Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Кафедра вычислительной математики и механики

**Лабораторная работа № 2**

**по дисциплине: «Интеллектуальные ИСИТ»**

Выполнил

студент группы ИСТ-19-2б

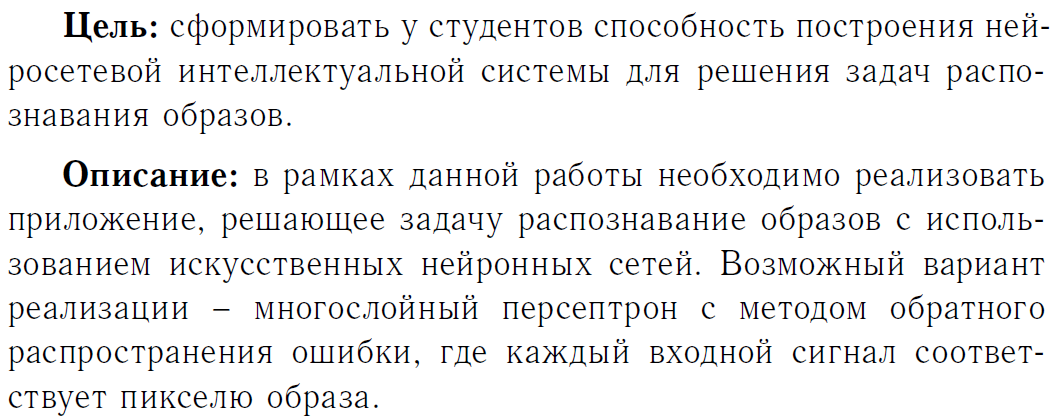
Лебедева А.А.

Проверил

ассистент кафедры ВММБ

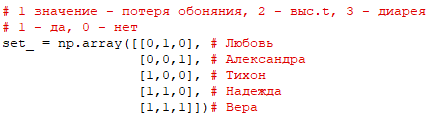
Нетбай Г.В.

Пермь, 2022



Допустим, что есть данные о людях и их симптомах covid-19.

Симптомы covid-19: потеря обоняния (первый показатель), высокая t (второй показатель), диарея (третий показатель)

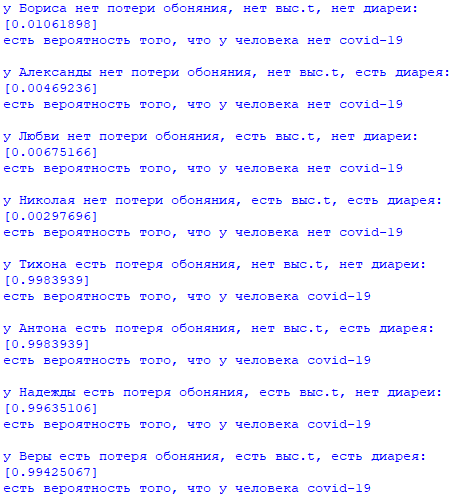


Кроме того, есть данные, болеют ли люди covid-19 (Любовь и Александра не болеют, Тихон, Надежда и Вера болеют).



Создадим нейронную сеть, чтобы узнать болеют ли другие люди covid-19, опираясь на симптомы.

По выполнению программы получаем результаты:



Если число приближенно к нулю, то результат – вероятность, что у человека covid-19 низка, если число приближенно к единице, то есть высокая вероятность, что человек болеет covid-19.

Визуальное представление обработки данных:



Где 3-5 столбец симптомы, которые приняты за входной сигнал, выходным сигналом является столбец с результатом.

import numpy as np

from tkinter import \*

class Table:

def \_\_init\_\_(self,root):

for i in range(t\_rows):

for j in range(t\_columns):

self.e = Entry(root, width = 15, fg = '#800080', font = ('Times',13,'bold'))

self.e.grid(row = i, column = j)

self.e.insert(END, lst[i][j])

def s(x):

return 1/(1 + np.exp(-x))

def s\_proiz(x):

return s(x)\*(1-s(x))

def res(a):

minim = float("inf")

list\_ = [0.0,1.0]

val = a

for i in list\_:

if abs(i - val) < minim:

final\_val = i

minim = abs(i - val)

