

Track 7

Inteligencia Artificial Aplicada



3. Introduction to Neural Network (TF Hands-On)

Prof. Marcelo José Rovai
rovai@unifei.edu.br

UNIFEI - Universidade Federal de Itajubá, Brazil



First Neural Network

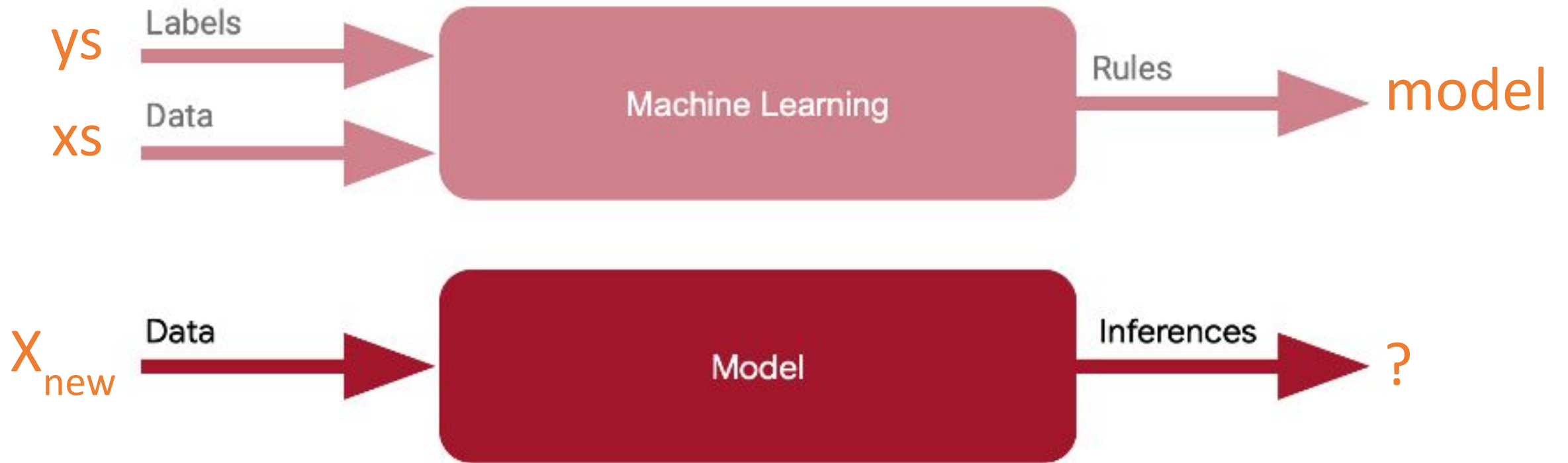
Putting it all together

$X \rightarrow -1, 0, 1, 2, 3, 4$

$Y \rightarrow -3, -1, 1, 3, 5, 7$



