- Corrida con

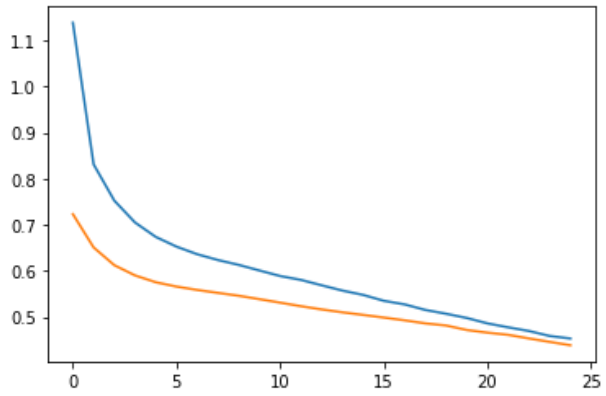
```
model.compile(loss="binary_crossentropy", optimizer="rmsprop",
metrics=[tf.keras.metrics.Accuracy()])
```

```
myInputSize = features_train.shape[1]

#arquitectura red neuronal
model = Sequential()
model.add(Dense(20, input_dim = myInputSize))
model.add(Activation("sigmoid"))
model.add(Dense(1))
model.add(Activation("relu"))

model.compile(loss="binary_crossentropy", optimizer="rmsprop", metrics=[tf.keras.metrics.Accuracy()])
```

```
[49] history = model.fit(features_train, target_train, epochs = 25, validation_split = 0.2)
```

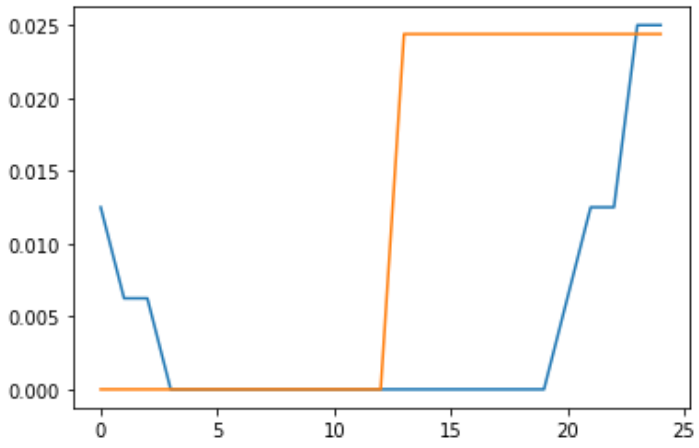
```
[50] plt.plot(history.history["loss"])
     plt.plot(history.history["val_loss"])
```

```
[<matplotlib.lines.Line2D at 0x7f21b728b210>]
```




```
plt.plot(history.history["accuracy"])
plt.plot(history.history["val_accuracy"])
```

```
[<matplotlib.lines.Line2D at 0x7f21b72459d0>]
```




```
[56] print(cm)
```

```
[[26  2]
 [ 3 20]]
```

ACCURACY = 46/51 = 0.9

- Corrida con

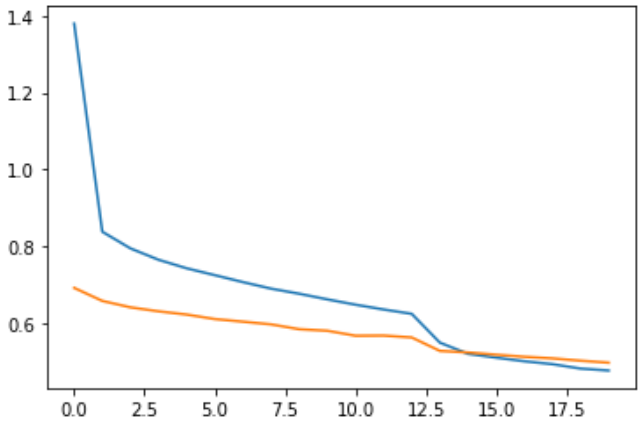
```
model.compile(loss="binary_crossentropy", optimizer="rmsprop", metrics=["accuracy"])
```

```
[64] myInputSize = features_train.shape[1]

     #arquitectura red neuronal
     model = Sequential()
     model.add(Dense(20, input_dim = myInputSize))
     model.add(Activation("sigmoid"))
     model.add(Dense(1))
     model.add(Activation("relu"))

     #model.compile(loss="binary_crossentropy", optimizer="rmsprop", metrics=[tf.keras.metrics.Accuracy()])
     model.compile(loss="binary_crossentropy", optimizer="rmsprop", metrics=["accuracy"])
```

```
[65] history = model.fit(features_train, target_train, epochs = 20, validation_split = 0.2)
```

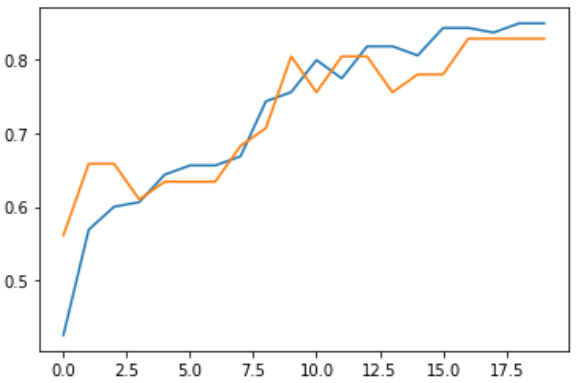
```
plt.plot(history.history["loss"])
plt.plot(history.history["val_loss"])
```

```
[<matplotlib.lines.Line2D at 0x7f21b3586c90>]
```




```
plt.plot(history.history["accuracy"])
plt.plot(history.history["val_accuracy"])
```

```
[<matplotlib.lines.Line2D at 0x7f21b352f8d0>]
```




```
print(cm)
```

```
[[19  2]
 [10 20]]
```

ACCURACY = 39/51= 0.765