Министерство науки и высшего образованияРоссийской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра «Электронные вычислительные машины»

Отчет по практическому занятию №2

на тему

«Проектирование нейронных сетей с использованием библиотеки Keras»

по дисциплине  
**«Машинное обучение»**

Выполнили:

Студенты группы №140

Бригада 7

Сафаров Д. А.

Тимохин Е. М.

Проверила:  
ас. Панина И.С.

**Цель работы**

Изучить простейшие модели активационных функций, научиться прогнозировать результат на основе анализа зависимости величин.

**Задание**

Построить простейшую нейронную сеть, предназначенную для определения вида цветка ириса по его размерам в соответствии с параметрами:

* Ширина чашелистика;
* Длина лепестка;
* Ширина лепестка.

**Практическая часть**

Код программы представлен в приложении А. Результат её работы представлен на рисунке 1.



Рисунок 1 – Результат работы нейросети

Здесь 1 – ирис щетинистый, 2 – ирис разноцветный, 3 – ирис вергинский. На рисунке 2 представлен фрагмент исходных данных.

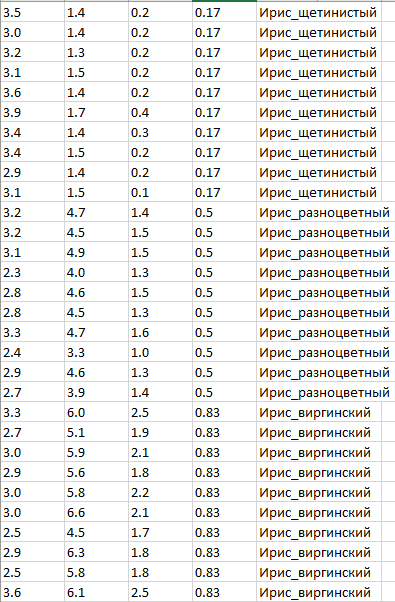


Рисунок 2 – Исходные данные

Приложение А. Текст программы

import os.path

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Activation

from array import array

data\_train = np.loadtxt('C:\Iris\_Train.txt', delimiter = ';',  
 dtype=np.float64)

X = data\_train[:,[0,1,2]]

Y = data\_train[:,[3]]

model = Sequential()

model.add(Dense(units = 8, input\_shape = (3,)))

model.add(Activation("sigmoid"))

model.add(Dense(units = 24))

model.add(Activation("sigmoid"))

model.add(Dense(units = 12))

model.add(Activation("sigmoid"))

model.add(Dense(1))

model.add(Activation("sigmoid"))

model.compile(optimizer = 'RMSprop', loss = 'binary\_crossentropy', metrics = ['accuracy'])

model.fit(X,Y, epochs = 1000, batch\_size = 80)

data\_test = np.loadtxt('C:\Iris\_Test.txt', delimiter = ';', dtype=np.float64)

res = model.predict(data\_test)

for var in res:

if (abs(var[0] - 0.17) <= 0.17): var[0] = 1

elif (abs(var[0] - 0.5) < 0.17): var[0] = 2

elif (abs(var[0] - 0.83) <= 0.17): var[0] = 3

print(res)