Inference -> `model.predict (Xnew)`

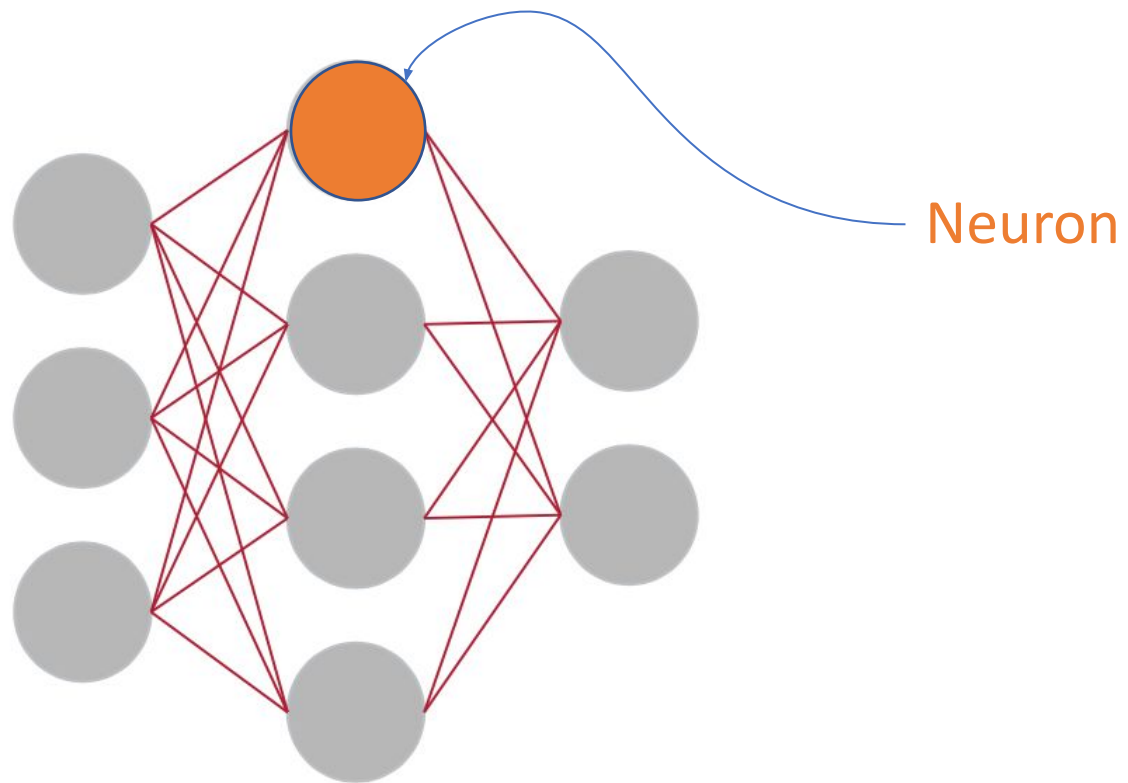


```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])  
model.compile(optimizer='sgd', loss='mean_squared_error')
```

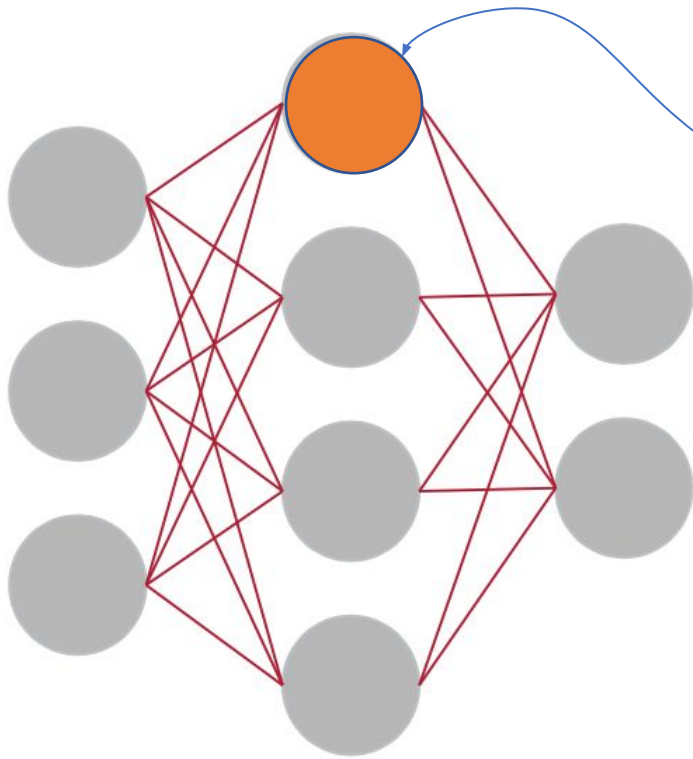
```
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)  
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)
```

