bib34-saba-attar-dl-lab-1-keras

[1]:

```
# !pip install tensorflow
# Step 1 - Load the dataset
from numpy import loadtxt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

[4]:

```
## INPUT Variables ##
# x1 - Number of times pregnant
# x2 - plasma glucose
# x3 - diastolic blood pressure
# x4 - Triceps skin fold thickness
# x5 - 2-hour serum insulin
# x6 - bmi
# x7 - diabetes pedigree function
# x8 - age (yrs)
## Output Variable ##
# Class Variable - 0 or 1
dataset = loadtxt('pima-indians-diabetes.csv',delimiter=',')
dataset
```

[4]:

```
array([[  6.   , 148.   ,  72.   , …,   0.627,  50.   ,   1.   ],
       [  1.   ,  85.   ,  66.   , …,   0.351,  31.   ,   0.   ],
       [  8.   , 183.   ,  64.   , …,   0.672,  32.   ,   1.   ],
       …,
       [  5.   , 121.   ,  72.   , …,   0.245,  30.   ,   0.   ],
       [  1.   , 126.   ,  60.   , …,   0.349,  47.   ,   1.   ],
       [  1.   ,  93.   ,  70.   , …,   0.315,  23.   ,   0.   ]])
```

[5]:

```
# [:,:] - first : is range of rows and second : is columns
# [start:end] - begins at start, ends at end-1
x = dataset[:,0:8]
print(type(x))
print(x.shape)
y = dataset[:,8]
print(y)
```

```
<class 'numpy.ndarray'>
```

```
(768, 8)
[1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 1. 1.
 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0.
 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0.
 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.
 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.
 1. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0.
 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 0.
 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 1.
 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0.
 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0.
 1. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 0. 1. 0. 1. 1. 0. 0.
 1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0.
 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1.
 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0.
 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0.
 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.
 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0.
 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0.
 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1.
 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0.
 0. 0. 0. 1. 1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1.
 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0.]
```

[6]:

```python
# Step 2 - Creating or define the Keras Model
# Sequential Model
# Layer1 -> Layer2 -> Layer3
model = Sequential()
# The model expects row of data with 8 variables
# 12 = nodes
model.add(Dense(12, input_shape=(8,), activation='relu'))
# Hidden Layer
# 8 = nodes
model.add(Dense(8, activation='relu'))
# Output layer
model.add(Dense(1,activation='sigmoid'))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87:
```

```
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

[7]:

```
# Step 3 - Compile the Keras model
# loss (error)
# optimizer (adam)
# metrics = accuracy
model.compile(loss='binary_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
```

