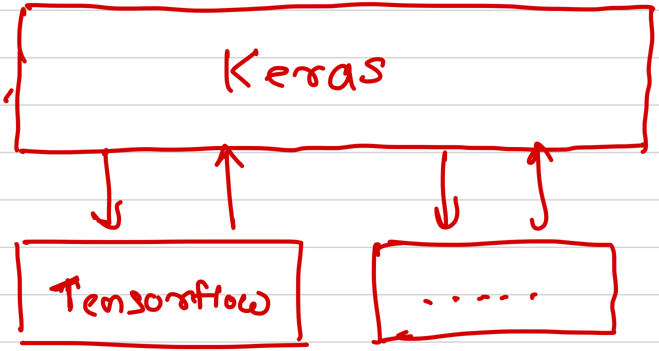
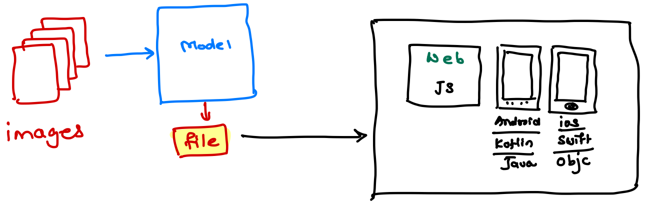
Machine Learning

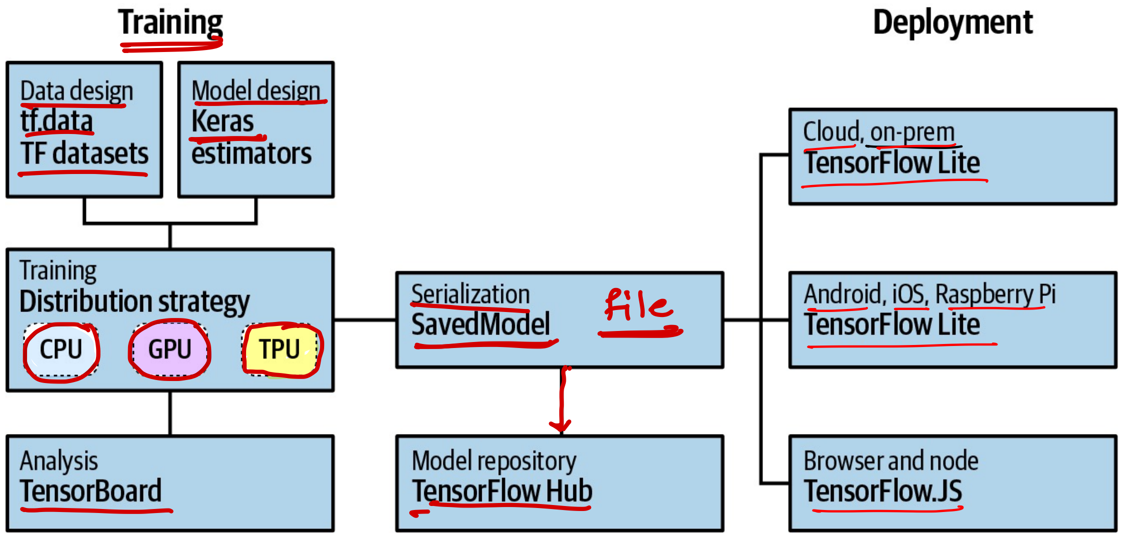
TensorFlow


Keras
Tensorflow
.....

TensorFlow

- TensorFlow is an open source platform for creating and using machine learning models
- It implements many of the common algorithms and patterns needed for machine learning, saving you from needing to learn all the underlying math and logic and enabling you to just to focus on your scenario
- It's aimed at everyone from hobbyists, to professional developers, to researchers pushing the boundaries of artificial intelligence
- Importantly, it also supports deployment of models to the web, cloud, mobile, and embedded systems





