**Лущик Никита КС-44**

**Сеть на основе радиально-симметричной функций**

**Листинг программы:**

**import** **numpy** **as** **np**

**from** **math** **import** sqrt

**from** **Common** **import** E

**def** **Gauss**(InputArray, vecC, R): # Non-iterative algoritm

Alfa = **1** / (**2**\*R\*\***2**)

X = []

Y = []

h = []

W = []

**for** i **in** InputArray:

X.append(i[**0**])

Y.append(i[**1**])

Y = np.array(Y, float)

W = np.array(W, float)

**print**("X:", X)

**print**("vecC:", vecC)

H = []

**for** i **in** range(len(vecC)):

**for** j **in** range(len(X)):

h.append(E\*\*(-Alfa\*sqrt((X[j]-vecC[i])\*\***2**)\*\***2**))

H.append(h.copy())

h.clear()

H = np.array(H, float)

H = H.T

**print**("H:**\n**", H)

HT = H.T

xt = **1**

W = np.dot(np.dot(np.linalg.inv(np.dot(HT, H)), HT), Y.T)

**print**("W:", W)

**def** **check**():

Ycalc = []

tmp = **0**

**for** j **in** range(len(X)):

**for** i **in** range(len(vecC)):

tmp += E\*\*(-Alfa\*sqrt((X[j]-vecC[i])\*\***2**)\*\***2**)\*W[i]

Ycalc.append(round(tmp,**2**))

tmp = **0**

Ycalc = np.array(Ycalc, float)

Yerr = abs(Y - Ycalc)

**print**("Y:", Y)

**print**("Y calc:", Ycalc)

**print**("Y errors:", Yerr)

**print**("Total error:", round(sum(Yerr)/len(Yerr)\***100**, **2**), "%")

check()

**def** **main**():

InputArray = [[-**2.0**, -**0.48**],

[-**1.5**, -**0.78**],

[-**1.0**, -**0.83**],

[-**0.5**, -**0.67**],

[**0.0**, -**0.20**],

[**0.5**, **0.70**],

[**1.0**, **1.48**],

[**1.5**, **1.17**],

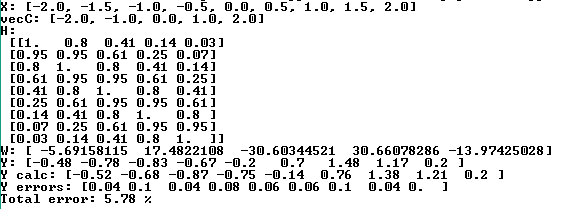
[**2.0**, **0.20**]]

vecC = [-**2.0**, -**1.0**, **0.0**, **1.0**, **2.0**] # Vector of centers

Gauss(InputArray, vecC, **1.5**)

main()

**Вывод программы:**



**Вывод:**

При использованных настройках нейронной сети средняя относительная ошибка аппроксимации, рассчитанная для экспериментальных точек, составила 5,78%. Повысить точность решения задачи аппроксимации с использованием сети на основе радиально-симметричных функций можно за счет добавления новых радиальных элементов или изменения настроек имеющихся скрытых нейронов.