[9]:

```
#Step 4 - Fit / Train the model
#1 = Epochs - number of iterations / passes
#2 - Batch - sample data
model.fit(x,y, epochs=150, batch_size=10)
```

```
Epoch 1/150
77/77                 1s 1ms/step -
accuracy: 0.4761 - loss: 6.2794
Epoch 2/150
77/77                 0s 1ms/step -
accuracy: 0.5047 - loss: 3.7406
Epoch 3/150
77/77                 0s 1ms/step -
accuracy: 0.4671 - loss: 2.3635
Epoch 4/150
77/77                 0s 1ms/step -
accuracy: 0.5006 - loss: 1.4537
Epoch 5/150
77/77                 0s 1ms/step -
accuracy: 0.4864 - loss: 1.2309
Epoch 6/150
77/77                 0s 2ms/step -
accuracy: 0.5298 - loss: 0.9965
Epoch 7/150
77/77                 0s 2ms/step -
accuracy: 0.5756 - loss: 0.8653
Epoch 8/150
77/77                 0s 2ms/step -
accuracy: 0.5822 - loss: 0.8102
Epoch 9/150
77/77                 0s 2ms/step -
accuracy: 0.6370 - loss: 0.7866
Epoch 10/150
77/77                 0s 2ms/step -
accuracy: 0.6364 - loss: 0.7399
Epoch 11/150
```

```
77/77             0s 2ms/step -
accuracy: 0.6329 - loss: 0.6746
Epoch 12/150
77/77             0s 2ms/step -
accuracy: 0.6311 - loss: 0.7134
Epoch 13/150
77/77             0s 2ms/step -
accuracy: 0.6552 - loss: 0.6758
Epoch 14/150
77/77             0s 2ms/step -
accuracy: 0.6541 - loss: 0.7116
Epoch 15/150
77/77             0s 2ms/step -
accuracy: 0.6573 - loss: 0.6806
Epoch 16/150
77/77             0s 2ms/step -
accuracy: 0.6690 - loss: 0.6701
Epoch 17/150
77/77             0s 1ms/step -
accuracy: 0.6675 - loss: 0.6545
Epoch 18/150
77/77             0s 1ms/step -
accuracy: 0.6618 - loss: 0.6704
Epoch 19/150
77/77             0s 1ms/step -
accuracy: 0.6635 - loss: 0.6410
Epoch 20/150
77/77             0s 2ms/step -
accuracy: 0.6542 - loss: 0.6384
Epoch 21/150
77/77             0s 2ms/step -
accuracy: 0.6969 - loss: 0.6067
Epoch 22/150
77/77             0s 1ms/step -
accuracy: 0.6550 - loss: 0.6636
Epoch 23/150
77/77             0s 1ms/step -
accuracy: 0.6806 - loss: 0.6312
Epoch 24/150
77/77             0s 1ms/step -
accuracy: 0.6852 - loss: 0.5988
Epoch 25/150
77/77             0s 1ms/step -
accuracy: 0.6916 - loss: 0.6028
Epoch 26/150
77/77             0s 1ms/step -
accuracy: 0.6731 - loss: 0.6181
Epoch 27/150
```

```
77/77             0s 2ms/step -
accuracy: 0.6916 - loss: 0.6159
Epoch 28/150
77/77             0s 1ms/step -
accuracy: 0.6829 - loss: 0.6038
Epoch 29/150
77/77             0s 1ms/step -
accuracy: 0.7010 - loss: 0.5952
Epoch 30/150
77/77             0s 1ms/step -
accuracy: 0.6839 - loss: 0.6271
Epoch 31/150
77/77             0s 1ms/step -
accuracy: 0.7439 - loss: 0.5731
Epoch 32/150
77/77             0s 1ms/step -
accuracy: 0.6954 - loss: 0.5885
Epoch 33/150
77/77             0s 1ms/step -
accuracy: 0.6937 - loss: 0.6175
Epoch 34/150
77/77             0s 1ms/step -
accuracy: 0.7213 - loss: 0.5658
Epoch 35/150
77/77             0s 1ms/step -
accuracy: 0.7245 - loss: 0.5749
Epoch 36/150
77/77             0s 1ms/step -
accuracy: 0.6860 - loss: 0.5729
Epoch 37/150
77/77             0s 1ms/step -
