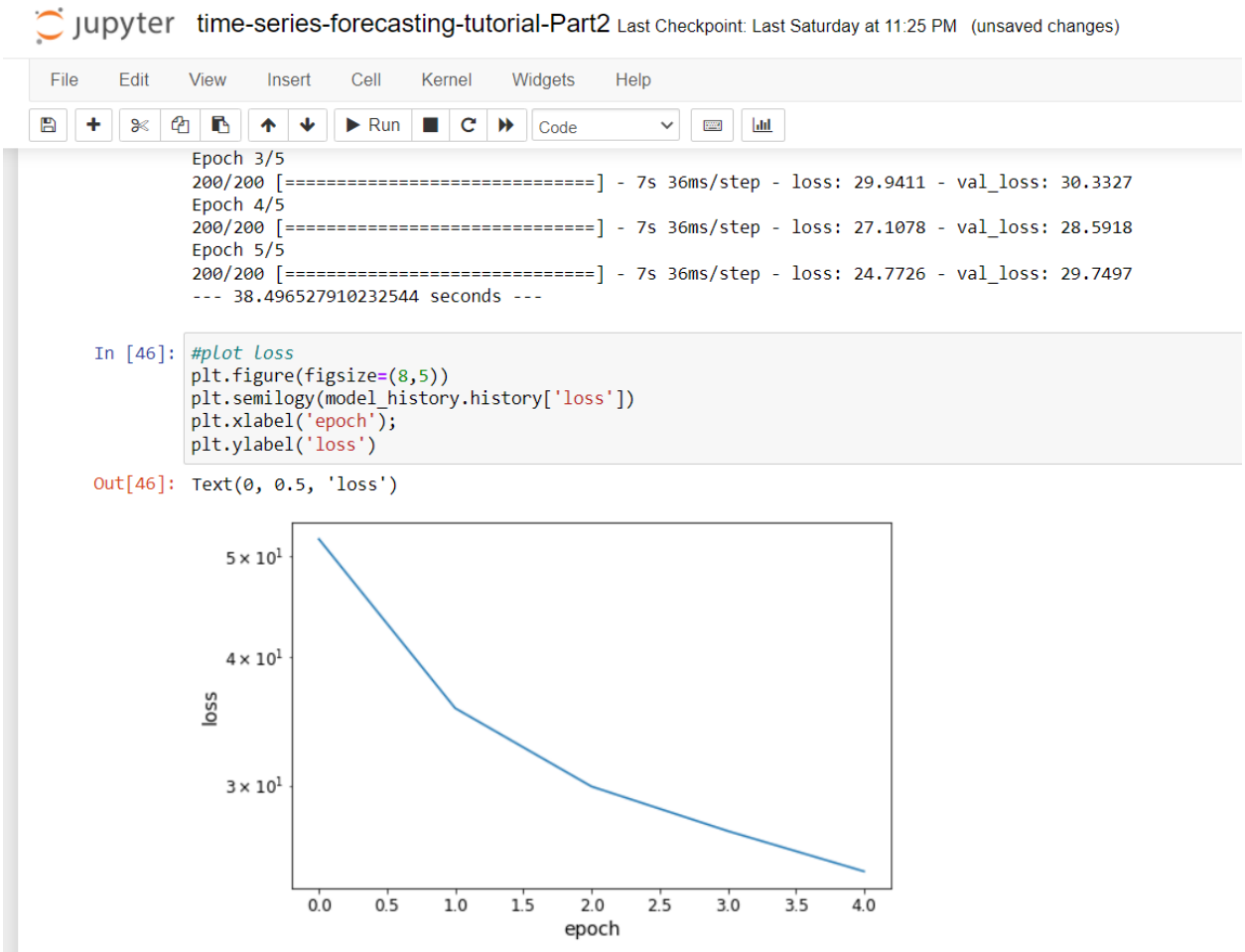
Time Series
Results and Explanation

Mahshid Marashi

Explanation Of Changing Epochs:

Increasing the number of epochs makes runtime worse, although the value of loss is getting better.

