# TF & KERAS. РОБОТА із ЗОБРАЖЕННЯМИ

## Файл: TF_KERAS_Image_01_001

### Визначення Тензорів

```
import tensorflow as tf
print(tf.__version__)
```

```
2.15.0
```

```
print(tf.executing_eagerly() )
```

```
True
```

```
import numpy as np
```

## ТЕНЗОР

Основний об'єкт TensorFlow - **ТЕНЗОР** – в деякому сенсі спадкоємець багатовимірних масивів в мовах програмування (наприклад, в пакеті *numpy*).

Тезор можна уявляти як деяку колекцію (багатовимірний список **list**) скалярів, веторів, матриць та інших тензорів меншого рангу.
Але, слід відзначити головну відмінність тензору від списку: **всі елементи тензору повинні бути одного типу**.

Один з основних параметрів тензору, це його ранг (*rank*), який визначає його вимірність (кількість осей *axis*).

### Ранг тензору

| Ранг | Математичне представлення |
|---|---|
| 0 | скаляр |
| 1 | вектор |
| 2 | матриця = вектор векторів |
| 3 | тензор = вектор векторів векторів (3-вимірний масив) |
| N | тензор = вектор векторів ... векторів(N-вимірний масив) |

Скаляр, це деяка змінна, наприклад $x$

Вектор $\vec{x}$- одновимірний список. Наприклад, вектор, що має 4 компоненти:

$$[x_0, x_1, x_2, x_3] \Longleftrightarrow \vec{x}$$

Матриця $X$, двовимірний список, або список списків. Наприклад, матриця розміром $3 * 4$, еквівалентна вектору із 3-х векторів:

$$\begin{bmatrix} [x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}] \\ [x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}] \\ [x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}] \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \vec{x_0} \\ \vec{x_1} \\ \vec{x_2} \end{bmatrix} \Longleftrightarrow X$$

Тензор $\mathcal{T}$ рангу 3 це список, що складається із списків, які в свою чергу теж є списками. Наприклад, тензор розмром $3 * 3 * 3$ :

$$\begin{bmatrix} \begin{bmatrix} [x_{0,0}, x_{0,1}, x_{0,2}] \\ [x_{1,0}, x_{1,1}, x_{1,2}] \\ [x_{2,0}, x_{2,1}, x_{2,2}] \end{bmatrix} \\ \begin{bmatrix} [y_{0,0}, y_{0,1}, y_{0,2}] \\ [x_{1,0}, y_{1,1}, y_{1,2}] \\ [y_{2,0}, y_{2,1}, y_{2,2}] \end{bmatrix} \\ \begin{bmatrix} [z_{0,0}, z_{0,1}, z_{0,2}] \\ [z_{1,0}, z_{1,1}, z_{1,2}] \\ [z_{2,0}, z_{2,1}, z_{2,2}] \end{bmatrix} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \begin{bmatrix} \vec{x_0} \\ \vec{x_1} \\ \vec{x_1} \end{bmatrix} \\ \begin{bmatrix} \vec{y_0} \\ \vec{y_1} \\ \vec{y_1} \end{bmatrix} \\ \begin{bmatrix} \vec{z_0} \\ \vec{z_1} \\ \vec{z_1} \end{bmatrix} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Longleftrightarrow \mathcal{T}$$

Та, нарешті, тензор $\mathcal{Q}$ рангу 4 - це список, що складається із списків, які в свою чергу теж є списками списків. Наприклад, тензор розмром $2 * 3 * 2 * 4$ елементів:

$$\begin{bmatrix} \begin{bmatrix} \begin{bmatrix} [x_{0,0}, \ldots, x_{0,3}] \\ [x_{1,0}, \ldots, x_{1,3}] \end{bmatrix} \\ \begin{bmatrix} [y_{0,0}, \ldots, y_{0,3}] \\ [x_{1,0}, \ldots, y_{1,3}] \end{bmatrix} \\ \begin{bmatrix} [z_{0,0}, \ldots, z_{0,3}] \\ [z_{1,0}, \ldots, z_{1,3}] \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} \begin{bmatrix} [x_{0,0}, \ldots, x_{0,3}] \\ [x_{1,0}, \ldots, x_{1,3}] \end{bmatrix} \\ \begin{bmatrix} [y_{0,0}, \ldots, y_{0,3}] \\ [y_{1,0}, \ldots, y_{1,3}] \end{bmatrix} \\ \begin{bmatrix} [z_{0,0}, \ldots, z_{0,3}] \\ [z_{1,0}, \ldots, z_{1,3}] \end{bmatrix} \end{bmatrix} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} \vec{x_0} \\ \vec{x_1} \end{bmatrix} \\ \begin{bmatrix} \vec{y_0} \\ \vec{y_1} \end{bmatrix} \\ \begin{bmatrix} \vec{z_0} \\ \vec{z_1} \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} \begin{bmatrix} \vec{x_0} \\ \vec{x_1} \end{bmatrix} \\ \begin{bmatrix} \vec{y_0} \\ \vec{y_1} \end{bmatrix} \\ \begin{bmatrix} \vec{z_0} \\ \vec{z_1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \\ \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \begin{bmatrix} \mathcal{T}_0 \\ \mathcal{T}_1 \end{bmatrix} \end{bmatrix} \Longleftrightarrow \mathcal{Q}$$

