

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: «Фундаментальная информатика и информационные
технологии»

Магистерская программа: «Компьютерная графика»

Образовательный курс «Глубокое обучение»

ОТЧЕТ

по лабораторной работе №1

Реализация метода обратного распространения ошибки для двухслойной полностью связанной нейронной сети

Выполнил:

студент группы 381706-2м
Привалов Даниил

Нижний Новгород
2018

Содержание

| | |
|--|---|
| Постановка задачи | 3 |
| Цели..... | 3 |
| Задачи..... | 3 |
| Общая последовательность действий..... | 3 |
| Теоретическое основание и реализация | 5 |
| Разработанные программы/скрипты | 6 |
| Результаты экспериментов | 7 |
| Литература..... | 8 |

Постановка задачи

Цели

Цель настоящей работы состоит в том, чтобы изучить метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двухслойной полностью связанной сети (один скрытый слой).

Задачи

Выполнение практической работы предполагает решение следующих задач:

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.
5. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

В процессе выполнения лабораторной работы предполагается, что сеть ориентирована на решение задачи классификации одноканальных изображений. Типичным примером такой задачи является задача классификации рукописных цифр. Именно ее предлагается использовать в качестве тестовой задачи на примере набора данных MNIST [1].

Метод обратного распространения ошибки разрабатывается, исходя из следующих предположений:

1. На входе сети имеется $w \times h$ 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.

Общая последовательность действий

Работа предусматривает выполнение следующей последовательности действий:

1. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
2. Подготовка пошагового описания метода обратного распространения ошибки с выводом всех математических формул для сети, описанной в разделе «Задачи». Примечание: для удобства может быть подготовлен псевдокод.
3. Проектирование и разработка программной реализации метода обратного распространения ошибки. Примечание: на этапе разработки имеет смысл активно использовать изученные в курсе «Инструменты программирования» инструменты (система контроля версий Git, Google Tests, CMake и другие).
4. Разработка тестового приложения для классификации рукописных цифр на примере базы MNIST. Для работы с матрицами и векторами можно воспользоваться типом данных Mat библиотеки OpenCV [2], а для чтения изображений – функцией imread указанной библиотеки. Более подробную информацию по работе с библиотекой можно получить из материалов Летней межвузовской школы 2016 [3]. Примечание:

- приложение должно обеспечивать обучение и тестирование сети, получая на вход пути до директорий с соответствующими выборками. Также в качестве параметров приложение должно принимать количество итераций, выполняемых в ходе обучения, и значение параметра скорости обучения.
5. Обучение сети на тренировочной выборке и тестирование на тестовой выборке набора данных MNIST. Измерение ошибки классификации и сбор результатов экспериментов при разном

Теоретическое основание и реализация

Многослойная нейронная сеть содержит нейроны, которые распределены по слоям. В простейшем случае в сети существует входной и выходной слои, и сеть называется однослойной. Если сигнал проходит от нейронов входного слоя к нейронам выходного, то такая сеть называется сетью прямого распространения. В общем случае сеть может содержать множество промежуточных слоев, называемых скрытыми слоями, и является многослойной.

Алгоритм обратного распространения ошибки является одним из методов обучения многослойных нейронных сетей прямого распространения, называемых также многослойными персептронами. Многослойные персептроны успешно применяются для решения многих сложных задач.

Обучение алгоритмом обратного распространения ошибки предполагает два прохода по всем слоям сети: **прямого** и **обратного**.

При прямом проходе входной вектор подается на входной слой нейронной сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синаптические веса сети фиксированы.

Во время обратного прохода все синаптические веса настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синаптических связей. Отсюда и название — алгоритм обратного распространения ошибки. Синаптические веса настраиваются с целью максимального приближения выходного сигнала сети к желаемому.

Алгоритм обратного распространения ошибки следующий:

1. Инициализировать синаптические веса маленькими случайными значениями.
2. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
3. Вычислить выход сети.
4. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
5. Подкорректировать веса сети для минимизации ошибки (как см. ниже).
6. Повторять шаги с 2 по 5 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Разработанные программы/скрипты

В репозитории `mnist_DL_lab.py` содержит:

1. Класс `Network` – для работы с нейронной сетью, инициализация, обучение, тестирование.
2. Функция `load_mnist_dataset` для загрузки данных, есть 3 вида данных:
 - a. Тренировочные,
 - b. Для тестирования,
 - c. Для валидации (во время тренировки)
3. Функция `createParser()` создает парсер аргументов, позволяющий задавать параметры сети без правки скрипта.

Также есть файлы с выводом для нескольких запусков сети: `results_good.txt`, `results.txt`, параметры тестируемых сетей описаны внутри файлов.

Результаты экспериментов

Параметры РС:
OS: Windows 10
CPU: Intel Core i5-5200U (2.2 GHz)
GPU: NVidia 920M (1 GB)
RAM: 4 GB

batch_size=1,
Количество эпох – 20,
Скорость обучения – 0.05.

| № | Количество скрытых слоев | Количество нейронов на слоях | Функции активации | Результат | |
|---|-----------------------------|------------------------------------|----------------------|-------------------------------------|--|
| | | | | Точность на тренировочном множестве | |
| 1 | 3 | 784-392-10 | sigmoid-sigmoid | 0.96 | |

batch_size=32,
Количество эпох – 20,
Скорость обучения – 0.1.

| № | Количество скрытых слоев | Количество нейронов на слоях | Функции активации | Результат | |
|---|-----------------------------|------------------------------------|----------------------|-------------------------------------|--|
| | | | | Точность на тренировочном множестве | |
| 1 | 3 | 784-392-10 | sigmoid-sigmoid | 0.33 | |

Литература

1. MNIST dataset [<http://yann.lecun.com/exdb/mnist>].
2. OpenCV [<http://opencv.org>].
3. Материалы Летней межвузовской школы 2016 [<https://github.com/itseez-academy/itseez-ss-2016theory>], [<https://github.com/itseez-academy/itseez-ss-2016-practice>].
4. Лекции по глубокому обучению:
<https://sites.google.com/site/kustikovaValentina/studentam/kursglubokoe-obucenie>,
2018.