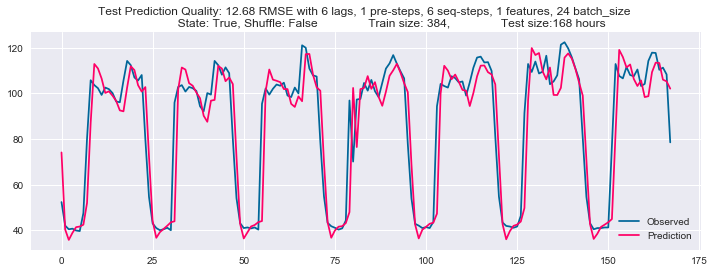
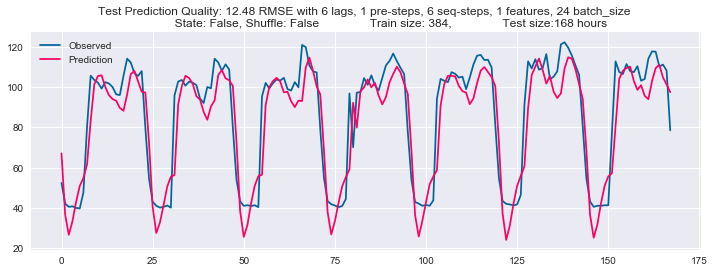
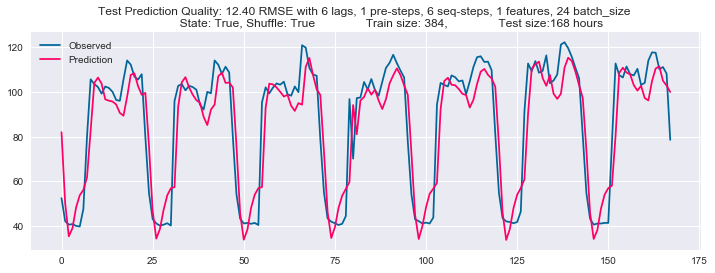
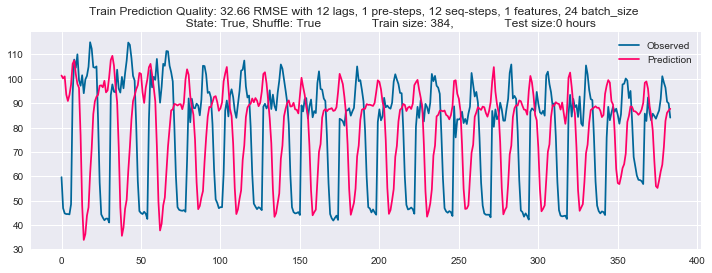


Using Parameters Class



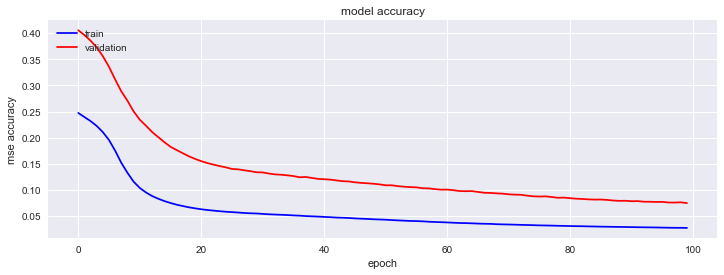




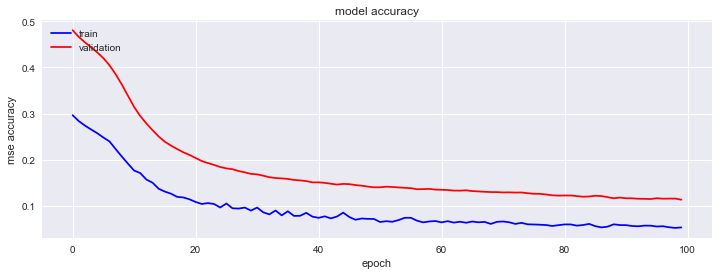


6 pre-steps with "No" state and "True" shuffle:

No Recurrent Ratio = 0.0

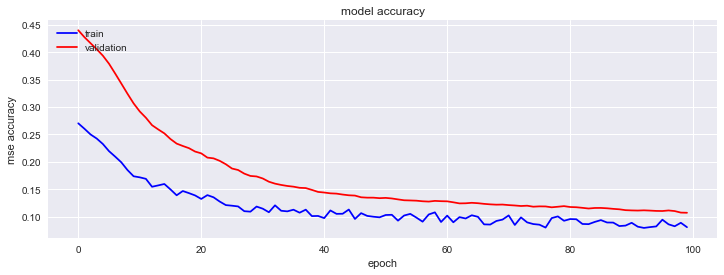


Recurrent Dropout = 0.4



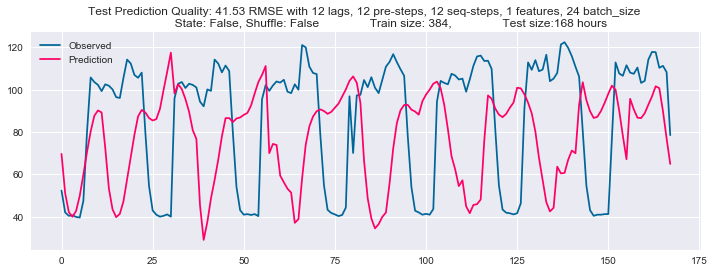
Recurrent Dropout = 0.4

input\_dropout = 0.2



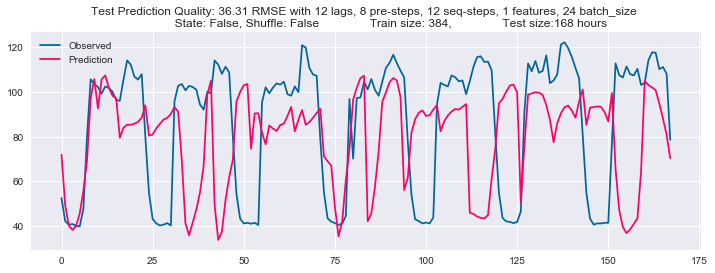
12 pre-steps out of 12 lags with "No" state and "False" shuffle:

