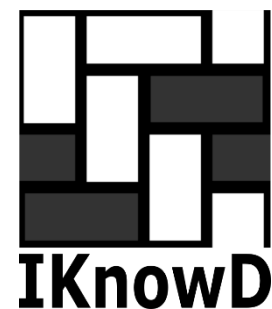




# MADEIRA INTERNATIONAL WORKSHOP IN MACHINE LEARNING



2021

# WHAT IS TENSORFLOW?

# WHAT IS TENSORFLOW?




**TensorFlow**


# WHAT IS TENSORFLOW?

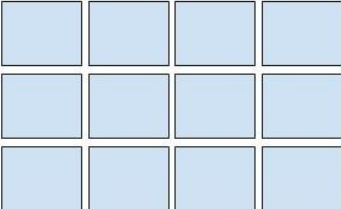
Open source platform for numerical computation

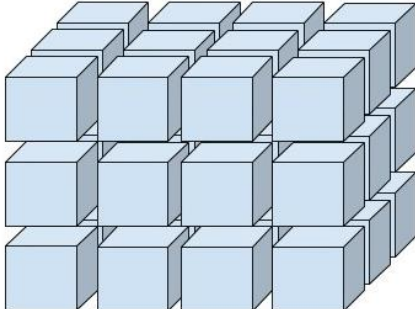
Computations based on **tensors**

- Tensors are multi-dimensional arrays with a uniform type

Rank 0:   
(scalar)

Rank 1:   
(vector)

Rank 2: (matrix)  


Rank 3: 

# 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:  $z = (x \times y) - (x + y)$

```
1 # import the library
2 import tensorflow as tf
3
4 # function to be traced
5 @tf.function # tensorflow graph function
6 def myFunction(x, y):
7     o1 = tf.add(x, y)
8     o2 = tf.multiply(x, y)
9     o3 = tf.subtract(o2, o1)
10    return o3
```

# WHAT IS TENSORFLOW?

## Simple example

- With  $x = 5$  and  $y = 7$

```
11  
12 # inputs  
13 x = 5  
14 y = 7  
15
```

# WHAT IS TENSORFLOW?

## Simple example

- Track the graph

```
16 # set up logging
17 stamp = datetime.now().strftime("%Y%m%d-%H%M%S")
18 logdir = 'logs/func/%s' % stamp
19 writer = tf.summary.create_file_writer(logdir)
20
21 # track the graph
22 tf.summary.trace_on(graph=True, profiler=True)
```

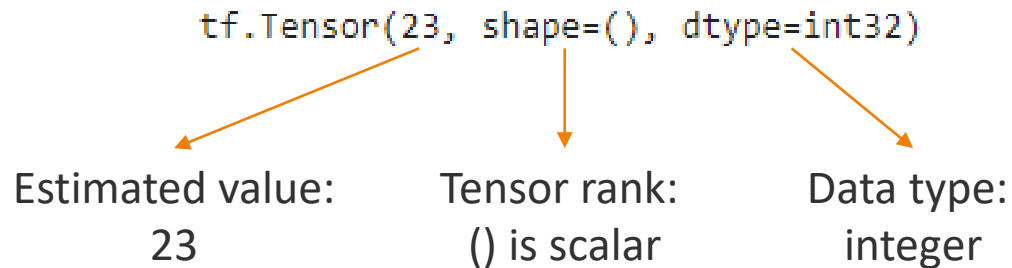


# WHAT IS TENSORFLOW?

## Simple example

- Estimate  $z$  (it is  $(5 \times 7) - (5 + 7) = 23$ )