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

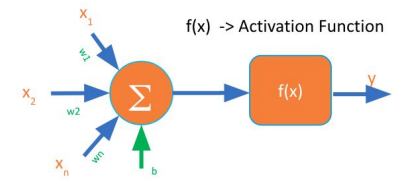
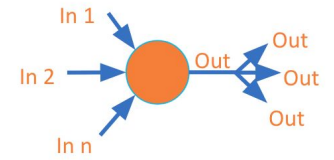
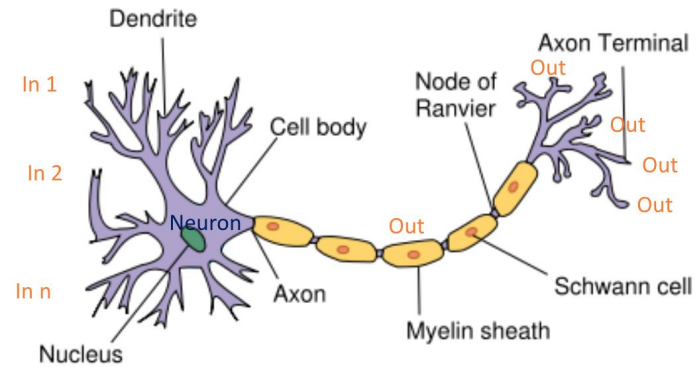
```
print(model.predict([10.0]))
```



Dense Neural Network (DNN)



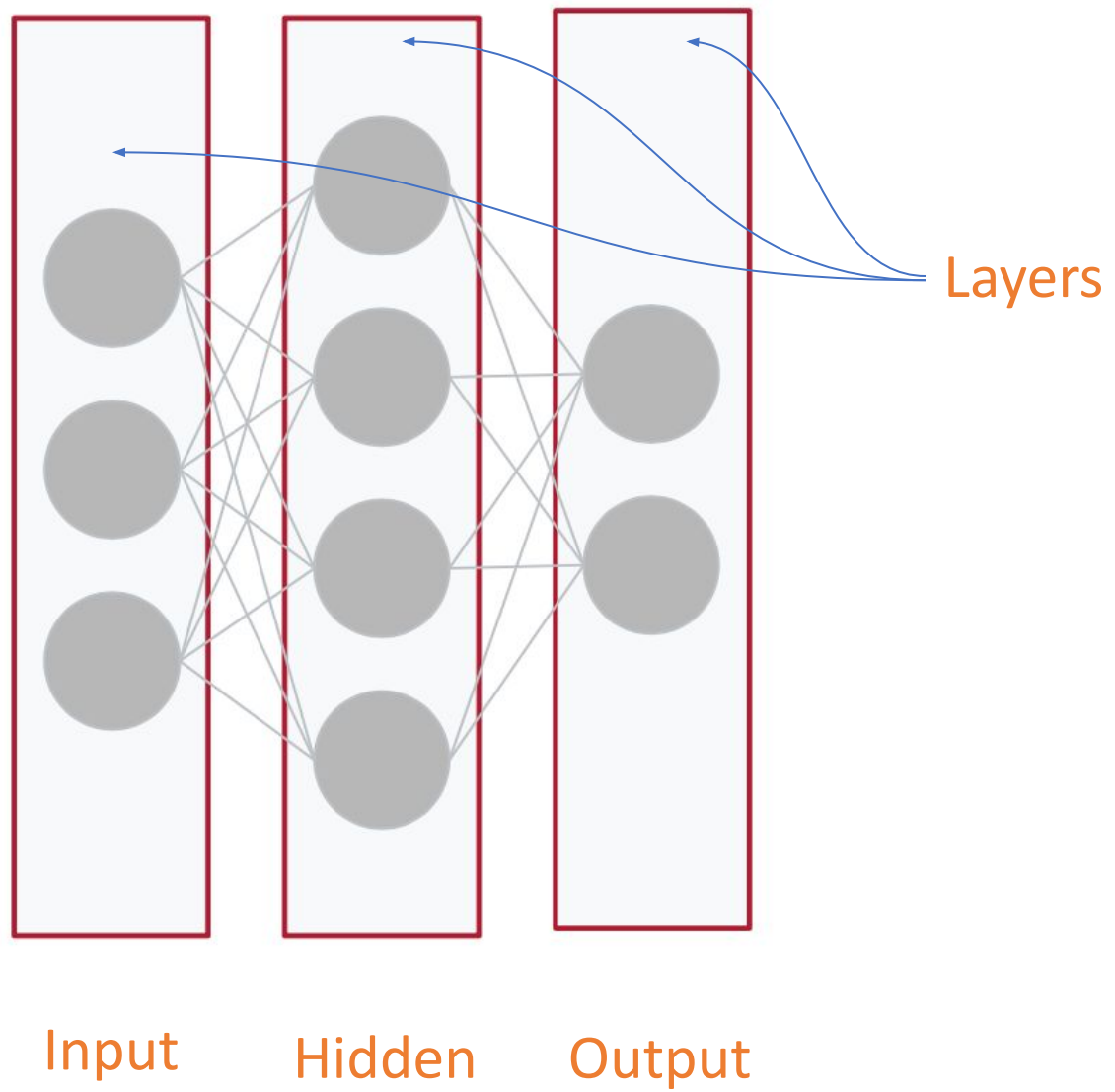
Neuron (Perceptron)

