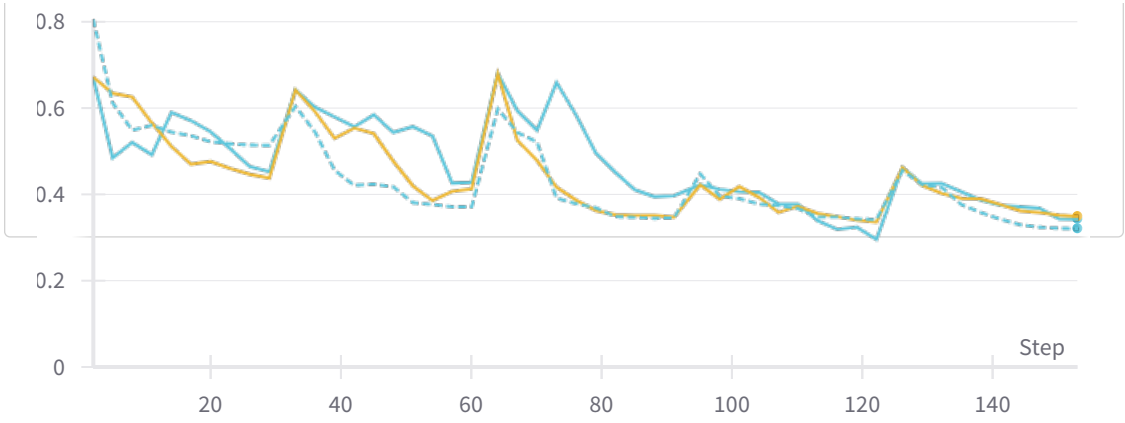
Share Comment Star

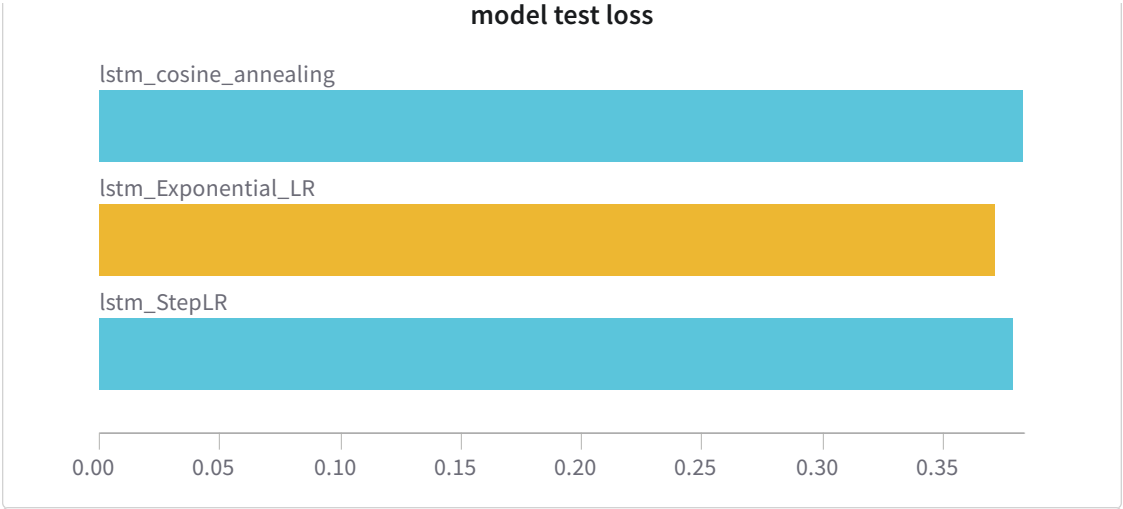
A comparison of LR Schedulers

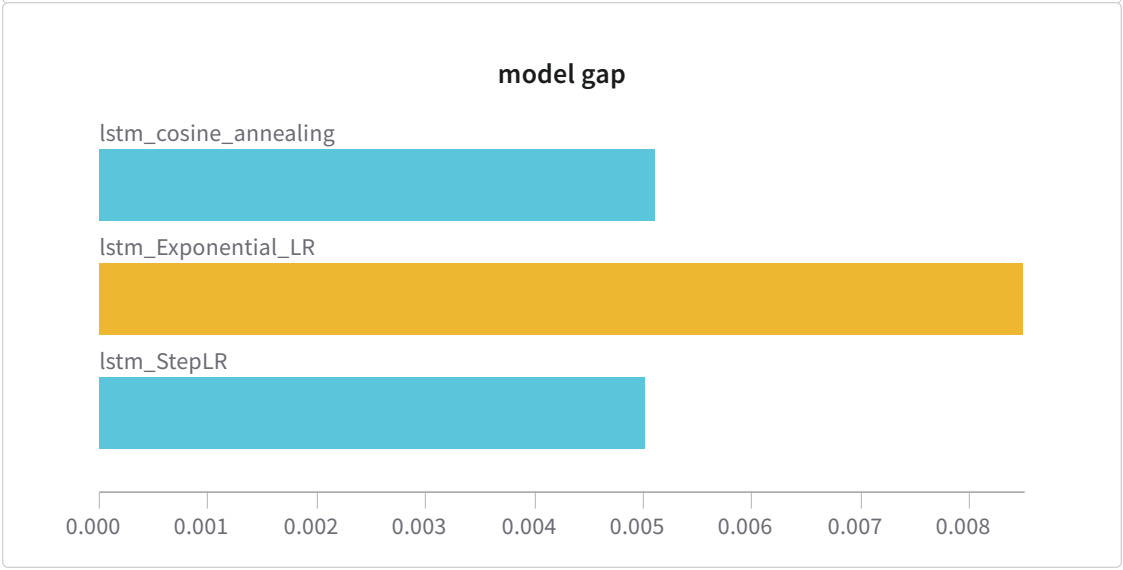
A comparison of learning rate schedulers. Findings were surprising:
- Not a great deal of impact on overall performance basted on average model test loss.
- a large impact on generalization gap. Exponential was much worse than step which surprised me
- A visual inspection of the train - test loss curves shows that StepLR is the most stable. This is not at all what i expected

Michael Kingston

Section 1


0.8
0.6
0.4
0.2
0
Step
20
40
60
80
100
120
140


model test loss