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



# WHAT IS TENSORFLOW?

## Simple example

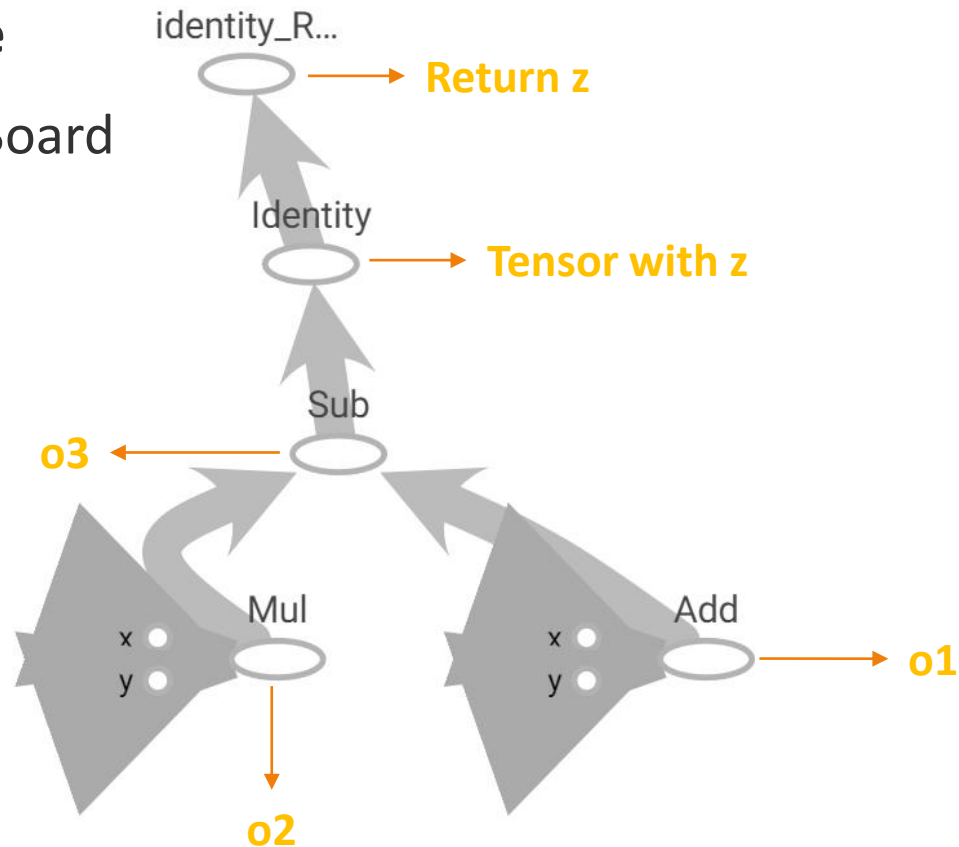
- Check TensorBoard

```
26
27 with writer.as_default():
28     tf.summary.trace_export(
29         name="myFunctionTrace",
30         step=0,
31         profiler_outdir=logdir)
32
33 %tensorboard --logdir logs/func
```

# WHAT IS TENSORFLOW?

## Simple example

- Check TensorBoard



# WHAT IS TENSORFLOW?

**Tensors flow through the graph,  
thus the name TensorFlow**

# WHAT IS TENSORFLOW?

## How to create a neural network:

```
1 class NeuralNetwork:
2     def __init__(self, layers):
3         self.layers = layers
4         self.L = len(layers)
5         self.num_features = layers[0]
6         self.num_classes = layers[-1]
7         self.W = {}
8         self.b = {}
9         # ...
10    def setup(self):
11        for i in range(1, self.L):
12            self.W[i] = tf.Variable(tf.random.normal(shape=(self.layers[i],self.layers[i-1])))
13            self.b[i] = tf.Variable(tf.random.normal(shape=(self.layers[i],1)))
14    def predict(self, X):
15        data = tf.convert_to_tensor(X, dtype=tf.float32)
16        for i in range(1, self.L):
17            Z = tf.matmul(data,tf.transpose(self.W[i])) + tf.transpose(self.b[i])
18            if i != self.L-1:
19                data = tf.nn.relu(Z)
20            else:
21                data = Z
22        return tf.argmax(tf.nn.softmax(data), axis=1)
23    # ...
```

### Include:

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

# WHAT IS TENSORFLOW?

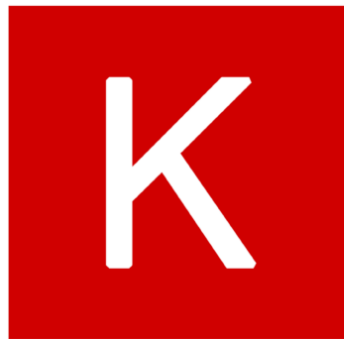
Is there any alternative?



