

Практические задания по курсу "Нейронные сети"

Слеповичев И.И. 01.10.2021

Общие требования к выполнению заданий

1. Допустимые языки реализации: C#, Python, C++.
2. Каждое задание реализуется отдельной программой. Рекомендуемые названия - nntask1, nntask2, и т.д.
3. Программа запускается в консольном режиме с параметрами:
nntask1 input1=имя_входного_файла1 input2=имя_входного_файла2
output1=имя_выходного_файла1 output2=имя_выходного_файла2
Если второй входной файл или второй выходной файл отсутствуют – соответствующий параметр пропускается.
4. Игнорировать лишние пробелы и символы табуляции во входном файле. Кодировка входного/выходного файла задания - UTF-8.
5. Результат выполнения практических заданий должен быть оформлен в виде отчета. Форма отчета: титульный лист, описания задач, пример исполнения программы по каждому заданию.

Описание заданий

1. Создание ориентированного графа

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:

а) Ориентированный граф с именованными вершинами и линейно упорядоченными дугами (в соответствии с порядком из текстового файла).

б) Сообщение об ошибке в формате файла, если ошибка присутствует.

Способ проверки результата:

а) Сериализованная структура графа в формате XML или JSON.

Пример:

```
<graph>
  <vertex>v1</vertex>
  <vertex>v2</vertex>
  <vertex>v3</vertex>
  <arc>
    <from>v1</from>
    <to>v3</to>
    <order>1</order>
  </arc>
  <arc>
    <from>v2</from>
    <to>v3</to>
    <order>2</order>
  </arc>
</graph>
```

б) Сообщение об ошибке с указанием номера строки с ошибкой во входном файле.

2. Создание функции по графу

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

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

$$A1(B1(C1(...),..., Cn(...)),..., Vn(...))$$

Способ проверки результата:

- а) выгрузка в текстовый файл результата преобразования графа в имя функции.
- б) сообщение о наличии циклов в графе, если они присутствуют.

3. Вычисление значения функции на графе

На входе:

а) Текстовый файл с описанием графа в виде списка дуг (смотри задание 1).

б) Текстовый файл соответствий арифметических операций именам вершин:

a_1 : операция_1

a_2 : операция_2

...

a_n : операция_n

где a_i - имя i-й вершины, операция_i - 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 а) и файлу б).

Способ проверки результата: результат вычисления, выведенный в файл.

4. Построение многослойной нейронной сети

На входе:

а) Текстовый файл с набором матриц весов межнейронных связей:

$M1 : [M1[1,1], M1[1,2], \dots, M1[1,n]], \dots, [M1[m,1], M1[m,2], \dots, M1[m,n]]$

$M2 : [M2[1,1], M2[1,2], \dots, M2[1,n]], \dots, [M2[m,1], M2[m,2], \dots, M2[m,n]]$

...

$Mr : [Mr[1,1], Mr[1,2], \dots, Mr[1,n]], \dots, [Mr[m,1], Mr[m,2], \dots, Mr[m,n]]$

б) Текстовый файл с входным вектором в формате:

$x_1, x_2, \dots, x_n.$

На выходе:

а) Сериализованная многослойная нейронная сеть (в формате XML или JSON) с полносвязной межслойной структурой.

Файл с выходным вектором – результатом вычислений НС в формате:

$y_1, y_2, \dots, y_n.$

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

5. Реализация метода обратного распространения ошибки для многослойной НС

На входе:

а) Текстовый файл с описанием НС (формат см. в задании 4).

б) Текстовый файл с обучающей выборкой:

$[x_{11}, x_{12}, \dots, x_{1n}] \rightarrow [y_{11}, y_{12}, \dots, y_{1m}]$

...

$[x_{k1}, x_{k2}, \dots, x_{kn}] \rightarrow [y_{k1}, y_{k2}, \dots, y_{km}]$

Формат описания входного вектора x и выходного вектора y соответствует формату из задания

4.

в) Число итераций обучения (в строке параметров).

На выходе:

Текстовый файл с историей N итераций обучения методом обратного распространения ошибки:

1 : Ошибка1

2 : Ошибка2

...

N : Ошибка N