lstm_cosine_annealing
lstm_Exponential_LR
lstm_StepLR
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35


model gap
lstm_cosine_annealing
lstm_Exponential_LR
lstm_StepLR
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008

Fold 4 Loss Curves

Fold 3 Loss Curves

lstm_cosine_annealing lstm_Exponential_LR lstm_StepLR

Fold 2 Loss Curves

lstm_cosine_annealing lstm_Exponential_LR lstm_StepLR

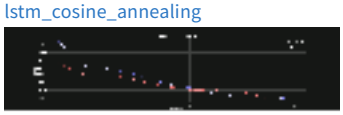
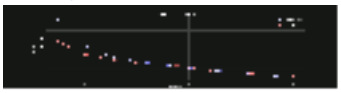
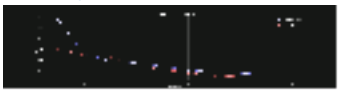
Fold 1 Loss Curves

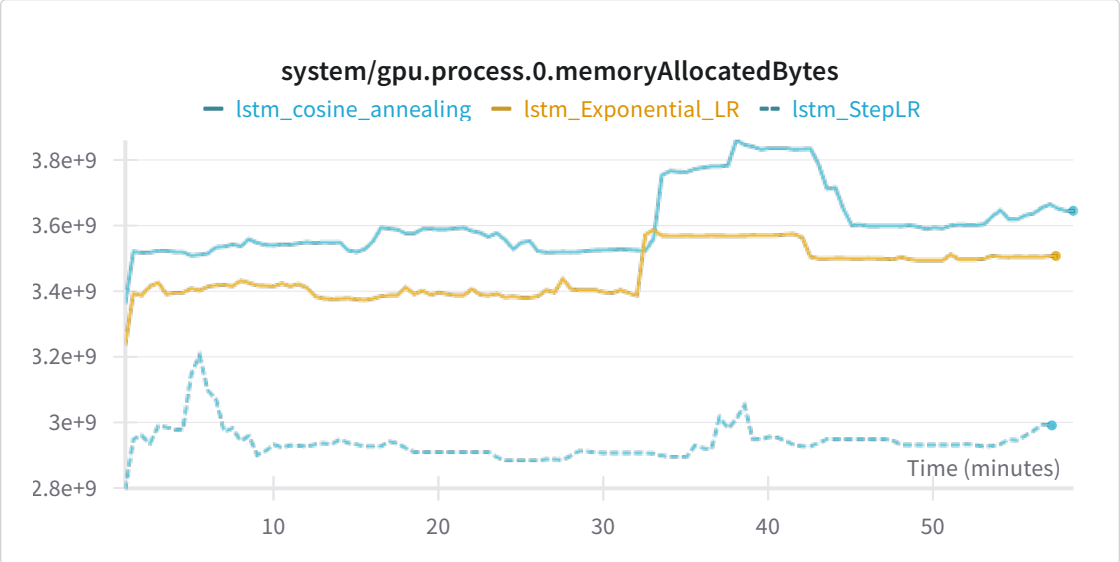
lstm_cosine_annealing lstm_Exponential_LR lstm_StepLR

