```
import tensorflow as tf
from tensorflow import keras
```

```
tf.__version__
```

```
'2.4.0'
```

```
import numpy as np
import matplotlib.pyplot as plt
```

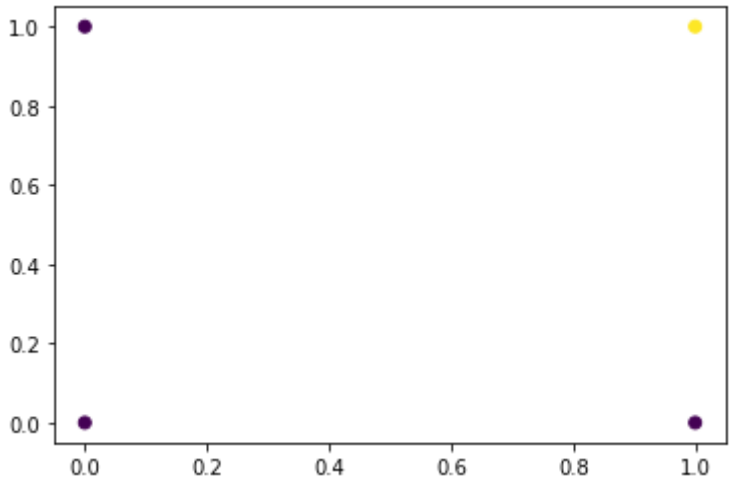
```
x_data = [[0, 0],
          [0, 1],
          [1, 0],
          [1, 1]]

y_data = [0,
          0,
          0,
          1]

x_data = np.array(x_data, dtype=np.float32)
y_data = np.array(y_data, dtype=np.float32)
```

```
x_data.shape, y_data.shape
```

```
((4, 2), (4,))
```

```
plt.scatter(x_data[:, 0], x_data[:, 1], c=y_data)
```

```
<matplotlib.collections.PathCollection at 0x7f61588dc8d0>
```

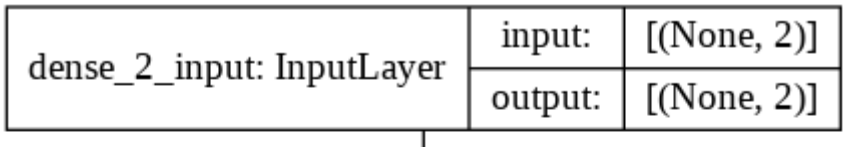



```
from tensorflow.keras import layers
from tensorflow.keras import activations
from tensorflow.keras import optimizers
from tensorflow.keras import models
```

```
model = models.Sequential()
model.add(layers.Dense(2, input_dim=2))
model.add(layers.Activation('tanh'))
model.add(layers.Dense(1))
model.add(layers.Activation('sigmoid'))


sgd = optimizers.SGD(lr=0.1)
model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])


model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_2 (Dense)              (None, 2)                 6         
_________________________________________________________________
activation_2 (Activation)    (None, 2)                 0         
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 3         
_________________________________________________________________
activation_3 (Activation)    (None, 1)                 0         
=================================================================
Total params: 9
Trainable params: 9
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow.keras.utils import plot_model
plot_model(model, to_file='model_and.png', show_shapes=True)
```



