



Advanced Microprocessors

FIRST NEURAL NETWORK

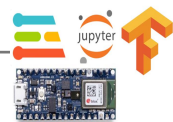
Dennis A. N. Gookyi

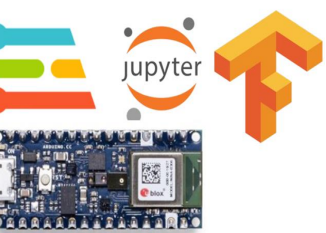




CONTENTS

❖ First Neural Network

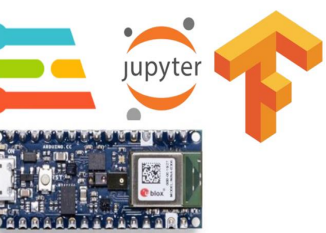




FIRST NEURAL NETWORK

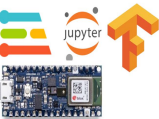
❖ First neural network

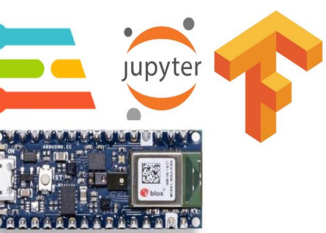
$X \rightarrow -1, 0, 1, 2, 3, 4$
 $Y \rightarrow -3, -1, 1, 3, 5, 7$



FIRST NEURAL NETWORK

❖ First neural network

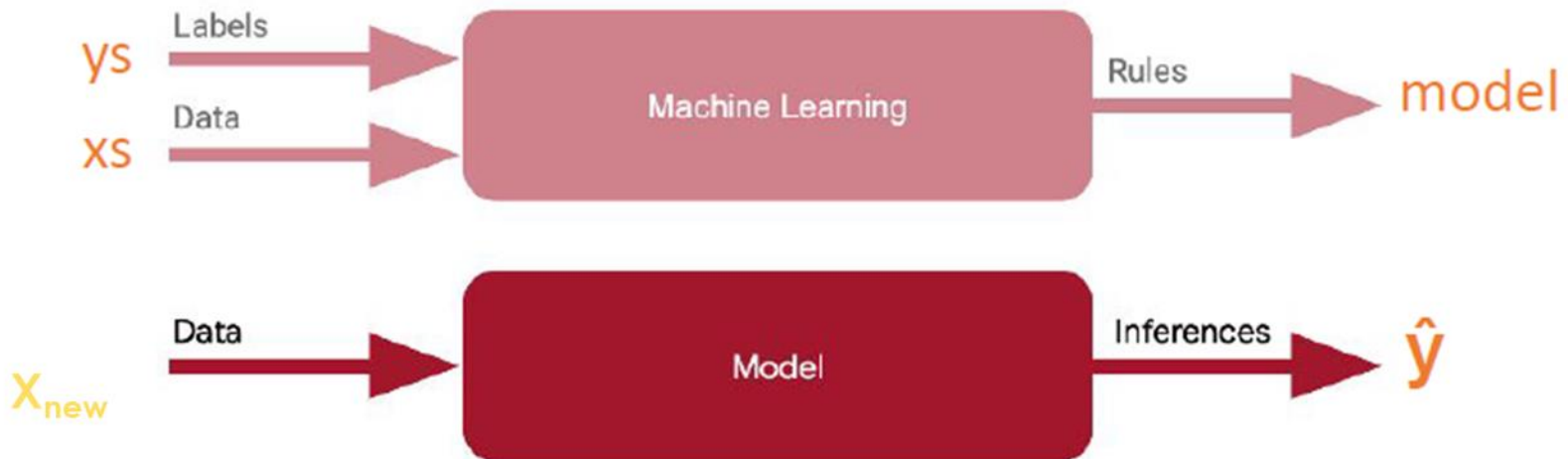


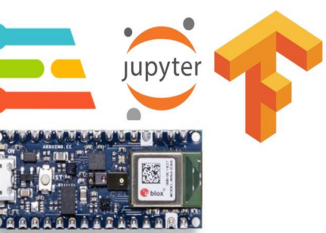


FIRST NEURAL NETWORK

❖ First neural network

Inference -> predict output (\hat{y}) for new input (X_{new})