All of these experiments and runtime have done in a specific system and same conditions.



INTERVAL200_EPOCHS5

jupyter time-series-forecasting-tutorial-Part2 Last Checkpoint: an hour ago (unsaved changes)

Logout

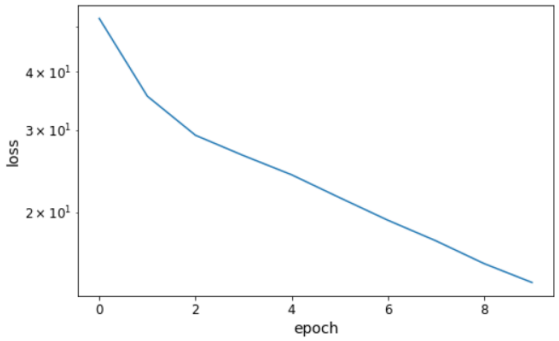
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

```
200/200 [==============================] - 7s 36ms/step - loss: 15.5186 - val_loss: 32.9087
Epoch 10/10
200/200 [==============================] - 7s 35ms/step - loss: 14.1308 - val_loss: 33.8960
--- 80.93412399291992 seconds ---
```

In [46]:
```
#plot loss
plt.figure(figsize=(8,5))
plt.semilogy(model_history.history['loss'])
plt.xlabel('epoch');
plt.ylabel('loss')
```

Out[46]: Text(0, 0.5, 'loss')



INTERVAL200_EPOCHS10

jupyter time-series-forecasting-tutorial-Part2 Last Checkpoint: an hour ago (autosaved)

Logout

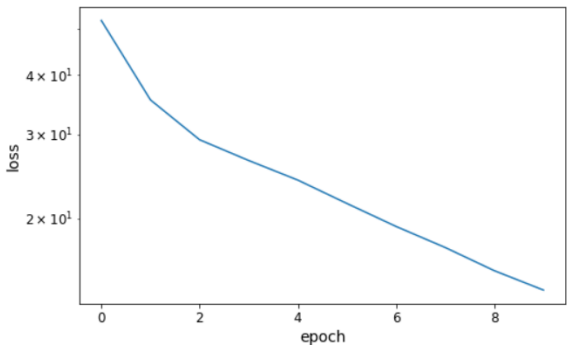
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