Training

Data design
tf.data
TF datasets

Model design
Keras
estimators

Training
Distribution strategy
CPU GPU TPU

Analysis
TensorBoard

Serialization
SavedModel

Model repository
TensorFlow Hub

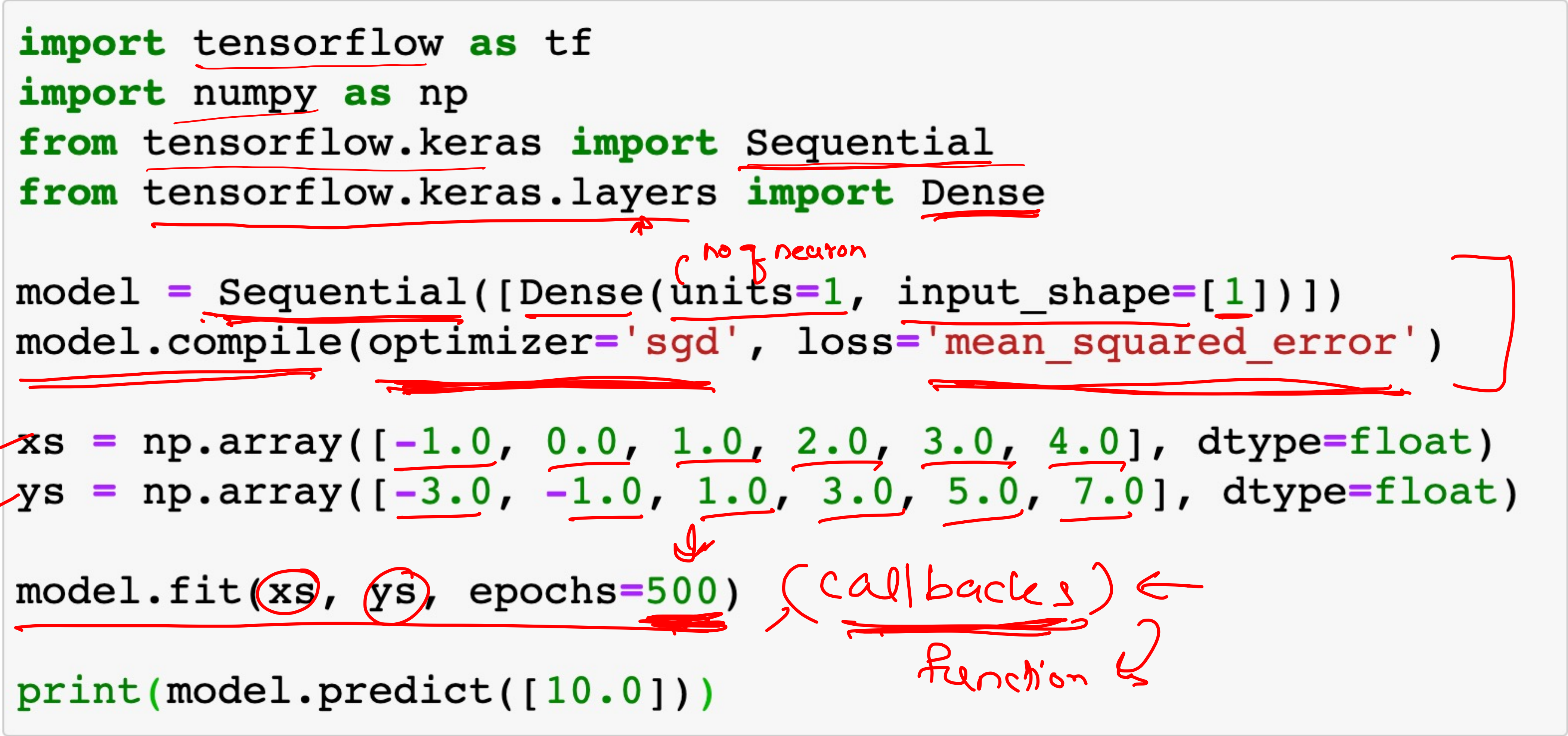
Deployment

Cloud, on-prem
TensorFlow Lite

Android, iOS, Raspberry Pi
TensorFlow Lite

Browser and node
TensorFlow.JS

Hello World



```
import tensorflow as tf
import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

no of neuron

(callbacks)

function

x	y
-1	-3
0	-1
1	1
2	3
3	5
4	7
10	?

$$\boxed{y = 2x - 1}$$

epoch = 1

$$\underline{y = \textcircled{1}x\,\textcircled{-1}}$$

$x = -1$, $\hat{y} = \underline{\underline{-2}}$ ✗

$x = 0$, $\hat{y} = -1$

$x = 1$ $\hat{y} = 0$

©

φ

epoch = 2

$\downarrow$

$$\underline{y = 2x - 1}$$

$x = -1$, $\hat{y} = -3$

$x = 0$, $\hat{y} = -1$

$x = 1$ $\hat{y} = 1$

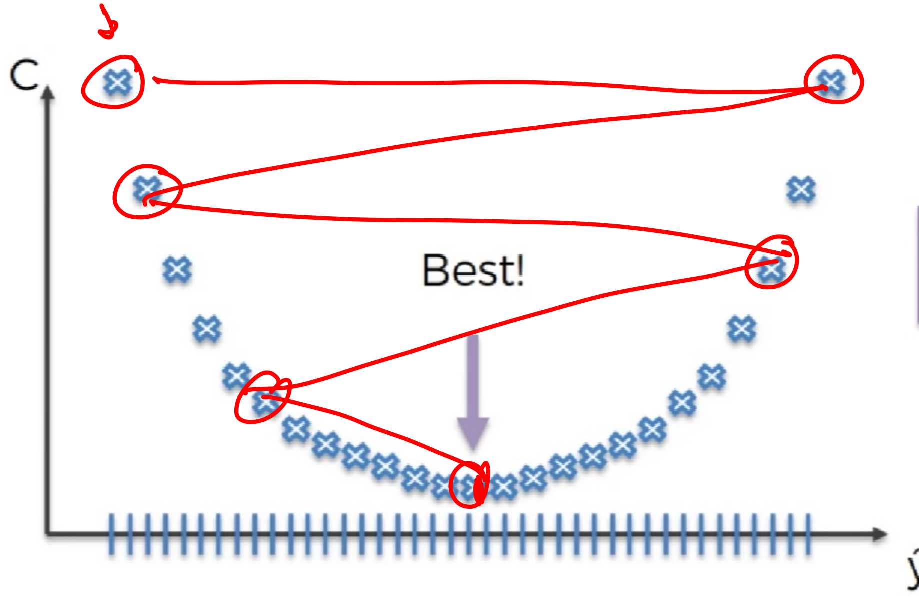
©

Hello World Explained

- Sequential
  - When using TensorFlow, you define your layers using Sequential
  - Inside the Sequential, you then specify what each layer looks like
- Layer
  - A layer is represented as Dense in TensorFlow
  - "Dense" means a set of fully (or densely) connected neurons where where every neuron is connected to every neuron in the next layer
- Compiling model
  - The computer starts guessing to create the model (formula) and comparing with the observed value
  - The optimizer used in this stage helps the machine to optimize the guess
