Fundamentals of Neuroengineering - Exercise 1

Philipp Spiess

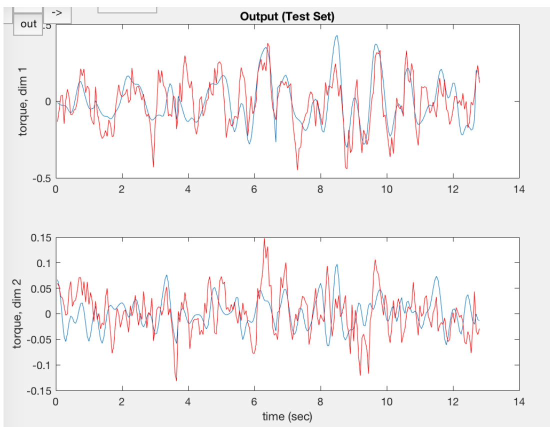
March 2019

Introduction

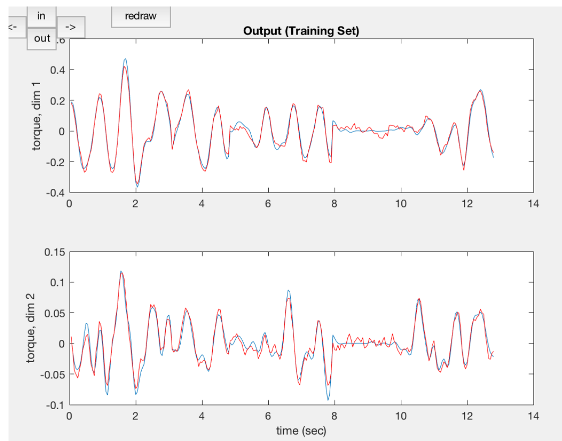
In this lab, we evaluated the performances of modeling an arm motion predictor by using in different ways a Monkey's data that had been collected and analysized in the university of Chicago. The data set cointains 13000 samples which pair arm motion with neuron activation. The purpose of this lab was to evaluate the performance of the model when changing the configuration of the software provided. The performance resulted in the fraction of variance accounted which goes from +1 (perfect prediction) to -infinity (worst case). The model parameters that we could change were the number of training and testing sets, the type of arm motion to predict (Cartesian position, velocity, acceleration, angle, angular speed, acceleration, and torque) and the time of measurement of each neurons. It was also possible to evalute the performance after setting on the additional proprioceptive samples or after performing a compression of data with PCA.

Task 1 - Basic Initialization

For this first task, we trained the model with set 1 and tested it with set 2. It makes perfect sens that the model achieves a higher performance when we test the performance on the same training set because it was designed for it. As expected, the performance of the testing set was poor because one training set is not enough do make a general mapping model. In figure 1, we display the real and the predicted torque of the shoulder and the elbow. On the left, set 2 was tested and on the right, the training set was tested. For this one, we can observe how accurate the prediction is (the original data is shown in blue and the prediction in red). In addition to what we see on the plots, the fraction of variance accounted for was given for both cases and explains with numbers how poorly the prediction is done when we test set 2 and how accurate the prediction is when we test the training set.



(a) Performance of testing set 1: Fraction of Variance Accounted For: 0.1762 -0.9060



(b) Performance of testing set 1: Fraction of Variance Accounted For: 0.9781 0.9400

Figure 1: Widows of training and testing perfomances

Task 2 - Training Set Effects

For torque prediction, we took as training set: set 1 and as testing set: set 10 and we got the following performances:
Training Fraction of Variance Accounted For:
0.9781 0.9400

Test Fraction of Variance Accounted For:
0.2607 -0.6840

We then used for training the sets 1 and 2 and tested as before the set 10:
We got the following performances:
Training Fraction of Variance Accounted For: 0.9260 0.8038

Test Fraction of Variance Accounted For: 0.7583 0.4114

The prediction performance of the training sets decreased as the model was generalized but this generalization has the effect that less artifacts are taken into account and thus, not surprisingly, the performance of the testing set increased.
In figure 2, the plot shows the performance as a function of the number of training sets used, while testing with the set number 10. We see clearly that the ideal number of folds to take is 2 for both shoulder and elbow. In fact, taking 9 folds also gives an equally good performance, but we chose 2 because it requires less computations.

