Лабораторная работа №7. Искусственные нейронные сети.

В данной работе вы ознакомитесь с работой искусственной нейронной сети и реализуете алгоритм прямого распространения для вычисления отклика сети на входной вектор значений. Созданные процедуры будут использованы для классификации рукописных цифр по изображениям из базы данных MNIST (https://ru.wikipedia.org/wiki/MNIST), которая уже использовалась в предыдущей лабораторной работе.

1. Загрузка и отображение данных

На этом шаге загружается файл ex3data1.npy, который содержит обучающий набор данных в виде матрицы *X* и вектора *y* в формате NumPy.

Как и в прошлой лабораторной работе, матрица *X* размером 5000×400 содержит 5000 изображений. Строка матрицы *X* длиной 400 элементов хранит изображение 20×20 (всего 400 пикселей) в следующем виде: сначала 20 чисел значений яркости первой строки изображение, затем второй и т.д. до последней строки. Таким образом вместо матрицы 20×20 один обучающий пример задается вектором из 400 параметров.

Вектор *y* длиной 5000 элементов для каждого обучающего примера X_i хранит индекс класса. Класс y_i=0 соответствует цифре «0», y_i=1 соответствует цифре «1» и т.д., y_i=9 соответствует цифре «9».

После загрузки выбирается случайных 100 изображений из обучающего набора и передаются в функцию display_data для отображения на экране. Функция принимает следующие параметры:

X – матрица элементов для отображения размером (width×height) строк и (el_width×el_height) столбцов;

width – сколько изображений разместить по горизонтали;

height – сколько изображений разместить по вертикали;

el_width – размер одного изображения по горизонтали;

el_height – размер одного изображения по вертикали.

Найдите функцию в программе и допишите ее содержимое, чтобы она отображала таблицу width×height изображений рукописных цифр на одной картинке.

Когда функция готова, запустите программу, вы должны увидеть изображение, похожее на рис. 1.

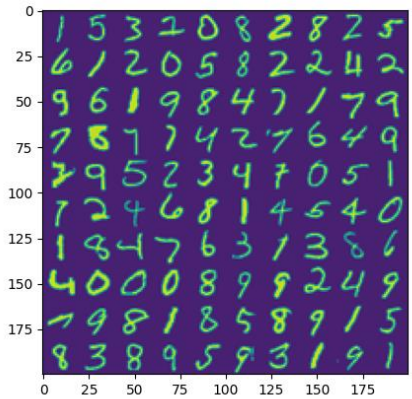

Рис. 1. Пример вывода функции display_data

Удалите оператор return после первого шага в функции main, чтобы перейти ко второму шагу.

2. Загрузка весов предобученной нейронной сети

В данной лабораторной работе вы не будете обучать нейронную сеть, а воспользуетесь готовыми весами уже обученной заранее сети. Для распознавания рукописных символов мы будем использовать нейронную сеть следующей архитектуры, показанной на рис. 2.

Сеть состоит из трех слоев. Входной слой $a^{(1)}$ составляют 400 входных параметров X (значения признаков каждого символа). Также неявно у нас присутствует входной параметр $a_0^{(1)}=1$. Второй скрытый слой $a^{(2)}$ составляют 25 нейронов. Третий выходной слой $a^{(3)}$ содержит 10 нейронов (по одному на каждый класс). В скрытом слое неявно присутствует нейрон $a_0^{(2)}=1$.

Десять выходных значений формируют выходной вектор длиной 10, элементы которого соответствуют десяти распознаваемым классам изображений рукописных символов. Первый нейрон (элемент вектора с индексом 0) выдает значение вероятности того, что на вход был подан символ ‘0’, второй нейрон – символ ‘1’ и т.д., десятый нейрон соответствует символу ‘9’. Мы будем относить распознаваемое изображение к тому классу, нейрон которого выдал наибольшее значение.

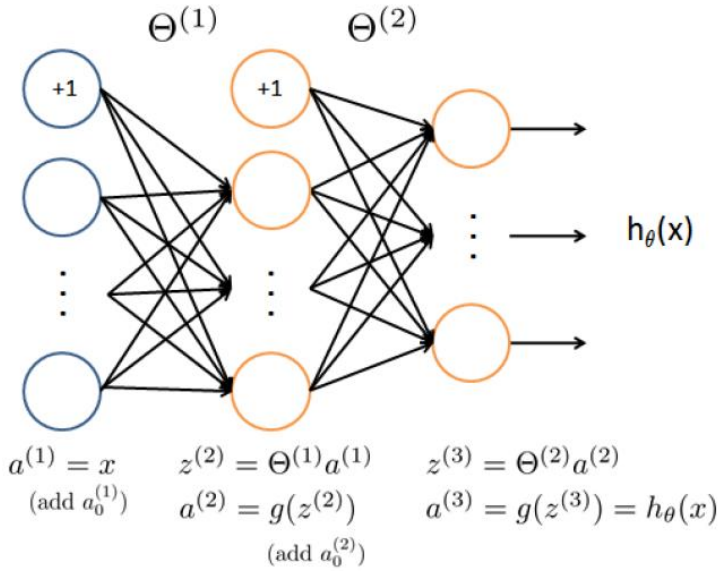

Рис. 2. Архитектура нейронной сети

На этом шаге выполняется загрузка весов обученной нейронной сети из файла ex3weights.npy. Из файла загружаются матрицы Theta1 и Theta2 параметров для вычисления второго и третьего слоя нейронной сети.

Поскольку второй слой имеет 25 нейронов и получает 400 входных параметров, матрица Theta1 имеет размер 25×401 (плюс параметр под один элемент +1). Третий слой имеет 10 нейронов и получает на вход выходы от 25 нейронов второго слоя, поэтому Theta2 имеет размер 10×26.

3. Алгоритм прямого распространения

На данном шаге нам необходимо запрограммировать алгоритм прямого распространения нейронной сети для того, чтобы распознать рукописный символ по его изображению. Найдите в программе функцию sigmoid и дополните необходимым кодом для вычисления логистической функции от вектора значений.

Далее найдите функцию predict и допишите в ней алгоритм прямого распространения для нейронной сети, представленной на рис. 2. Функция принимает на

вход следующие параметры:

Theta1 – матрица весов для вычисления второго слоя нейронной сети;

Theta2 – матрица весов для вычисления третьего слоя нейронной сети;

X – матрица элементов тестовой выборки.

Выдавать функция должна вектор p с ответами, к какому классу относится каждый элемент из X. Размер p равен числу строк в матрице X, а значение каждого k-го элемента p соответствует идентификатору класса X_k (целое число от 0 до 9).

Запрограммируйте функцию predict так, чтобы последовательно вычислить выходные значения нейронов второго слоя от входных значений, затем выходы нейронов выходного слоя от выходов второго слоя. После того, как получены все 10 значений нейронов выходного слоя, необходимо определить класс символа изображения, он равен индексу выходного нейрона с максимальным значением.

После того, как функция готова, запустите программу. В конце программы вычисляется точность на обучающем наборе. Если все было сделано без ошибок, точность должна получиться больше, чем для логистической регрессии при решении той же задачи в прошлой лабораторной работе.

4. Предсказание нейронной сетью случайных цифр

На последнем шаге программа использует написанный вами алгоритм прямого распространения для классификации случайных изображений из обучающей выборки. Нажмите в командном окне клавишу Enter, будет выбрано случайное изображение из набора данных, показано на экране и классифицировано. Результат классификации (к какому классу нейросеть относить символ) написан в командном окне. Остановить процесс можно вводом символа q и нажатием Enter.

На этом выполнение лабораторной работы №7 завершается.