Not accurate



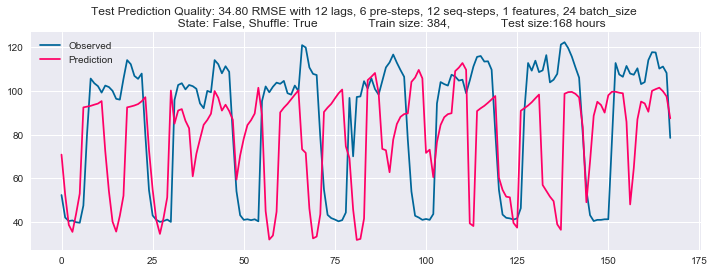
8 pre-steps out of 12 lags with "No" state and "False" shuffle:

more fit



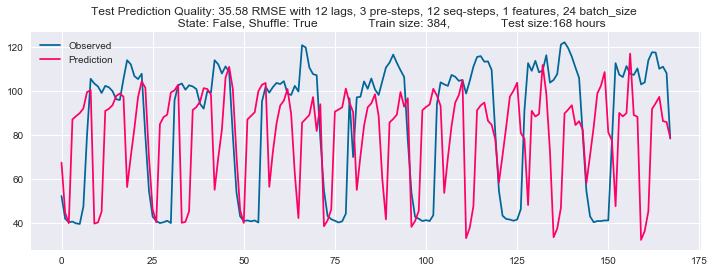
6 pre-steps out of 12 lags with "No" state and "True" shuffle:

2 cycles in 24 hours



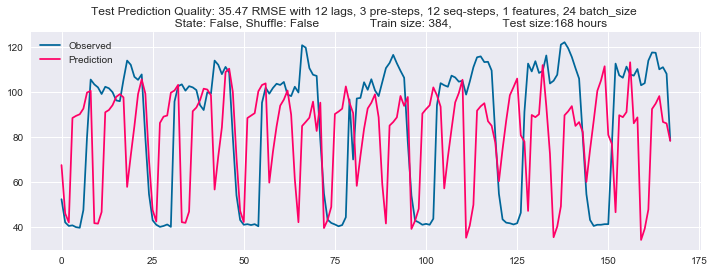
3 pre-steps out of 12 lags with "No" state and "True" shuffle:

3 cycles in 24 hours



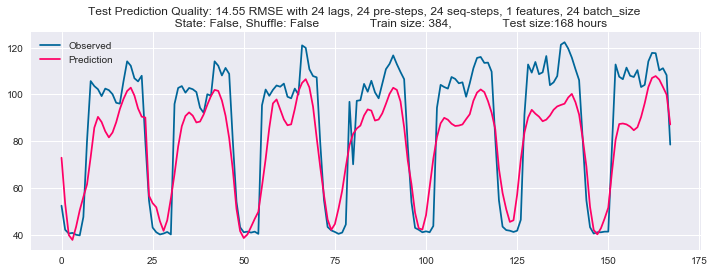
3 pre-steps with "No" state and "False" shuffle:

3 cycles in 24 hours with NO CHANGE



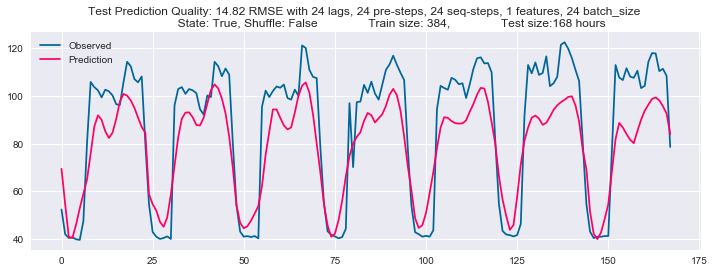
24 pre-steps out of 24 lags with "No" state and "False" shuffle:

Fitter than 12 lags



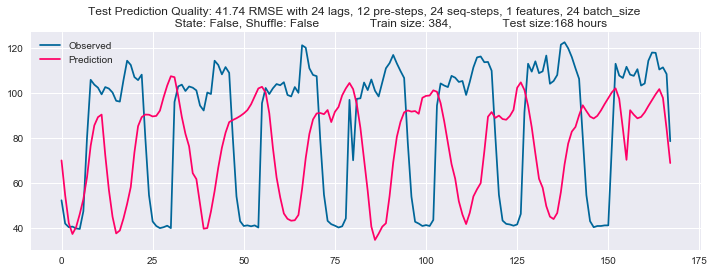
24 pre-steps out of 24 lags with "True" state and "False" shuffle:

similar to being No state



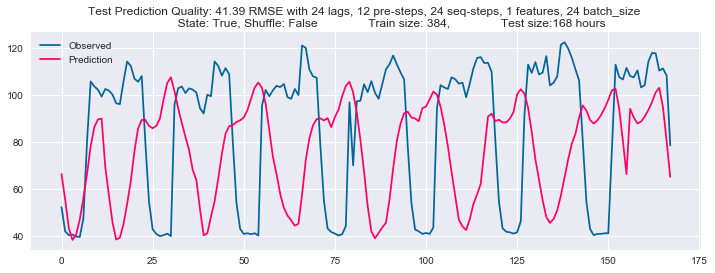
12 pre-steps out of 24 lags with "False" state and "False" shuffle:

2 cycles in 24 hours



12 pre-steps out of 24 lags with "True" state and "False" shuffle:

Similar to being "False" state



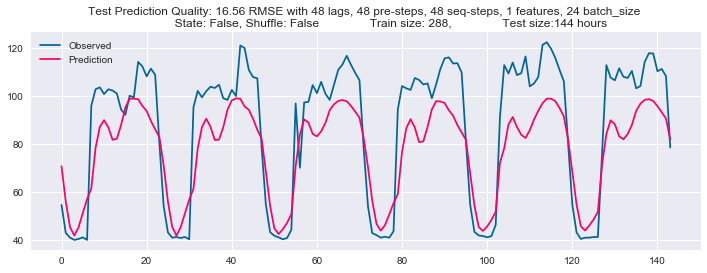
6 pre-steps out of 24 lags with "False" state and "False" shuffle:

more cycles (2 cycles ) than 12 pre-steps



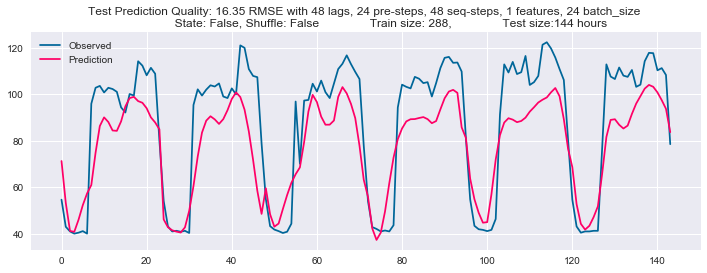
48 pre-steps out of 48 lags with "False" state and "False" shuffle:

16.56 rmse that is less than 14.82 of 24 pre-steps out of 24 lags



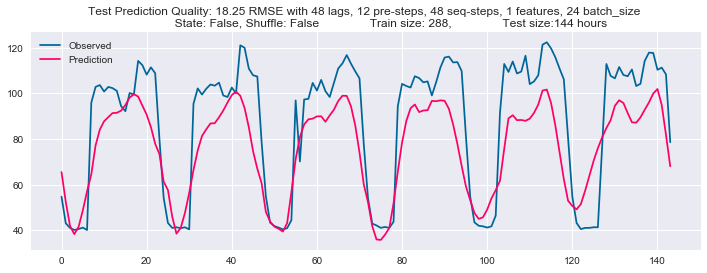
24 pre-steps out of 48 lags with "False" state and "False" shuffle:

16.35 rmse that is less than 14.82 of 24 pre-steps out of 24 lags



12 pre-steps out of 48 lags with "False" state and "False" shuffle:

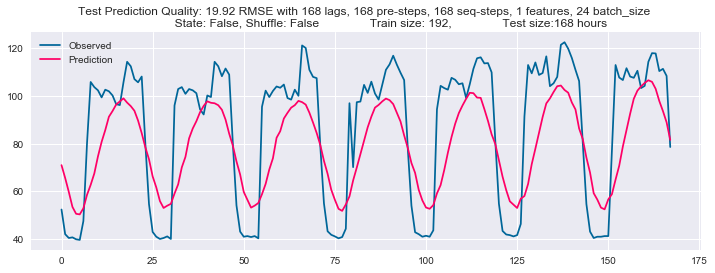
18.35 rmse that is less than 14.82 of 24 pre-steps out of 24 lags



168 pre-steps out of 168 lags with "False" state and "False" shuffle:

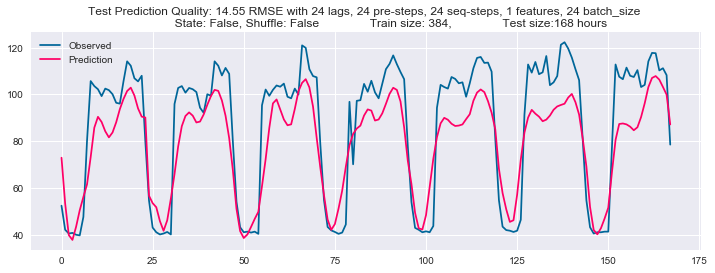
contain just seven smoothing cycles.

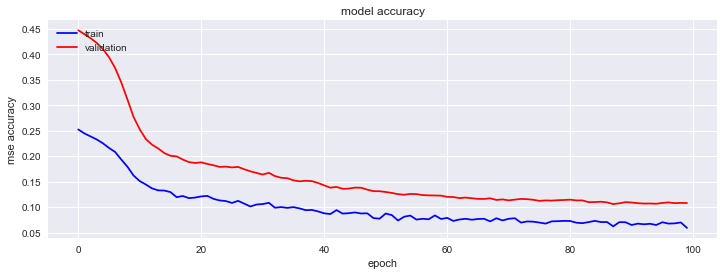
Thus, using 4 weeks’ data, 24 pre-steps out of 24 lags are the best.



24 pre-steps out of 24 lags are the best

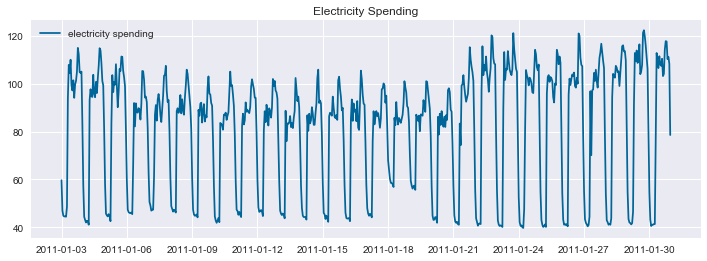
So, I will focus on the input structure



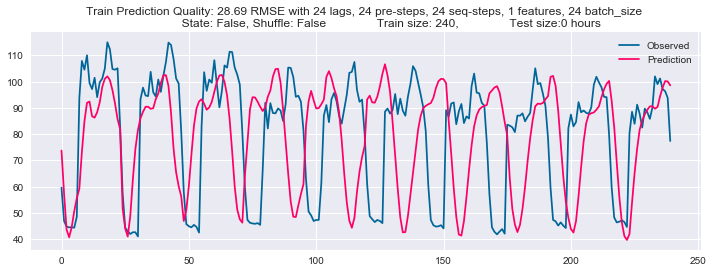


The following is a full data set of 4 weeks. Since Jan 21, spending is higher than before.

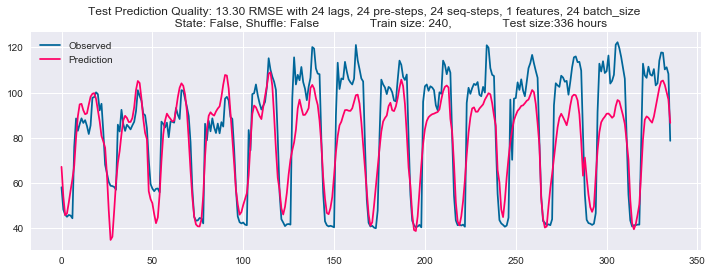
if data before Jan 21 as training is used, data after Jan 21 cannot learn.