# WHAT IS TENSORFLOW?

**Use Keras API:**

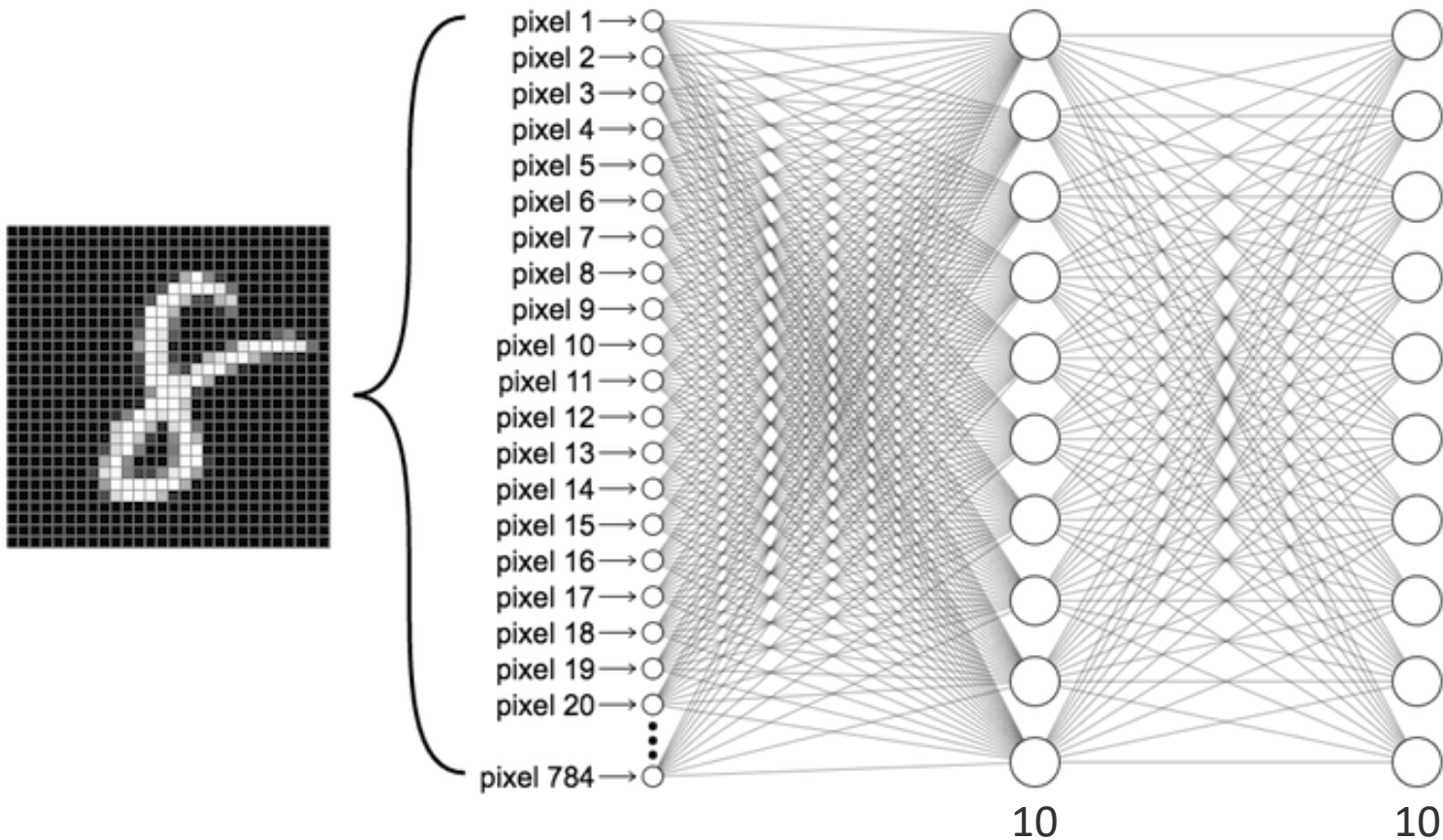
“API designed for human beings, not machines”