#print(final\_val)

if final\_val == list\_[0]:

print("есть вероятность того, что у человека нет covid-19")

return final\_val

else:

print("есть вероятность того, что у человека covid-19")

return final\_val

# 1 значение - потеря обоняния, 2 - выс.t, 3 - диарея

# 1 - да, 0 - нет

set\_ = np.array([[0,1,0], # Любовь

[0,0,1], # Александра

[1,0,0], # Тихон

[1,1,0], # Надежда

[1,1,1]])# Вера

# есть ли у людей covid-19(1 - да, 0 - нет)

lab = np.array([[0,0,1,1,1]])

lab = lab.reshape(5,1)

np.random.seed(42)

# вектор веса

w = np.random.rand(3,1)

# смещение

b = np.random.rand(1)

# скорость обучения

n = 0.05

for i in range(21000):

inp = set\_

wx = np.dot(set\_, w) + b

z = s(wx)

err = z - lab

u = err

dz = s\_proiz(z)

delta = u \* dz

inp = set\_.T

w -= n \* np.dot(inp, delta)

for j in delta:

b -= n \* j

cov = []

# у Бориса нет потери обоняния, нет выс.t, нет диареи

set1 = np.array([0,0,0])

res1 = s(np.dot(set1, w) + b)

print("у Бориса нет потери обоняния, нет выс.t, нет диареи:")

print(res1)

cov.append(res(res1))

print()

# у Александы нет потери обоняния, нет выс.t, есть диарея

set2 = np.array([0,0,1])

res2 = s(np.dot(set2, w) + b)

print("у Александы нет потери обоняния, нет выс.t, есть диарея:")

print(res2)

cov.append(res(res2))

print()

# у Любви нет потери обоняния, есть выс.t, нет диареи

set3 = np.array([0,1,0])

res3 = s(np.dot(set3, w) + b)

print("у Любви нет потери обоняния, есть выс.t, нет диареи:")

print(res3)

cov.append(res(res3))

print()

# у Николая нет потери обоняния, есть выс.t, есть диарея

set4 = np.array([0,1,1])

res4 = s(np.dot(set4, w) + b)

print("у Николая нет потери обоняния, есть выс.t, есть диарея:")

print(res4)

cov.append(res(res4))

print()

# у Тихона есть потеря обоняния, нет выс.t, нет диареи

set5 = np.array([1,0,0])

res5 = s(np.dot(set5, w) + b)

print("у Тихона есть потеря обоняния, нет выс.t, нет диареи:")

print(res5)

cov.append(res(res5))

print()

# у Антона есть потеря обоняния, нет выс.t, есть диарея

set6 = np.array([1,0,1])

res6 = s(np.dot(set6, w) + b)

print("у Антона есть потеря обоняния, нет выс.t, есть диарея:")

print(res5)

cov.append(res(res5))

print()

# у Надежды есть потеря обоняния, есть выс.t, нет диареи

set7 = np.array([1,1,0])

res7 = s(np.dot(set7, w) + b)

print("у Надежды есть потеря обоняния, есть выс.t, нет диареи:")

print(res6)

cov.append(res(res6))

print()

# у Веры есть потеря обоняния, есть выс.t, есть диарея

set8 = np.array([1,1,1])

res8 = s(np.dot(set8, w) + b)

print("у Веры есть потеря обоняния, есть выс.t, есть диарея:")

print(res8)

cov.append(res(res8))

print()

lst = [('№','Имя','потеря обоняния','высокая t°C','диарея','результат','covid-19'),

(1,'Борис',set1[0], set1[1], set1[2], res1, cov[0]),

(2,'Александра',set2[0], set2[1], set2[2], res2, cov[1]),

(3,'Любовь',set3[0], set3[1], set3[2], res3, cov[2]),

(4,'Николай',set4[0], set4[1], set4[2], res4, cov[3]),

(5,'Тихон',set5[0], set5[1], set5[2], res5, cov[4]),

(6,'Антон',set6[0], set6[1], set6[2], res6, cov[5]),

(7,'Надежда',set7[0], set7[1], set7[2], res7, cov[6]),

(8,'Вера',set8[0], set8[1], set8[2], res8, cov[7])]

t\_rows = len(lst)

t\_columns = len(lst[0])

root = Tk()

root.title("Results")

t = Table(root)

root.mainloop()