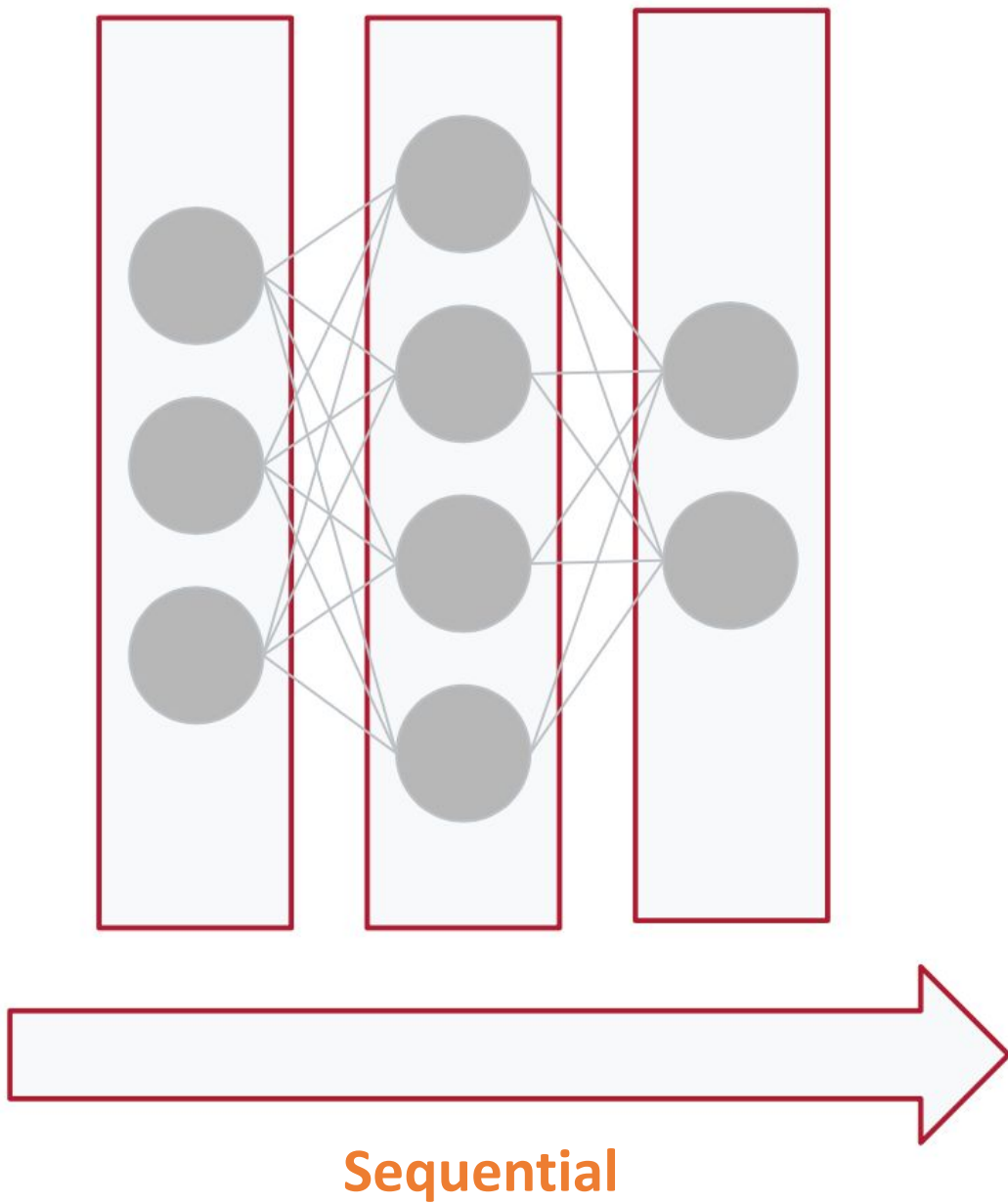


Parameters

$$y = f\left(\sum_{i=1}^n x_i w_i + b\right)$$

Dense Neural Network (DNN)





```
model = keras.Sequential([keras.layers.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]))
```

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
```

1 Layer

```
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]))
```

1 Neuron

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

1 Layer

```