accuracy: 0.6662 - loss: 0.6037
Epoch 38/150
77/77             0s 1ms/step -
accuracy: 0.7177 - loss: 0.5642
Epoch 39/150
77/77             0s 1ms/step -
accuracy: 0.7164 - loss: 0.5715
Epoch 40/150
77/77             0s 1ms/step -
accuracy: 0.7014 - loss: 0.6005
Epoch 41/150
77/77             0s 2ms/step -
accuracy: 0.7268 - loss: 0.5544
Epoch 42/150
77/77             0s 1ms/step -
accuracy: 0.6926 - loss: 0.5782
Epoch 43/150
```

```
77/77            0s 1ms/step -
accuracy: 0.7267 - loss: 0.5683
Epoch 44/150
77/77            0s 1ms/step -
accuracy: 0.7258 - loss: 0.5780
Epoch 45/150
77/77            0s 1ms/step -
accuracy: 0.7006 - loss: 0.5666
Epoch 46/150
77/77            0s 2ms/step -
accuracy: 0.6955 - loss: 0.5821
Epoch 47/150
77/77            0s 1ms/step -
accuracy: 0.6830 - loss: 0.5948
Epoch 48/150
77/77            0s 1ms/step -
accuracy: 0.6994 - loss: 0.5803
Epoch 49/150
77/77            0s 1ms/step -
accuracy: 0.7158 - loss: 0.5513
Epoch 50/150
77/77            0s 1ms/step -
accuracy: 0.7218 - loss: 0.5553
Epoch 51/150
77/77            0s 1ms/step -
accuracy: 0.6865 - loss: 0.5960
Epoch 52/150
77/77            0s 2ms/step -
accuracy: 0.7104 - loss: 0.5523
Epoch 53/150
77/77            0s 1ms/step -
accuracy: 0.7275 - loss: 0.5451
Epoch 54/150
77/77            0s 1ms/step -
accuracy: 0.7373 - loss: 0.5468
Epoch 55/150
77/77            0s 1ms/step -
accuracy: 0.7062 - loss: 0.5950
Epoch 56/150
77/77            0s 1ms/step -
accuracy: 0.6887 - loss: 0.5796
Epoch 57/150
77/77            0s 1ms/step -
accuracy: 0.7119 - loss: 0.5431
Epoch 58/150
77/77            0s 1ms/step -
accuracy: 0.7246 - loss: 0.5499
Epoch 59/150
```

```
77/77             0s 2ms/step -
accuracy: 0.7266 - loss: 0.5555
Epoch 60/150
77/77             0s 1ms/step -
accuracy: 0.7447 - loss: 0.5408
Epoch 61/150
77/77             0s 1ms/step -
accuracy: 0.7430 - loss: 0.5357
Epoch 62/150
77/77             0s 1ms/step -
accuracy: 0.7301 - loss: 0.5292
Epoch 63/150
77/77             0s 1ms/step -
accuracy: 0.7172 - loss: 0.5599
Epoch 64/150
77/77             0s 1ms/step -
accuracy: 0.7061 - loss: 0.5684
Epoch 65/150
77/77             0s 2ms/step -
accuracy: 0.7536 - loss: 0.5236
Epoch 66/150
77/77             0s 1ms/step -
accuracy: 0.7522 - loss: 0.5230
Epoch 67/150
77/77             0s 1ms/step -
accuracy: 0.7426 - loss: 0.5530
Epoch 68/150
77/77             0s 1ms/step -
accuracy: 0.7413 - loss: 0.5329
Epoch 69/150
77/77             0s 1ms/step -
accuracy: 0.7422 - loss: 0.5360
Epoch 70/150
77/77             0s 1ms/step -
accuracy: 0.7309 - loss: 0.5253
Epoch 71/150
77/77             0s 1ms/step -
accuracy: 0.7447 - loss: 0.5360
Epoch 72/150
77/77             0s 1ms/step -
accuracy: 0.7052 - loss: 0.5693
Epoch 73/150
77/77             0s 2ms/step -
accuracy: 0.7302 - loss: 0.5362
Epoch 74/150
77/77             0s 1ms/step -
accuracy: 0.7431 - loss: 0.5183
Epoch 75/150
```

```
77/77             0s 1ms/step -
accuracy: 0.7284 - loss: 0.5314
Epoch 76/150
77/77             0s 1ms/step -
accuracy: 0.7186 - loss: 0.5426