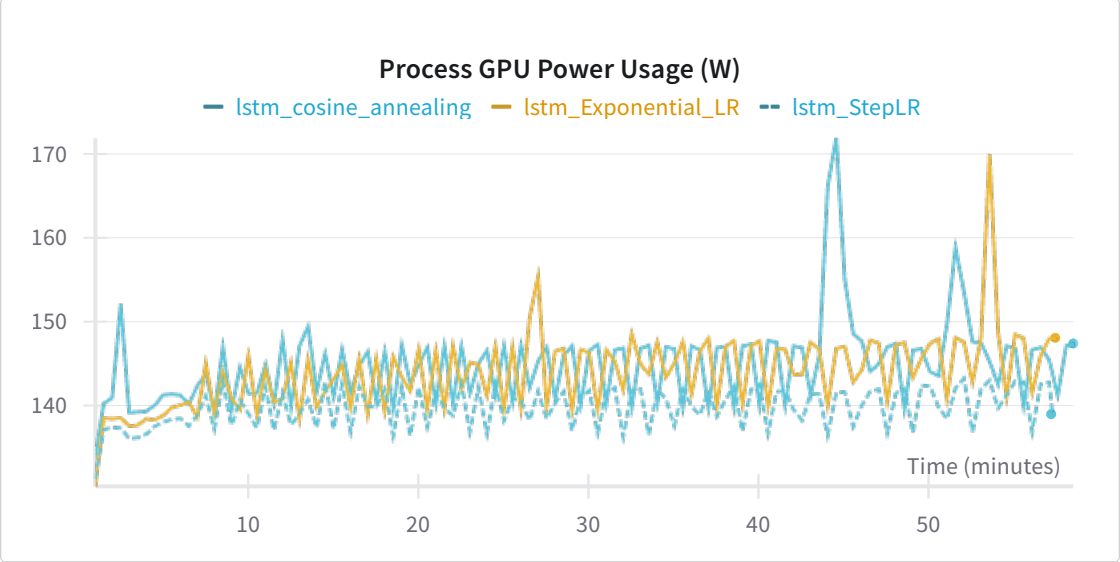
Fold 0 Loss Curves

lstm_cosine_annealing lstm_Exponential_LR lstm_StepLR

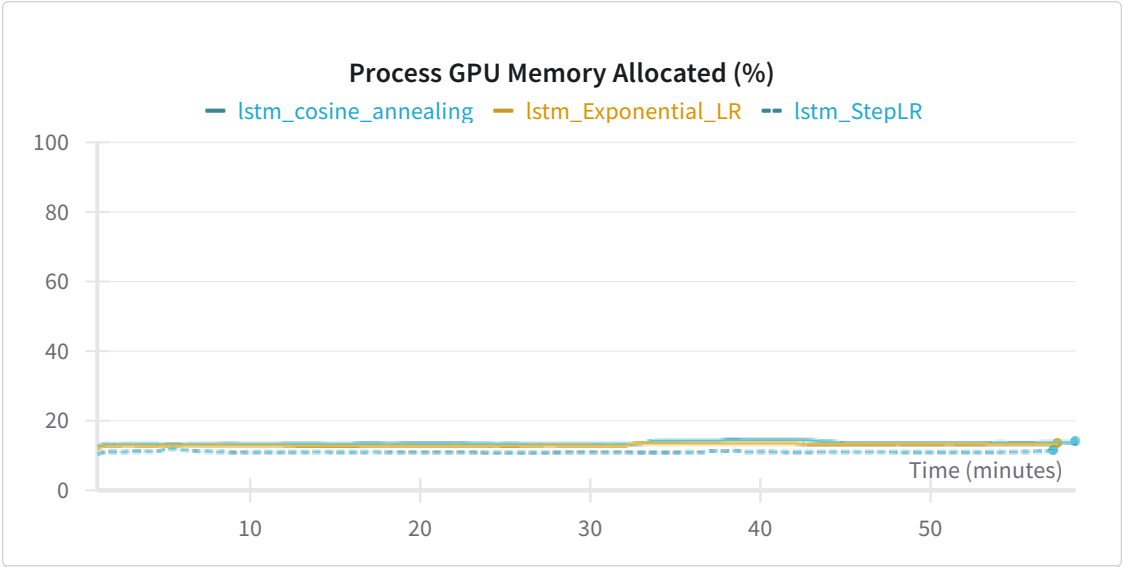
lstm_cosine_annealing lstm_Exponential_LR lstm_StepLR