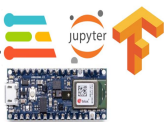




FIRST NEURAL NETWORK

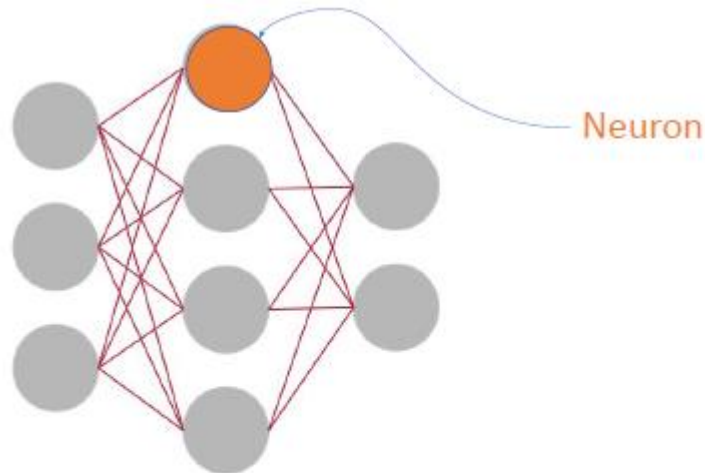
❖ First neural network

```
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]))
```



FIRST NEURAL NETWORK

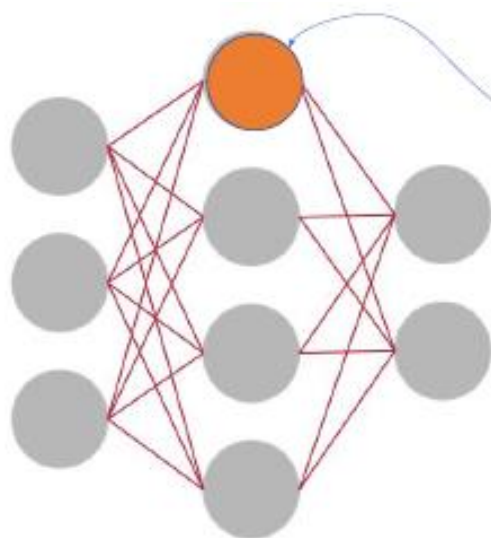
❖ First neural network



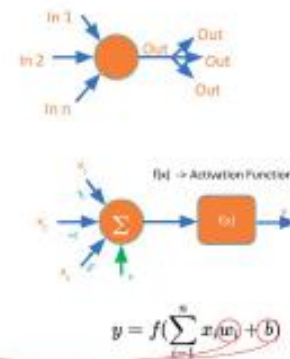
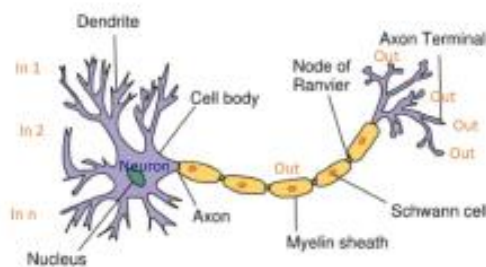
Dense Neural Network (DNN)

FIRST NEURAL NETWORK

❖ First neural network



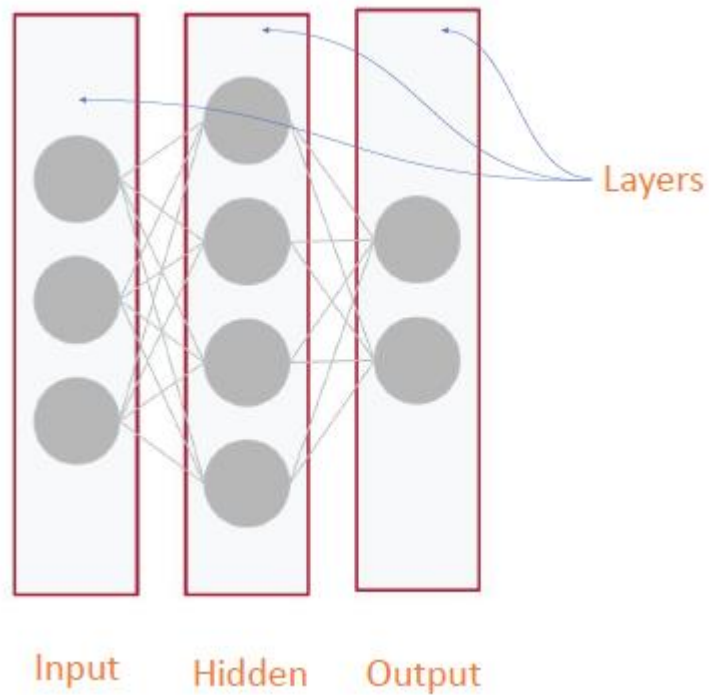
Neuron (Perceptron)

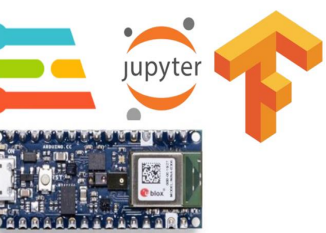


Dense Neural Network (DNN)