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]))
```

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

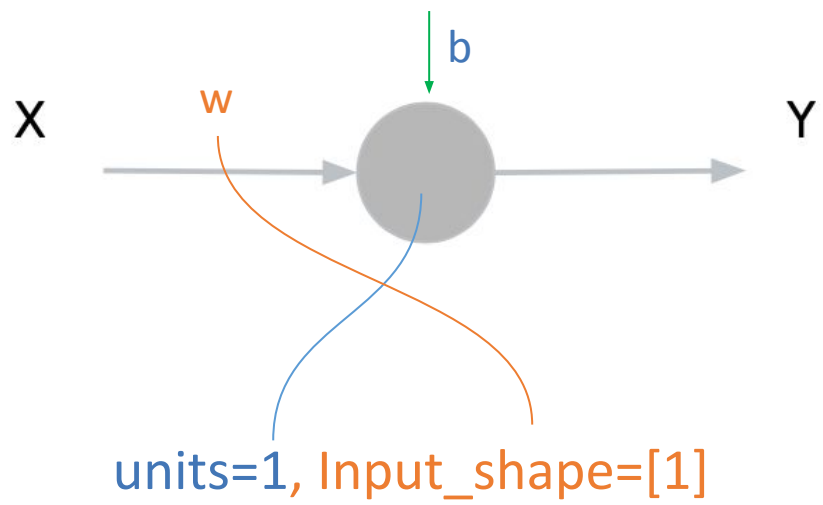
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

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

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

1 Neuron

1 Input



```
model = keras.Sequential([keras.layers.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]))
```



