Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 1 семестр

По дисциплине: «МиАПР»

Тема: «Нелинейные ИНС в задачах прогнозирования»

Выполнил:

Студент 2 курса

Группы ПО-4(2)

Яковчик И.А

Проверил:

Крощенко А.А.

2020

**Лабораторная работа №4**

Нелинейные ИНС в задачах прогнозирования

Цель работы: Изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов

**Вариант 10**

**Код программы:**

import numpy as np

import sys

import math

def func(x):

a = 0.2

b = 0.4

c = 0.09

d = 0.4

return a \* math.cos(b \* x) + c \* math.sin(d \* x)

class Network:

def \_\_init\_\_(self, learning\_rate = 0.25):

self.weights\_0\_1 = np.random.normal(0.0, 2 \*\* -0.5, (2, 6))

self.weights\_1\_2 = np.random.normal(0.0, 1, (1, 2))

self.tanh\_mapper = np.vectorize(self.tanh)

self.learning\_rate = np.array([learning\_rate])

def tanh(self, x):

return np.tanh(x)

def predict(self, inputs):

inputs\_1 = np.dot(self.weights\_0\_1, inputs)

outputs\_1 = self.tanh\_mapper(inputs\_1)

inputs\_2 = np.dot(self.weights\_1\_2, outputs\_1)

outputs\_2 = self.tanh(inputs\_2)

return outputs\_2

def train(self, inputs, expected\_predict):

inputs\_1 = np.dot(self.weights\_0\_1, inputs)

outputs\_1 = self.tanh\_mapper(inputs\_1)

inputs\_2 = np.dot(self.weights\_1\_2, outputs\_1)

outputs\_2 = self.tanh(inputs\_2)

actual\_predict = outputs\_2[0]

error\_layer\_2 = np.array([actual\_predict - expected\_predict])

gradient\_layer\_2 = -(self.tanh(actual\_predict) \*\* 2) - 1

weights\_delta\_layer\_2 = error\_layer\_2 \* gradient\_layer\_2

self.weights\_1\_2 -= (np.dot(weights\_delta\_layer\_2, outputs\_1.reshape(1, len(outputs\_1)))) \* self.learning\_rate

# Адаптивный шаг

self.learning\_rate = (np.sum(np.square(error\_layer\_2) \* (np.ones((1, 2)) - np.square(outputs\_2))))/((1 + np.sum(np.square(outputs\_1)))\*(np.sum(np.square(error\_layer\_2)\*np.square(np.ones((1, 2)) - np.square(outputs\_2)))))

error\_layer\_1 = weights\_delta\_layer\_2 \* self.weights\_1\_2

gradient\_layer\_1 = -(self.tanh(outputs\_1) \*\* 2) - 1

weights\_delta\_layer\_1 = error\_layer\_1 \* gradient\_layer\_1

self.weights\_0\_1 -= np.dot(inputs.reshape(len(inputs), 1), weights\_delta\_layer\_1).T \* self.learning\_rate

def MSE(y, Y):

return np.mean((y - Y) \*\* 2)

step = 0.1

counter = 0

train = []

for i in range(-15, 15):

combol = []

inputs = []

for j in range(6):

x = counter \* step

inputs.append(func(x))

counter += 1

combol.append(inputs)

x = counter \* step

combol.append(func(x))

combo = tuple(combol)

train.append(combo)

learning\_rate = 0.05

network = Network(learning\_rate)

losses = {'train':[], 'validation':[]}

Emin = 1e-5

train\_loss = 0

epoch = 0

while True:

inputs = []

correct\_predictions = []

for input\_stat, correct\_predict in train:

network.train(np.array(input\_stat), correct\_predict)

inputs.append(np.array(input\_stat))

correct\_predictions.append(np.array(correct\_predict))

train\_loss = MSE(network.predict(np.array(inputs).T), np.array(correct\_predictions))

epoch += 1

if train\_loss <= Emin:

break

print("epochs: {}, training loss: {}".format(

str(epoch),

str(train\_loss)

