Interactive
Technologies Institute
LARSyS

arditi agência regional para o desenvolvimento da investigação tecnologia e inovação

UNIVERSIDADE da MADEIRA

# MADEIRA INTERNATIONAL WORKSHOP IN MACHINE LEARNING

IKnowD

FLOATINGPARTICLE
RESEARCH & DEVELOPMENT

ams OSRAM

asseco
PST

SEEHEALTH

Automatisierung
Bildverarbeitung
Software
ABS

2021

# WHAT IS TENSORFLOW?

# WHAT IS TENSORFLOW?

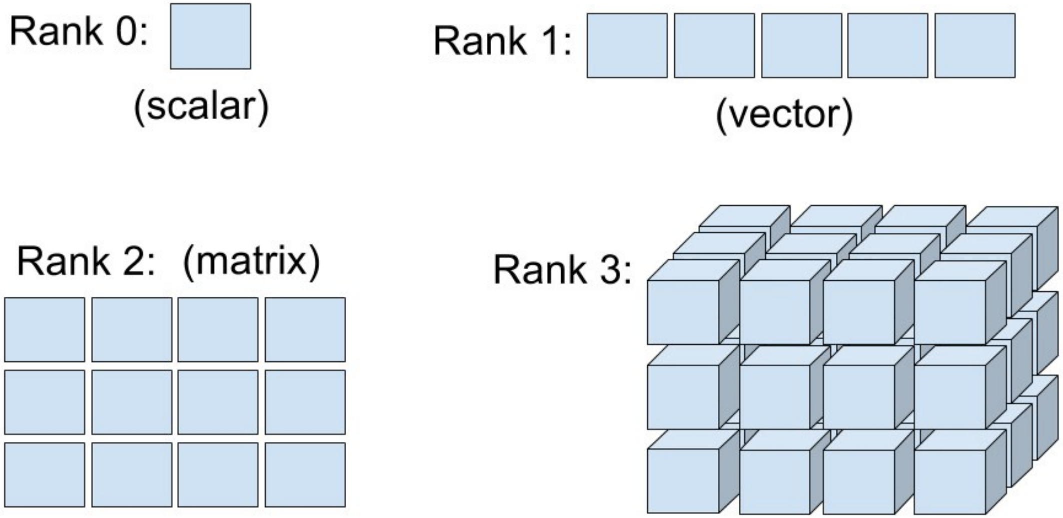
The TensorFlow logo and any related marks are trademarks of Google Inc..

# WHAT IS TENSORFLOW?

**Open source platform for numerical computation**

**Computations based on tensors**

- Tensors are multi-dimensional arrays with a uniform type



# WHAT IS TENSORFLOW?

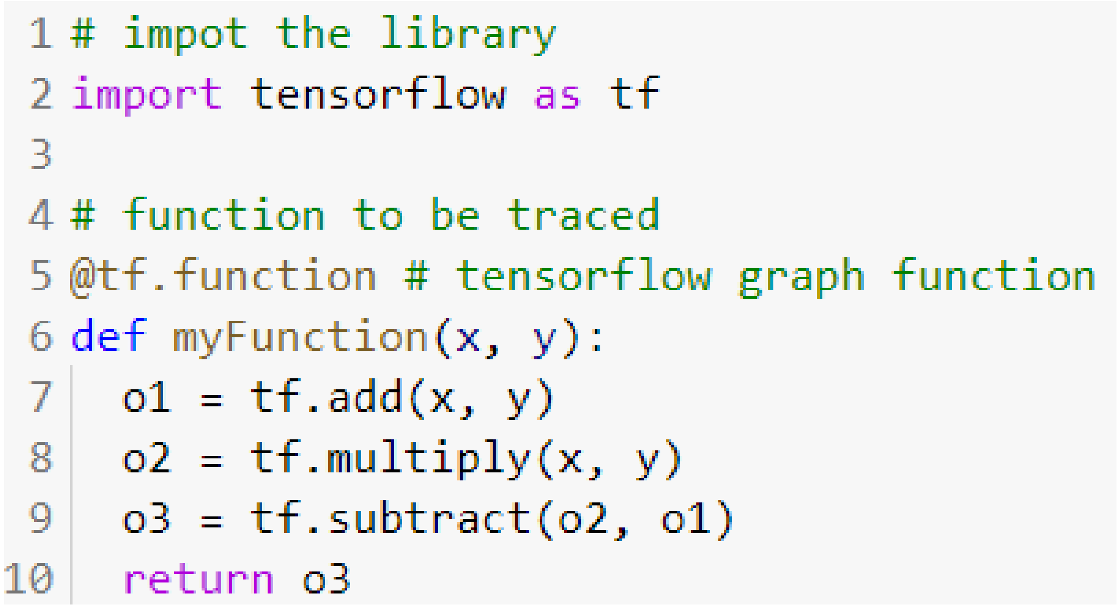
**Concept:**

- Create a directed graph to represent the computation
- Mathematical operations are represented by nodes
- Edges represent the data flow between nodes (the tensors)

# WHAT IS TENSORFLOW?

## Simple example

- Compute: z = (x × y) - (x + y)

