

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

ОТЧЁТ ПО ПРАКТИЧЕСКИМ ЗАДАНИЯМ ПО КУРСУ
«НЕЙРОННЫЕ СЕТИ»

ЛАБОРАТОРНАЯ РАБОТА

студента 4 курса 431 группы
специальности 10.05.01 — Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Гущина Андрея Юрьевича

Проверил
доцент

И. И. Слеповичев

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Создание ориентированного графа	4
2 Создание функции по графу	6
3 Вычисление значения функции на графе	9
4 Построение многослойной нейронной сети	11

ВВЕДЕНИЕ

Задания выполнены на языке программирования Rust с использованием библиотек

- clap
- serde
- serde_json
- burn
- xml

Для сборки задания N необходимо перейти в директорию `nntaskN` и выполнить команду

```
1 cargo build --release
```

В результате сборки загрузятся соответствующие библиотеки и появится директория `target` в текущей директории. В ней будет находиться исполняемый файл `nntaskN` (или `nntaskN.exe` на Windows). Примеры запуска каждого из заданий приведены ниже.

1 Создание ориентированного графа

Вход

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.

Выход

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

Пример исполнения программы

Пусть задан файл `tests/t1_input.txt` со следующим содержимым:

```
1 (f, b, 1), (b, a, 2), (d, c, 4), (e, c, 5), (c, a, 3)
```

При запуске программы командой

```
1 ./nntask1 --input1 ./tests/t1_input.txt --output1 ./tests/t1_output.xml
```

в результате получим файл `tests/t1_output.xml`

```
1 <graph>
2   <vertex>a</vertex>
3   <vertex>b</vertex>
4   <vertex>c</vertex>
5   <vertex>d</vertex>
6   <vertex>e</vertex>
7   <vertex>f</vertex>
8   <arc>
9     <from>f</from>
10    <to>b</to>
11    <order>1</order>
12  </arc>
13  <arc>
14    <from>b</from>
15    <to>a</to>
16    <order>2</order>
```

```
17  </arc>
18  <arc>
19    <from>c</from>
20    <to>a</to>
21    <order>3</order>
22  </arc>
23  <arc>
24    <from>d</from>
25    <to>c</to>
26    <order>4</order>
27  </arc>
28  <arc>
29    <from>e</from>
30    <to>c</to>
31    <order>5</order>
32  </arc>
33 </graph>
```

2 Создание функции по графу

Вход

Ориентированный граф с именованными вершинами как описано в задании 1.

Выход

- Линейное представление функции, реализуемой графом в префиксной скобочной записи, либо
- Сообщение об ошибке в формате файла, если ошибка присутствует, либо
- Сообщение о наличии в графе циклов.

Пример исполнения программы

Пусть задан файл `tests/t1_output.xml` со следующим содержимым:

```
1 <graph>
2   <vertex>a</vertex>
3   <vertex>b</vertex>
4   <vertex>c</vertex>
5   <vertex>d</vertex>
6   <vertex>e</vertex>
7   <vertex>f</vertex>
8   <arc>
9     <from>f</from>
10    <to>b</to>
11    <order>1</order>
12  </arc>
13  <arc>
14    <from>b</from>
15    <to>a</to>
16    <order>2</order>
17  </arc>
18  <arc>
19    <from>c</from>
20    <to>a</to>
21    <order>3</order>
22  </arc>
23  <arc>
24    <from>d</from>
25    <to>c</to>
26    <order>4</order>
```

```

27     </arc>
28     <arc>
29         <from>e</from>
30         <to>c</to>
31         <order>5</order>
32     </arc>
33 </graph>

```

При запуске программы командой

```
1 ./nntask2 --input1 ./tests/t1_output.xml --output1 ./tests/t2_output.txt
```

в результате получим файл tests/t2_output.txt

```
1 a(b(), c(d(), e()))
```

Если в качестве ввода задан граф с циклом (файл tests/t2_input_cycle.xml):

```

1 <graph>
2     <vertex>a</vertex>
3     <vertex>b</vertex>
4     <vertex>c</vertex>
5     <vertex>d</vertex>
6     <vertex>e</vertex>
7     <arc>
8         <from>a</from>
9         <to>b</to>
10        <order>1</order>
11    </arc>
12    <arc>
13        <from>a</from>
14        <to>c</to>
15        <order>2</order>
16    </arc>
17    <arc>
18        <from>c</from>
19        <to>d</to>
20        <order>3</order>
21    </arc>
22    <arc>
23        <from>c</from>
24        <to>e</to>
25        <order>4</order>

```

```
26     </arc>
27     <arc>
28         <from>e</from>
29         <to>c</to>
30         <order>5</order>
31     </arc>
32 </graph>
```

То при запуске программа выведет сообщение:

- 1 Некорректный ввод - в графе есть циклы

3 Вычисление значения функции на графе

Вход

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

```
1 {  
2     "a_1": "операция_1",  
3     "a_2": "операция_2",  
4     ...,  
5     "a_n": "операция_n",  
6 }
```

где a_i — — — i — , i — — , a_i .