## ВИЗНАЧЕННЯ ТЕНЗОРУ в TENSORFLOW

Усі тензорні об'єкти в Tensorflow мають наступні властивості:

- ***name*** — ім'я тензору;
- ***value*** — значення тензору;
- ***rank*** — вимірність тензору ;
- ***shape*** — кількість елементів, яку містить тензор у кожному вимірі;
- ***dtype*** — тип даних, до якого належать усі елементи в тензорі.

Тип елементу *dtype* повністю аналогічен *numpy*. Наприклад:

- float128, float64, float32,
- int64, int32, int16, int8
- complex128, complex64
- string (!! навіть строки)

## Відмінності LIST <> Numpy Array <> Tensor

```
# Python List
m1 = [[1.0, 2.0], [3.0, 4.0]]
```

```
print(type(m1))
```

```
<class 'list'>
```

```
# Numpy Martrix
m2 = np.array([[1.0, 2.0], [3.0, 4.0]], dtype=np.float32)
print(type(m2), m2.shape)
```

```
<class 'numpy.ndarray'> (2, 2)
```

```
# Tensorflow Tensor
m3 = tf.constant([[1.0, 2.0], [3.0, 4.0]])
print(type(m3), m3.shape)
```

```
<class 'tensorflow.python.framework.ops.EagerTensor'> (2, 2)
```

## Перетворення до тензору

```
t1 = tf.convert_to_tensor(m1, dtype=tf.float32)
print(type(t1), t1.shape)
```

```
<class 'tensorflow.python.framework.ops.EagerTensor'> (2, 2)
```

```
t2 = tf.convert_to_tensor(m2, dtype=tf.float32)
print(type(t2), t2.shape)
```

```
<class 'tensorflow.python.framework.ops.EagerTensor'> (2, 2)
```

```
t3 = tf.convert_to_tensor(m3, dtype=tf.float32)
print(type(t3), t3.shape)
```

```
<class 'tensorflow.python.framework.ops.EagerTensor'> (2, 2)
```

## Приклади створення константних тензорів

```
scalar1 = tf.constant(2, dtype=tf.int8)
scalvec = tf.constant(2, shape = (1), dtype=tf.int16)
scalmatr = tf.constant(2, shape = (1,1), dtype=tf.int32)
```

```
print(scalar1)
print(scalvec)
print(scalmatr)
```

```
tf.Tensor(2, shape=(), dtype=int8)
tf.Tensor([2], shape=(1,), dtype=int16)
tf.Tensor([[2]], shape=(1, 1), dtype=int32)
```

```
# RANK - РАНГ
print(tf.rank(scalar1))
print(tf.rank(scalvec))
print(tf.rank(scalmatr))
```

```
tf.Tensor(0, shape=(), dtype=int32)
tf.Tensor(1, shape=(), dtype=int32)
tf.Tensor(2, shape=(), dtype=int32)
```

```
# SHAPE - РОЗМІРНІСТЬ
print(scalar1.shape)
print(scalvec.shape)
print(scalmatr.shape)
```

```
()
(1,)
(1, 1)
```

```
vector1 = tf.constant([1, 2, 3, 4], dtype=tf.float32)
vector2 = tf.constant([10, 20, 30, 40, 50], dtype=tf.float32)
```

```
print(vector1)
print('Ранг vec1',tf.rank(vector1))
print('Розмір vec1',vector1.shape)
print(vector2)
print('Ранг vec2',tf.rank(vector2))
print('Розмір vec2',vector2.shape)
```

```
tf.Tensor([1. 2. 3. 4.], shape=(4,), dtype=float32)
Ранг vec1 tf.Tensor(1, shape=(), dtype=int32)
Розмір vec1 (4,)
tf.Tensor([10. 20. 30. 40. 50.], shape=(5,), dtype=float32)
Ранг vec2 tf.Tensor(1, shape=(), dtype=int32)
Розмір vec2 (5,)
```

```
matrix1 = tf.constant([[1, 2, 3, 4], [3, 4, 5, 6], [6, 7, 8, 9]],
dtype=tf.float64)
matrix2 = tf.constant([[1, 2, 3], [3, 4, 5], [6, 7, 8], [7, 8, 9]],
dtype=tf.float64)
```

```
print(matrix1)
print('Ранг matr1',tf.rank(matrix1))
print('Розмір matr1',matrix1.shape)
print(matrix2)
print('Ранг matr2',tf.rank(matrix2))
print('Розмір matr2',matrix2.shape)
```

```
tf.Tensor(
[[1. 2. 3. 4.]
 [3. 4. 5. 6.]
 [6. 7. 8. 9.]], shape=(3, 4), dtype=float64)
Ранг matr1 tf.Tensor(2, shape=(), dtype=int32)
Розмір matr1 (3, 4)
tf.Tensor(
[[1. 2. 3.]
 [3. 4. 5.]
 [6. 7. 8.]
 [7. 8. 9.]], shape=(4, 3), dtype=float64)
Ранг matr2 tf.Tensor(2, shape=(), dtype=int32)
Розмір matr2 (4, 3)
```