))

print("\nРЕЗУЛЬТАТЫ ОБУЧЕНИЯ:")

for input\_stat, correct\_predict in train:

print("the prediction is: {}, expected: {}, mistake: {}".format(

str(network.predict(input\_stat)),

str(correct\_predict),

str(network.predict(input\_stat) - correct\_predict)

))

predict = []

for i in range(30, 45):

combol = []

inputs = []

for j in range(6):

x = counter \* step

inputs.append(func(x))

counter += 1

combol.append(inputs)

x = counter \* step

combol.append(func(x))

combo = tuple(combol)

predict.append(combo)

print("\nРЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ")

for input\_stat, correct\_predict in predict:

print("the prediction is: {}, expected: {}, mistake: {}".format(

str(network.predict(input\_stat)),

str(correct\_predict),

str(network.predict(input\_stat) - correct\_predict)

))

Результат выполнения без адаптивного шага:

epochs: 1044, training loss: 9.97835174129353e-06

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ:

the prediction is: [0.21340099], expected: 0.21566083134884803, mistake: [-0.00225984]

the prediction is: [0.21760229], expected: 0.21895911035459031, mistake: [-0.00135682]

the prediction is: [0.20984561], expected: 0.20970576630561158, mistake: [0.00013984]

the prediction is: [0.19040051], expected: 0.18843123836158115, mistake: [0.00196927]

the prediction is: [0.16003835], expected: 0.15635506863238505, mistake: [0.00368328]

the prediction is: [0.12015031], expected: 0.11531599308488084, mistake: [0.00483432]

the prediction is: [0.07284188], expected: 0.06766653774985257, mistake: [0.00517534]

the prediction is: [0.0209112], expected: 0.016138162401499512, mistake: [0.00477303]

the prediction is: [-0.03234832], expected: -0.03631531778004107, mistake: [0.003967]

the prediction is: [-0.08349926], expected: -0.08668705685864561, mistake: [0.0031878]

the prediction is: [-0.12935762], expected: -0.13208954272987805, mistake: [0.00273192]

the prediction is: [-0.16730164], expected: -0.16992012101009504, mistake: [0.00261848]

the prediction is: [-0.19541003], expected: -0.19801018972723708, mistake: [0.00260016]

the prediction is: [-0.21243903], expected: -0.2147495123693462, mistake: [0.00231048]

the prediction is: [-0.21771438], expected: -0.2191785231633661, mistake: [0.00146414]

the prediction is: [-0.21102923], expected: -0.21104333327177924, mistake: [1.40999841e-05]

the prediction is: [-0.19261239], expected: -0.19081028472909783, mistake: [-0.0018021]

the prediction is: [-0.16318558], expected: -0.15963921782762297, mistake: [-0.00354636]

the prediction is: [-0.12407932], expected: -0.11931698437419269, mistake: [-0.00476234]

the prediction is: [-0.0773334], expected: -0.0721550181073362, mistake: [-0.00517838]

the prediction is: [-0.02568897], expected: -0.020856833953390233, mistake: [-0.00483214]

the prediction is: [0.02759245], expected: 0.03163694839911395, mistake: [-0.00404449]

the prediction is: [0.07906984], expected: 0.08231717273037709, mistake: [-0.00324733]

the prediction is: [0.1255216], expected: 0.12827864331182462, mistake: [-0.00275704]

the prediction is: [0.16426697], expected: 0.16688666249216988, mistake: [-0.00261969]

the prediction is: [0.19332061], expected: 0.1959280622380926, mistake: [-0.00260745]

the prediction is: [0.21138163], expected: 0.21373807188989244, mistake: [-0.00235644]

the prediction is: [0.21772937], expected: 0.21929574955851877, mistake: [-0.00156638]

the prediction is: [0.21211757], expected: 0.2122825066497665, mistake: [-0.00016493]

the prediction is: [0.19473474], expected: 0.1931003706528747, mistake: [0.00163437]

РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ

the prediction is: [0.16625397], expected: 0.1628489392965125, mistake: [0.00340503]