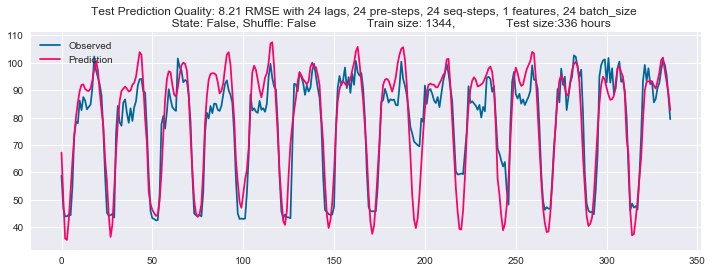
As a training step, a model learns a pattern that is until around 100 spending.



As a test step, until around 100 spending per hour can be predicted. However over 100, it cannot predict because the model would not learn the point over 100



As data size increases to 3 months, the rmse is 8.21

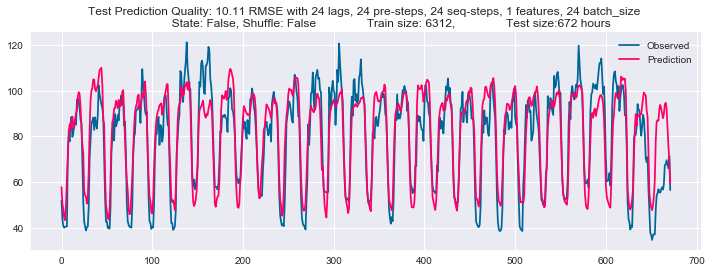


As data increases to 52 weeks, it seems like fitted.

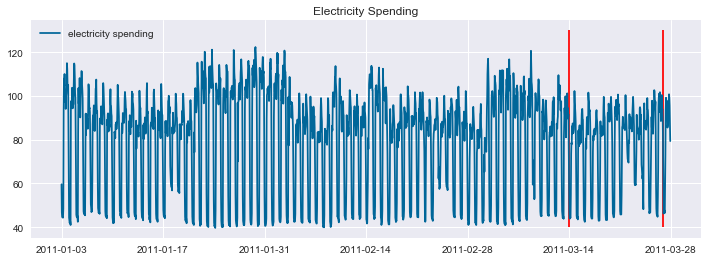
With 10 neurons on one LSTM

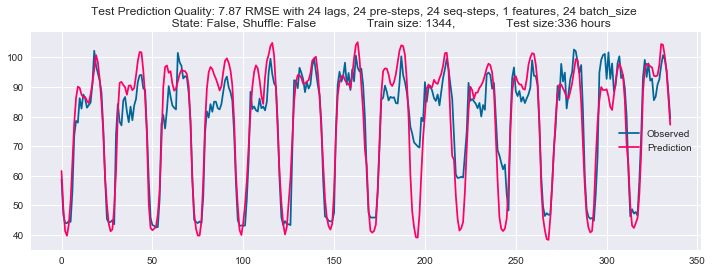


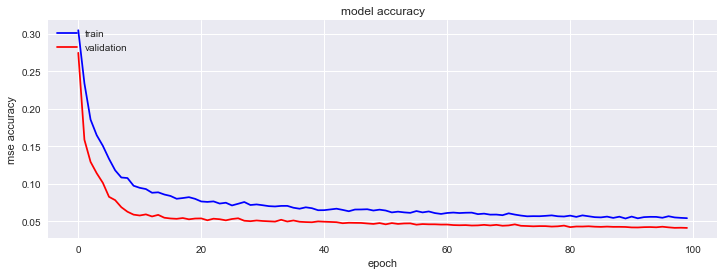
With 20 neurons on one LSTM



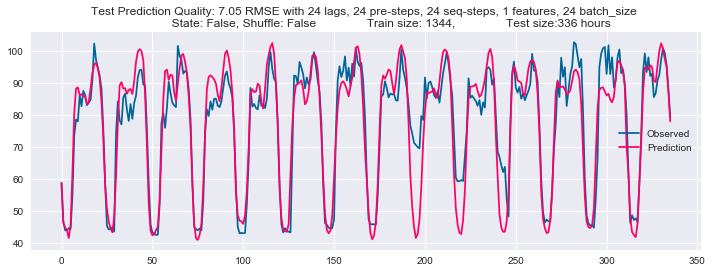
**Because there is no big difference between 3 months and 1 year data, Let me return to 3 months for more optimization**

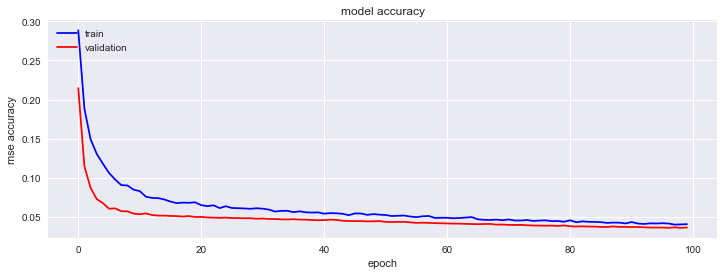






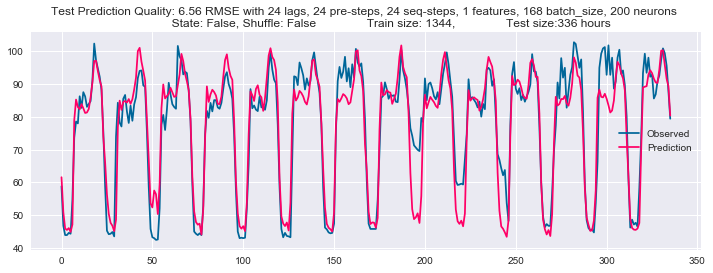
**With two stacked LSTM**





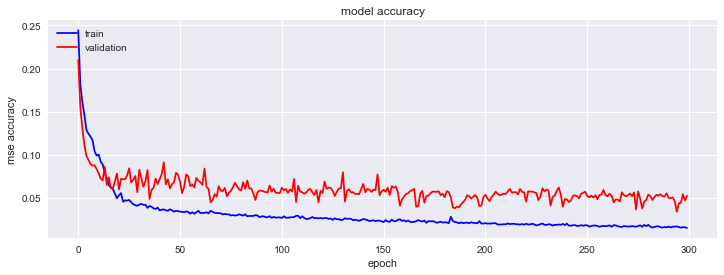


**Recurrent Dropout Ratio = 0.0s**



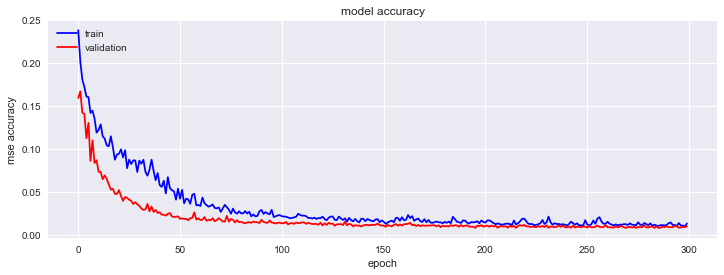
**12 pre-steps out of 12 lags on two LSTM**



