

```
In [1]: import numpy as np
        from keras.models import Sequential
        from keras.layers import Dense

        #AND Operation
        training_data = np.array([[0,0],[0,1],[1,0],[1,1]], "float32")
        target_data = np.array([[0],[0],[0],[1]], "float32")

        model = Sequential()
        model.add(Dense(8, input_dim = 2, activation='relu'))
        model.add(Dense(1, activation='sigmoid'))
        model.compile(loss='mean_squared_error',
                      optimizer='adam',
                      metrics=['binary_accuracy'])

        model.fit(training_data, target_data, epochs=1000)
        scores = model.evaluate(training_data, target_data)

        print("\ns: %.2f%%" % (model.metrics_names[1], scores[1]*100))
        print(model.predict(training_data).round())
```

```

1/1 [=====] - 0s 6ms/step - loss: 0.0262 - binary_accuracy:
1.0000
Epoch 991/1000
1/1 [=====] - 0s 8ms/step - loss: 0.0262 - binary_accuracy:
1.0000
Epoch 992/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0261 - binary_accuracy:
1.0000
Epoch 993/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0260 - binary_accuracy:
1.0000
Epoch 994/1000
1/1 [=====] - 0s 10ms/step - loss: 0.0260 - binary_accurac
y: 1.0000
Epoch 995/1000
1/1 [=====] - 0s 13ms/step - loss: 0.0259 - binary_accurac
y: 1.0000
Epoch 996/1000
1/1 [=====] - 0s 10ms/step - loss: 0.0258 - binary_accurac
y: 1.0000
Epoch 997/1000
1/1 [=====] - 0s 7ms/step - loss: 0.0258 - binary_accuracy:
1.0000
Epoch 998/1000
1/1 [=====] - 0s 8ms/step - loss: 0.0257 - binary_accuracy:
1.0000
Epoch 999/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0256 - binary_accuracy:
1.0000
Epoch 1000/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0256 - binary_accuracy:
1.0000
1/1 [=====] - 0s 197ms/step - loss: 0.0255 - binary_accurac
y: 1.0000

binary_accuracy: 100.00%
1/1 [=====] - 0s 129ms/step
[[0.]
 [0.]
 [0.]
 [1.]]

```