Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

за 1 семестр

По дисциплине: «МиАПР»

Тема: «Нелинейные ИНС в задачах прогнозирования»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Кречко К.А

Проверил:

Крощенко А.А.

2020

**Лабораторная работа №3**

Нелинейные ИНС в задачах прогнозирования

Цель работы: Изучить обучение и функционирование нелинейной ИНС при решении задач прогнозирования.

**Задание:**

Написать на любом ЯВУ программу моделирования прогнозирующей нелинейной ИНС. Для тестирования использовать функцию:

Y = a\*cos(b\*x)+c\*sin(d\*x)

**Код программы:**

import numpy as np

def func(x):

a, b, c, d, step = 0.2, 0.6, 0.05, 0.6, 0.1

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

def activation(x):

return np.tanh(x)

def derivative(x):

return 1 - (activation(x) \*\* 2)

def error(y,Y):

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

def training(inputs,predict,weights\_hidden,weights\_input):

learning\_rate = 0.5 # скорость обучения

inputs\_hidden = np.dot(weights\_hidden,inputs)

outputs\_hidden = activation\_mapper(inputs\_hidden)

inputs\_input = np.dot(weights\_input,outputs\_hidden)

outputs\_input = activation(inputs\_input)

error\_input = np.array([outputs\_input[0] - predict])

gradient\_input = derivative(outputs\_input[0])

delta\_input = error\_input \* gradient\_input

weights\_input -= learning\_rate \* np.dot(delta\_input,outputs\_hidden.reshape(1,len(outputs\_hidden)))

error\_hidden = delta\_input \* weights\_input

gradient\_hidden = derivative(outputs\_hidden)

delta\_hidden = error\_hidden \* gradient\_hidden

weights\_hidden -= learning\_rate \* np.dot(inputs.reshape(len(inputs),1),delta\_hidden).T

return weights\_hidden,weights\_input

def prediction(inputs,weights\_hidden,weights\_input):

inputs\_hidden = np.dot(weights\_hidden,inputs)

outputs\_hidden = activation\_mapper(inputs\_hidden)

inputs\_input = np.dot(weights\_input,outputs\_hidden)

outputs\_input = activation(inputs\_input)

return outputs\_input

activation\_mapper = np.vectorize(activation)

learning = []

predictions = []

epoch = 0

epoch\_maximum = 1000

error\_minimum = 1e-5 # минимальная ошибка

n\_input = 10 # количество входов

n\_hidden = 4 # количество элементов скрытого слоя

n\_train = 30 # размер выборки для обучения

n\_predict = 15 # размер выборки для прогназированния

param = 0

w\_hidden = np.random.normal(0.0,2 \*\* -0.5,(n\_hidden,n\_input))

w\_input = np.random.normal(0.0,1,(1,n\_hidden))

for i in range(n\_train):

inp, com = [], []

for j in range(n\_input):

inp.append(func(param))

param += 1

com.append(inp)

com.append(func(param))

learning.append(tuple(com))

while True:

inputs, predicts = [], []

for sample,predict in learning:

w\_hidden, w\_input = training(np.array(sample),predict,w\_hidden,w\_input)

inputs.append(np.array(sample))

predicts.append(np.array(predict))

error\_learning = error(prediction(np.array(inputs).T,w\_hidden,w\_input),np.array(predicts))

epoch += 1

if error\_learning <= error\_minimum or epoch > epoch\_maximum:

break

print("Ошибка: {}, Эпохи: {}".format(str(error\_learning),str(epoch)))

for i in range(n\_train,n\_train + n\_predict):

inp, com = [], []

for j in range(n\_input):

inp.append(func(param))

param += 1

com.append(inp)

com.append(func(param))

predictions.append(tuple(com))

print("\nРЕЗУЛЬТАТЫ ОБУЧЕНИЯ:")

for sample,predict in learning:

output = prediction(sample,w\_hidden,w\_input)

print("прогноз: {:<20} ожидаемый: {:<30} погрешность: {:<20}".format(str(output),str(np.array(predict)),str(output - predict)))

print("\nРЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ:")

for sample,predict in predictions:

output = prediction(sample,w\_hidden,w\_input)

print("прогноз: {:<20} ожидаемый: {:<30} погрешность: {:<20}".format(str(output),str(np.array(predict)),str(output - predict)))

Результат:

Ошибка: 9.986984931341595e-06, Эпохи: 202

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ:

прогноз: [0.18938193] ожидаемый: 0.19329924665168743 погрешность: [-0.00391732]

прогноз: [0.12127885] ожидаемый: 0.119073505193696 погрешность: [0.00220535]

прогноз: [0.00257492] ожидаемый: 0.0032519626052923342 погрешность: [-0.00067704]

