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 программы: «Вычислительные методы и суперкомпьютерные технологии»

Образовательный курс «Глубокое обучение»

ОТЧЕТ

по лабораторной работе №1

Реализация метода обратного распространения ошибки для двуслойной полностью связанной нейронной сети

Выполнила:
студентка группы 381703-3м
Митрохина Юлия

Нижний Новгород
2018

Содержание

Цели 3
Постановка задачи 4
Описание метода обратного распространения ошибки. Вывод математических формул 5
Алгоритм метода обратного распространения ошибки 6
Описание программной реализации 8
Результаты 9
Выводы 10
Приложение 11

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).

Постановка 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 1.

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.

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.

Описание метода обратного распространения ошибки. Вывод математических формул

Описание метода обратного распространения ошибки и вывод математических формул приведены в Приложении.

Алгоритм метода обратного распространения ошибки

1. Инициализация весов w некоторыми значениями
2. $for\ epoch\ = \overline{1, maxEpochs}$
3. $for\ i\ = \overline{0, dataSize}$
4. Прямой проход нейронной сети
5. Обратный проход
6. Шаги 3-5 повторяются до тех пока, пока не выполнится критерий остановки – либо максимальное число эпох, либо достигнутая точность обучения.

Прямой проход.

На вход подаются $x_{i,}\ i\ = \overline{1, N}$, производится свертка, а затем с применением функции активации (в данной реализации используется *логистическая функция*) вычисляются значения выходных сигналов нейронов скрытого слоя v_j , $j\ = \overline{1, S}$, где S – количество нейронов на скрытом слое. Затем производится сверка выходных сигналов скрытого слоя и применяется функция активации *softmax*, чтобы вычислить выходные сигналы нейронов выходного слоя u_k , $k\ = \overline{1, K}$, где K – количество классов изображений.

Обратный проход:

Вычисляются значения градиентов функции ошибки кросс-энтропия, начиная с выходного:

for $k\ = \overline{1, K}$

$$\delta_k^{(2)} = u_k - y_k, \frac{\partial \mathrm{E(w)}}{w_{jk}^{(2)}} = \delta_k^{(2)} \cdot v_j$$

Скрытый слой:

for $j\ = \overline{1, S}$

$$\delta_j^{(1)} = -\frac{\partial \varphi_1}{\partial \mathrm{z_j}} \left[\sum_{k=1}^{K} \delta_k^{(2)} \cdot w_{jk}^{(2)}\right], \frac{\partial \mathrm{E(w)}}{\partial w_{ij}^{(1)}} = \delta_j^{(1)} \cdot x_i$$

И производится пересчет весов:

$$w_{ij}^{(1)}(r+1) = w_{ij}^{(1)}(\mathrm{r}) - \eta \frac{\partial \mathrm{E(w)}}{\partial w_{ij}^{(1)}},$$

$$w_{jk}^{(2)}(r+1) = w_{jk}^{(2)}(\mathrm{r}) - \eta \frac{\partial \mathrm{E(w)}}{w_{jk}^{(2)}}.$$

Качество решения выбранной задачи оценивается с использованием известной метрики – отношение числа верно классифицированных изображений к общему числу изображений в выборке:

$$\text{точность} = \frac{\text{верноклассифицированные}}{\text{всеизображения}}$$

Описание программной реализации

Структура проекта

- Network.h – заголовочный файл с описанием класса нейронной сети
- Network.cpp – реализация методов для работы с нейронной сетью
- ReadMNIST.h – файл с методами для чтения данных MNIST
- main.cpp – приложение для запуска сети. В нем происходит загрузка данных MNIST, создание нейронной сети и подача данных в созданную сеть для обучения и теста.

Основные параметры:

1. data_train – тренировочные данные
2. labels_train – разметка тренировочных данных
3. data_test – тестовые данные
4. labels_test – разметка тестовых данных
5. number_of_images_test – объем тестовой выборки (= 10000)
6. number_of_images_train – объем тренировочной выборки (= 60000)
7. image_size – размер картинок (28 * 28)
8. numberOfEpochs – число эпох для расчета (по умолчанию = 10)
9. crossError – точность обучения для критерия остановки (по умолчанию = 0.005)
10. learningRate – скорость обучения (по умолчанию = 0.01)
11. hiddenNeurons– число нейронов скрытого слоя
12. classNumber – количество классов / количество нейронов на выходном слое (в задаче распознавания цифр = 10)

Результаты

Достигнутая точность при проведении различных экспериментов представлена в таблице ниже.