```
model = keras.Sequential([keras.layers.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]))
```



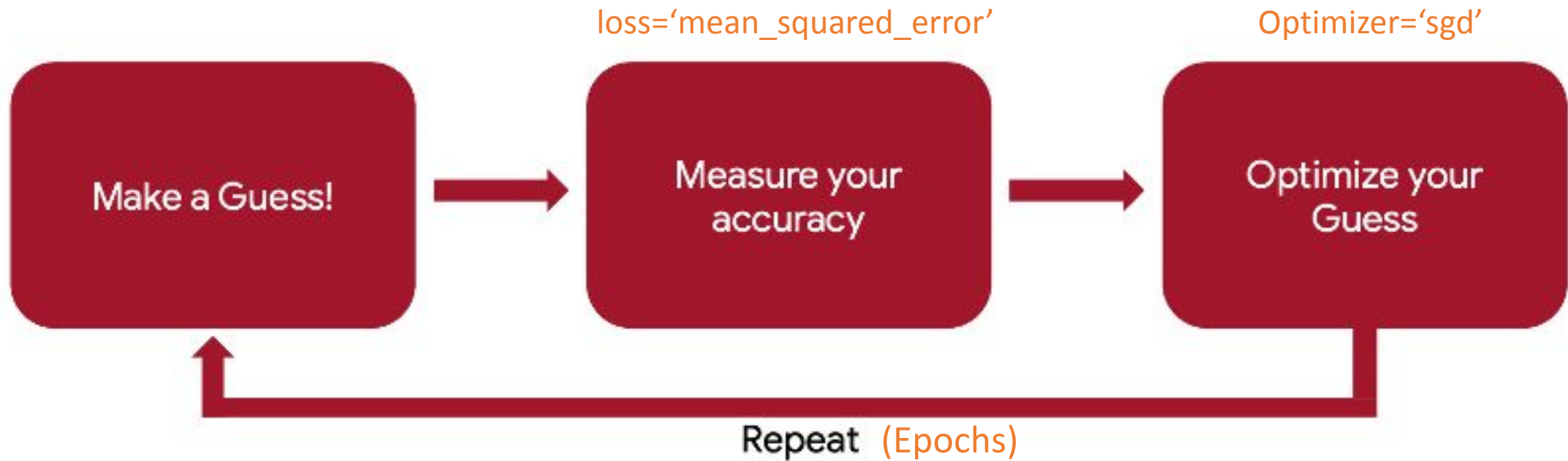
```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])  
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
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]))
```

Training -> `model.fit(xs, ys, epochs=500)`



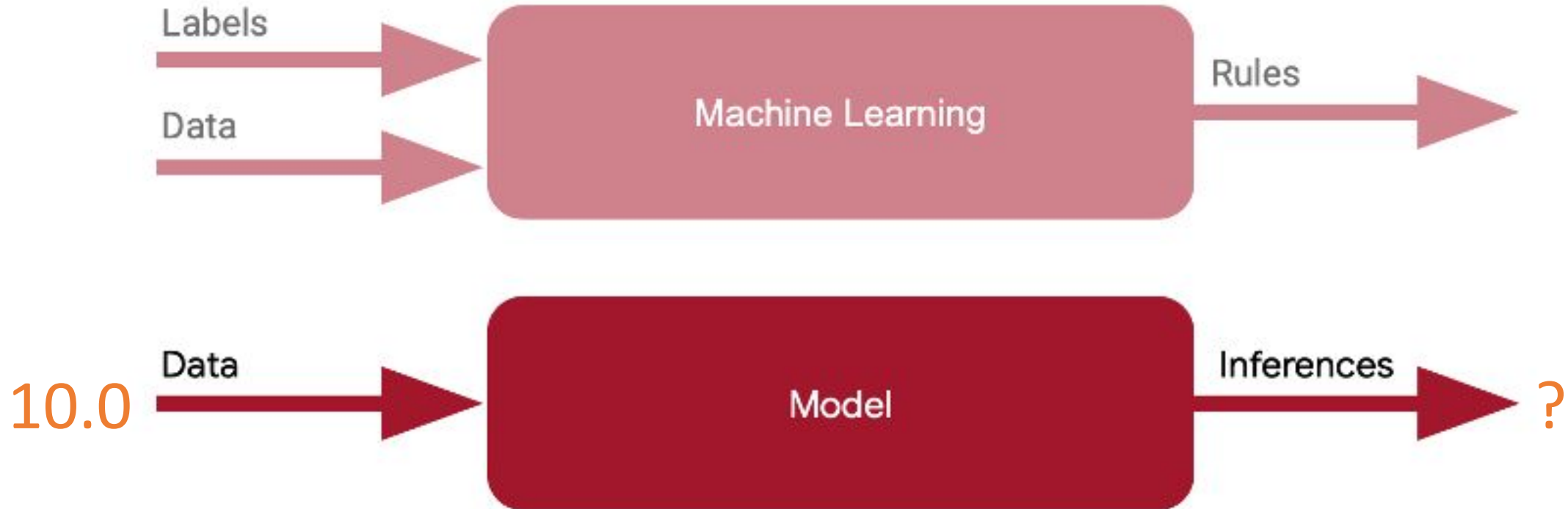
```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])  
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
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]))
```

Inference -> `model.predict([10.0])`



First Neural Network with TF2

Code Time!

[TF_First_Neural_Network.ipynb](#)



Thanks



UNIFEI