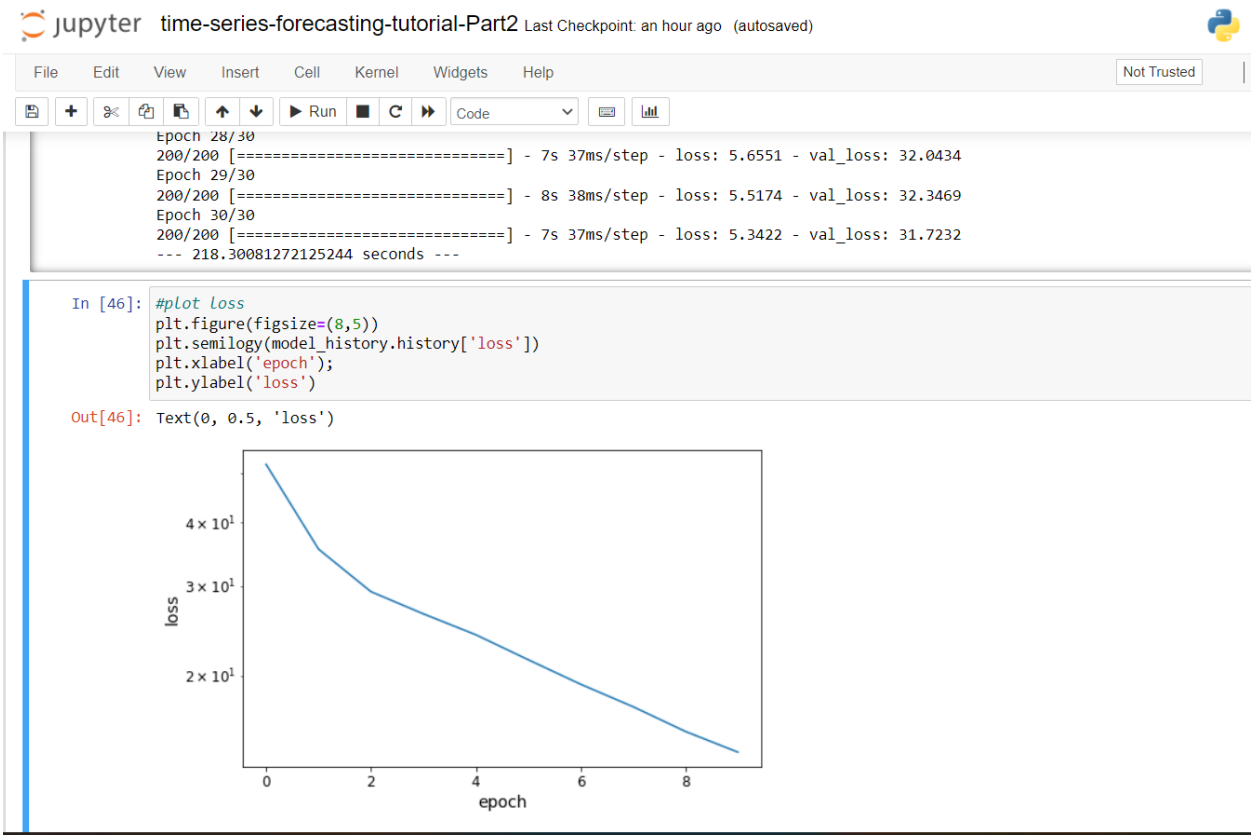
```
Epoch 18/20
200/200 [==============================] - 7s 36ms/step - loss: 8.0482 - val_loss: 34.8575
Epoch 19/20
200/200 [==============================] - 7s 35ms/step - loss: 7.8017 - val_loss: 33.3901
Epoch 20/20
200/200 [==============================] - 7s 36ms/step - loss: 7.4168 - val_loss: 33.1731
--- 145.10245823860168 seconds ---
```

In [46]:
```
#plot loss
plt.figure(figsize=(8,5))
plt.semilogy(model_history.history['loss'])
plt.xlabel('epoch');
plt.ylabel('loss')
```

Out[46]: Text(0, 0.5, 'loss')