Epoch 77/150
77/77             0s 1ms/step -
accuracy: 0.7398 - loss: 0.5434
Epoch 78/150
77/77             0s 2ms/step -
accuracy: 0.7608 - loss: 0.5476
Epoch 79/150
77/77             0s 2ms/step -
accuracy: 0.7382 - loss: 0.5157
Epoch 80/150
77/77             0s 2ms/step -
accuracy: 0.7207 - loss: 0.5495
Epoch 81/150
77/77             0s 2ms/step -
accuracy: 0.7377 - loss: 0.5464
Epoch 82/150
77/77             0s 2ms/step -
accuracy: 0.7580 - loss: 0.5133
Epoch 83/150
77/77             0s 2ms/step -
accuracy: 0.7367 - loss: 0.5462
Epoch 84/150
77/77             0s 2ms/step -
accuracy: 0.7292 - loss: 0.5224
Epoch 85/150
77/77             0s 3ms/step -
accuracy: 0.7407 - loss: 0.5489
Epoch 86/150
77/77             0s 2ms/step -
accuracy: 0.7189 - loss: 0.5331
Epoch 87/150
77/77             0s 2ms/step -
accuracy: 0.7287 - loss: 0.5385
Epoch 88/150
77/77             0s 2ms/step -
accuracy: 0.7376 - loss: 0.5316
Epoch 89/150
77/77             0s 2ms/step -
accuracy: 0.7370 - loss: 0.5315
Epoch 90/150
77/77             0s 1ms/step -
accuracy: 0.7096 - loss: 0.5675
Epoch 91/150
```

```
77/77             0s 1ms/step -
accuracy: 0.7487 - loss: 0.5136
Epoch 92/150
77/77             0s 1ms/step -
accuracy: 0.7378 - loss: 0.5400
Epoch 93/150
77/77             0s 1ms/step -
accuracy: 0.7224 - loss: 0.5571
Epoch 94/150
77/77             0s 1ms/step -
accuracy: 0.7447 - loss: 0.5322
Epoch 95/150
77/77             0s 1ms/step -
accuracy: 0.6993 - loss: 0.5742
Epoch 96/150
77/77             0s 2ms/step -
accuracy: 0.7357 - loss: 0.5320
Epoch 97/150
77/77             0s 2ms/step -
accuracy: 0.7449 - loss: 0.5154
Epoch 98/150
77/77             0s 1ms/step -
accuracy: 0.7418 - loss: 0.5136
Epoch 99/150
77/77             0s 1ms/step -
accuracy: 0.7475 - loss: 0.5302
Epoch 100/150
77/77             0s 1ms/step -
accuracy: 0.7320 - loss: 0.5069
Epoch 101/150
77/77             0s 1ms/step -
accuracy: 0.7616 - loss: 0.5157
Epoch 102/150
77/77             0s 1ms/step -
accuracy: 0.7143 - loss: 0.5377
Epoch 103/150
77/77             0s 2ms/step -
accuracy: 0.7539 - loss: 0.5058
Epoch 104/150
77/77             0s 1ms/step -
accuracy: 0.7349 - loss: 0.5213
Epoch 105/150
77/77             0s 1ms/step -
accuracy: 0.7489 - loss: 0.5166
Epoch 106/150
77/77             0s 1ms/step -
accuracy: 0.7356 - loss: 0.5477
Epoch 107/150
```

```
77/77             0s 1ms/step -
accuracy: 0.7622 - loss: 0.5013
Epoch 108/150
77/77             0s 1ms/step -
accuracy: 0.7574 - loss: 0.5195
Epoch 109/150
77/77             0s 2ms/step -
accuracy: 0.7610 - loss: 0.5027
Epoch 110/150
77/77             0s 2ms/step -
accuracy: 0.7648 - loss: 0.5028
Epoch 111/150
77/77             0s 1ms/step -
accuracy: 0.7818 - loss: 0.4767
Epoch 112/150
77/77             0s 1ms/step -
accuracy: 0.7237 - loss: 0.5204
Epoch 113/150
77/77             0s 1ms/step -
accuracy: 0.7423 - loss: 0.5246
Epoch 114/150
77/77             0s 1ms/step -
accuracy: 0.7395 - loss: 0.5161
Epoch 115/150
77/77             0s 2ms/step -
accuracy: 0.7653 - loss: 0.4972