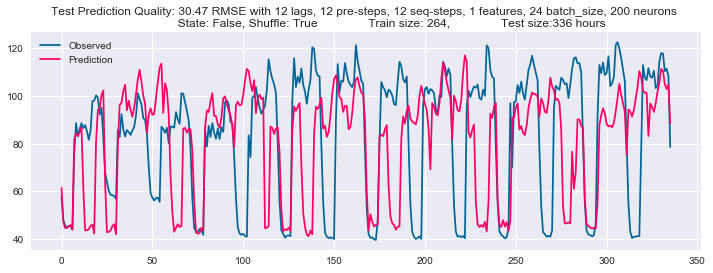


**Because of a learning time, decrease data to 4 weeks with 200 neurons**

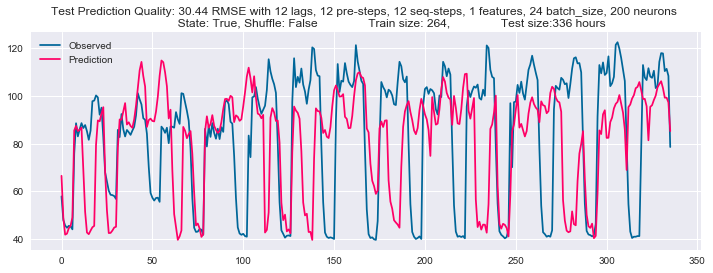




**Shuffle = True**

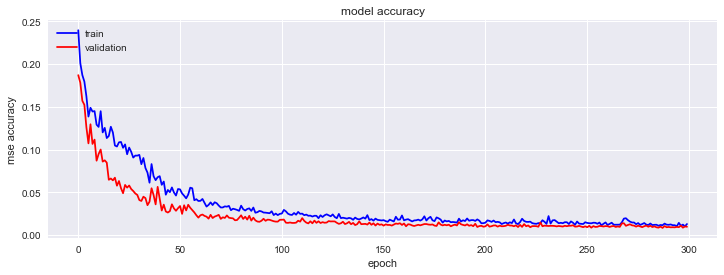


**State = True, Shuffle = False**

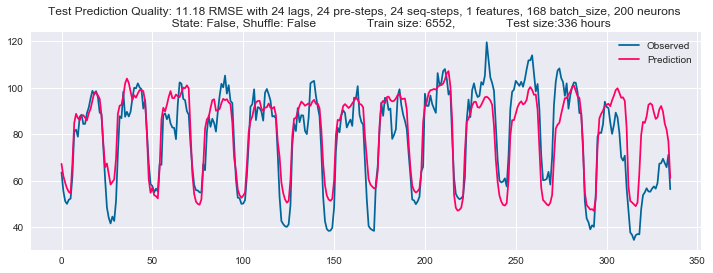


**State = True, Shuffle = True**

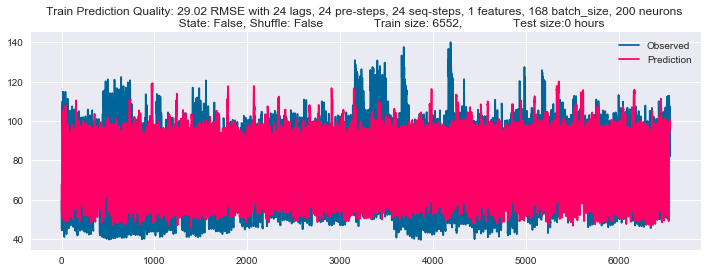


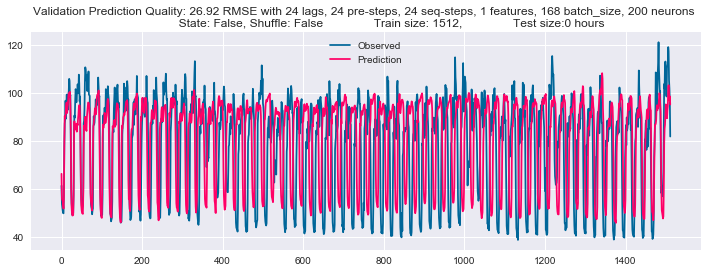


**52 weeks data**

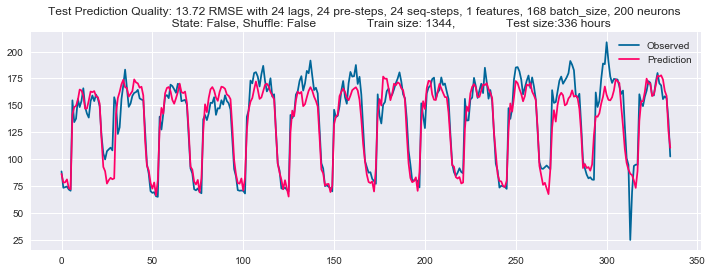


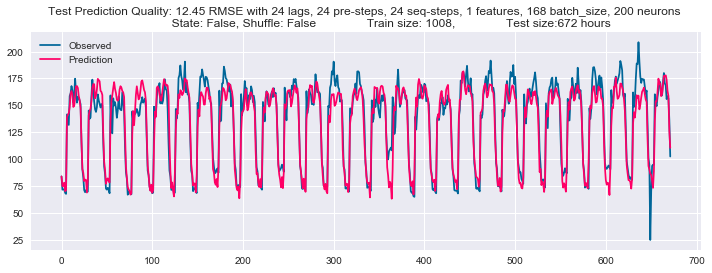




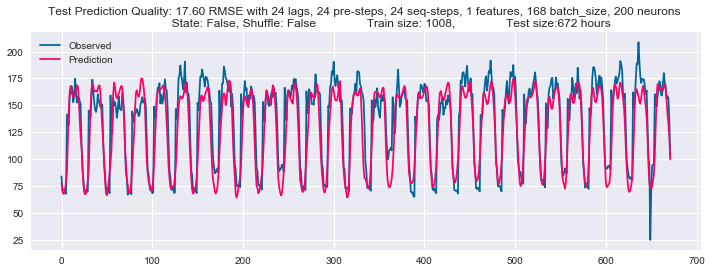


**e235**

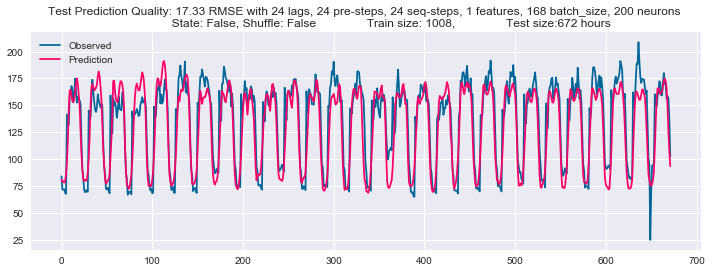




