|  |
| --- |
|  |
| Задачник по TensorFlow |
|  |

|  |
| --- |
|  |

Оглавление

[Тензоры 2](#_Toc169910645)

[Задание 1 2](#_Toc169910646)

[Задание 2 2](#_Toc169910647)

[Задание 3 2](#_Toc169910648)

[Задание 4 3](#_Toc169910649)

[Задание 5 3](#_Toc169910650)

[Задание 6 3](#_Toc169910651)

[Переменные тензора 4](#_Toc169910652)

[Задание 1 4](#_Toc169910653)

[Операции над тензорами 5](#_Toc169910654)

[Задание 1 5](#_Toc169910655)

[Задание 2 5](#_Toc169910656)

[Задание 3 5](#_Toc169910657)

[Задание 4 5](#_Toc169910658)

[Задание 5 6](#_Toc169910659)

[Индексирование с помощью фрагмента : 7](#_Toc169910660)

[Задание 1 7](#_Toc169910661)

[Многоосевое индексирование 8](#_Toc169910662)

[Задание 1 8](#_Toc169910663)

[Управление фигурами 9](#_Toc169910664)

[Задание 1 9](#_Toc169910665)

# Тензоры

**Normal tf.Tensor objects are immutable**

### Задание 1

Создать 2 тензора размерностью 2 на 2.

Назвать первый тензор именем first.

Второй тензор именем second.

Заполнить их значениями 1 2 3 4

Вывести в консоль тип данных и форму тензоров

### Задание 2

Создать 2 тензора размерностью 2 на 2.

Назвать первый тензор именем first\_int. Содержимое массива должно иметь тип int32

Второй тензор именем second\_float. Содержимое массива должно иметь тип float32

Заполнить их значениями 1 2 3 4

Вывести в консоль тип данных и форму тензоров

### Задание 3

Создать 3 одномерных тензора.

Заполнить их числами от 1 до 30, по 10 значений в каждом.

Назвать тензоры first second third .

first должен содержать значения от 1 до 10, second – 11-20 и так далее.

Вывести в консоль тип данных и форму тензоров

### Задание 4

Создать 3 одномерных тензора.

Заполнить их любым одним значением.

Вывести в консоль тип данных и форму тензора

### Задание 5

Создать тензор с целыми значениями, но при создании указать тип данных «число с плавающей точкой»

Вывести в консоль тип данных и форму тензора

### Задание 6

Создать трехмерный тензор, заполнить любыми значениями.

Вывести в консоль тип данных и форму тензора, сам тензор.

## Переменные тензора

### Задание 1

Создать любой тензор, заполнить любыми значениями.

Вывести в консоль:

* тип данных
* число осей
* форму
* количество элементов на каждой оси
* количество элементов на последней оси (предположим, что вы не знаете размерность тензора)
* общее число элементов в тензоре

## Операции над тензорами

### Задание 1

Создать любой тензор, заполнить любыми значениями.

Создать массив numpy на основе этого тензора.

Результат вывести в консоль.

Выполнить задание 2-мя способами.

### Задание 2

Создать 2 тензора.

Выполнить их сложение, поэлементное умножение и матричное умножение.

Результат вывести в консоль.

Выполнить задание 2-мя способами.

### Задание 3

Создать любой тензор, заполнить любыми значениями.

Найти наибольшее значение в тензоре.

Результат вывести в консоль.

### Задание 4

Создать любой тензор, заполнить любыми значениями.

Найти индекс наибольшего значения в тензоре.

Результат вывести в консоль.

### Задание 5

Создать любой тензор, заполнить любыми значениями.

Найти softmax.

Результат вывести в консоль.

## Индексирование с помощью фрагмента :

### Задание 1

Создать любой тензор, заполнить любыми значениями.

Используя индексирование с помощью фрагмента : вывести в консоль указанные ниже данные.

print("Everything:", ???)  
print("Before 4:", ???)  
print("From 4 to the end:", ???)  
print("From 2, before 7:", ???)

print("Every other item:", ???)  
print("Reversed:", ???)

## Многоосевое индексирование

### Задание 1

Создать любой тензор, заполнить любыми значениями.

Используя многоосевое индексирование вывести в консоль указанные ниже данные.

# Get row and column tensors  
print("Second row:", ???)  
print("Second column:", ???)  
print("Last row:", ???)

print("First item in last column:", ???)  
print("Skip the first row:", ???)

## Управление фигурами

### Задание 1

Создать любой тензор, заполнить любыми значениями.