Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра информатики

Отчет по лабораторной работе #1

«Логистическая регрессия в качестве нейронной сети»

Карп Александр Игоревич

магистрант кафедры информатики

группа №858641

Минск 2019

**Задание 1.**

Загрузите данные и отобразите на экране несколько из изображений с помощью языка Python;

Добавление изображение будем совершать только в случае отсутствия дубликатов. Для этого будем хранить хэши уже загруженных изображений в сэте и совершать добавление нового только в случае отсутствия дубликатов.

def remove\_duplicates():  
 start = time.time()  
 hashes = set()  
 i = 0  
 copies = 0  
 for dir in dirs:  
 print(dir)  
 images = os.listdir(os.path.join(IMAGES\_PATH, dir))  
  
 for img in images:  
 path = os.path.join(IMAGES\_PATH, dir, img)  
 try:  
 image =Image.open(path)  
 except:  
 print('Bad image')  
 continue  
  
 hash = imagehash.phash(image)  
 if hash in hashes:  
 copies += 1  
 os.remove(path)  
 else:  
 hashes.add(hash)  
  
 i += 1  
 print('Time:', time.time() - start)  
 print('Count of duplciates:', copies)  
 print('Dataset length: ', len(X))

В выборку будем добавлять изображение в формате одномерного массива. Таким образом, изображение размерности 28х28 пикселей будет представлять собой массив размерности (784,1).

**Задание 2.**

Проверьте, что классы являются сбалансированными, т.е. количество изображений, принадлежащих каждому из классов, примерно одинаково (В данной задаче 10 классов).

В качестве количества значений элементов для каждого класса будем использовать минимальное число по всем классам

def get\_balance\_value(IMAGES\_PATH):  
 minimum = math.inf  
 dirs = os.listdir(IMAGES\_PATH)  
 for dir in dirs:  
 images = os.listdir(os.path.join(IMAGES\_PATH, dir))  
 if len(images) < minimum:  
 minimum = len(images)  
 return minimum

**Задание 3.**

Разделите данные на три подвыборки: обучающую (200 тыс. изображений), валидационную (10 тыс. изображений) и контрольную (тестовую) (19 тыс. изображений);

Для разделения будем использовать функцию из пакета sklearn train\_test\_split

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, y, train\_size=size, test\_size=10000)

**Задание 4.**

Проверьте, что данные из обучающей выборки не пересекаются с данными из валидационной и контрольной выборок. Другими словами, избавьтесь от дубликатов в обучающей выборке.

Данный пункт выполняется ввиду п1, так как только уникальные элементы добавляются в выборку.

**Задание 5.**

Постройте простейший классификатор (например, с помощью логистической регрессии). Постройте график зависимости точности классификатора от размера обучающей выборки (50, 100, 1000, 50000).

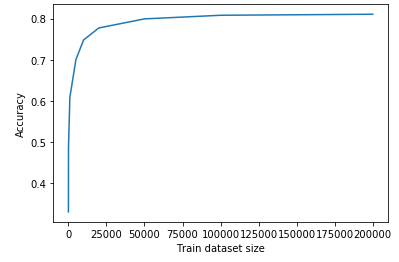
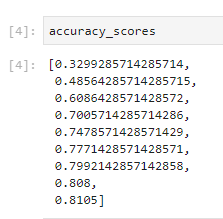


Рис. 1. График зависимости точности классификатора от размера тренировочной выборки

Из графика видно, что с увеличением количества элементов в тренировочном сете растет точность классификатора. Наибольшее достигнутое значение точности: 0.81



**Выводы**

Логистическую регрессию можно использовать для решения задач классификации изображений. Она проста в использовании, дает удовлетворительную точность. Однако есть решения, более приспособленные для работы с изображениями, дающие большую точность (например, CNN)