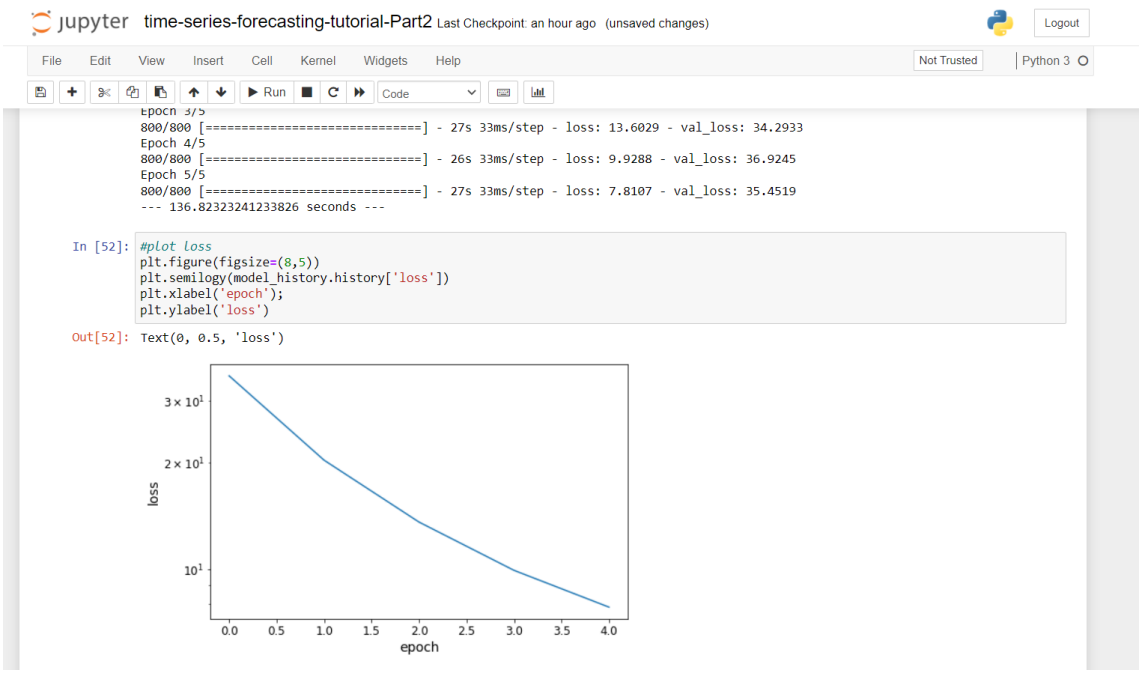
INTERVAL200_EPOCHS20



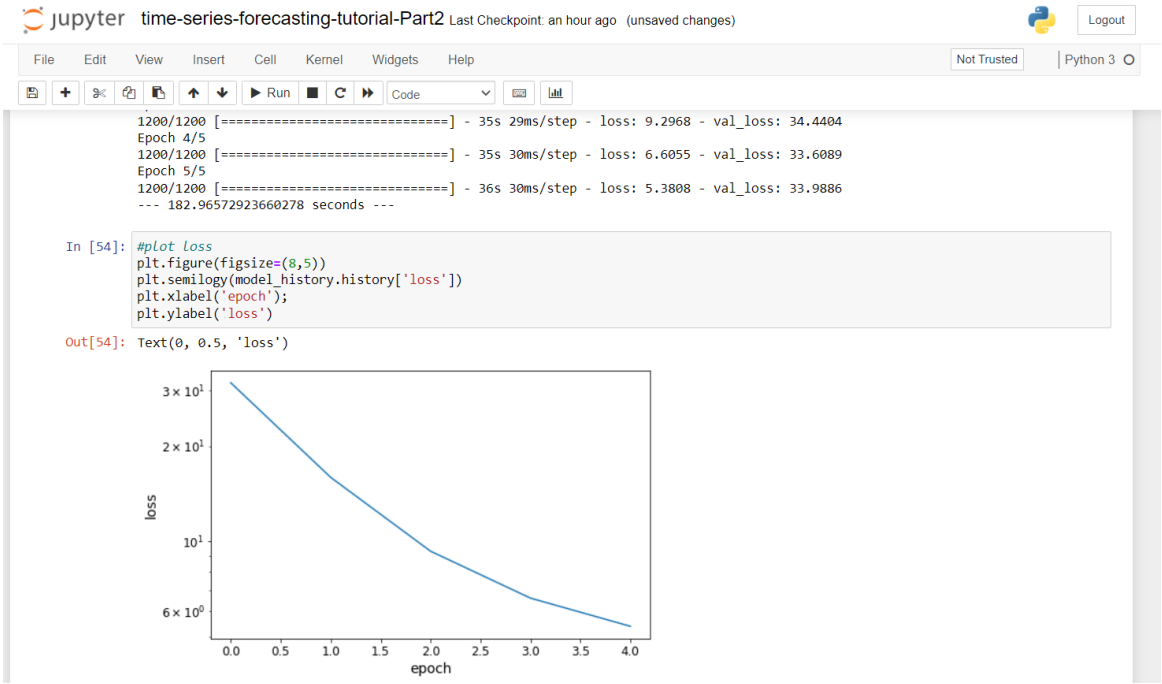
INTERVAL200_EPOCHS30

Explanation Of Changing Intervals:

Like previous part, growing number of Intervals have a positive effect on Loss (sometimes it affects more than increasing epochs), but as you know It would be harmful for runtime as much as increasing epochs.



INTERVAL800_EPOCHS5



INTERVAL1200_EPOCHS5

Adding Dense Layer:

I added a dense layer with 64 and 128 units in model that they could reduce Loss significantly and did not meaningful bad effect on runtime. Also the layer with 64 units reduced Loss more than other.