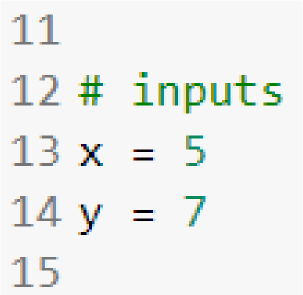


```python
# impot the library
import tensorflow as tf

# function to be traced
@tf.function # tensorflow graph function
def myFunction(x, y):
  o1 = tf.add(x, y)
  o2 = tf.multiply(x, y)
  o3 = tf.subtract(o2, o1)
  return o3
```

# WHAT IS TENSORFLOW?

## Simple example

- With  x = 5 and y = 7

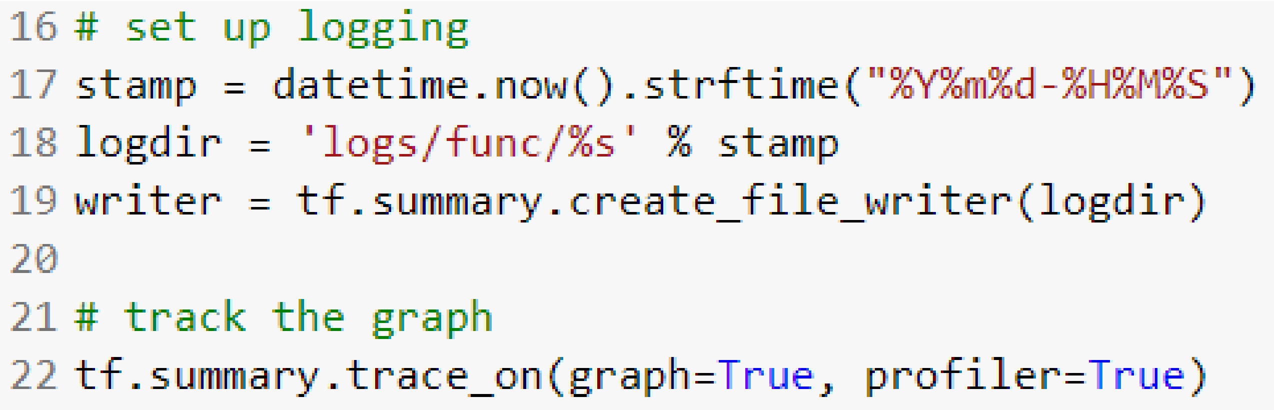


```

# inputs
x = 5
y = 7

```

# WHAT IS TENSORFLOW?

## Simple example

- Track the graph

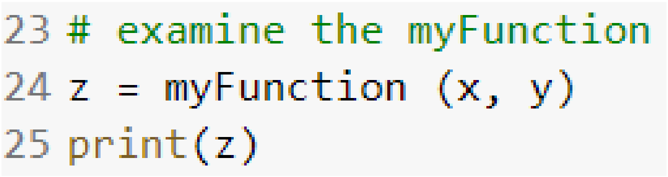


```
# set up logging
stamp = datetime.now().strftime("%Y%m%d-%H%M%S")
logdir = 'logs/func/%s' % stamp
writer = tf.summary.create_file_writer(logdir)

# track the graph
tf.summary.trace_on(graph=True, profiler=True)
```

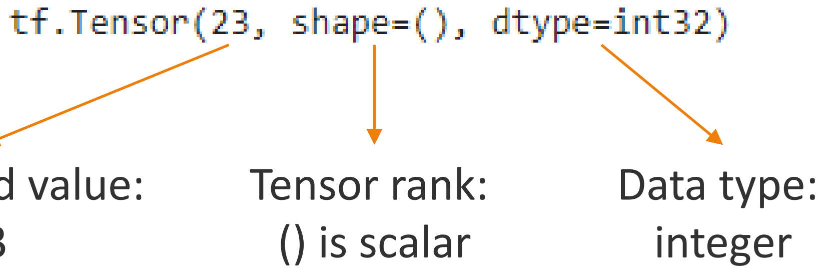
# WHAT IS TENSORFLOW?

## Simple example

- Estimate z (it is (5 × 7) - (5 + 7) = 23)



```
# examine the myFunction
z = myFunction (x, y)
print(z)
```



```
tf.Tensor(23, shape=(), dtype=int32)
```

