Министерство образования Республики Беларусь

Учреждение образования

"Белорусский Государственный университет информатики

и радиоэлектроники"

Лабораторная работа №1

“**Логистическая регрессия в качестве нейронной сети**”

по учебной дисциплине “Машинное обучение”

|  |  |
| --- | --- |
| Выполнил: | Студент гр. 956241 Дубовик Н.О. |
|  |  |

Минск 2020

**Данные:** В работе предлагается использовать набор данных notMNIST, который состоит из изображений размерностью 28×28 первых 10 букв латинского алфавита (A … J, соответственно). Обучающая выборка содержит порядка 500 тыс. изображений, а тестовая – около 19 тыс.

Данные можно скачать по ссылке:

* <https://commondatastorage.googleapis.com/books1000/notMNIST_large.tar.gz> (большой набор данных);
* [https://commondatastorage.googleapis.com/books1000/notMNIST\_small.tar.gz](https://commondatastorage.googleapis.com/books1000/notMNIST_large.tar.gz) (маленький набор данных);

Описание данных на английском языке доступно по ссылке:

<http://yaroslavvb.blogspot.sg/2011/09/notmnist-dataset.html>

Результат выполнения заданий опишите в отчете.

В ходе выполнения лабораторной работы был использован датасет notMNIST\_large

**Задание 1.**

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

Следующие изображения, из предоставленного датасета, были показаны с помощью библиотеки matplotlib.pyplot, рисунок 1.

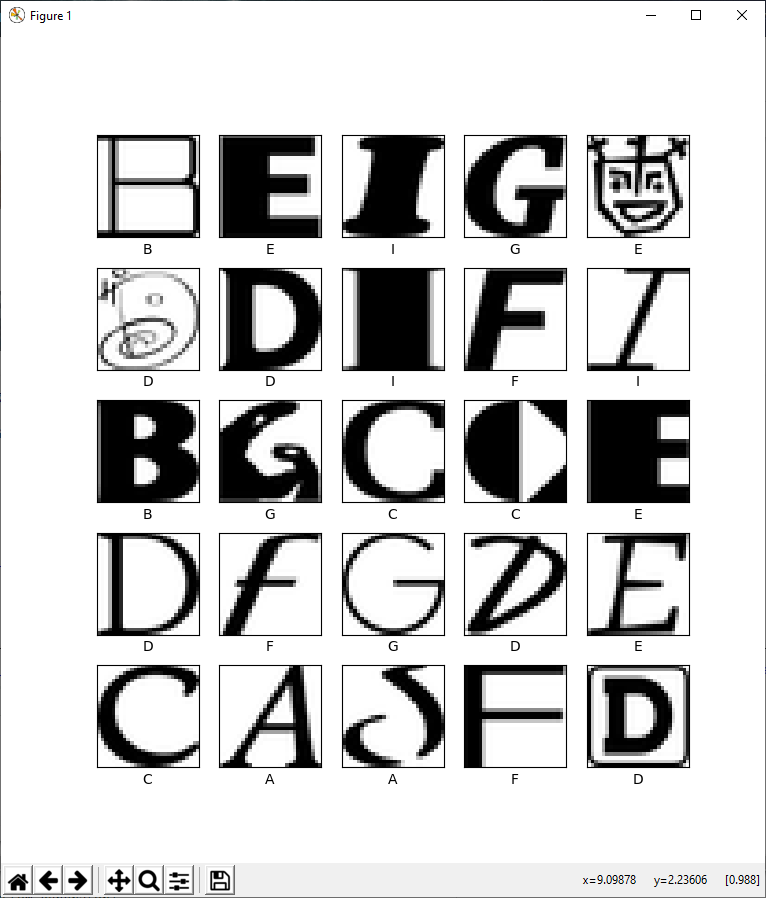


Рисунок 1 – Изображения из датасета

**Задание 2.**

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

Для этого задания была использована следующая функция:

def show\_percentages(Y,classes):

total=Y.shape[1]

for i in range(len(classes)):

count=np.count\_nonzero(Y==i)

print("{0} : {1:.2f}%".format(classes[i],count/total\*100))

И ее результат, рисунок 2.

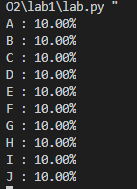
****

Рисунок 2 – Результат проверки на сбалансированность классов

**Задания 3/4.**

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

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

Для этих заданий использовался метод:

def split\_dataset(X,Y,train\_size, valid\_size,test\_size):

train\_index=train\_size

valid\_index=train\_index+valid\_size

test\_index=valid\_index+test\_size

p=np.random.permutation(X.shape[1])

X\_split=np.hsplit(X[:,p], [train\_index,valid\_index,test\_index])

Y\_split=np.hsplit(Y[:,p], [train\_index,valid\_index,test\_index])

return X\_split[0],X\_split[1],X\_split[2],Y\_split[0],Y\_split[1],Y\_split[2]

С аргументами:

split\_dataset(X,Y,200000,10000,19000)

**Задание 5.**

Постройте простейший классификатор (например, с помощью логистической регрессии). Постройте график зависимости точности классификатора от размера обучающей выборки (50, 100, 1000, 50000). Для построения классификатора можете использовать библиотеку SkLearn ([http://scikit-learn.org](http://scikit-learn.org/)).

Был использован классификатор OneVsRestClassifier из библиотеки sklearn.multiclass.

Сам метод выглядит следующим образом:

def train(X\_train,Y\_train,X\_valid,Y\_valid):

model=OneVsRestClassifier(LogisticRegression(solver="lbfgs",max\_iter=1000)).fit(X\_train.T,Y\_train.T)

print("train score: {}".format(model.score(X\_train.T,Y\_train.T)))

print("validation score: {}".format(model.score(X\_valid.T,Y\_valid.T)))

return model

Итоги его выполнения на выборках различного размера показаны на рисунке 3. По данному рисунку можно сделать вывод, что чем больше размер обучающей выборки, тем выше будет оценка.

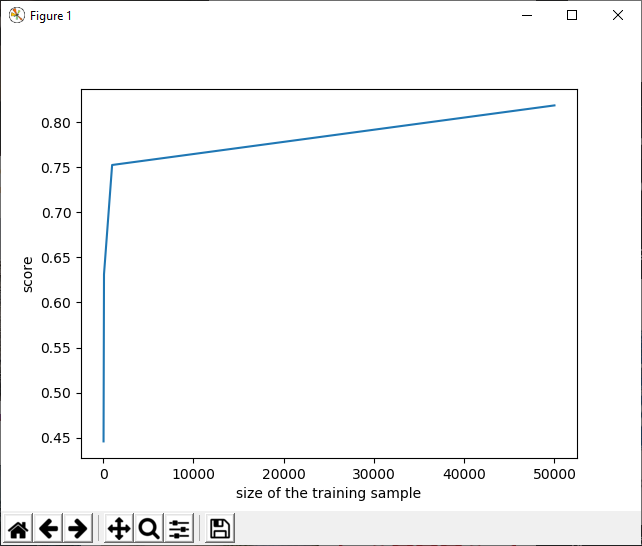


Рисунок 3 – Зависимость точности классификатора от размера обучающей выборки