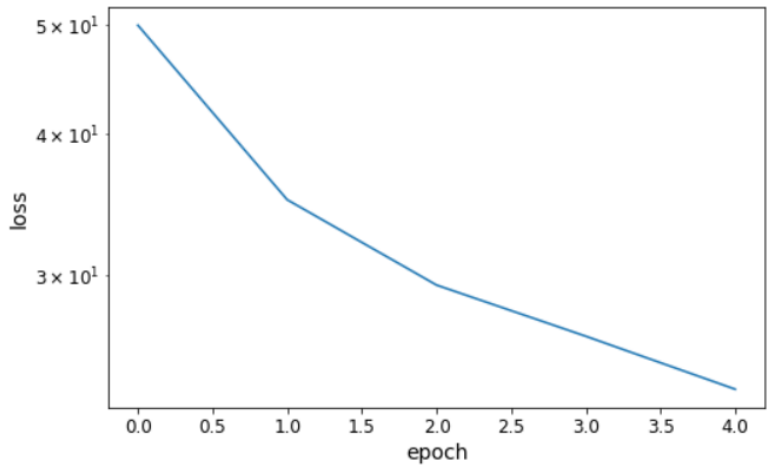
File Edit View Insert Cell Kernel Widgets Help

Run Code

```
Epoch 3/5
200/200 [==============================] - 7s 36ms/step - loss: 29.4022 - val_loss: 30.6426
Epoch 4/5
200/200 [==============================] - 7s 35ms/step - loss: 26.4877 - val_loss: 29.6445
Epoch 5/5
200/200 [==============================] - 7s 36ms/step - loss: 23.7752 - val_loss: 30.8402
--- 38.00011324882507 seconds ---
```

In [42]:
```
#plot loss
plt.figure(figsize=(8,5))
plt.semilogy(model_history.history['loss'])
plt.xlabel('epoch');
plt.ylabel('loss')
```

Out[42]: Text(0, 0.5, 'loss')



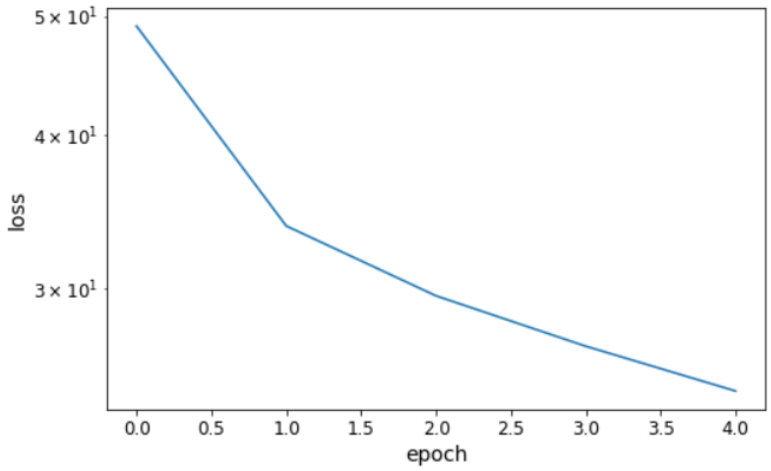
Dense 64

INTERVAL200_EPOCHS5

```
Epoch 3/5
200/200 [==============================] - 7s 35ms/step - loss: 29.6252 - val_loss: 30.2571
Epoch 4/5
200/200 [==============================] - 7s 34ms/step - loss: 26.9586 - val_loss: 29.4751
Epoch 5/5
200/200 [==============================] - 7s 36ms/step - loss: 24.7901 - val_loss: 29.5620
--- 37.251147985458374 seconds ---
```

In [44]:
```
#plot loss
plt.figure(figsize=(8,5))
plt.semilogy(model_history.history['loss'])
plt.xlabel('epoch');
plt.ylabel('loss')
```

Out[44]: Text(0, 0.5, 'loss')