Допустимы следующие символы операций:

- «+» — сумма значений,
- «*» — произведение значений,
- «exp» — экспонирование входного значения,
- «число» — любая числовая константа.

Выход

Значение функции, построенной по графу (1) и файлу (2).

Пример исполнения программы

Пусть задан файл `tests/t1_output.xml` с некоторым графом:

```
1 <graph>  
2   <vertex>a</vertex>  
3   <vertex>b</vertex>  
4   <vertex>c</vertex>  
5   <vertex>d</vertex>  
6   <vertex>e</vertex>  
7   <vertex>f</vertex>  
8   <arc>  
9     <from>f</from>  
10    <to>b</to>  
11    <order>1</order>  
12  </arc>  
13  <arc>  
14    <from>b</from>  
15    <to>a</to>
```

```

16     <order>2</order>
17 </arc>
18 <arc>
19     <from>c</from>
20     <to>a</to>
21     <order>3</order>
22 </arc>
23 <arc>
24     <from>d</from>
25     <to>c</to>
26     <order>4</order>
27 </arc>
28 <arc>
29     <from>e</from>
30     <to>c</to>
31     <order>5</order>
32 </arc>
33 </graph>

```

Также задан файл `tests/t3_ops.json` с соответствием операций вершинам:

```

1 {
2   "a": "+",
3   "b": "exp",
4   "c": "*",
5   "d": 5.0,
6   "e": 9.0,
7   "f": 2.0
8 }

```

При запуске программы командой

```

1 ./nntask3 --input1 ./tests/t1_output.xml --input2 ./tests/t3_ops.json
  ↪ --output1 ./tests/t3_output.txt

```

в результате получим файл `tests/t3_output.txt`

```

1 52.38905609893065

```

4 Построение многослойной нейронной сети

Вход

1. Текстовый файл с набором матриц весов межнейронных связей в формате:

```
1 {
2     "weights": [
3         [
4             [M1_11, M1_12, ..., M1_1n],
5             ...
6             [M1_m1, M1_m2, ..., M1_mn]
7         ],
8         ...,
9         [
10            [Mp_11, Mp_12, ..., Mp_1n],
11            ...
12            [Mp_m1, Mp_m2, ..., Mp_mn]
13        ]
14    ]
15 }
```

2. Текстовый файл с входным вектором в формате:

```
1 x_1, x_2, ..., x_n
```

Выход

1. Файл с выходным вектором – результатом вычислений НС в формате:

```
1 y_1, y_2, ..., y_n
```

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

Пример исполнения программы

Пусть задан файл tests/t4_w.json с весами НС:

```
1 {
2     "weights": [
3         [
4             [0.47519493033675375, 0.015705490366171526, 0.9433818257724572],
5             [0.48092032736144574, 0.13929695479782134, 0.6869903232566065],
6             [0.436988975888717, 0.20037642195993755, 0.17561406275527947]
7         ],
8         [
```

```

9          [0.042224071742743785, 0.15331022315027187, 0.464635658411239],
10         [0.6000159964796773, 0.22606113281552231, 0.5301212736820182],
11         [0.19651133783303198, 0.7498835958139106, 0.28721556978456597]
12     ],
13     [
14         [0.11837615025116721, 0.00927217999098906, 0.7504596929897048],
15         [0.5675946231090779, 0.9748635791740536, 0.30501309542663524],
16         [0.8574872089946126, 0.3047120321509168, 0.3376899733092712]
17     ]
18 ]
19 }

```

Также задан файл `tests/t4_x.txt` с входным вектором:

```
1 1, 2, 3
```

При запуске программы командой

```
1 ./nntask3 --input1 ./tests/t4_w.json --input2 ./tests/t4_x.txt --output1
  ↪ ./tests/t4_output.txt
```

в результате получим файл `tests/t4_output.txt`

```
1 [0.62035125, 0.5470768, 0.5459341]
```