FashionMIST

В данном задании вам предстоит реализовать инференс ML модели компьютерного зрения для классификации изображений предметов гардероба по типу. В качестве набора данных будем использовать Fashion MNIST. В данном наборе данных представлены изображения в разрешении 28x28 пикселей. Каждый пиксел кодируется одним байтом, который задает оттенок серого цвета (0 - черный, 255 - белый). Примеры изображений представлены на рисунке 1.

Рис 1. Примеры изображений из Fashion MNIST с указанием классов

Всего в наборе данных представлено 60,000 изображений, относящихся к 10 классам, которые закодированы следующим образом:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress

- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boot

Специалист отдела Data Science по имени СкайНет уже подготовил модели, обученные с помощью 4-х различных алгоритмов и оценил их метрику качества на проверочной выборке. В итоге получились следующие результаты:

- Логистическая регрессия - 0.85
- CatBoost - 0.87
- Многослойный перцептрон - 0.89
- Сверточная нейронная сеть на Tensorflow - 0.91

В архиве 12_homework.zip приложены следующие файлы

- test.csv - проверочная выборка в формате csv, для которой нужно вычислить точность модели. В каждой строке 785 целых чисел. 1-е число в строке - номер класса объекта. остальные числа - байты изображения
- logreg_coef.txt - коэффициенты логистической регрессии. Матрица вещественных чисел размером 10x785
- model.cbm - модель CatBoost
- w1.txt - коэффициенты 1-го слоя сети. Матрица вещественных чисел 784x128
- w2.txt - коэффициенты 2-го слоя сети. Матрица вещественных чисел 128x10
- saved_model/ - директория со сохраненной моделью Tensorflow
- test_data_logreg.txt - предсказания логистической регрессии для тестов
- test_data_catboost.txt - предсказания CatBoost для тестов
- test_data_mlp.txt - предсказания MLP для тестов
- test_data_cnn.txt - предсказания CNN для тестов
- FashionMNIST.ipynb - jupyter notebook, в котором код обучения моделей
- data - данные для обучения

Руководитель проекта просит вас самостоятельно выбрать, какую из моделей будет проще всего реализовать в коде на C++. При этом он намекает, что чем более качественная модель будет запущена в проде, тем больше премию получит отдел в этом году.

Задание

Решение должно представлять из себя исполняемый файл, который принимает в качестве параметра файл с тестовыми данными и файл с моделью. Тестовые данные представляют

собой текстовый файл в формате CSV без заголовка. В первом столбце - номер класса, в последующих 784 столбцах закодированы пиксели изображения (матрица 28x28 развернута в одномерный массив по строкам).

В результате исполнения программы должна выводиться метрика качества модели - доля правильных ответов (accuracy).

Пример запуска:
./fashio_mnist test.csv model
0.855

ВАЖНО! В данном задании достаточно реализовать инференс **любой одной** из предлагаемых моделей.

Указание к решению

Результаты обучения каждой модели сохранены в файл и приложены к заданию.

Логистическая регрессия

Поскольку стоит задача множественной классификации, то модель представляет собой 10 моделей логистической регрессии, каждая из которых обучена как 1 vs rest. В файле с моделью сохранена матрица 10 x 785, где каждая строка - отдельная модель, которая предсказывает вероятность своего класса, а столбцы - коэффициенты скалярного произведения (1-й элемент - свободный член).

Чтобы выбрать конкретный класс из всех 10-ти предсказаний нужно выбрать номер классификатора, который предсказывает класс с наибольшей вероятностью.

Подсказка 1: можно просто использовать 10 логистических регрессий из примера на вебинаре.

Подсказка 2: для улучшения производительности можно использовать матричные вычисления. Поскольку каждая из моделей логистической регрессии представляет собой скалярное произведение, то можно сразу рассчитать все 10 скалярных произведений, если представить их в виде **Ax**, где **A** - матрица коэффициентов всех моделей (10 x 785), а **x** (785 x 1)- расширенная матрица признаков (в качестве первого элемента добавлена 1).

CatBoost

Для хранения модели в виде файла используется собственный формат библиотеки CatBoost (.cbm). Для подключения данной библиотеки к проекту можно использовать следующие способы:

- готовый docker образ sdukshis/cppml, в котором уже собрана динамическая библиотека /usr/local/lib/libcatboost.so и ее заголовочные файлы лежат в /usr/local/include/catboost
- собрать библиотеку самостоятельно по инструкции от разработчиков

Для задачи классификации на несколько классов необходимо использовать функцию CalcModelPredictionSingle

Многослойный перцептрон

Данная модель содержит в себе 2 слоя нейронов. Коэффициенты этих слоев сохранены в файлах w1.txt (матрица 784x128) и w2.txt (128x10). Обратите внимание, что в файле матрицы хранятся в транспонированном виде. Перед умножением нужно будет поменять местами строки и столбцы (выполнить транспонирование). В качестве функции активации первого слоя используется сигмоида, а на втором слое softmax, чтобы привести выход модели к виду распределения вероятностей по классам.
В модели предполагается, что все входные данные x нормированы от 0 до 1.
Итого для вычисления предсказанного класса нужно выполнить следующие вычисления

$$z_1 = W_1 x$$
$$o_1 = \sigma(z)$$
$$z_2 = W_2 o_1$$
$$o_2 = softmax(z_2)$$

Ниже приведены определения функций:

$$\sigma(z) = 1/(1 + exp(-z))$$

$$softmax(z) = exp(z_i)/\sum_i exp(z_i)$$

Подсказка. Можно использовать готовую библиотеку для матричных вычислений, например eigen.

Сверточная нейронная сеть на Tensorflow

Модель сохранена в стандартном для библиотеки tensorflow представлении. Для загрузки модели и инференса следует использовать функции библиотеки tensorflow.

В модели предполагается, что все входные данные x нормированы от 0 до 1. В качестве примера можно переиспользовать код из вебинара. Для подключения библиотеки предлагаются следующие варианты:

- использовать готовый docker образ sdukshis/cppml, а котором установлена библиотека (/usr/local/lib/tensorflow*) и ее заголовочные файлы для C API (/usr/local/include/tensorflow)
- скачать собранную под вашу платформу библиотеку с C API по инструкции.