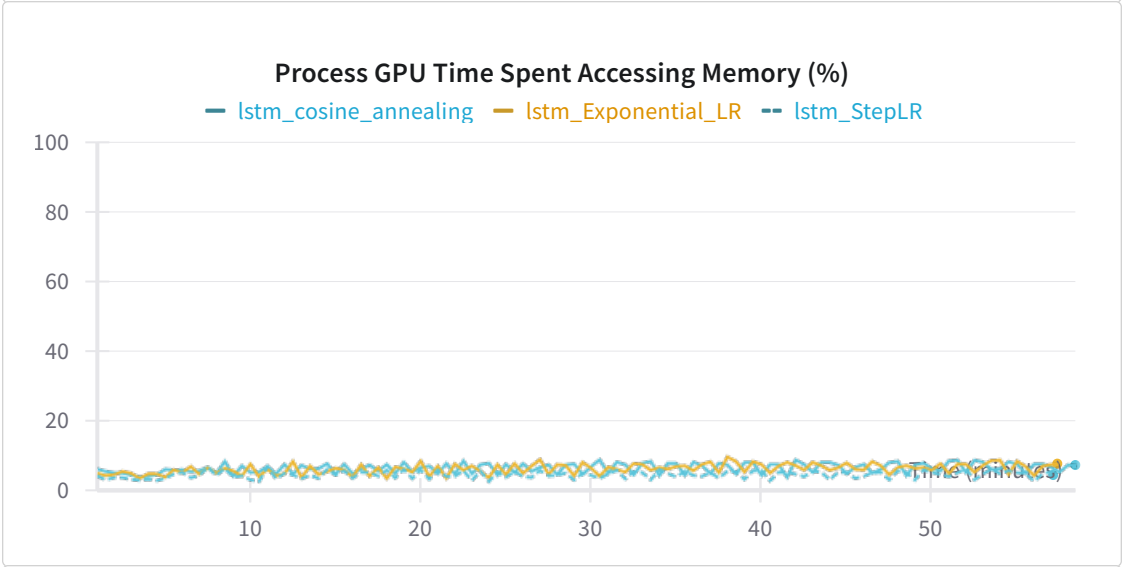




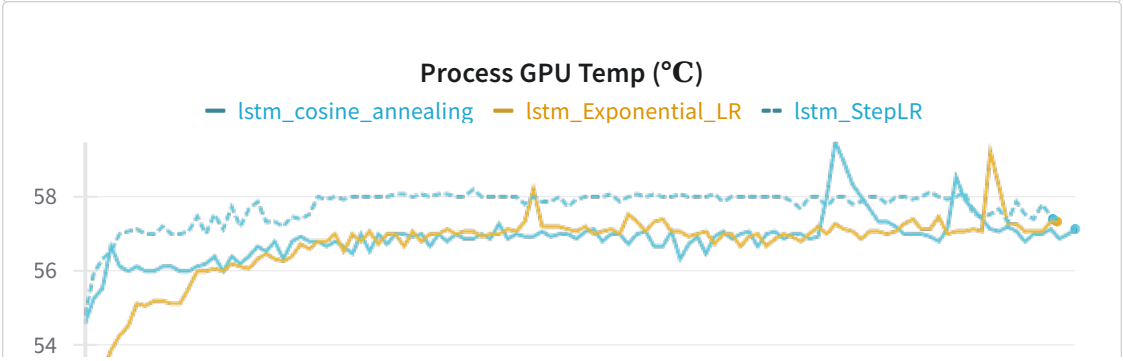

system/gpu.process.0.memoryAllocatedBytes
lstm_cosine_annealing
lstm_Exponential_LR
lstm_StepLR
3.8e+9
3.6e+9
3.4e+9
3.2e+9
3e+9
2.8e+9
Time (minutes)
10
20
30
40
50

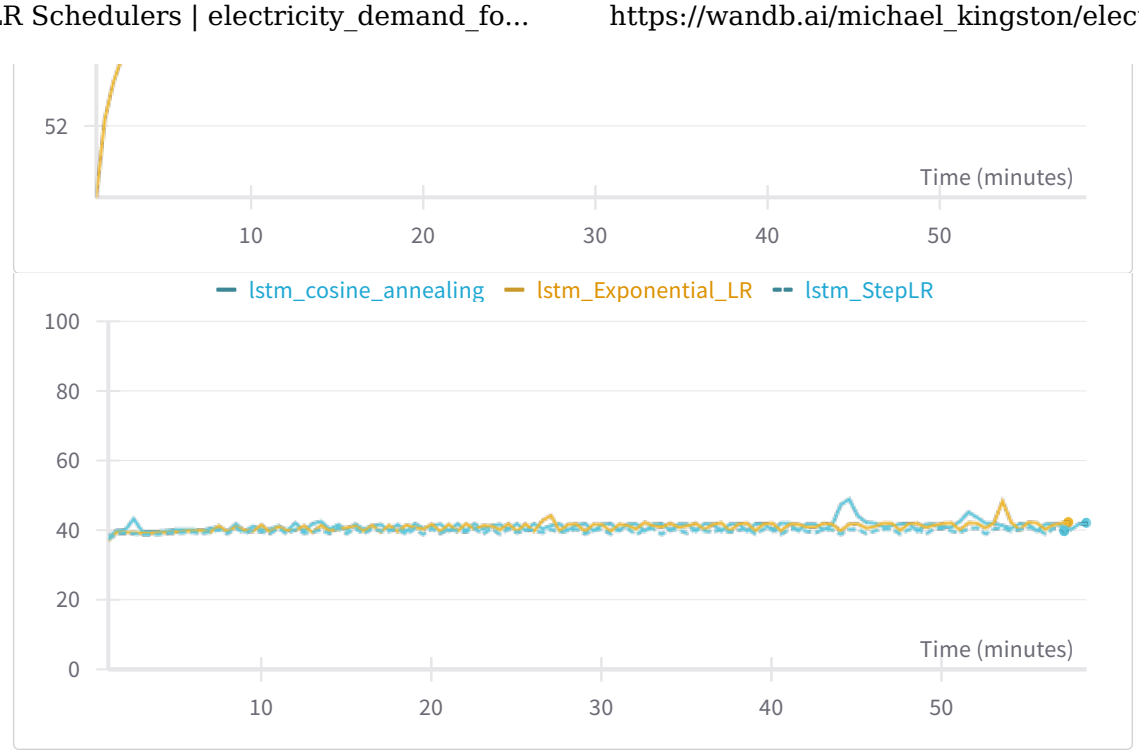

Process GPU Power Usage (W)
lstm_cosine_annealing
lstm_Exponential_LR
lstm_StepLR
170
160
150
140
Time (minutes)
10
20
30
40
50

Process GPU Power Usage (%)








52
Time (minutes)
10
20
30
40
50
lstm_cosine_annealing
lstm_Exponential_LR
lstm_StepLR
100
80
60
40
20
0
Time (minutes)
10
20
30
40
50

Created with ❤ on Weights & Biases.

https://wandb.ai/michael_kingston/electricity_demand_forecasting/reports/A-comparison-of-LR-Schedulers--Vmlldzo3NTM1MzMw