Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

за 1 семестр

По дисциплине: «МиАПР»

Тема: «Линейная искусственная нейронная сеть. Правило Видроу-Хоффа»

Выполнил:

Студент 2 курса

Группы ПО-4(2)

Яковчик И.А.

Проверил:

Крощенко А.А.

2020

Лабораторная работа №1

Линейная искусственная нейронная сеть. Правило Видроу-Хоффа

Цель работы: изучить обучение и функционирование линейной ИНС при решении

задач прогнозирования.

Вариант 1

Задание:

Написать на любом ЯВУ программу моделирования прогнозирующей линейной

ИНС. Для тестирования использовать функцию

y = a\*sin(bx) + d

a = 1, b = 5 d = 0.1, кол-во входов ИНС = 3.

Обучение и прогнозирование производить на 30 и 15 значениях соответственно

табулируя функцию с шагом 0.1. Скорость обучения выбирается студентом

самостоятельно, для чего моделирование проводится несколько раз для разных .

Результаты оцениваются по двум критериям - скорости обучения и минимальной

достигнутой ошибке. Необходимо заметить, что эти критерии в общем случае

являются взаимоисключающими, и оптимальные значения для каждого критерия

достигаются при разных a.

Код программы:

import math

import random

def function(x, a, b, d):

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

a = 1

b = 5

d = 0.1

a\_inputs = 3 # Количество входов нс

a\_testing = 30 # Кол-во элементов для обучения нс

a\_learning = 15 # Количество элементов для тестирования нс

min\_error = 0.0001 # Минимальная среднеквадратичная ошибка

step = 0.1

s\_training = 0.2 # Скорость обучения

T = random.uniform(0.5, 1) # Порог нс

synaptic\_weights = [] # Синаптические веса

#рандомно генерируем веса

for i in range(a\_inputs):

synaptic\_weights.append(random.uniform(0, 1))

training\_outputs = [] # Эталонные выходные значения

for i in range(a\_testing + a\_learning):

x = i \* step

training\_outputs.append(function(x, a, b, d))

error = 1

print(T)

print(synaptic\_weights)

while(error > min\_error):

error = 0 #Суммарная среднеквадратичная ошибка

for i in range(a\_testing):

output = 0

# Вычисляем выходное значение нейронной сети

for j in range(a\_inputs):

output += (synaptic\_weights[j] \* training\_outputs[j + i])

output -= T

#Корректируем порог нейронной сети, веса и ошибку

for j in range(a\_inputs):

synaptic\_weights[j] -= s\_training \* (output - training\_outputs[i + a\_inputs]) \* training\_outputs[i + j]

T += s\_training \* (output - training\_outputs[i + a\_inputs])

error += 0.5 \* ((output - training\_outputs[i + a\_inputs]) \*\* 2)

outputs = []

print("Результаты обучения:")

for i in range(a\_testing):

outputs.append(0)

for j in range(a\_inputs):

outputs[i] += synaptic\_weights[j] \* training\_outputs[j + i - a\_inputs]

outputs[i] -= T

print(str(i) + " " + str(outputs[i]) + " " + str(training\_outputs[i]) + " " + str(outputs[i] - training\_outputs[i]))

print("Результаты прогнозирования:")

for i in range(a\_learning):

outputs.append(0)

for j in range(a\_inputs):

outputs[i + a\_testing] += synaptic\_weights[j] \* training\_outputs[a\_testing - a\_inputs + j + i]

outputs[i + a\_testing] -= T

print(str(i + a\_testing) + " " + str(outputs[i + a\_testing]) + " " + str(training\_outputs[i + a\_testing]) + " " + str(outputs[i + a\_testing] - training\_outputs[i + a\_testing]))



Вывод: В ходе выполнения данной работы я спроектировал линейную искусственную нейронную сеть, используя правило обучения Видроу-Хоффа.