Epoch 116/150
77/77             0s 1ms/step -
accuracy: 0.7453 - loss: 0.5088
Epoch 117/150
77/77             0s 1ms/step -
accuracy: 0.7535 - loss: 0.5209
Epoch 118/150
77/77             0s 1ms/step -
accuracy: 0.7678 - loss: 0.4802
Epoch 119/150
77/77             0s 1ms/step -
accuracy: 0.7745 - loss: 0.4844
Epoch 120/150
77/77             0s 1ms/step -
accuracy: 0.7482 - loss: 0.5191
Epoch 121/150
77/77             0s 2ms/step -
accuracy: 0.7405 - loss: 0.5076
Epoch 122/150
77/77             0s 1ms/step -
accuracy: 0.7573 - loss: 0.5113
Epoch 123/150
```

```
77/77             0s 2ms/step -
accuracy: 0.7745 - loss: 0.4975
Epoch 124/150
77/77             0s 1ms/step -
accuracy: 0.7560 - loss: 0.5047
Epoch 125/150
77/77             0s 1ms/step -
accuracy: 0.7435 - loss: 0.5195
Epoch 126/150
77/77             0s 1ms/step -
accuracy: 0.7584 - loss: 0.5284
Epoch 127/150
77/77             0s 1ms/step -
accuracy: 0.7877 - loss: 0.4698
Epoch 128/150
77/77             0s 2ms/step -
accuracy: 0.7537 - loss: 0.4971
Epoch 129/150
77/77             0s 2ms/step -
accuracy: 0.7490 - loss: 0.5128
Epoch 130/150
77/77             0s 1ms/step -
accuracy: 0.7496 - loss: 0.5155
Epoch 131/150
77/77             0s 1ms/step -
accuracy: 0.7657 - loss: 0.4977
Epoch 132/150
77/77             0s 1ms/step -
accuracy: 0.7383 - loss: 0.5295
Epoch 133/150
77/77             0s 2ms/step -
accuracy: 0.7777 - loss: 0.4872
Epoch 134/150
77/77             0s 2ms/step -
accuracy: 0.7469 - loss: 0.5060
Epoch 135/150
77/77             0s 1ms/step -
accuracy: 0.7400 - loss: 0.5272
Epoch 136/150
77/77             0s 1ms/step -
accuracy: 0.7632 - loss: 0.4775
Epoch 137/150
77/77             0s 1ms/step -
accuracy: 0.7632 - loss: 0.4804
Epoch 138/150
77/77             0s 1ms/step -
accuracy: 0.7529 - loss: 0.4887
Epoch 139/150
```

```
77/77             0s 2ms/step -
accuracy: 0.7582 - loss: 0.5150
Epoch 140/150
77/77             0s 2ms/step -
accuracy: 0.7379 - loss: 0.5041
Epoch 141/150
77/77             0s 1ms/step -
accuracy: 0.7643 - loss: 0.4994
Epoch 142/150
77/77             0s 1ms/step -
accuracy: 0.7501 - loss: 0.5199
Epoch 143/150
77/77             0s 1ms/step -
accuracy: 0.7777 - loss: 0.4968
Epoch 144/150
77/77             0s 2ms/step -
accuracy: 0.7440 - loss: 0.5082
Epoch 145/150
77/77             0s 1ms/step -
accuracy: 0.7834 - loss: 0.4723
Epoch 146/150
77/77             0s 2ms/step -
accuracy: 0.7448 - loss: 0.4953
Epoch 147/150
77/77             0s 1ms/step -
accuracy: 0.7777 - loss: 0.4749
Epoch 148/150
77/77             0s 2ms/step -
accuracy: 0.7714 - loss: 0.4911
Epoch 149/150
77/77             0s 2ms/step -
accuracy: 0.7418 - loss: 0.4972
Epoch 150/150
77/77             0s 2ms/step -
accuracy: 0.7958 - loss: 0.4624
```

[9]:
```
<keras.src.callbacks.history.History at 0x7b4e1e64bca0>
```

[10]:
```python
# Step 5 - evaluate the model
model.evaluate(x,y)
```

```
24/24             0s 1ms/step -
accuracy: 0.7436 - loss: 0.4986
```

[10]:
```
[0.480383962392807, 0.7708333134651184]
```