

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА** - **Российский технологический университет"**  **РТУ МИРЭА** |

**Институт** Информационных Технологий

**Кафедра** Вычислительной Техники

**ПРАКТИЧЕСКАЯ РАБОТА № 8**

**по дисциплине**

**«Проектирование систем поддержки принятия решений»**

Студент группы: ИКБО-14-20 Вежновец Ф.Ю. *(Ф.И.О.студента)*

Руководитель \_\_Семенов Р. Э.\_\_

*(Ф.И.О. преподавателя)*

Москва 2022

Практическая работа 8

Постановка задачи

Приобрести навыки проектирования и программной реализации поставленной задачи предметной области при помощи логических нейронных сетей.

Результат работы

Опишем предметную область.

Высказывания:

– сборщик установил систему охлаждения;

– сборщик одел кулер на башню;

– сборщик подключил кулер;

– сборщик прикрутил башню;

a – сборщик нанёс термопасту.

Аксиомы:

– чтобы написать экзамен, надо получить справку и найти кабинет;

Целевая логическая функция:

Для данной логической функции напишем таблицу истинности (Таблица 1).

Таблица 1 – Таблица истинности для целевой истинности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

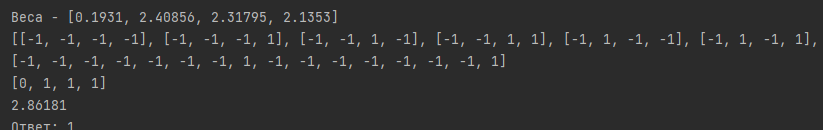


Рисунок 1 – Результат обучения

Листинг 1 – Функция main

if \_\_name\_\_ == '\_\_main\_\_':

last\_error = 0

step = 0.01

epochs = 1

mass\_value = [[-1, -1, -1, -1],

[-1, -1, -1, 1],

[-1, -1, 1, -1],

[-1, -1, 1, 1],

[-1, 1, -1, -1],

[-1, 1, -1, 1],

[-1, 1, 1, -1],

[-1, 1, 1, 1],

[1, -1, -1, -1],

[1, -1, -1, 1],

[1, -1, 1, -1],

[1, -1, 1, 1],

[1, 1, -1, -1],

[1, 1, -1, 1],

[1, 1, 1, -1],

[1, 1, 1, 1]]

mass\_answers = [-1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1]

brain = Neural(mass\_value,

random\_weight(len(mass\_value[0])),

step,

mass\_answers,

last\_error,

epochs)

brain.start([0, 1, 1, 1])

Листинг 2 – Описание класса нейронной сети Neural

class Neural:

def \_\_init\_\_(self, inputs, weight, step, answers, last\_error):

self.inputs = inputs

self.weight = weight

self.step = step

self.answers = answers

self.last\_error = last\_error

self.t = 0

Листинг 3 – Реализация метода обучения нейронной сети

def train(self):

eph = 0

count = 0

while count != len(self.inputs):

self.summ\_to\_active = 0

for i in range(len(self.inputs[count])):

self.summ\_to\_active += self.inputs[count][i] \* self.weight[i]

self.summ\_to\_active -= self.t

if self.summ\_to\_active > 0:

self.summ\_to\_active = 1

else:

self.summ\_to\_active = -1

if self.summ\_to\_active != self.answers[count]:

for i in range(len(self.weight)):

self.weight[i] = round(self.weight[i] + self.inputs[count][i] \* self.answers[count],

Продолжение Листинг 3

5)

# print(self.weight[i], i)

self.t = self.t - self.answers[count]

count = 0

else:

count += 1

eph += 1

print(f'Эпоха №{eph} - ', self.weight)

Листинг 4 – Реализация метода для распознавания заданной картинки

def start(self, new\_value):

self.train()

print('Веса -', self.weight)

final\_answer = 0

for i in range(len(new\_value)):

final\_answer += new\_value[i] \* self.weight[i]

final\_answer -= self.t

if final\_answer >= 0:

final\_answer = 1

print('Ответ:', final\_answer)

else:

print('Ответ:', -1)