Estimated value:
23

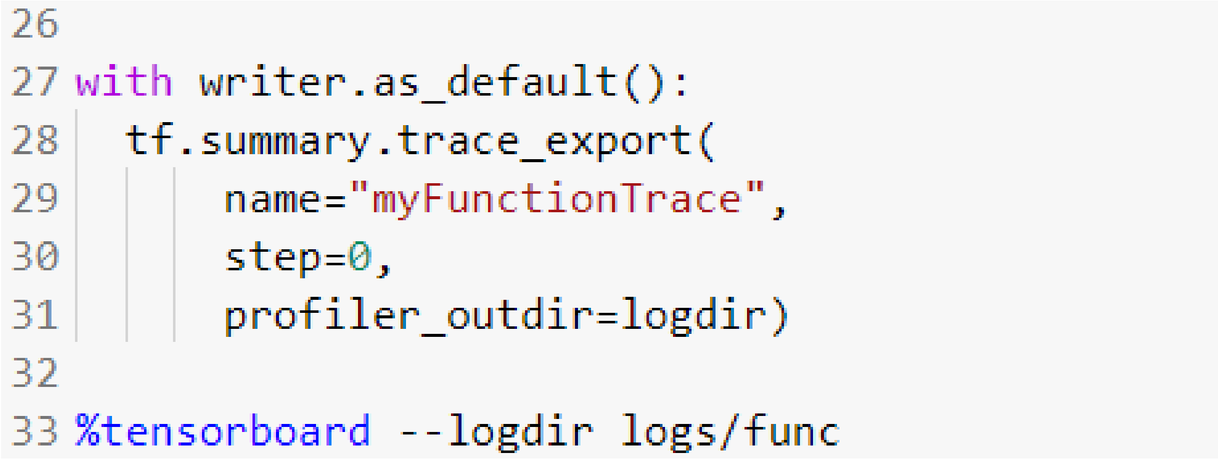
Tensor rank:
() is scalar

Data type:
integer

# WHAT IS TENSORFLOW?

### Simple example

- Check TensorBoard

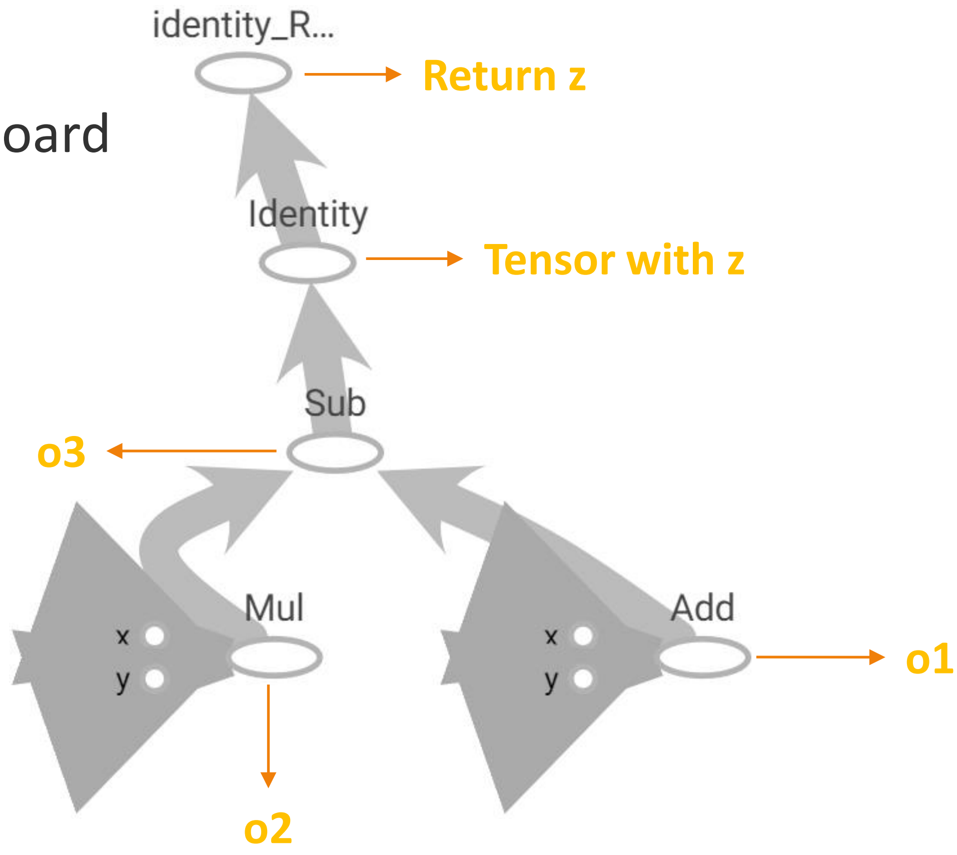


```

with writer.as_default():
  tf.summary.trace_export(
      name="myFunctionTrace",
      step=0,
      profiler_outdir=logdir)

%tensorboard --logdir logs/func
```

