

# Constants and variables

Puteaux, Fall/Winter 2020-2021

```
#####  
##                               ##  
## Deep Learning in Python ##  
##                               ##  
#####
```

## §2 Introduction to TensorFlow in Python

### §2.1 Introduction to TensorFlow

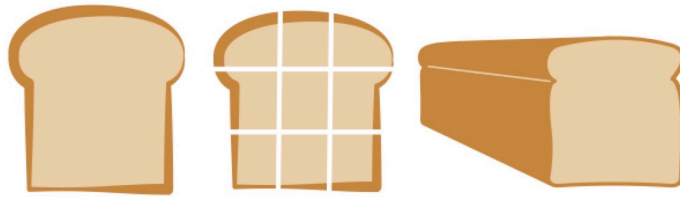
## 1 Constants and variables

### 1.1 What is TensorFlow?

- An open-source library for graph-based numerical computation:
  - developed by the Google Brain team
- Has both low and high-level APIs:
  - can be performed for addition, multiplication, differentiation
  - can be used to design and train machine learning models
- Important changes in TensorFlow 2.0:
  - eager execution is now available by default, which allows users to write simple and more intuitive code
  - model building is now centered around high-level APIs Keras and Estimators

### 1.2 What is a tensor?

- It is a generalization of vectors and matrices to potentially higher dimensions.
- It is a collection of numbers, which is arranged into a specific shape.



Source: Public Domain Vectors

### 1.3 Code of defining tensors in TensorFlow:

```
[1]: import tensorflow as tf

# 0D Tensor
d0 = tf.ones((1, ))

d0
```

```
[1]: <tf.Tensor: shape=(1,), dtype=float32, numpy=array([1.], dtype=float32)>
```

```
[2]: # 1D Tensor
d1 = tf.ones((2, ))

d1
```

```
[2]: <tf.Tensor: shape=(2,), dtype=float32, numpy=array([1., 1.], dtype=float32)>
```

```
[3]: # 2D Tensor
d2 = tf.ones((2, 2))

d2
```

```
[3]: <tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[1., 1.],
       [1., 1.]], dtype=float32)>
```

```
[4]: # 3D Tensor
d3 = tf.ones((2, 2, 2))

d3
```

```
[4]: <tf.Tensor: shape=(2, 2, 2), dtype=float32, numpy=
array([[[1., 1.],
       [1., 1.]],
      [[1., 1.],
       [1., 1.]]], dtype=float32)>
```

```
[5]: # Print the 3D tensor
print(d3.numpy())
```

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

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

## 1.4 How to define constants in TensorFlow?

- A constant is the simplest category of tensor:
  - cannot be changed and not trainable
  - can have any dimension

## 1.5 Code of defining constants in TensorFlow:

```
[6]: from tensorflow import constant

# Define a 2x3 constant.
a = constant(3, shape=[2, 3])

a
```

```
[6]: <tf.Tensor: shape=(2, 3), dtype=int32, numpy=
array([[3, 3, 3],
       [3, 3, 3]], dtype=int32)>
```

```
[7]: # Define a 2x2 constant.
b = constant([1, 2, 3, 4], shape=[2, 2])

b
```

```
[7]: <tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[1, 2],
       [3, 4]], dtype=int32)>
```

## 1.6 How to use convenience functions to define constants?

Operation	Example
<code>tf.constant()</code>	<code>constant([1, 2, 3])</code>
<code>tf.zeros()</code>	<code>zeros([2, 2])</code>
<code>tf.zeros_like()</code>	<code>zeros_like(input_tensor)</code>
<code>tf.ones()</code>	<code>ones([2, 2])</code>
<code>tf.ones_like()</code>	<code>ones_like(input_tensor)</code>
<code>tf.fill()</code>	<code>fill([3, 3], 7)</code>

## 1.7 Code of defining and initializing variables:

```
[8]: import tensorflow as tf
```

```
# Define a variable
```

```
a0 = tf.Variable([1, 2, 3, 4, 5, 6], dtype=tf.float32)
```

```
a1 = tf.Variable([1, 2, 3, 4, 5, 6], dtype=tf.int16)
```

```
a0, a1
```

```
[8]: (<tf.Variable 'Variable:0' shape=(6,) dtype=float32, numpy=array([1., 2., 3., 4., 5., 6.], dtype=float32)>,  
      <tf.Variable 'Variable:0' shape=(6,) dtype=int16, numpy=array([1, 2, 3, 4, 5, 6], dtype=int16)>)
```

```
[9]: # Define a constant
```

```
b = tf.constant(2, tf.float32)
```

```
b
```

```
[9]: <tf.Tensor: shape=(), dtype=float32, numpy=2.0>
```

```
[10]: # Compute their product
```

```
c0 = tf.multiply(a0, b)
```

```
c1 = a0 * b
```

```
c0, c1
```

```
[10]: (<tf.Tensor: shape=(6,), dtype=float32, numpy=array([ 2.,  4.,  6.,  8., 10., 12.], dtype=float32)>,  
      <tf.Tensor: shape=(6,), dtype=float32, numpy=array([ 2.,  4.,  6.,  8., 10., 12.], dtype=float32)>)
```

## 1.8 Practice exercises for constants and variables:

### ► Package pre-loading:

```
[11]: import pandas as pd
import numpy as np
```

### ► Data pre-loading:

```
[12]: df = pd.read_csv('ref3. UCI credit card.csv', dtype=np.float64)
credit_numpy = df[['EDUCATION', 'MARRIAGE', 'AGE', 'BILL_AMT1']].to_numpy()
```

### ► Constants defining practice:

```
[13]: # Import constant from TensorFlow
from tensorflow import constant

# Convert the credit_numpy array into a tensorflow constant
credit_constant = constant(credit_numpy)

# Print constant datatype
print('The datatype is:', credit_constant.dtype)

# Print constant shape
print('The shape is:', credit_constant.shape)
```

```
The datatype is: <dtype: 'float64'>
The shape is: (30000, 4)
```

### ► Variables defining practice:

```
[14]: import tensorflow as tf

# Define the 1-dimensional variable A1
A1 = tf.Variable([1, 2, 3, 4])

# Print the variable A1
print(A1)

# Convert A1 to a numpy array and assign it to B1
B1 = A1.numpy()

# Print B1
print(B1)
```

```
<tf.Variable 'Variable:0' shape=(4,) dtype=int32, numpy=array([1, 2, 3, 4],
dtype=int32)>
[1 2 3 4]
```