**Integrated Model, State=False, Shuffle = False**

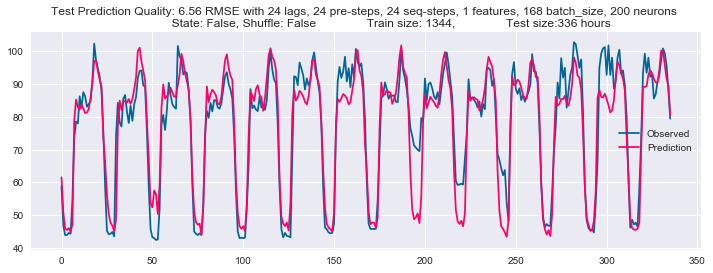


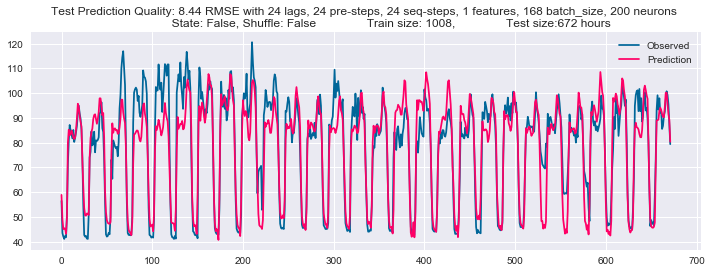
**On an integrated model, state = False, shuffle = True**



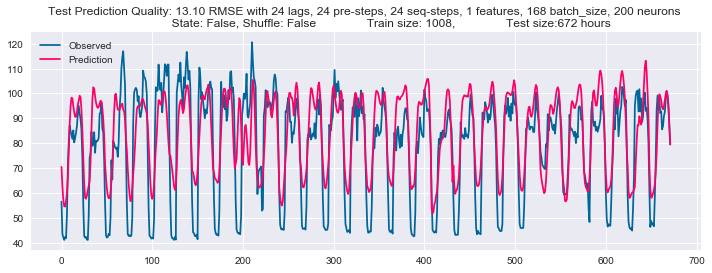
**With two feature such as electricity spending and person\_id**

**e250**

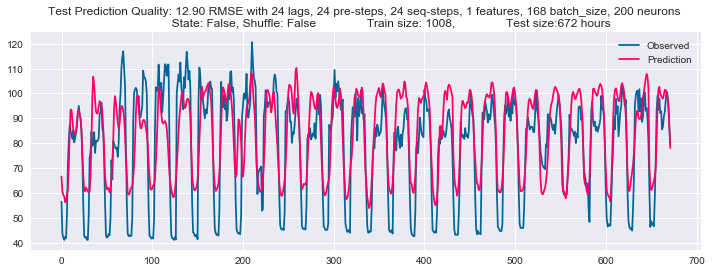




**On an integrated model, state = False, shuffle = False**

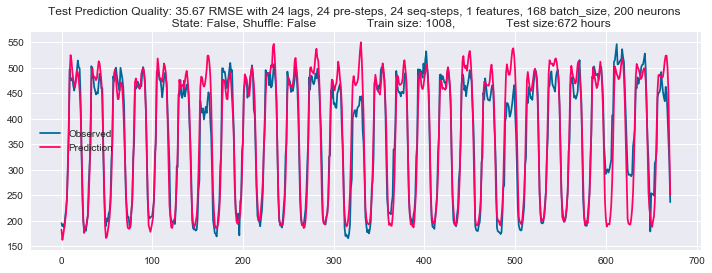


**On an integrated model, state = False, shuffle = True**

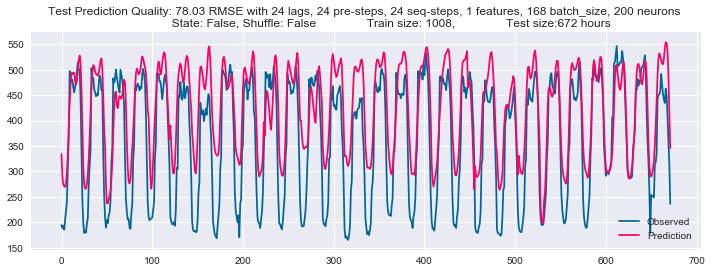


**e252**

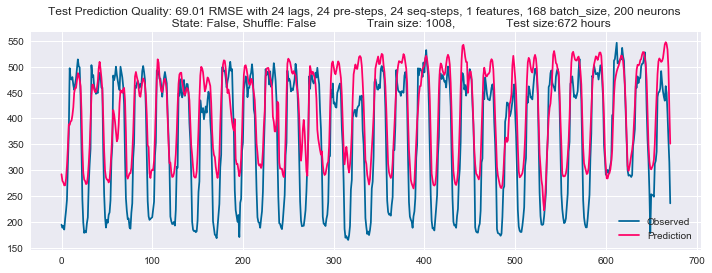




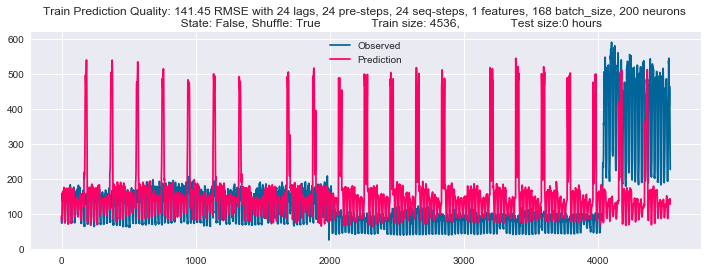
**On a integrated model, state = False, shuffle = False**