# WHAT IS TENSORFLOW?

**Simple example**

- Check TensorBoard



# WHAT IS TENSORFLOW?

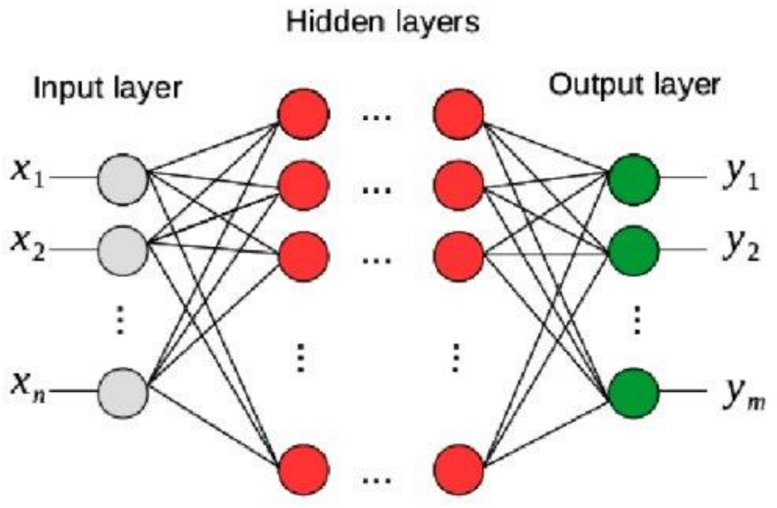
**Tensors flow trough the graph,**

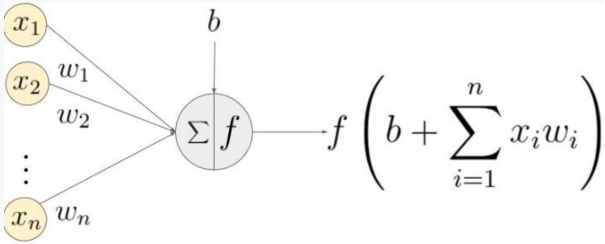
**thus the name TensorFlow**

# WHAT IS TENSORFLOW?

## How to create a neural network:

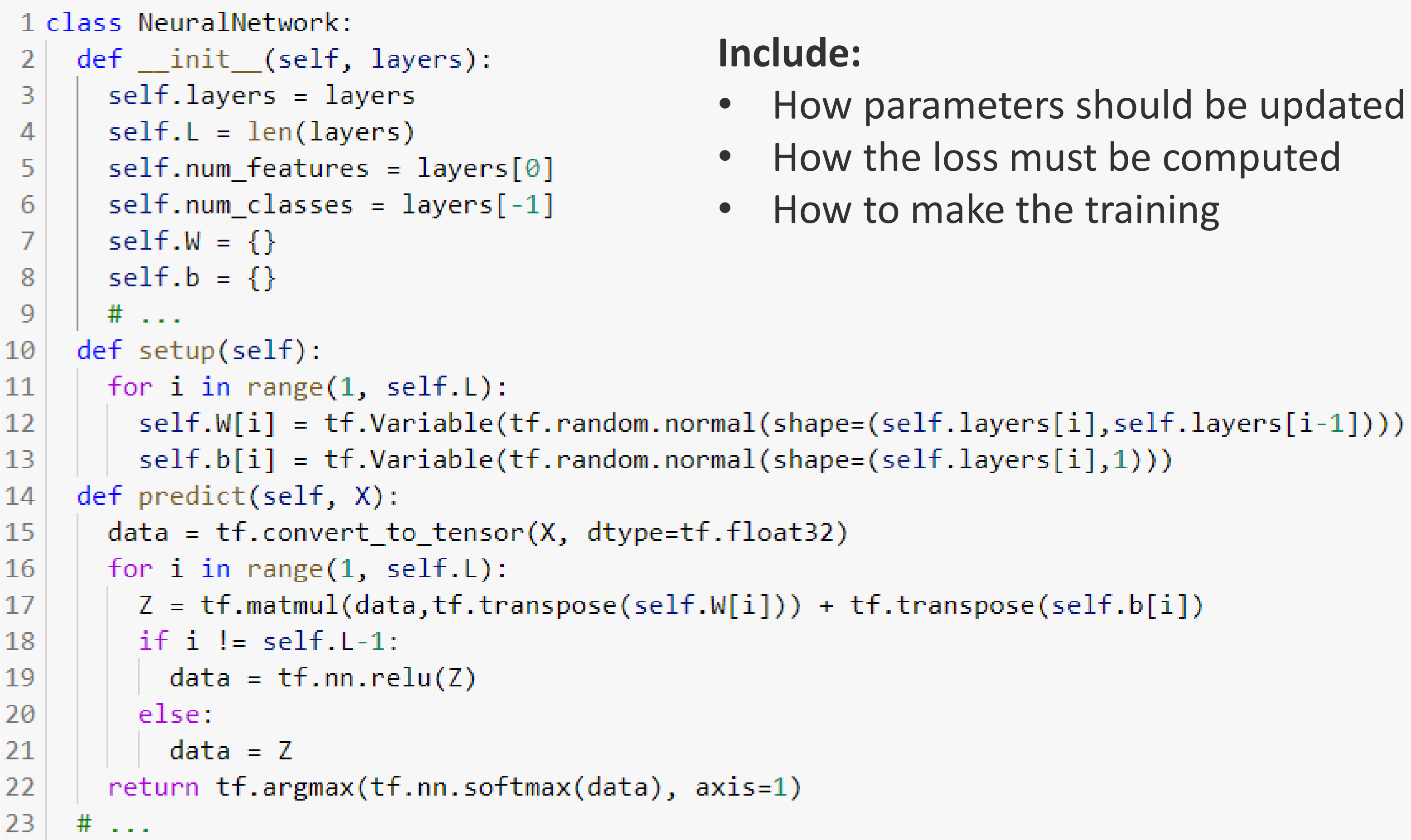
```
class NeuralNetwork:
  def __init__(self, layers):
    self.layers = layers
    self.L = len(layers)
    self.num_features = layers[0]
    self.num_classes = layers[-1]
    self.W = {}
    self.b = {}
    # ...
  def setup(self):
    for i in range(1, self.L):
      self.W[i] = tf.Variable(tf.random.normal(shape=(self.layers[i],self.layers[i-1])))
      self.b[i] = tf.Variable(tf.random.normal(shape=(self.layers[i],1)))
  def predict(self, X):
    data = tf.convert_to_tensor(X, dtype=tf.float32)
    for i in range(1, self.L):
      Z = tf.matmul(data,tf.transpose(self.W[i])) + tf.transpose(self.b[i])
      if i != self.L-1:
        data = tf.nn.relu(Z)
      else:
        data = Z
    return tf.argmax(tf.nn.softmax(data), axis=1)
  # ...
```