    - SGD: stochastic gradient descent, a complex mathematical function that, when given the values, the previous guess, and the results of calculating the errors (or loss) on that guess, can then generate another one
    - ADAM: the optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments

Stochastic Gradient Descent

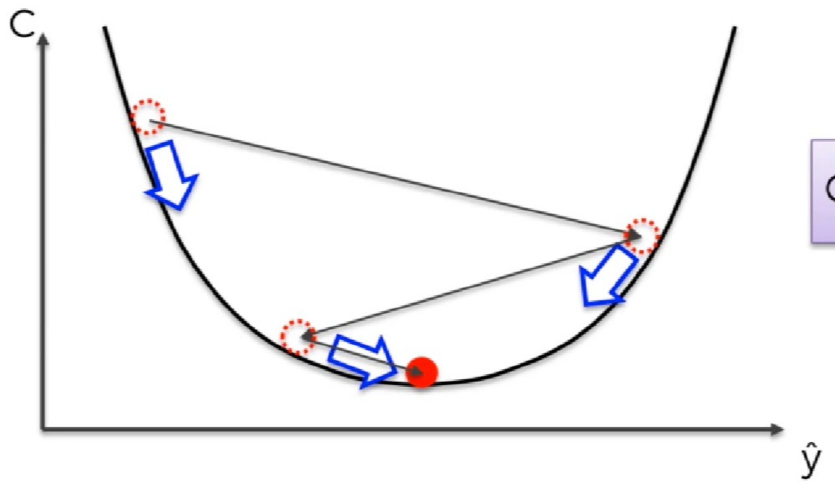
- Gradient, in plain terms means slope or slant of a surface
- So gradient descent literally means descending a slope to reach the lowest point on that surface
- Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function



$C = \frac{1}{2}(\hat{y} - y)^2$

Stochastic Gradient Descent - Steps

- Randomly initialize the weights to small numbers close to 0 (!= 0)
- Input the first observation in the input layer, each feature in one input node
- Forward Propagation:
  - From left to right
  - The neurons are activated in a way that the impact of each neuron's activation is limited by the weights
  - Propagate until getting the predicted result
- Compare the predicted result to the actual result
- Measure the generated error (cost)



$C = \frac{1}{2}(\hat{y} - y)^2$

Stochastic Gradient Descent - Steps

- Back Propagation
    - From right to left
    - The error is back propagated
    - Update the weights according to how much they are responsible for the error
    - The learning rate decides by how much we update the weights
- Repeat the steps and update the weights
    - after each observation
    - After a batch of observations
- When the whole training set passes through the ANN, that makes an epoch. Redo more epochs.