**On a integrated model, state = False, shuffle = True**



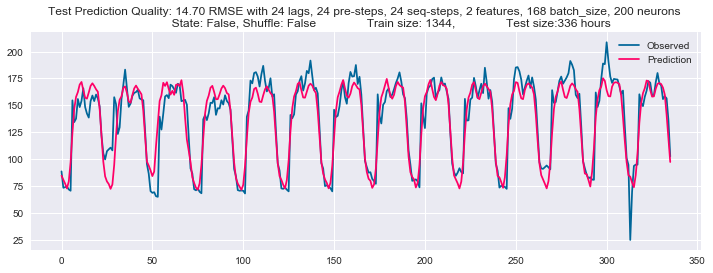
**With one feature such as electricity spending with e235, e250, e252**

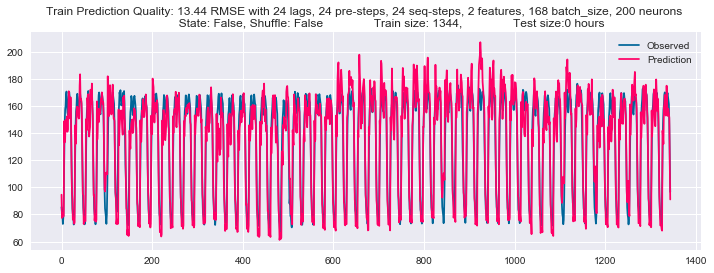


**With two feature such as electricity spending and person\_id**

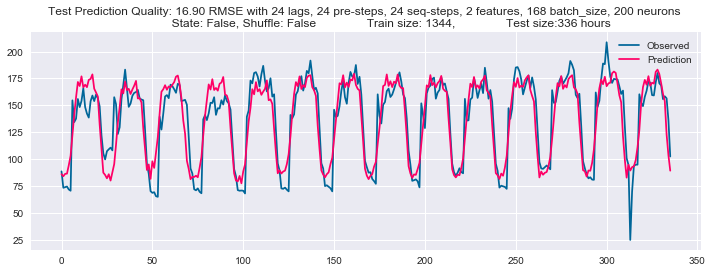
**e235**

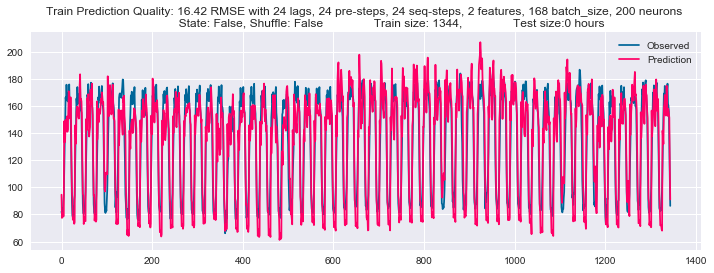
A prediction of a train data is much better than before. person\_id may remove some noise





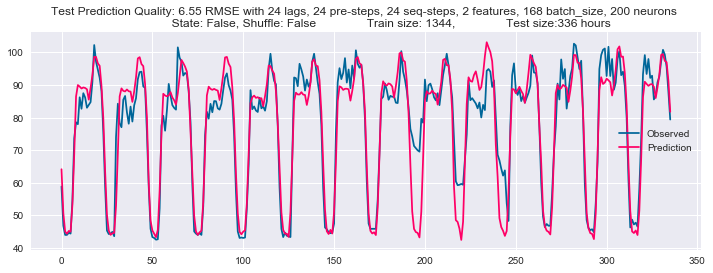
**On a integrated model, state = False, shuffle = False and True**

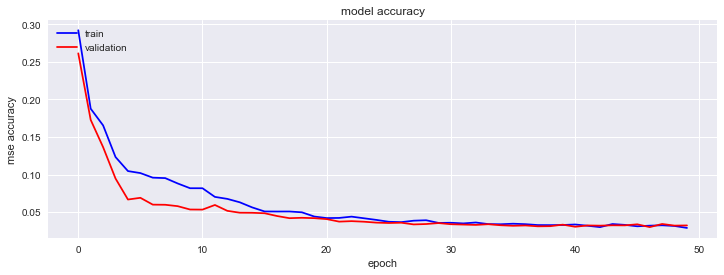


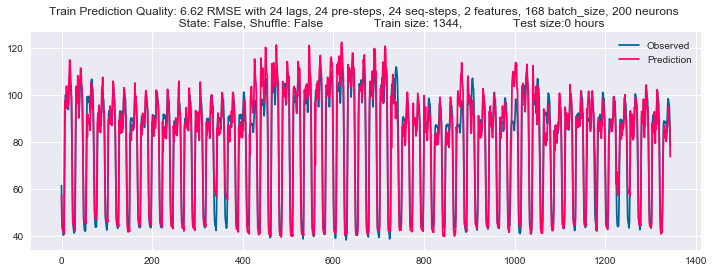


**e250**

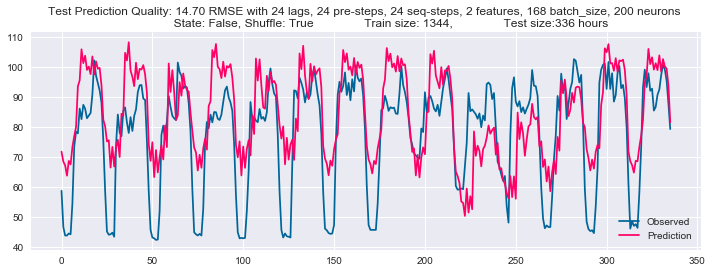
Compared to before, Test and Train prediction are better. Also, a loss graph of a train and validation are more sensible to learn