# WHAT IS TENSORFLOW?

## How to create a neural network:



```python
class NeuralNetwork:
  def __init__(self, layers):
    self.layers = layers
    self.L = len(layers)
    self.num_features = layers[0]
    self.num_classes = layers[-1]
    self.W = {}
    self.b = {}
    # ...
  def setup(self):
    for i in range(1, self.L):
      self.W[i] = tf.Variable(tf.random.normal(shape=(self.layers[i],self.layers[i-1])))
      self.b[i] = tf.Variable(tf.random.normal(shape=(self.layers[i],1)))
  def predict(self, X):
    data = tf.convert_to_tensor(X, dtype=tf.float32)
    for i in range(1, self.L):
      Z = tf.matmul(data,tf.transpose(self.W[i])) + tf.transpose(self.b[i])
      if i != self.L-1:
        data = tf.nn.relu(Z)
      else:
        data = Z
    return tf.argmax(tf.nn.softmax(data), axis=1)
  # ...
```

**Include:**

- How parameters should be updated
- How the loss must be computed
- How to make the training

# WHAT IS TENSORFLOW?

**Is there any alternative?**

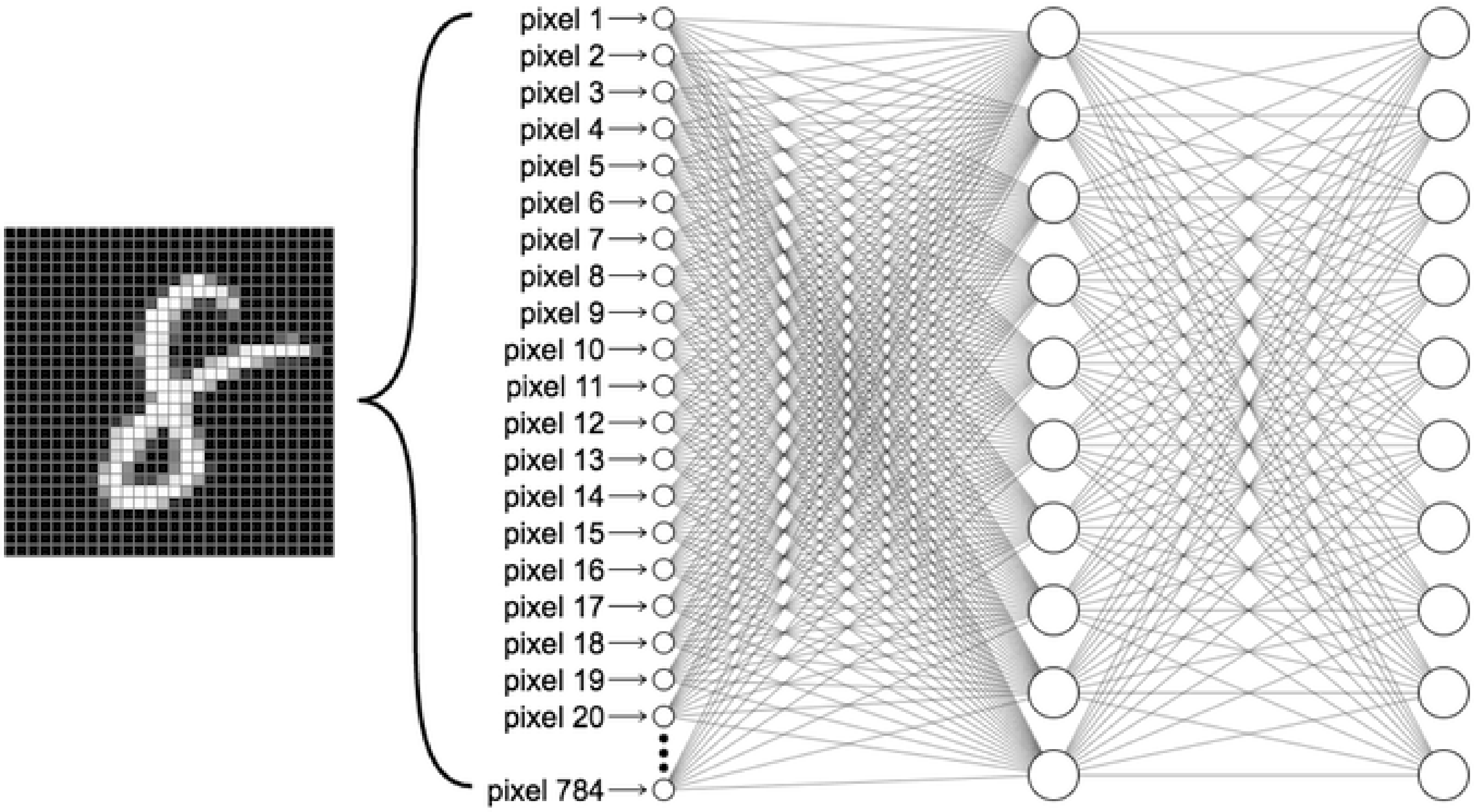
# WHAT IS TENSORFLOW?