Число нейронов скрытого слоя	Точность классификации на тестовой выборке	Точность классификации на обучающей выборке
60	0.8375	0.895
80	0.8969	0.9186
100	0.9189	0.927
120	0.9341	0.9455
140	0.9493	0.9542

Табл.1 Результаты экспериментов

Приведенные результаты показывают, что разработанная программа решает поставленную с задачу с достаточной точностью (в большинстве экспериментов > 90% точности).

Выводы

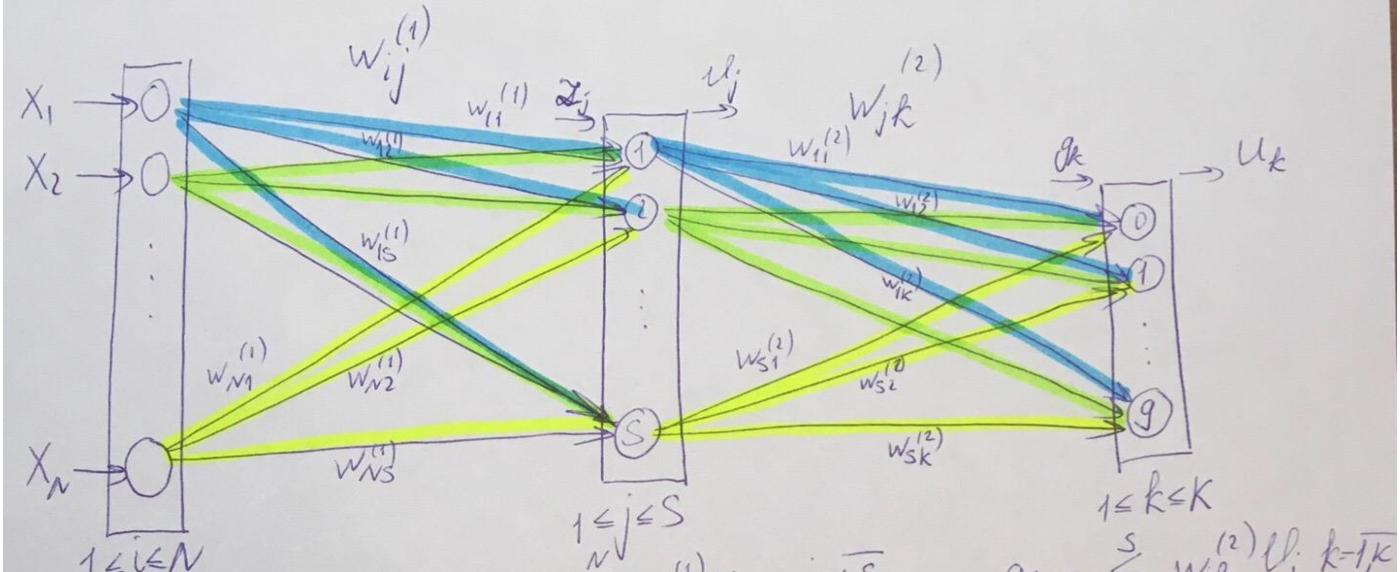
В ходе выполнения лабораторной работы была изучена общая схема метода обратного распространения ошибки для обучения глубоких нейронных сетей на примере двухслойной полностью связанной сети (один скрытый слой). Выведены математические формулы для вычисления градиентов функции ошибки по параметрам нейронной сети и формулы коррекции весов. Разработана программа на языке программирования C++ для решения задачи распознавания рукописных цифр на примере набора данных MNIST.

