Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Модели решения задач в интеллектуальных системах»

Тема: «Сверточные сети»

**Выполнил:**

Студент 3 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Туз И. С.

Брест 2024

**Цель:** реализовать и обучить свёрточную нейронную сеть для классификации изображений.

**Ход работы**

**Вариант 7**

****

import numpy as np

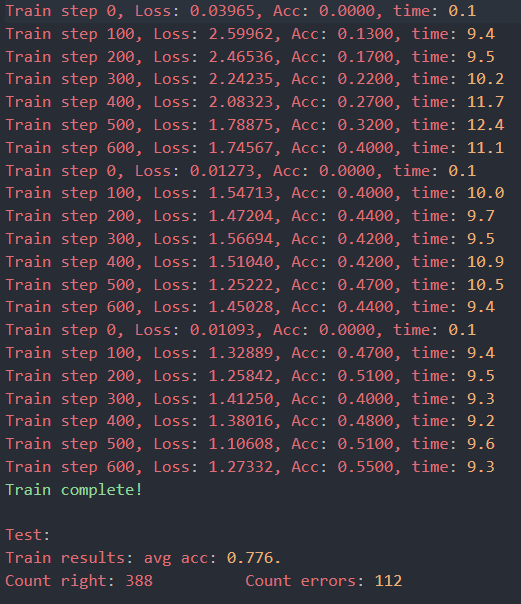
from models import \*

class CnnFromScratch:

def \_\_init\_\_(self, load\_weights=False):

self.conv1 = Conv2d(1, 1, 3, 1)

self.conv2 = Conv2d(1, 1, 2, 2, padding=1)

 self.max\_pool = Maxpool2d(2, 2, padding=1)

self.fc1 = Linear(64, 30)

self.fc2 = Linear(30, 10)

self.flatten = Flatten()

self.relu = ReLU()

self.sigmoid1 = Sigmoid()

self.sigmoid2 = Sigmoid()

self.softmax = Softmax()

if load\_weights:

with open('MRZIS/lab4/4\_for\_danik/weights.json') as file:

data = json.load(file)

self.fc1.fc\_w = np.array(data['fc1']['w'])

self.fc1.fc\_b = np.array(data['fc1']['b'])

self.fc2.fc\_w = np.array(data['fc2']['w'])

self.fc2.fc\_b = np.array(data['fc2']['b'])

def \_\_call\_\_(self, x):

x = self.conv1(x)

x = self.relu(x)

x = self.max\_pool(x)

x = self.conv2(x)

x = self.sigmoid1(x)

x = self.flatten.matrices2vector(x)

x = self.fc1(x)

x = self.sigmoid2(x)

x = self.fc2(x)

x = self.softmax(x)

return x

def backprop(self, x, lr=0.01):

x = self.softmax.backprop(x)

x = self.fc2.backprop(x, lr)

x = self.sigmoid2.backprop(x)

x = self.fc1.backprop(x, lr)

x = self.flatten.vector2matrices(x)

x = self.sigmoid1.backprop(x)

x = self.conv2.backprop(x, lr)

x = self.max\_pool.backprop(x)

x = self.relu.backprop(x)

x = self.conv1.backprop(x, lr)

**Вывод:** в ходе лабораторной работы я реализовал сверточную нейронную сеть для классификации изображений.