**Use Keras API:**

“API designed for human beings, not machines”

# WHAT IS TENSORFLOW?

**Methods to build a Keras model in TensorFlow:**



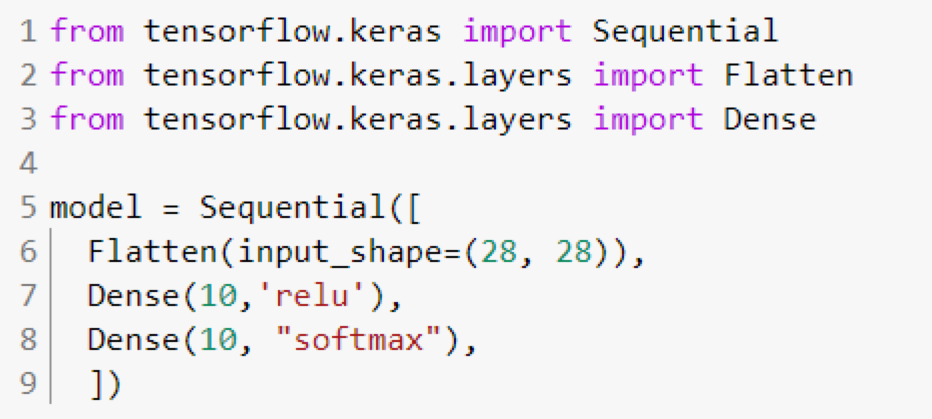
10

10

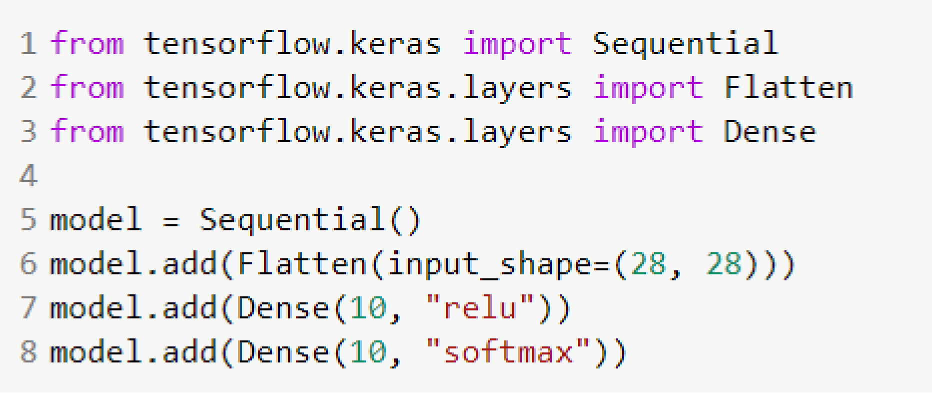
# WHAT IS TENSORFLOW?

**Methods to build a Keras model in TensorFlow:**

- Sequential API



```python
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense

model = Sequential([
  Flatten(input_shape=(28, 28)),
  Dense(10,'relu'),
  Dense(10, "softmax"),
  ])
```

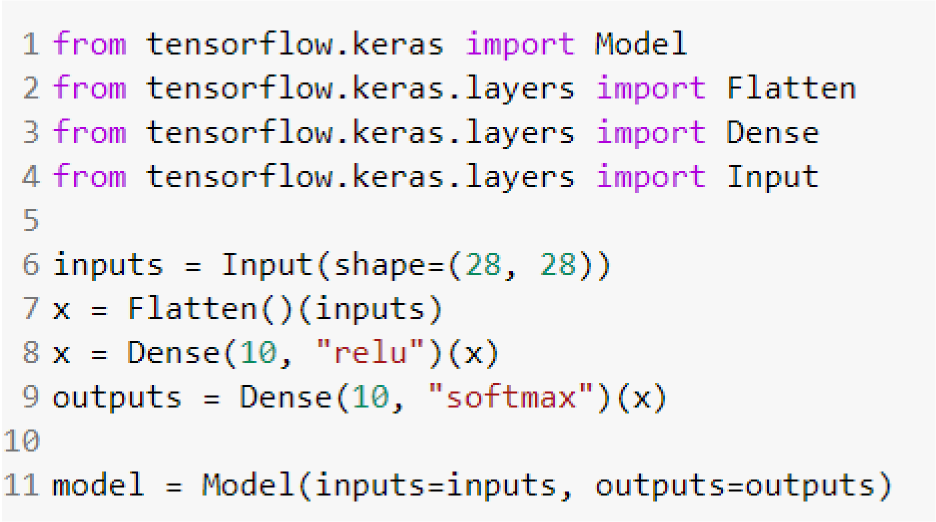


```python
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense

model = Sequential()
model.add(Flatten(input_shape=(28, 28)))
model.add(Dense(10, "relu"))
model.add(Dense(10, "softmax"))
```

