed– скорость обучения (по умолчанию = 0.1)
4. neurons – число нейронов скрытого слоя.
5. gradient =10^(-20) – норма вектора изменения весов

Результаты

Общие параметры для всех экспериментов:

min\_error = 0.01

speed= 0.1

Результаты представлены в таблице ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Число нейронов скрытого слоя | Число эпох | Точность классификации на тестовом наборе | Точность классификации на обучающем множестве | Время, с |
| 50 | 28 | 0.9669 | 0.9966 | 225 |
| 100 | 22 | 0.9752 | 0.9986 | 294 |
| 150 | 22 | 0.9778 | 0.9995 | 380 |
| 200 | 18 | 0.9786 | 0.9998 | 1920 |
|  |  |  |  |  |