МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра Вычислительной техники

ОТЧЁТ

по лабораторной работе №2

«Представление данных и библиотека NumPy»

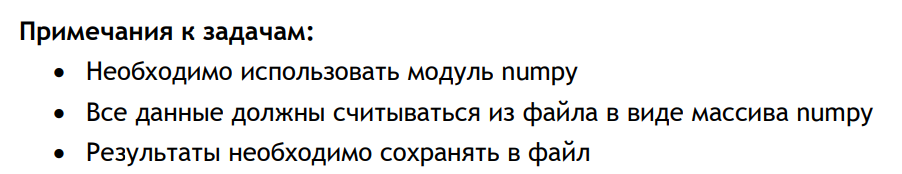
«Операции с тензорами в библиотеке Keras»

по дисциплине: «Машинное обучение»

|  |  |
| --- | --- |
| Выполнил:  Студент гр. АПИМ-25, АВТФ:  Клименко К. В. | Преподаватель:  Гаврилов А. В. |

Новосибирск, 2025

**Задание:** написать функцию, преобразовывающую вектор чисел в матрицу бинарных представлений.



**Ход работы:**

Рисунок 1. Полный листинг кода

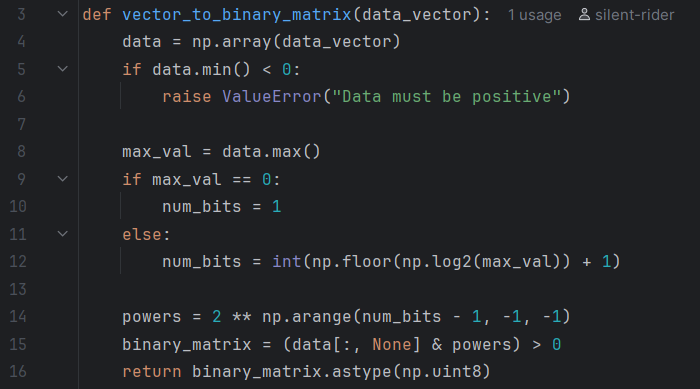


Для выполнения работы потребовалось создать функцию vector\_to\_binary\_matrix с входным параметром data\_vector типа ndarray и возвращаемым значением в виде матрицы бинарных представлений типа ndarray. Выполняется проверка на наличие отрицательных значений.

Затем определяется максимальное значение в векторе, на основе которого вычисляется необходимое количество битов для представления всех чисел. Если максимальное значение равно нулю, используется один бит; в противном случае количество битов рассчитывается как целая часть логарифма по основанию 2 от максимального значения, увеличенная на единицу.

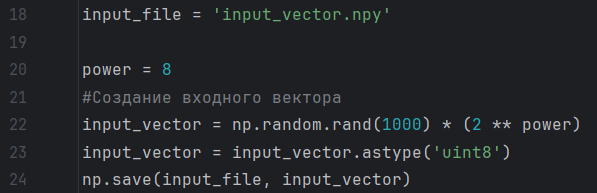
После этого создаётся массив степеней двойки в порядке убывания (от старшего бита к младшему). С помощью побитовой операции И (&) между каждым элементом исходного вектора и массивом степеней двойки формируется бинарная матрица: каждый элемент результирующей матрицы указывает, установлен ли соответствующий бит в двоичном представлении числа. Полученная логическая матрица преобразуется в целочисленный тип uint8, где True становится 1, а False — 0, и возвращается как результат.

Рисунок 2. Функция vector\_to\_binary\_matrix



Для проверки корректности работы функции был подготовлен тестовый входной файл input\_vector.npy. С этой целью сгенерирован вектор из 1000 случайных чисел, равномерно распределённых в диапазоне от 0 до 28 − 1, с последующим приведением к целочисленному беззнаковому типу uint8. Полученный вектор сохранён в файл с помощью функции np.save.

Рисунок 3. Создание входного вектора целых чисел



Из файла input\_vector.npy загружается входной вектор с помощью функции np.load. Для первичной проверки корректности загрузки на экран выводятся первые пять элементов вектора. Далее вызывается функция vector\_to\_binary\_matrix, которая преобразует исходный вектор неотрицательных целых чисел в двумерную матрицу их бинарных представлений. С целью визуального контроля корректности преобразования выводятся первые три строки результирующей матрицы. После этого полученная бинарная матрица сохраняется в файл output\_matrix.npу с использованием функции np.save, что полностью соответствует требованию задания о сохранении результатов в файл.

Рисунок 4. Загрузка и преобразование вектора

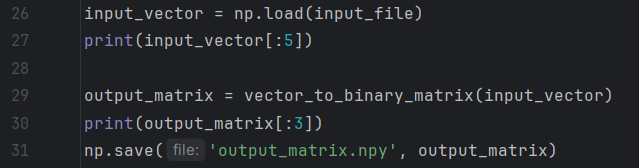
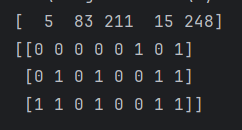


Рисунок 5. Результаты работы функции vector\_to\_binary\_matrix



Как видно на рисунке 5, результаты работы функции vector\_to\_binary\_matrix полностью соответствуют ожидаемым.

**Задание:** реализовать нейронную сеть, вычисляющую результат заданной логической операции. Затем реализовать функции, которые будут симулировать работу построенной модели. Функции должны принимать тензор входных данных и список весов.

Должно быть реализовано 2 функции:

1. Функция, в которой все операции реализованы как поэлементные

операции над тензорами

2. Функция, в которой все операции реализованы с использованием

операций над тензорами из NumPy

**Выводы**

В ходе выполнения лабораторной работы была изучена архитектура и реализована простая полносвязная нейронная сеть для решения задачи бинарной классификации. Была создана последовательная модель, состоящая из трёх полносвязных (Dense) слоёв: два скрытых слоя по 16 нейронов с активационной функцией ReLU и выходной слой с одним нейроном и сигмоидной активацией, обеспечивающей вероятностную интерпретацию результата. В качестве функции потерь использовалась бинарная кросс-энтропия, оптимизатор — RMSprop, а для оценки качества обучения отслеживалась метрика точности (accuracy).