# WHAT IS TENSORFLOW?

**Methods to build a Keras model in TensorFlow:**

- Functional API

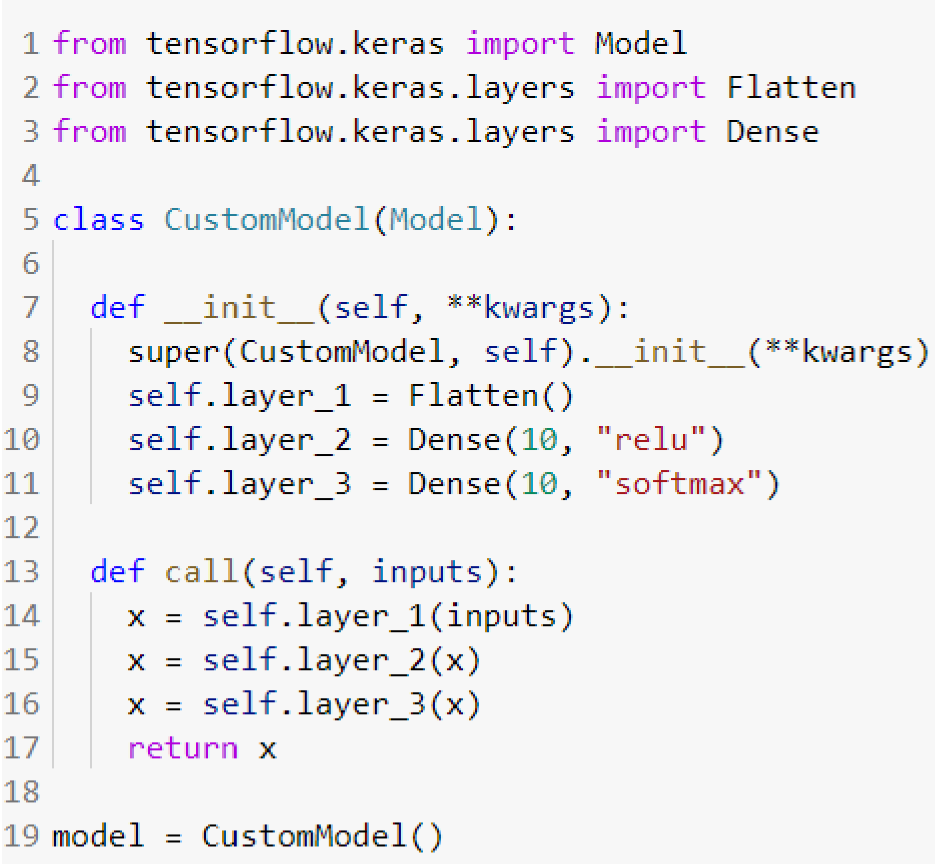


```python
from tensorflow.keras import Model
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input

inputs = Input(shape=(28, 28))
x = Flatten()(inputs)
x = Dense(10, "relu")(x)
outputs = Dense(10, "softmax")(x)

model = Model(inputs=inputs, outputs=outputs)
```

# WHAT IS TENSORFLOW?

**Methods to build a Keras model in TensorFlow:**

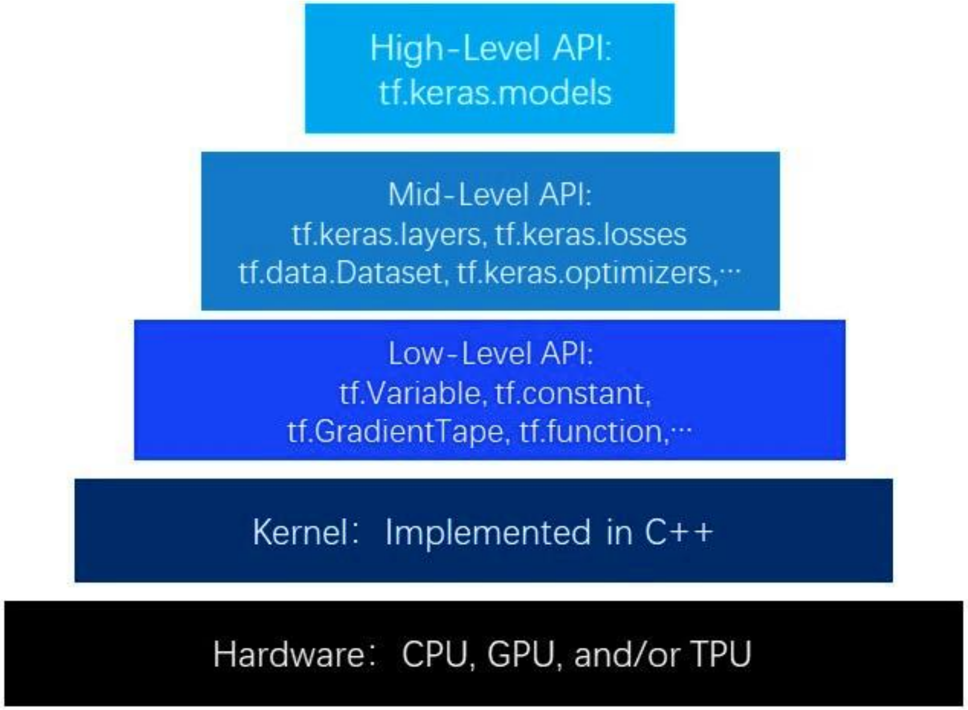
- Model Subclassing



```
from tensorflow.keras import Model
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense

class CustomModel(Model):

  def __init__(self, **kwargs):
    super(CustomModel, self).__init__(**kwargs)
    self.layer_1 = Flatten()
    self.layer_2 = Dense(10, "relu")
    self.layer_3 = Dense(10, "softmax")

  def call(self, inputs):
    x = self.layer_1(inputs)
    x = self.layer_2(x)
    x = self.layer_3(x)
    return x

model = CustomModel()
```

# WHAT IS TENSORFLOW?

**Keras was defined the standard high-level API of TensorFlow 2**



High-Level API:
tf.keras.models

Mid-Level API:
tf.keras.layers, tf.keras.losses
tf.data.Dataset, tf.keras.optimizers,···

