

МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

ПРАКТИЧЕСКИЕ ЗАДАНИЯ ПО КУРСУ «НЕЙРОННЫЕ СЕТИ»
ОТЧЕТ О ПРАКТИКЕ

студента 5 курса 531 группы
направления 100501 — Компьютерная безопасность
факультета КНиИТ
Окуньков Сергей Викторович

Проверил
доцент

И. И. Слеповичев

СОДЕРЖАНИЕ

1	Задание 1: Создание ориентированного графа	3
1.1	Описание	3
1.2	Пример исполнения программы	3
2	Задание 2: Создание функции по графу	5
2.1	Описание	5
2.2	Пример исполнения программы	5
3	Задание 3: Вычисление значение функции на графе	7
3.1	Описание	7
3.2	Пример исполнения программы	7
4	Задание 4: Построение многослойной нейронной сети	9
4.1	Описание	9
4.2	Пример исполнения программы	9

1 Задание 1: Создание ориентированного графа

1.1 Описание

На входе: текстовый файл с описанием графа в виде списка дуг:

$$(a_1, b_1, n_1), (a_2, b_2, n_2), \dots, (a_k, b_k, n_k),$$

где a_i — начальная вершина дуги i , b_i — конечная вершина дуги i , n_i — порядковый номер дуги в списке всех заходящих в вершину b_i дуг.

На выходе: Ориентированный граф с именованными вершинами и линейно упорядоченными дугами (в соответствии с порядком из текстового файла). Сообщение об ошибке в формате файла, если ошибка присутствует.

1.2 Пример исполнения программы

Содержимое файла ‘example.txt’:

$$(v_1, v_2, 1), (v_2, v_3, 2), (v_1, v_4, 1)$$

Запускаем программу с помощью консоли следующим образом:

```
python3 task1.py --input_file tests/example.txt --output_file
↪ tests/example.xml
```

В качестве результата получаем файл ‘example.xml’ со следующим содержанием:

```
<graph>
  <vertex>v_4</vertex>
  <vertex>v_3</vertex>
  <vertex>v_2</vertex>
  <vertex>v_1</vertex>
  <arc>
    <from>v_1</from>
    <to>v_2</to>
    <order>1</order>
  </arc>
  <arc>
    <from>v_1</from>
    <to>v_4</to>
    <order>1</order>
  </arc>
```

```
<arc>
  <from>v_2</from>
  <to>v_3</to>
  <order>2</order>
</arc>
</graph>
```

2 Задание 2: Создание функции по графу

2.1 Описание

На входе: ориентированный граф с именованными вершинами как описано в задании 1.

На выходе: линейное представление функции, реализуемой графом в префиксной скобочной записи:

$$A_1(B_1(C_1(\dots), \dots, C_m(\dots)), \dots, B_n(\dots))$$

2.2 Пример исполнения программы

Содержимое файла ‘example.xml’:

```
<graph>
  <vertex>v_4</vertex>
  <vertex>v_3</vertex>
  <vertex>v_2</vertex>
  <vertex>v_1</vertex>
  <arc>
    <from>v_1</from>
    <to>v_2</to>
    <order>1</order>
  </arc>
  <arc>
    <from>v_1</from>
    <to>v_4</to>
    <order>1</order>
  </arc>
  <arc>
    <from>v_2</from>
    <to>v_3</to>
    <order>2</order>
  </arc>
</graph>
```

Запускаем программу с помощью консоли следующим образом:

```
python3 task2.py --input_file tests/example.xml --output_file
↳ tests/example2.txt
```

В качестве результата получаем файл ‘example2.txt’ со следующим содержанием:

$$v_1(v_2(v_3), v_4)$$

3 Задание 3: Вычисление значения функции на графе

3.1 Описание

На входе:

1. Текстовый файл с описанием графа в виде списка дуг (смотри задание 1).
2. Текстовый файл соответствий арифметических операций именам вершин:

a_1 : 1-я операция

a_2 : 2-я операция

...

a_n : n -я операция,

где a_i – имя i -й вершины, i -я операция – символ операции, соответствующий вершине a_i .

Допустимы следующие символы операций:

$+$ – сумма значений,

$*$ – произведение значений,

exp – экспонирование входного значения,

число – любая числовая константа.

На выходе: значение функции, построенной по графу и файлу.

3.2 Пример исполнения программы

Содержимое ‘example.txt’ и ‘example.json’:

Содержимое файла ‘example.xml’:

```
<graph>
  <vertex>v_4</vertex>
  <vertex>v_3</vertex>
  <vertex>v_2</vertex>
  <vertex>v_1</vertex>
  <arc>
    <from>v_1</from>
    <to>v_2</to>
    <order>1</order>
  </arc>
  <arc>
    <from>v_1</from>
    <to>v_4</to>
```

```
        <order>1</order>
    </arc>
    <arc>
        <from>v_2</from>
        <to>v_3</to>
        <order>2</order>
    </arc>
</graph>
```

example.json

```
{
    "v_1": "+",
    "v_2": "exp",
    "v_3": 5,
    "v_4": 8
}
```

Запускаем программу с помощью консоли следующим образом:

```
python3 task3.py --input_file tests/example.xml --output_file
↳ tests/example3.txt --nodes_to_func_file tests/example.json
```

В качестве результата получаем файл 'example3.txt' со следующим содержанием:

```
[156.4131591025766]
```


4 Задание 4: Построение многослойной нейронной сети

4.1 Описание

На входе:

1. Файл с набором матриц весов межнейронных связей:

$$\begin{aligned} M_1 &: [a_{11}^1, a_{12}^1, \dots, a_{1n_1}^1], \dots, [a_{m_11}^1, a_{m_12}^1, \dots, a_{m_1n_1}^1] \\ M_2 &: [a_{11}^2, a_{12}^2, \dots, a_{1n_2}^2], \dots, [a_{m_21}^2, a_{m_22}^2, \dots, a_{m_2n_2}^2] \\ &\dots \\ M_p &: [a_{11}^p, a_{12}^p, \dots, a_{1n_p}^p], \dots, [a_{m_p1}^p, a_{m_p2}^p, \dots, a_{m_pn_p}^p] \end{aligned}$$

2. Файл с входным вектором в формате:

$$x_1, x_2, \dots, x_k.$$

На выходе:

1. Сериализованная многослойная нейронная сеть с полносвязной межслойной структурой. Файл с выходным вектором – результатом вычислений НС в формате:

$$y_1, y_2, \dots, y_k.$$

2. Сообщение об ошибке, если в формате входного вектора или файла описания НС допущена ошибка.

4.2 Пример исполнения программы

Содержимое файлов 'nn.json' и 'X.txt':

X.txt

1., 2., 3.

nn.json

```
{
  "layer1.sigmoid": [[1, 1, 1], [2, 2, 2]],
  "layer2.linear": [[1, 1], [2, 2], [3, 3]],
  "layer3.softmax": null,
  "layer4.argmax": null
}
```

Запускаем программу с помощью консоли следующим образом:

```
python3 task4.py --input_file tests/X.txt --weights_file  
↳ tests/nn.json --output_file tests/Y.txt
```

В качестве результата получаем файл ‘Y.txt’ со следующим содержимым:

```
{"Y": 2}
```