прогноз: [-0.11772182] ожидаемый: -0.11370558408069162 погрешность: [-0.00401624]

прогноз: [-0.18833451] ожидаемый: -0.19094249891709572 погрешность: [0.00260799]

прогноз: [-0.19644575] ожидаемый: -0.20147770543157204 погрешность: [0.00503196]

прогноз: [-0.14282196] ожидаемый: -0.1416309528888193 погрешность: [-0.00119101]

прогноз: [-0.03314866] ожидаемый: -0.03230843375390258 погрешность: [-0.00084023]

прогноз: [0.09253763] ожидаемый: 0.08830035081072758 погрешность: [0.00423728]

прогноз: [0.17731466] ожидаемый: 0.1780632824201269 погрешность: [-0.00074862]

прогноз: [0.2000851] ожидаемый: 0.20562358656737478 погрешность: [-0.00553849]

прогноз: [0.16105866] ожидаемый: 0.16135365609890856 погрешность: [-0.000295]

прогноз: [0.06271054] ожидаемый: 0.060718251381260024 погрешность: [0.00199228]

прогноз: [-0.06484238] ожидаемый: -0.061127785418923095 погрешность: [-0.0037146]

прогноз: [-0.16275641] ожидаемый: -0.16162012811484758 погрешность: [-0.00113628]

прогноз: [-0.20030952] ожидаемый: -0.20565391021997442 погрешность: [0.00534439]

прогноз: [-0.17586252] ожидаемый: -0.17784686478511696 погрешность: [0.00198434]

прогноз: [-0.09044746] ожидаемый: -0.08791279279439154 погрешность: [-0.00253467]

прогноз: [0.03530266] ожидаемый: 0.03273174698634439 погрешность: [0.00257092]

прогноз: [0.14469829] ожидаемый: 0.14194214584647669 погрешность: [0.00275614]

прогноз: [0.1971043] ожидаемый: 0.20156806946125788 погрешность: [-0.00446377]

прогноз: [0.18719833] ожидаемый: 0.19078046726345116 погрешность: [-0.00358214]

прогноз: [0.11570162] ожидаемый: 0.11334775906201464 погрешность: [0.00235386]

прогноз: [-0.00473669] ожидаемый: -0.0036805824152872993 погрешность: [-0.00105611]

прогноз: [-0.12327702] ожидаемый: -0.11942319056390842 погрешность: [-0.00385383]

прогноз: [-0.19043323] ожидаемый: -0.19344784242179097 погрешность: [0.00301462]

прогноз: [-0.19507891] ожидаемый: -0.19989559739237026 погрешность: [0.00481669]

прогноз: [-0.13799745] ожидаемый: -0.13651406916134773 погрешность: [-0.00148338]

прогноз: [-0.02594587] ожидаемый: -0.025444249037836104 погрешность: [-0.00050162]

прогноз: [0.09876768] ожидаемый: 0.09451397931023223 погрешность: [0.0042537]

РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ:

прогноз: [0.18025231] ожидаемый: 0.18145575550097895 погрешность: [-0.00120344]

прогноз: [0.19952993] ожидаемый: 0.20500981578036886 погрешность: [-0.00547989]

прогноз: [0.15703525] ожидаемый: 0.1569480492382424 погрешность: [8.7198203e-05]

прогноз: [0.05582645] ожидаемый: 0.054059813673469355 погрешность: [0.00176663]

прогноз: [-0.07161495] ожидаемый: -0.06771307011805086 погрешность: [-0.00390188]

прогноз: [-0.16653377] ожидаемый: -0.16583183040007735 погрешность: [-0.00070194]

прогноз: [-0.2005658] ожидаемый: -0.20602076131163946 погрешность: [0.00545496]

прогноз: [-0.17266029] ожидаемый: -0.17424071304252675 погрешность: [0.00158042]

прогноз: [-0.08406288] ожидаемый: -0.08159337077086955 погрешность: [-0.00246951]

прогноз: [0.04244412] ожидаемый: 0.0395568833670682 погрешность: [0.00288724]

прогноз: [0.14929887] ожидаемый: 0.14688878008620948 погрешность: [0.00241009]

прогноз: [0.19817778] ожидаемый: 0.2029081999045 погрешность: [-0.00473042]

прогноз: [0.18481977] ожидаемый: 0.18804594779058334 погрешность: [-0.00322618]

прогноз: [0.10995671] ожидаемый: 0.10749383599752856 погрешность: [0.00246287]

прогноз: [-0.01204169] ожидаемый: -0.010608965326542083 погрешность: [-0.00143272]

Вывод: В ходе выполнения работы спроектировал линейную ИНС с использованием адаптивного шага обучения.