

5043 Advanced Machine Learning - HW 1

Author: Enzo Durel

February 12, 2025



Contents

1	Design and Approach	1
2	Plots	2
2.1	Figure 1	2
2.2	Figure 2a	2
2.3	Figure 2b	3

List of Figures

1	True Acceleration vs Predicted Velocity Over Time	2
2	FVAF vs. Training Set Size	2
3	RMSE vs. Training Set Size	3

Design and Approach

Hidden activation function: I choose to use the elu activation because the outputs can be negative and positive.

Output activation function: I choose to use the linear activation because the outputs can be negative and positive but there is no specific range for an acceleration. In practice, I tried tanh but it did not work as well as the linear activation function.

Number of hidden layers: I chose to have a really little number of hidden neural networks ([8, 4]). I tried different size of neural networks and choosing a higher number of hidden layer does not improve the accuracy enough to make the model more suitable. We can see that the model tend to overfit quickly, having higher hidden layers make the model overfitting. Moreover, I think this kind of model will be run in a real-time environment. Having a little dnn makes it less ressources consumer, less power consumer and so more environment friendly. I chose hidden layers size a power of 2^i because there are some optimisations made on layer size power of 2.

Learning rate: I opt for a lrate = 0.0001.

Epochs: I opt for a maximal number of epochs of 300. The fitting process never exceed 150 epochs.

Patience: I opt to reduce patience (50) because the model tend to overfit quickly and I did not want to wait for nothing to get results.

Output type: I chose the "dtheta" output type because we want to predict the velocity.

Plots

2.1 Figure 1

Here is the figure plotting the true acceleration versus the predicted velocity on testing as a function of the timestamp. We can see that the model quite reproduce exactly what is the true acceleration.

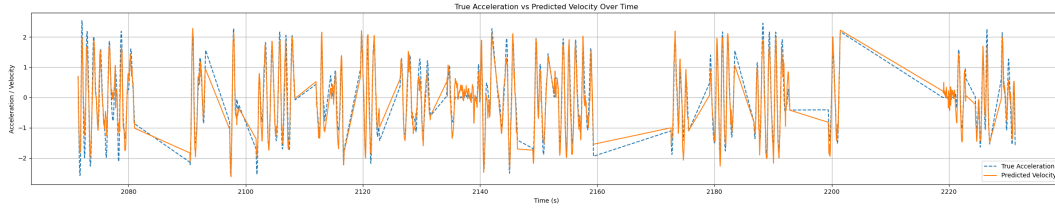


Fig. 1: True Acceleration vs Predicted Velocity Over Time

2.2 Figure 2a

Here is the figure plotting the FVAF metric of the training, testing and validation sets as a function of the training set size. With a little of training set, we can see that the model is very accurate on the training set but not on the others. It is because we do not have a lot of input data points and so the model tend to overfit.

As the number of training data points grows, the model is getting more accurate on validation and testing dataset but not on the training set because it tries to generalize.

Consequently, the three curves converge in a similar fvaf.

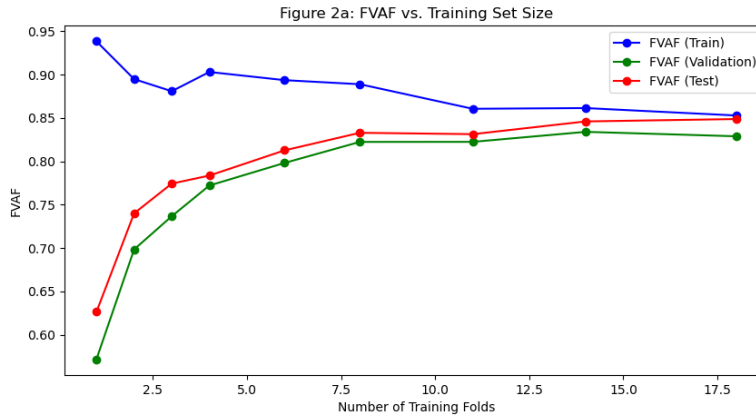


Fig. 2: FVAF vs. Training Set Size

2.3 Figure 2b

Here is the figure plotting the RMSE metric of the training, testing and validation sets as a function of the training set size. With a little of training set, we can see that the model is very accurate (the loss is lower) on the training set but not on the others. It is because we do not have a lot of input data points as I explained before.

As the number of training data points grows, the model is getting more accurate (the rmse ~ loss getting lower) on validation and testing dataset but not on the training set (it's getting higher) because it tries to generalize.

Consequently, the three curves converge in a similar rmse.

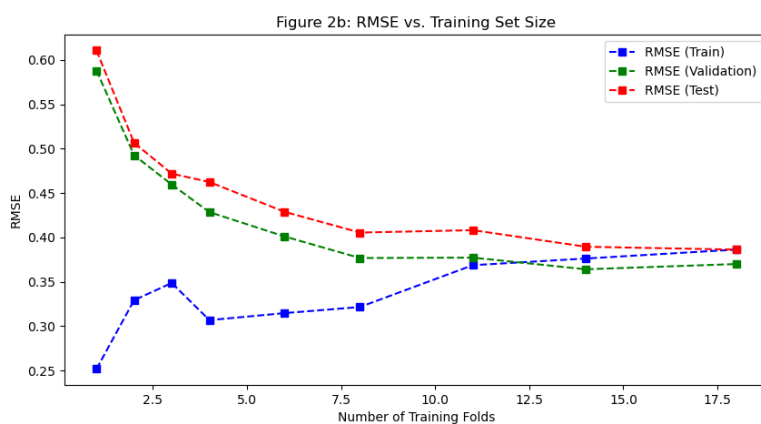


Fig. 3: RMSE vs. Training Set Size