Low-Level API:
tf.Variable, tf.constant,
tf.GradientTape, tf.function,···

Kernel: Implemented in C++

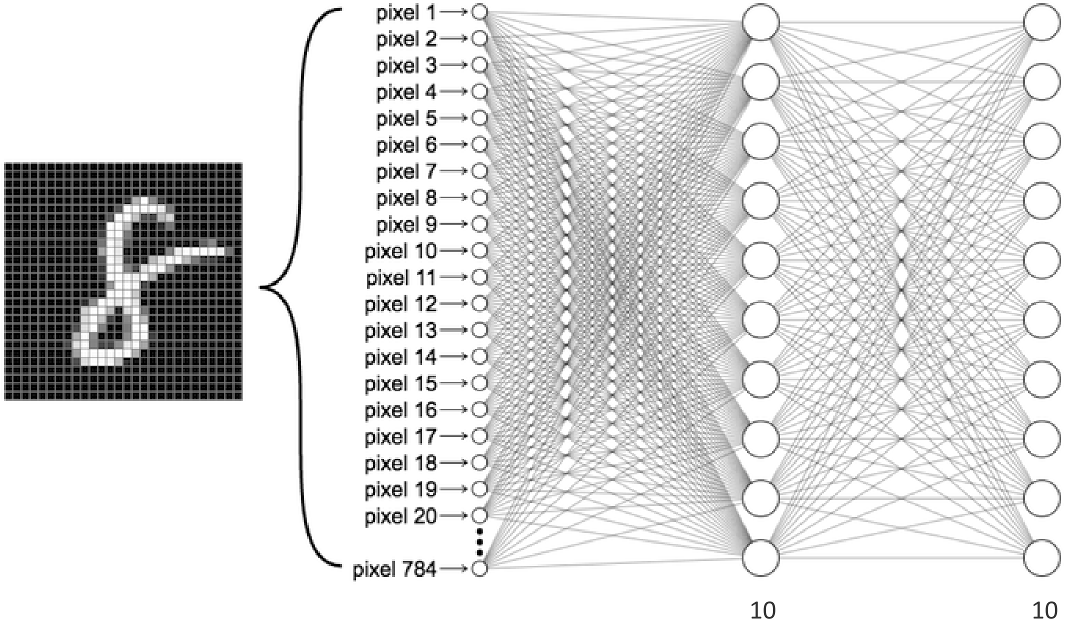
Hardware: CPU, GPU, and/or TPU

# PRACTICAL EXAMPLE

# PRACTICAL EXAMPLE

**FFNN Colab example: FFNN – MNIST**

- Handwritten dataset
- 70000 images
- All are 28x28
- 784 pixels in total



# WHY TENSORFLOW

# WHY TENSORFLOW

**Key advantages:**

- Allow the visualization of the computational graphs (nor possible in Scikit learn)
- Can be used from beginners (with simple to use interfaces) to researchers (with the high customization capabilities)
- Probably the easiest to use for deep learning
- Great portability (deploy on multiple devices)
- Easy to train in GPU and TPU
- Open-source backed by a large community (all the way from software engineers to data scientist)

# WHAT IS TENSORFLOW?

| Software | Open source | Written in | Interface | CUDA support | Pretrained models | Recurrent nets | Convolutional nets | Actively developed |
|---|---|---|---|---|---|---|---|---|
| Caffe | Yes | C++ | Python, MATLAB, C++ | Yes | Yes | Yes | Yes | No |
| Chainer | Yes | Python | Python | Yes | Yes | Yes | Yes | No |
| Deeplearning4j | Yes | C++, Java | Java, Scala, Clojure, Python (Keras), Kotlin | Yes | Yes | Yes | Yes | Yes |
| Flux | Yes | Julia | Julia | Yes | Yes | Yes | Yes | Yes |
| **Keras** | **Yes** | **Python** | **Python, R** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |
| MATLAB + Deep Learning Toolbox | No | C, C++, Java, MATLAB | MATLAB | Yes | Yes | Yes | Yes | Yes |
| Microsoft Cognitive Toolkit (CNTK) | Yes | C++ | Python (Keras), C++, Command line | Yes | Yes | Yes | Yes | No |
| Apache MXNet | Yes | C++ | C++, Python, Julia, Matlab, JavaScript | Yes | Yes | Yes | Yes | Yes |
| PlaidML | Yes | Python, C++, OpenCL | Python, C++ | No | Yes | Yes | Yes | Yes |
| **PyTorch** | **Yes** | **Python, C, C++, CUDA** | **Python, C++, Julia** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |
| Seq2SeqSharp | Yes | C#, C, C++, CUDA | C# | Yes | Yes | Yes | No | Yes |
| **TensorFlow** | **Yes** | **C++, Python, CUDA** | **Python, C/C++, Java, Go, JavaScript** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |
| Theano | Yes | Python | Python | Yes | With Lasagne | Yes | Yes | No |
| Wolfram Mathematica | No | C++, Wolfram Language, CUDA | Wolfram Language | Yes | Yes | Yes | Yes | Yes |

# SOURCES

# SOURCES

- http://clipart-library.com/
- https://medium.com/@schartz/the-shape-of-tensor-bab75001d7bc
- https://lyhue1991.github.io/eat_tensorflow2_in_30_days/english/Chapter3/
- https://zitaoshen.rbind.io/project/machine_learning/how-to-build-your-own-neural-net-from-the-scrach/
- https://en.wikipedia.org/w/index.php?title=Comparison_of_deep-learning_software&action=edit&section=1
- https://www.researchgate.net/publication/319901002_NeuRoute_Predictive_Dynamic_Routing_for_Software-Defined_Networks/figures
- https://abhigoku10.medium.com/activation-functions-and-its-types-in-artifical-neural-network-14511f3080a8