## Приклади створення вареативних тензорів

```
vmatr1 = tf.Variable([[1, 2, 3, 4], [3, 4, 5, 6], [6, 7, 8, 9]],
dtype=tf.float64)
vmatr2 = tf.Variable([[1, 2, 3], [3, 4, 5], [6, 7, 8], [7, 8, 9]],
dtype=tf.int32)
```

```
print(vmatr1)
print('Ранг vmatr1',tf.rank(vmatr1))
print('Розмір vmatr1',vmatr1.shape)
print(vmatr2)
print('Ранг vmatr2',tf.rank(vmatr2))
print('Розмір vmatr2',vmatr2.shape)
```

```
<tf.Variable 'Variable:0' shape=(3, 4) dtype=float64, numpy=
array([[1., 2., 3., 4.],
       [3., 4., 5., 6.],
       [6., 7., 8., 9.]])>
Ранг vmatr1 tf.Tensor(2, shape=(), dtype=int32)
Розмір vmatr1 (3, 4)
<tf.Variable 'Variable:0' shape=(4, 3) dtype=int32, numpy=
array([[1, 2, 3],
       [3, 4, 5],
       [6, 7, 8],
       [7, 8, 9]], dtype=int32)>
Ранг vmatr2 tf.Tensor(2, shape=(), dtype=int32)
Розмір vmatr2 (4, 3)
```

## Атрибути тензору

```
tensor_rank_4 = tf.zeros([3, 2, 4, 5])
```

```
print("Type of every element:", tensor_rank_4.dtype)
print("Number of axes:", tensor_rank_4.ndim)
print("Shape of tensor:", tensor_rank_4.shape)
print("Elements along axis 0 of tensor:", tensor_rank_4.shape[0])
print("Elements along the last axis of tensor:", tensor_rank_4.shape[-1])
print("Total number of elements (3*2*4*5): ", tf.size(tensor_rank_4).numpy())
```

```
Type of every element: <dtype: 'float32'>
Number of axes: 4
Shape of tensor: (3, 2, 4, 5)
Elements along axis 0 of tensor: 3
Elements along the last axis of tensor: 5
Total number of elements (3*2*4*5):  120
```

```
tensor_rank_4 = tf.ones([3, 2, 4, 5])
```

```
print(tensor_rank_4[0:1].numpy())
```

```
[[[[1. 1. 1. 1. 1.]
   [1. 1. 1. 1. 1.]
   [1. 1. 1. 1. 1.]
   [1. 1. 1. 1. 1.]]

  [[1. 1. 1. 1. 1.]
   [1. 1. 1. 1. 1.]
   [1. 1. 1. 1. 1.]
   [1. 1. 1. 1. 1.]]]]
```

```
tnrank4 = tf.constant([[[1,2,3],[4,5,6],[7,8,9],[10,11,12]],
  [[13,14,15],[16,17,18],[19,20,21],[22,23,24]],
  [[25,26,27],[28,29,30],[31,32,33],[34,35,36]]],dtype=tf.int8)
```

```
print(tnrank4[:,1,2].numpy())
```

```
[ 6 18 30]
```