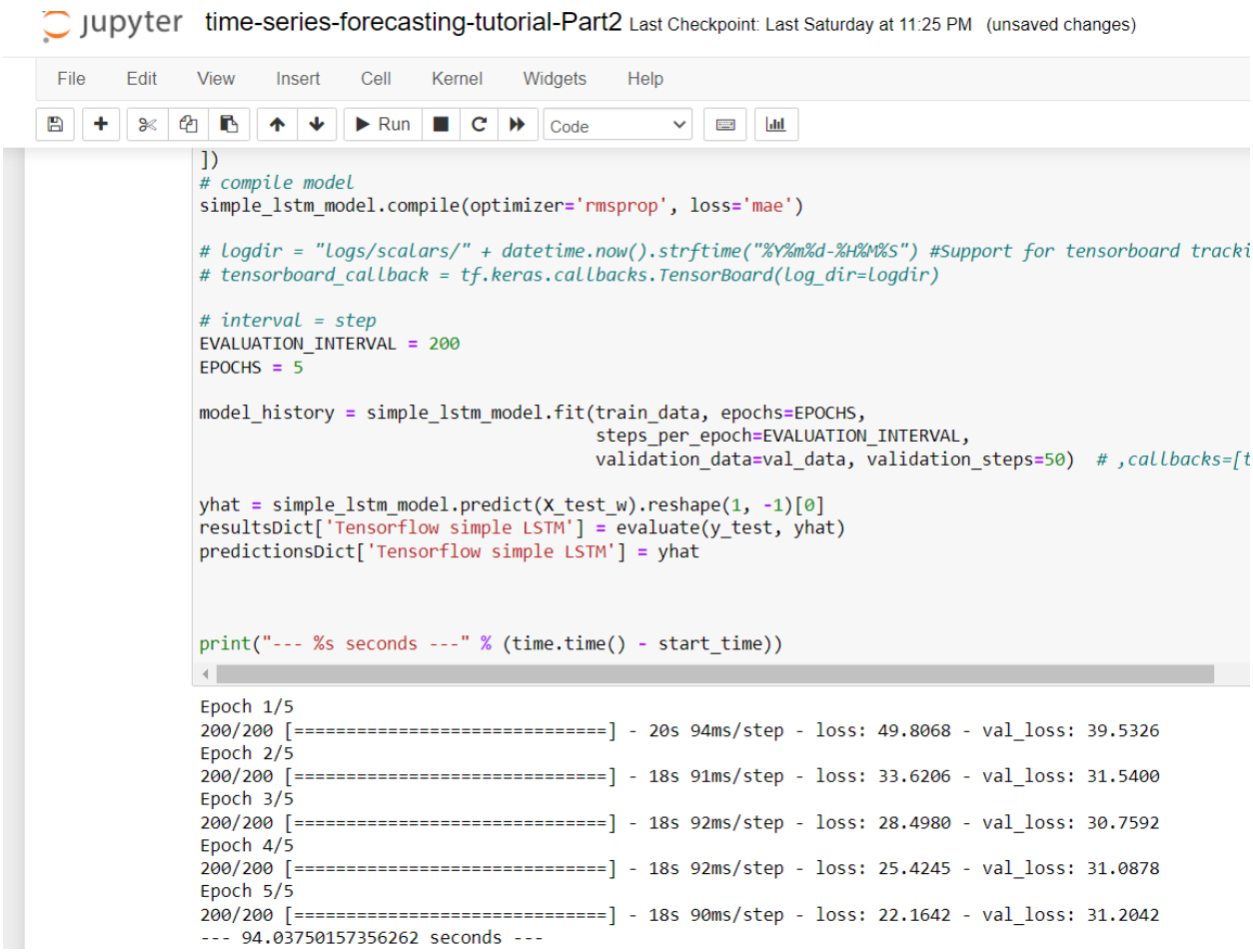


Dense 128

INTERVAL200_EPOCHS5

Adding Units In LSTM Layer:

I doubled units to 256 to see if it had positive effect or not and I realized that the Loss became much better but runtime became extremely worser than before.



LSTM 256

INTERVAL200_EPOCHS5