ResNet 50 + BiLSTM

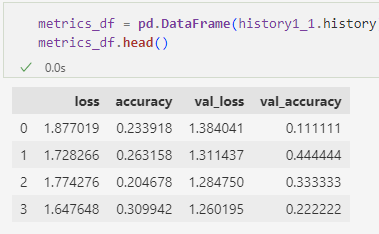
Try 1:

optimizer=**RMSprop**(learning\_rate=2e-5)

batch\_size = 2

epochs = 4

bilstm\_model.add(**Dropout**(0.3))



Try 2:

optimizer=**Adam**(learning\_rate=0.0001)

*# Define a learning rate schedule function*

from **keras**.**callbacks** import **LearningRateScheduler**

def **lr\_schedule**(epoch, current\_lr):

    if epoch < 3:

        return current\_lr  *# Keep the initial learning rate for the first 5 epochs*

    else:

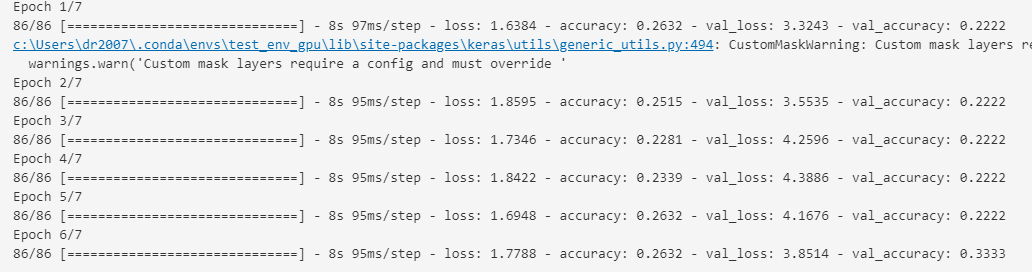
*# Increase the learning rate by 3 times after the 5th epoch*

        return current\_lr \* 0.1

*# Create a learning rate scheduler*

lr\_scheduler = **LearningRateScheduler**(**lr\_schedule**)

bilstm\_model.add(**Dropout**(0.5))

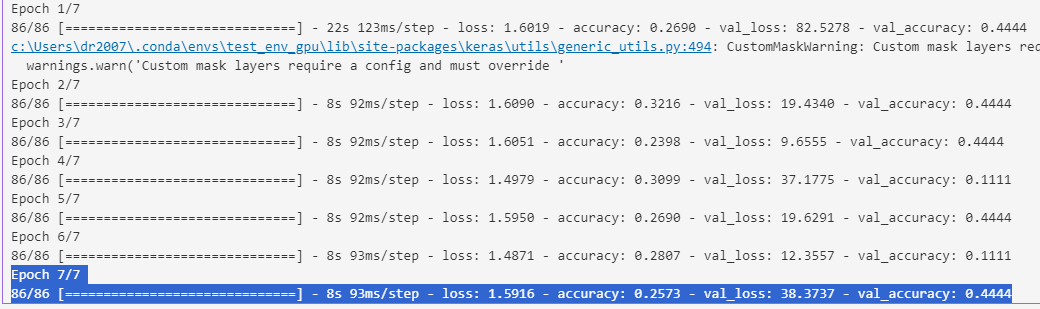


Try 3:

optimizer='adam'

no learning scheduler

bilstm\_model.add(**Dropout**(0.5))



ResNet 50 (Alone)

Try 1:

A screenshot of a computer

Description automatically generated

Try 2:

A table of numbers on a white background

Description automatically generated