FIRST NEURAL NETWORK

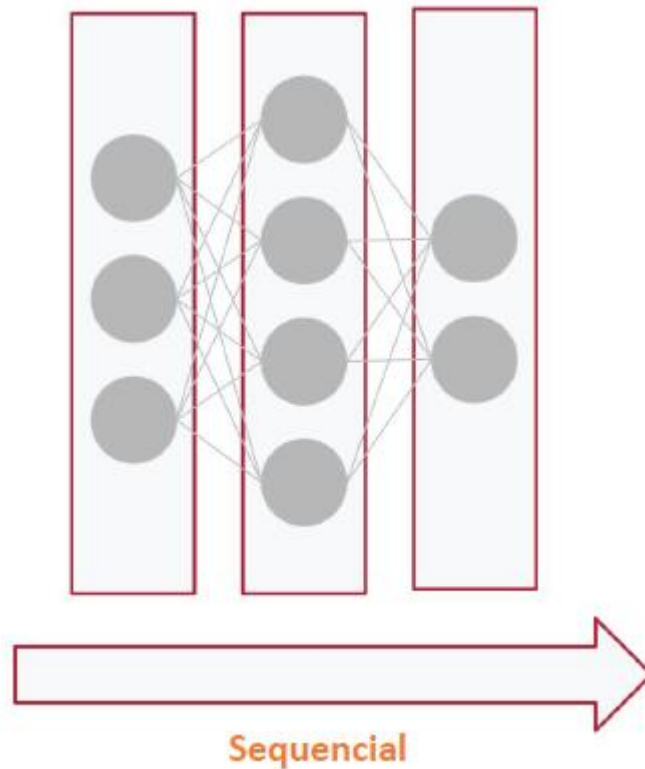
❖ First neural network

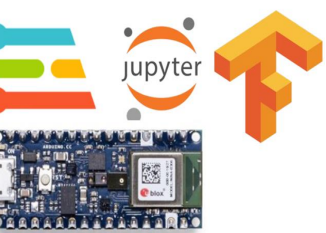




FIRST NEURAL NETWORK

❖ First neural network

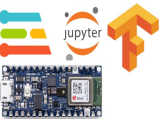




FIRST NEURAL NETWORK

❖ First neural network

```
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]))
```





FIRST NEURAL NETWORK

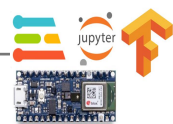
❖ First neural network

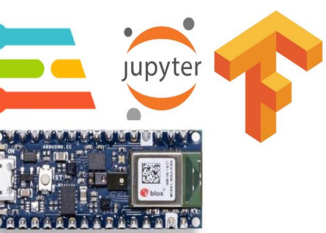
```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])]) 1 Layer
model.compile(optimizer='sgd', loss='mean_squared_error')

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]))
```





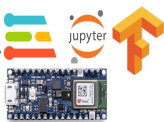
FIRST NEURAL NETWORK

❖ First neural network

```
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 Layer





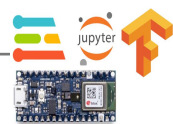
FIRST NEURAL NETWORK

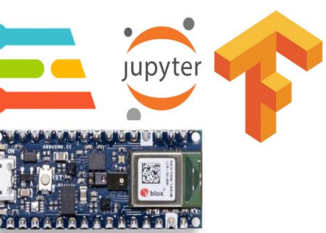
❖ First neural network

```
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

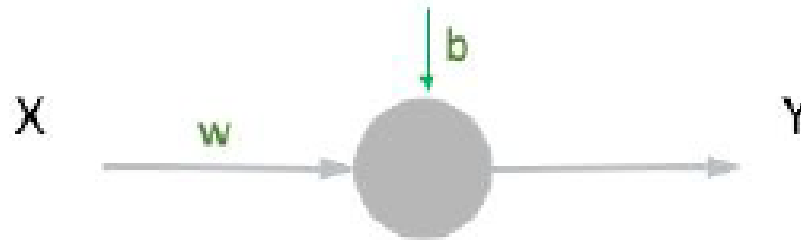
1 Input



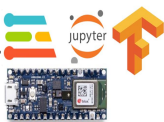


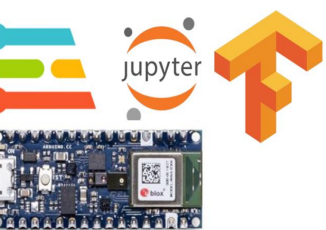
FIRST NEURAL NETWORK

❖ First neural network



`units=1, Input_shape=[1]`





FIRST NEURAL NETWORK

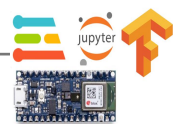
❖ First neural network

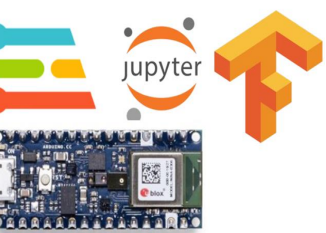
```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])  
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
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]))
```

