**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5.

Курс «Введение в анализ данных и машинное обучение»

Отчет по лабораторной работе №5

«Распознавание рукописных цифр»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-42 |  | преподаватель каф. ИУ5 |
| Егор Хаметов |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2018 г.

**Задание:**

Необходимо решить задачу распознавания рукописных цифр (датасет MNIST). Задача решается в рамках платформы онлайн-конкурсов по машинному обучению Kaggle.

**Текст программы:**

import pandas as pd

import numpy as np

dtrain = pd.read\_csv('./train.csv')

dtest = pd.read\_csv('./test.csv')

from sklearn import cross\_validation

x\_train, x\_valid, y\_train, y\_valid = cross\_validation.train\_test\_split(

dtrain.drop(('label'), axis=1).astype('float32')/255.0, #X

dtrain['label'], #Y засовываем ответы без one hot кодинга, модель их и так съест

test\_size = 0.2,

random\_state = 11)

from sklearn.linear\_model import LogisticRegression

LRmodel = LogisticRegression(fit\_intercept=True) # параметр multi\_class по дефолту равен 'ovr' (один-против-всех), модель сама сделает one hot encoding для ответов 'label' (10 классов)

LRmodel.fit(x\_train,y\_train)

y\_valid\_predictions = LRmodel.predict(x\_valid)

from sklearn import metrics

metrics.accuracy\_score(y\_valid\_predictions, y\_valid)

import pickle

filename = 'LearnedLRmodel.sav'

pickle.dump(LRmodel, open(filename, 'wb')) # сохранили модель

import pandas as pd

import numpy as np

dtrain = pd.read\_csv('./train.csv')

dtest = pd.read\_csv('./test.csv')

from sklearn import cross\_validation

x\_train, x\_valid, y\_train, y\_valid = cross\_validation.train\_test\_split(

dtrain.drop(('label'), axis=1).astype('float32')/255.0,

dtrain['label'],

test\_size = 0.2,

random\_state = 11)

from keras.models import Sequential

from keras.layers import Dense

from keras import regularizers

model = Sequential()

model.add(Dense(800, input\_dim=x\_train.shape[1], activation="relu", kernel\_initializer="normal", kernel\_regularizer=regularizers.l2(0.01)))

model.add(Dense(600, activation="relu", kernel\_initializer="normal"))

model.add(Dense(10, activation="softmax", kernel\_initializer="normal"))

model.compile(loss='sparse\_categorical\_crossentropy',

optimizer='adam',

metrics=['accuracy'])

history = model.fit(x\_train, y\_train, validation\_data=(x\_valid, y\_valid), epochs=10, batch\_size=400)

predictions = model.predict\_classes(dtest, verbose=0)

submissions=pd.DataFrame({"ImageId": list(range(1,len(predictions)+1)), "Label": predictions})

submissions.to\_csv("predictions.csv", index=False, header=True)

import pickle

LRmodel = pickle.load(open('LearnedLRmodel.sav', 'rb'))

from sklearn import metrics

metrics.accuracy\_score(LRmodel.predict(x\_valid), y\_valid)

import matplotlib.pyplot as plt

%matplotlib inline

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'valid'], loc='upper left')

plt.show()