# Keras

# WHAT IS TENSORFLOW?

Methods to build a Keras model in TensorFlow:





# WHAT IS TENSORFLOW?

## Methods to build a Keras model in TensorFlow:

- Sequential API

```
1 from tensorflow.keras import Sequential
2 from tensorflow.keras.layers import Flatten
3 from tensorflow.keras.layers import Dense
4
5 model = Sequential([
6     Flatten(input_shape=(28, 28)),
7     Dense(10, 'relu'),
8     Dense(10, "softmax"),
9 ])
```

```
1 from tensorflow.keras import Sequential
2 from tensorflow.keras.layers import Flatten
3 from tensorflow.keras.layers import Dense
4
5 model = Sequential()
6 model.add(Flatten(input_shape=(28, 28)))
7 model.add(Dense(10, "relu"))
8 model.add(Dense(10, "softmax"))
```

# WHAT IS TENSORFLOW?

## Methods to build a Keras model in TensorFlow:

- Functional API

```
1 from tensorflow.keras import Model
2 from tensorflow.keras.layers import Flatten
3 from tensorflow.keras.layers import Dense
4 from tensorflow.keras.layers import Input
5
6 inputs = Input(shape=(28, 28))
7 x = Flatten()(inputs)
8 x = Dense(10, "relu")(x)
9 outputs = Dense(10, "softmax")(x)
10
11 model = Model(inputs=inputs, outputs=outputs)
```

# WHAT IS TENSORFLOW?

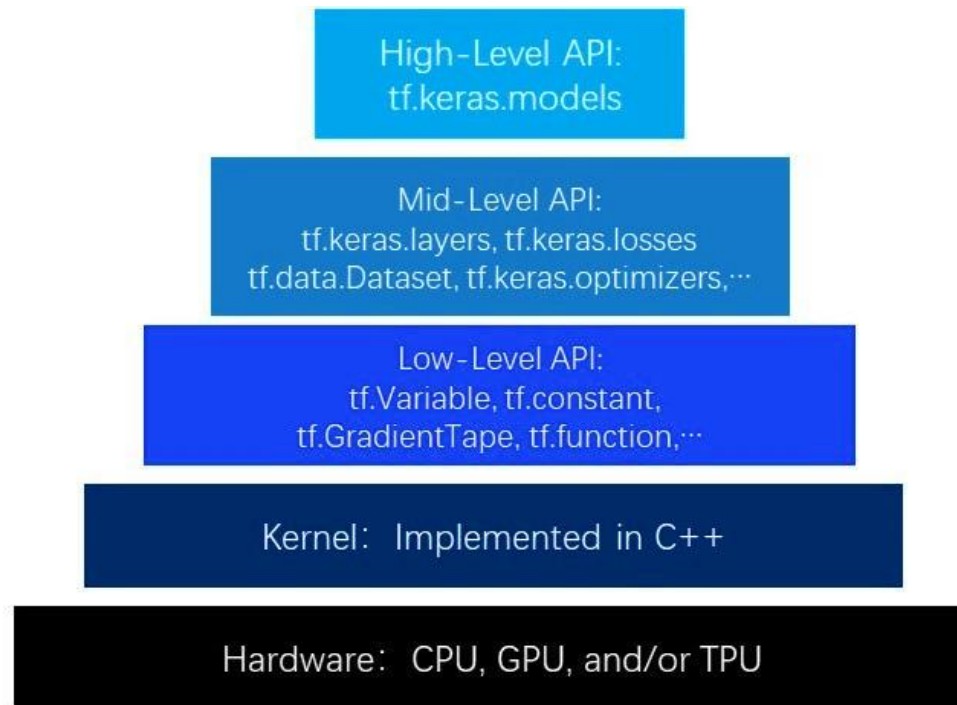
## Methods to build a Keras model in TensorFlow:

- Model Subclassing

```
1 from tensorflow.keras import Model
2 from tensorflow.keras.layers import Flatten
3 from tensorflow.keras.layers import Dense
4
5 class CustomModel(Model):
6
7     def __init__(self, **kwargs):
8         super(CustomModel, self).__init__(**kwargs)
9         self.layer_1 = Flatten()
10        self.layer_2 = Dense(10, "relu")
11        self.layer_3 = Dense(10, "softmax")
12
13    def call(self, inputs):
14        x = self.layer_1(inputs)
15        x = self.layer_2(x)
16        x = self.layer_3(x)
17        return x
18
19 model = CustomModel()
```

# WHAT IS TENSORFLOW?

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

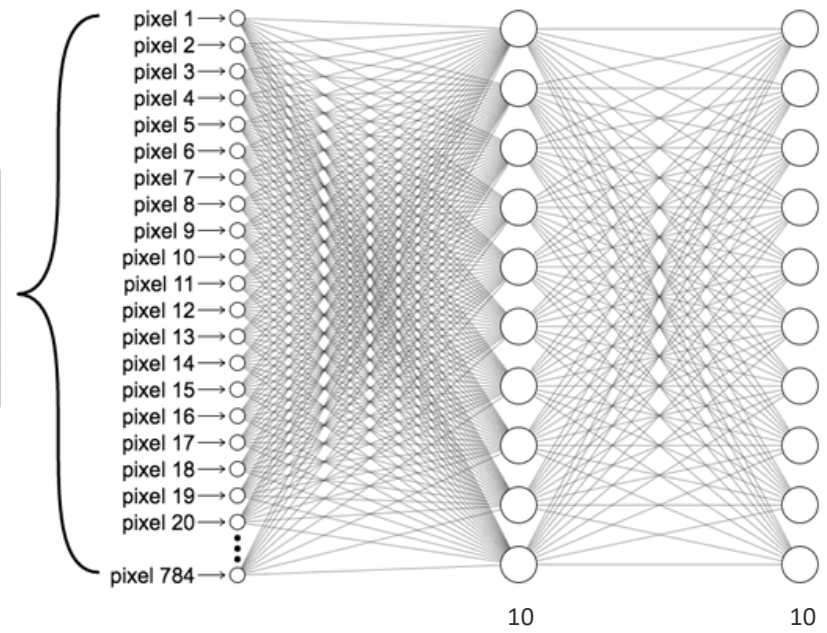
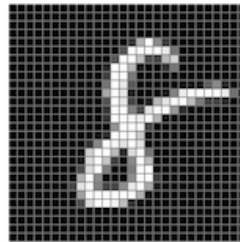


# PRACTICAL EXAMPLE

# 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 (not 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
<b>Keras</b>	<b>Yes</b>	<b>Python</b>	<b>Python, R</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
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
<b>PyTorch</b>	<b>Yes</b>	<b>Python, C, C++, CUDA</b>	<b>Python, C++, Julia</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
Seq2SeqSharp	Yes	C#, C, C++, CUDA	C#	Yes	Yes	Yes	No	Yes
<b>TensorFlow</b>	<b>Yes</b>	<b>C++, Python, CUDA</b>	<b>Python, C/C++, Java, Go, JavaScript</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
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://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-scratch/](https://zitaoshen.rbind.io/project/machine_learning/how-to-build-your-own-neural-net-from-the-scratch/)
- [https://en.wikipedia.org/w/index.php?title=Comparison\\_of\\_deep-learning\\_software&action=edit&section=1](https://en.wikipedia.org/w/index.php?title=Comparison_of_deep-learning_software&action=edit&section=1)