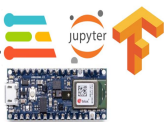




FIRST NEURAL NETWORK

❖ First neural network

```
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]))
```

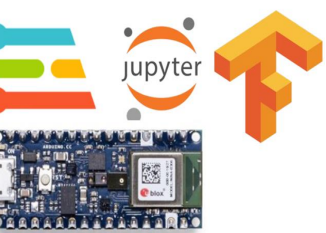




FIRST NEURAL NETWORK

❖ First neural network

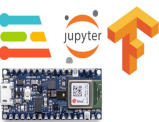


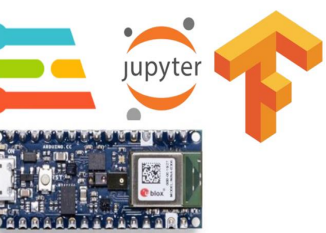


FIRST NEURAL NETWORK

❖ First neural network

```
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]))
```

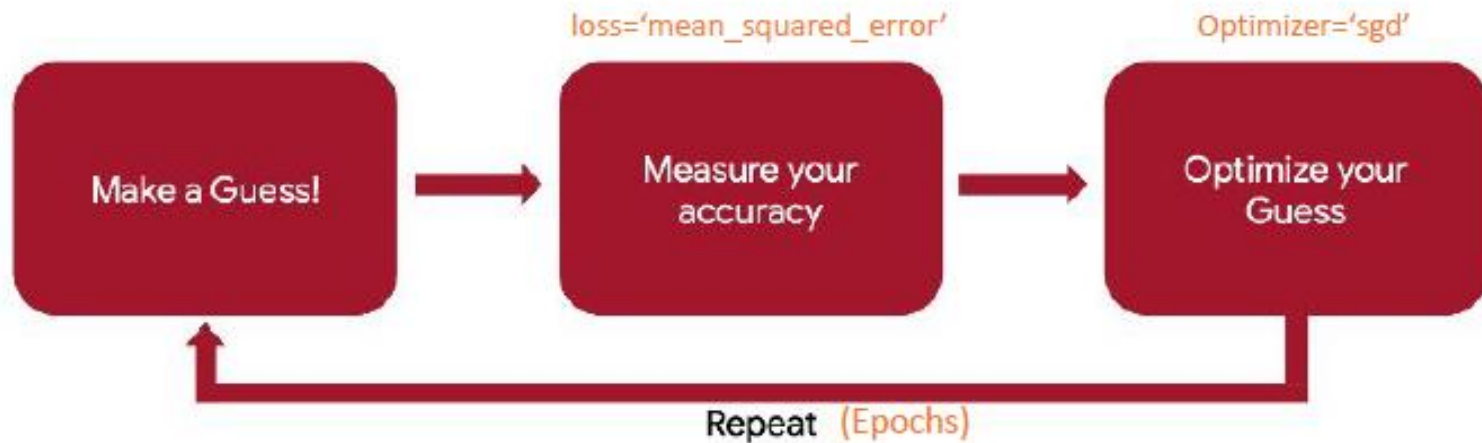


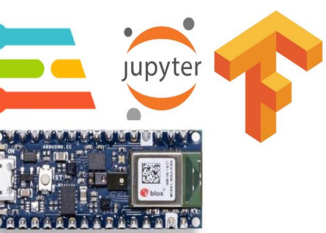


FIRST NEURAL NETWORK

❖ First neural network

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

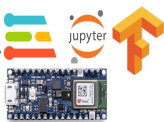


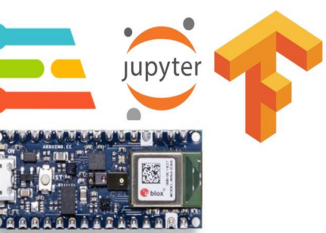


FIRST NEURAL NETWORK

❖ First neural network

```
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]))
```

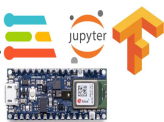


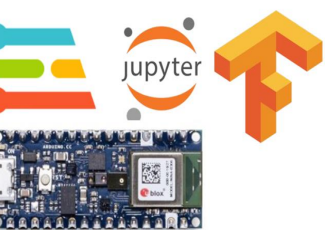


FIRST NEURAL NETWORK

❖ First neural network

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





FIRST NEURAL NETWORK

❖ First neural network

First Neural Network with TF2

Code Time!

