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.

Если параметр input или output не указаны, использовать стандартные названия файлов: input.txt, output.json (либо output.xml).

1. Игнорировать лишние пробелы и символы табуляции во входном файле. Кодировка входного/выходного файла задания - UTF-8.
2. 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), ..., (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:**

1. 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.

1. **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(...),..., Cm(...)),..., Bn(...))

Способ проверки результата:

a) выгрузка в текстовый файл результата преобразования графа в имя функции.

б) сообщение о наличии циклов в графе, если они присутствуют.

1. **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:

**+** – cумма 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:

М1 : [M1[1,1], M1[1,2],..., M1[1,n]], ..., [M1[m,1], M1[m,2],...,M1[m,n]]

М2 : [M2[1,1], M2[1,2],..., M2[1,n]], ..., [M2[m,1], M2[m,2],...,M2[m,n]]

...

Мp : [Mp[1,1], Mp[1,2],..., Mp[1,n]], ..., [Mp[m,1], Mp[m,2],...,Mp[m,n]]

б) Текстовый файл с входным вектором в формате:

x1, x2, …, xn.

**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:

y1, y2, …, yn.

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:

[x11, x12, ..., x1n] -> [y11, y12, ..., y1m]

...

[xk1, xk2, ..., xkn] -> [yk1, yk2, ..., ykm]

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 : ОшибкаN