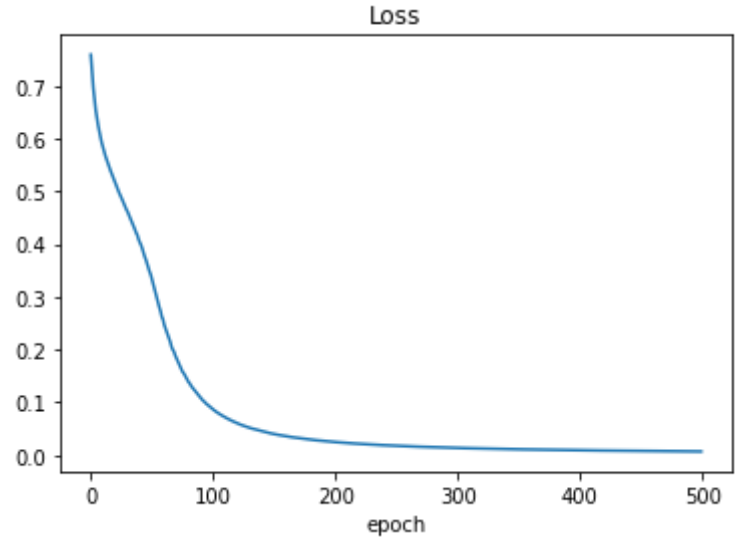

```
history = model.fit(x_data, y_data,  batch_size=1, epochs=500)
```

```
Epoch 1/500
4/4 [==============================] - 0s 2ms/step - loss: 0.8134 - accuracy: 0.4333
Epoch 2/500
4/4 [==============================] - 0s 2ms/step - loss: 0.6502 - accuracy: 0.5667
Epoch 3/500
4/4 [==============================] - 0s 2ms/step - loss: 0.7828 - accuracy: 0.2667
Epoch 4/500
4/4 [==============================] - 0s 2ms/step - loss: 0.6406 - accuracy: 0.4333
Epoch 5/500
4/4 [==============================] - 0s 2ms/step - loss: 0.7396 - accuracy: 0.4333
Epoch 6/500
4/4 [==============================] - 0s 2ms/step - loss: 0.5810 - accuracy: 0.9000
Epoch 7/500
4/4 [==============================] - 0s 2ms/step - loss: 0.5284 - accuracy: 0.8333
Epoch 8/500
4/4 [==============================] - 0s 2ms/step - loss: 0.5143 - accuracy: 0.8333
Epoch 9/500
4/4 [==============================] - 0s 2ms/step - loss: 0.5017 - accuracy: 0.8333
Epoch 10/500
4/4 [==============================] - 0s 2ms/step - loss: 0.7243 - accuracy: 0.5333
Epoch 11/500
4/4 [==============================] - 0s 2ms/step - loss: 0.6460 - accuracy: 0.7333
Epoch 12/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4691 - accuracy: 0.8333
Epoch 13/500
4/4 [==============================] - 0s 2ms/step - loss: 0.7217 - accuracy: 0.5333
Epoch 14/500
4/4 [==============================] - 0s 2ms/step - loss: 0.3957 - accuracy: 0.9000
Epoch 15/500
4/4 [==============================] - 0s 2ms/step - loss: 0.5308 - accuracy: 0.7333
Epoch 16/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4647 - accuracy: 0.9000
Epoch 17/500
4/4 [==============================] - 0s 2ms/step - loss: 0.3781 - accuracy: 0.9000
Epoch 18/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4958 - accuracy: 0.7333
Epoch 19/500
4/4 [==============================] - 0s 2ms/step - loss: 0.3729 - accuracy: 0.9000
Epoch 20/500
4/4 [==============================] - 0s 2ms/step - loss: 0.5029 - accuracy: 0.7333
Epoch 21/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4839 - accuracy: 0.7333
Epoch 22/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4768 - accuracy: 0.7333
Epoch 23/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4372 - accuracy: 0.8333
Epoch 24/500
4/4 [==============================] - 0s 2ms/step - loss: 0.6186 - accuracy: 0.5333
Epoch 25/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4239 - accuracy: 0.9000
Epoch 26/500
4/4 [==============================] - 0s 2ms/step - loss: 0.4706 - accuracy: 0.7333
Epoch 27/500
```

```
4/4 [==============================] - 0s 2ms/step - loss: 0.4407 - accuracy: 0.7333
Epoch 28/500
4/4 [==============================] - 0s 2ms/step - loss: 0.6143 - accuracy: 0.5333
Epoch 29/500
4/4 [==============================] - 0s 3ms/step - loss: 0.5850 - accuracy: 0.5333
Epoch 30/500
```

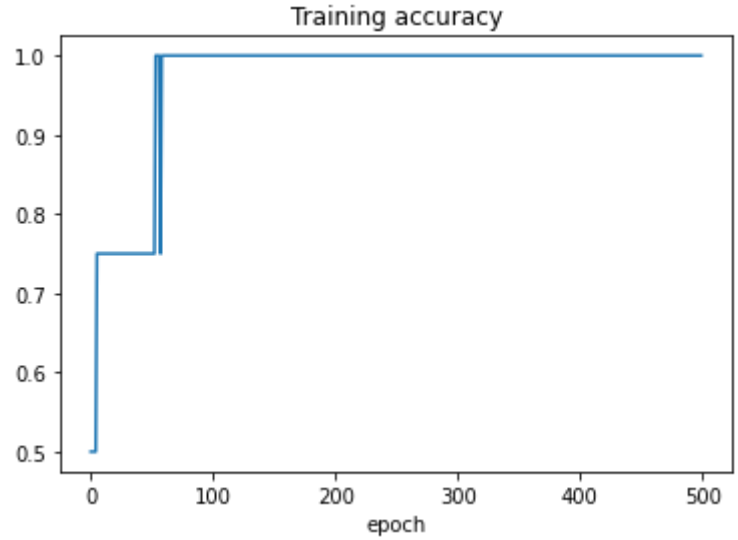
```
plt.plot(history.history['loss'])
plt.title('Loss')
plt.xlabel('epoch')
```

```
Text(0.5, 0, 'epoch')
```




```
plt.plot(history.history['accuracy'])
plt.title('Training accuracy')
plt.xlabel('epoch')
```

```
Text(0.5, 0, 'epoch')
```




```
hypothesis = model.predict(x_data)
print(hypothesis)
```

```
[[0.00154999]
 [0.00635552]
 [0.00623325]
 [0.9861352 ]]
```

```
predicted = hypothesis > 0.5
print(predicted)
```

```
[[False]
 [False]
 [False]
 [ True]]
```