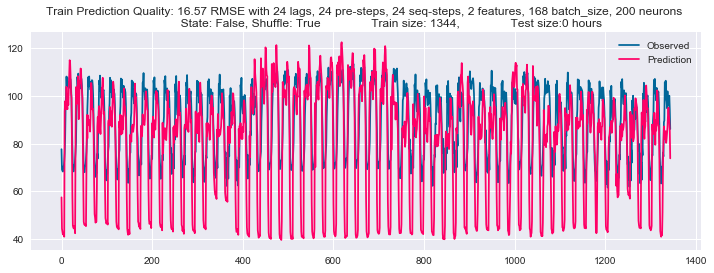




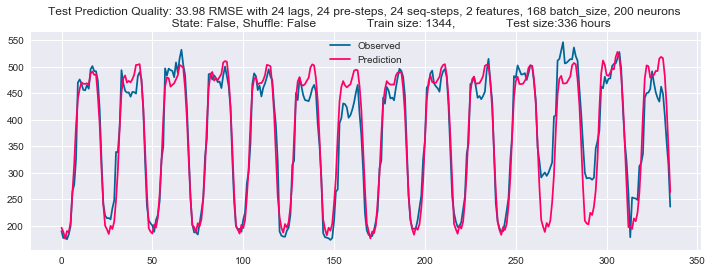


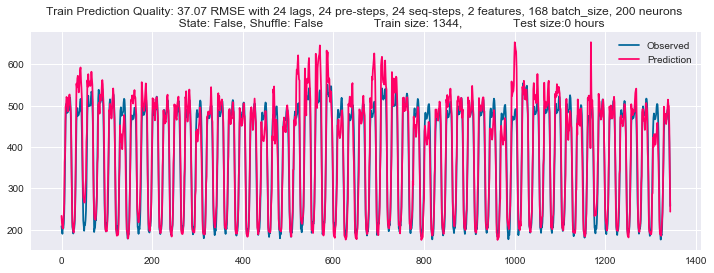
**On a integrated model, state = False, shuffle = False and True**



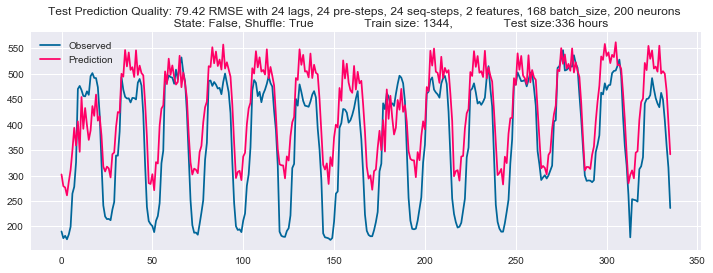


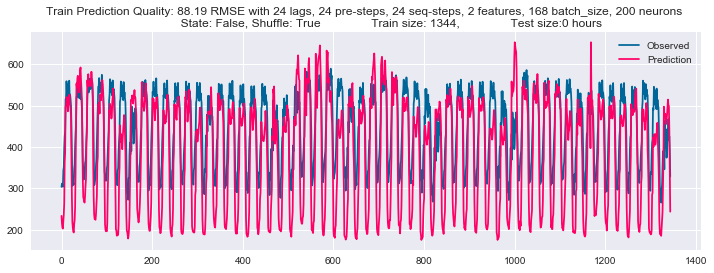
**e252**



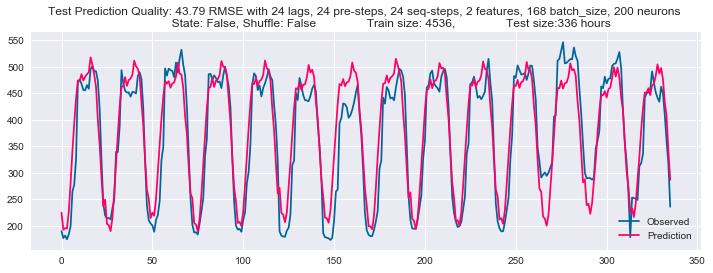


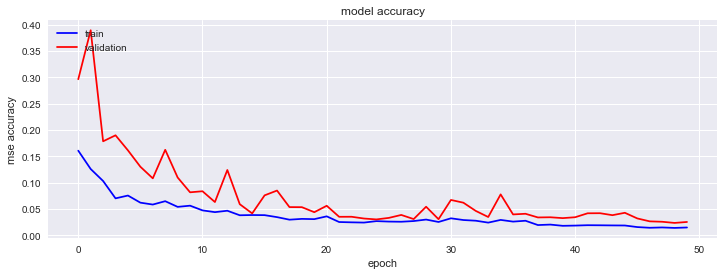
**On a integrated model, state = False, shuffle = True**

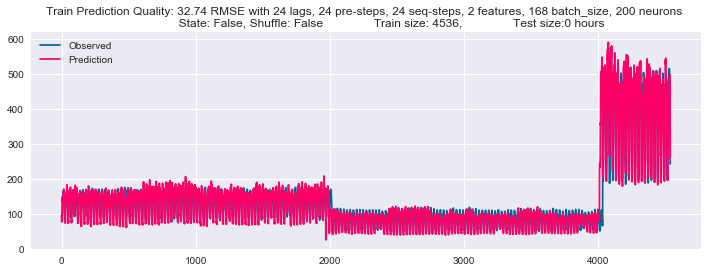


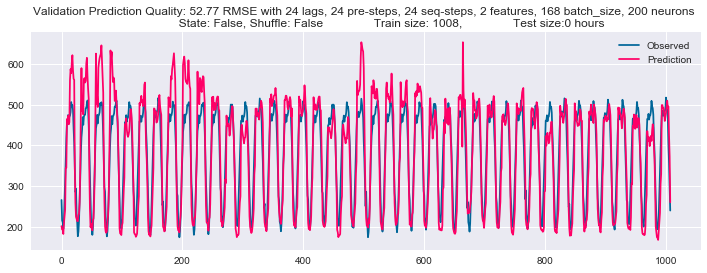


**e235, e250, e252 training with shuffle = False**









**e235, e250, e252 training with shuffle = True**