Приложение

Вход: $w \times h = N$ нейронов

Выход: K нейронов, $K = 10$

Скрытый слой: S нейронов



$$z_j = \sum_{i=1}^{N} w_{ij}^{(1)} x_i, \quad j = \overline{1,S}$$

$$v_j = \varphi_1(z_j)$$

$$\varphi_1 = \frac{1}{1+e^{-z_j}}$$ – логистическая ф-я активации

$$g_k = \sum_{j=1}^{S} w_{jk}^{(2)} v_j, \quad k = \overline{1,K}$$

$$u_k = \varphi_2(g_k)$$

softmax

$$\boxed{u_k = \frac{e^{g_k}}{\sum_{m=1}^{K} e^{g_m}}}$$

$x = (x_1, \dots, x_N)$ – вход

$u = (u_1, \dots, u_K)$ – полученный выход

$y = (y_1, \dots, y_K)$ – правильный выход

$\boxed{\text{Ф-я ошибки}}$

$$E(w) = -\sum_{k=1}^{K} y_k \cdot \ln(u_k)$$

$$E(w) = -\sum_{k=1}^{K} y_k \cdot \ln\left(\frac{e^{g_k}}{\sum_{m=1}^{K} e^{g_m}}\right) = -\sum_{k=1}^{K} y_k \left(g_k - \ln \sum_{m=1}^{K} e^{g_m}\right)$$

$$g_k = \sum_{j=1}^{S} w_{jk}^{(2)} \varphi_1\left(\sum_{i=1}^{N} w_{ij}^{(1)} \cdot x_i\right), \quad E(w) \to \min_w$$

Градиенты ф-и ошибки

$$\boxed{\frac{\partial E(w)}{\partial w_{jk}^{(2)}}} = \frac{\partial E}{\partial g_k} \cdot \frac{\partial g_k}{\partial w_{jk}^{(2)}} = \underbrace{\frac{\partial E}{\partial g_k}}_{\delta_k^{(2)}} \cdot v_j =$$

$$\delta_k^{(2)} = \frac{\partial E}{\partial g_k} = -\frac{\partial}{\partial g_k}\left(\sum_{k=1}^{K} y_k \left(g_k - \ln \sum_{m=1}^{K} e^{g_m}\right)\right) =$$

$$= -\frac{\partial}{\partial g_k}\left(y_1\left(g_1 - \ln \sum_{m=1}^{K} e^{g_m}\right) + y_2\left(g_2 - \ln \sum_{m=1}^{K} \cancel{g}\, e^{g_m}\right) + \right.$$

$$\left. + \ldots + y_k\left(g_k - \ln \sum_{m=1}^{K} e^{g_m}\right)\right) = -\left(-y_1 \cdot \frac{e^{g_k}}{\sum_{m=1}^{K} e^{g_m}} - \right.$$

$$\left. - y_2 \cdot \frac{e^{g_k}}{\sum_{m=1}^{K} e^{g_m}} + y_k\left(1 - \frac{e^{-g_k}}{\sum_{m=1}^{K} e^{g_m}}\right) - \ldots - y_K \cdot \frac{e^{g_K}}{\sum_{m=1}^{K} e^{g_m}}\right) =$$

$$= \underbrace{\left(\sum_{k=1}^{K} y_k\right)}_{=1} \cdot \frac{e^{g_k}}{\sum_{m=1}^{K} e^{g_m}} - y_k = u_k - y_k$$

$$= \boxed{(u_k - y_k) \cdot v_j}$$

$$\boxed{\frac{\partial E(w)}{\partial w_{ij}^{(1)}}} = \frac{\partial E}{\partial z_j} \cdot \overbrace{\frac{\partial z_j}{\partial w_{ij}^{(1)}}}^{x_i} =$$

$$\frac{\partial E}{\partial z_j} = \sum_{k=1}^{K} \underbrace{\frac{\partial E}{\partial g_k}}_{\delta_k^{(2)}} \cdot \frac{\partial g_k}{\partial v_j} \cdot \frac{\partial \varphi_1}{\partial z_j} = \frac{\partial \varphi_1}{\partial z_j} \sum_{k=1}^{K} \left(\delta_k^{(2)} \cdot w_{jk}^{(2)}\right)$$

$$= \boxed{\frac{\partial \varphi_1}{\partial z_j} \left(\sum_{k=1}^{K} \delta_k^{(2)} \cdot w_{jk}^{(2)}\right) \cdot x_i}$$

скорость обучения

Пересчет весов:

$$w_{ij}^{(1)}(r+1) = w_{ij}^{(1)}(r) - \eta \frac{\partial E(w)}{\partial w_{ij}^{(1)}}$$

$$w_{jk}^{(2)}(r+1) = w_{jk}^{(2)}(r) - \eta \frac{\partial E(w)}{\partial w_{jk}^{(2)}}$$