

```
In [2]: import numpy as np
from keras.models import Sequential
from keras.layers import Dense
#XOR operations
training_data = np.array([[0,0],[0,1],[1,0],[1,1]], "float32")
target_data = np.array([[0],[1],[1],[0]], "float32")

model = Sequential()
model.add(Dense(12, 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 10ms/step - loss: 0.0468 - binary_accu
racy: 1.0000
Epoch 991/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0466 - binary_accu
racy: 1.0000
Epoch 992/1000
1/1 [=====] - 0s 11ms/step - loss: 0.0465 - binary_accu
racy: 1.0000
Epoch 993/1000
1/1 [=====] - 0s 8ms/step - loss: 0.0464 - binary_accu
racy: 1.0000
Epoch 994/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0463 - binary_accu
racy: 1.0000
Epoch 995/1000
1/1 [=====] - 0s 10ms/step - loss: 0.0461 - binary_accu
racy: 1.0000
Epoch 996/1000
1/1 [=====] - 0s 10ms/step - loss: 0.0460 - binary_accu
racy: 1.0000
Epoch 997/1000
1/1 [=====] - 0s 10ms/step - loss: 0.0459 - binary_accu
racy: 1.0000
Epoch 998/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0458 - binary_accu
racy: 1.0000
Epoch 999/1000
1/1 [=====] - 0s 8ms/step - loss: 0.0457 - binary_accu
racy: 1.0000
Epoch 1000/1000
1/1 [=====] - 0s 9ms/step - loss: 0.0455 - binary_accu
racy: 1.0000
1/1 [=====] - 0s 153ms/step - loss: 0.0454 - binary_accu
racy: 1.0000

binary_accuracy: 100.00%
1/1 [=====] - 0s 63ms/step
[[0.]
 [1.]
 [1.]
 [0.]]

```