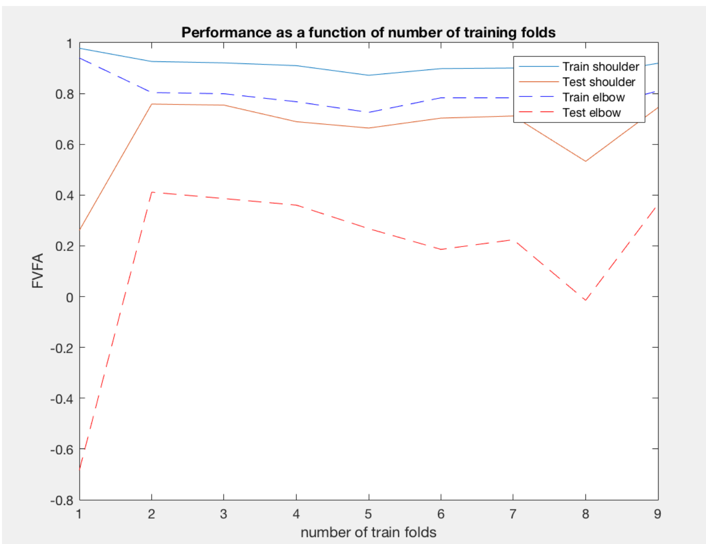


Figure 2: Performances for shoulder and elbow motion prediction as a function of the number of training sets used

Task 3 - Comparing output prediction

After changing set-type() to Cartesian values 'X' to predict the endpoint of the arm with x,y coordinates, we got:

Training Fraction of Variance Accounted For: 0.8607 0.8914

Test Fraction of Variance Accounted For: 0.4095 0.2907

The torque experiment seems more relevant with test values of 0.7583 0.4114 which are higher.

We then changed to velocity prediction values with 'dX', we got better performances than for Cartesian values:

Training Fraction of Variance Accounted For: 0.8885 0.9097

Test Fraction of Variance Accounted For: 0.5497 0.6560

Here, the prediction is about as good as the torque prediction.

Figure 3 compares the four types of predictors: 'X', 'dX', 'ddX' and 'torque'. The relevant performance is given by the FVAF of the test applied to the testing set 10. We can see that the best performances are given by the velocity with good results of about 0.6 for both shoulder and elbow and by the torque which gives a very good prediction for the shoulder motion. The worst prediction appears to be the Cartesian one.

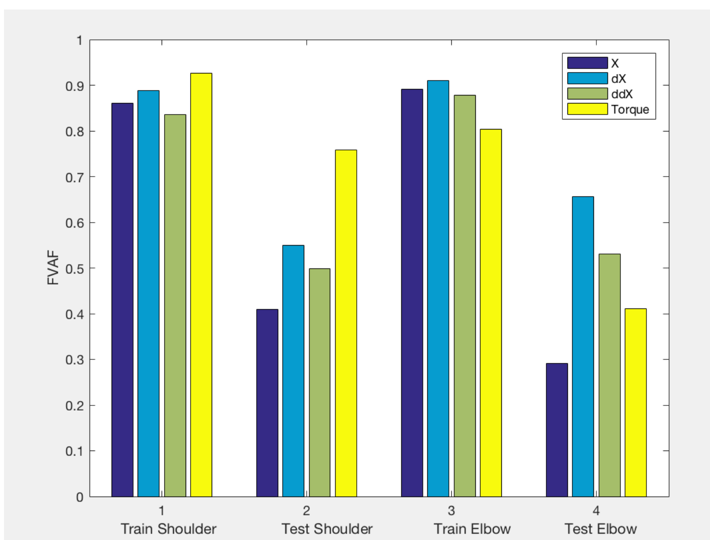


Figure 3: FVAF for shoulder and elbow for different types of predictors

Task 4 - Principal Components Preprocessing

In this task, we analyzed the effect of reducing the number of principal components with PCA. In fact, the problem is to correctly predict 2 variables from 960 input variables (the neurons evaluated * periods of evaluation of each neuron). It is clear that some of these variables give redundant information. So we ploted the FVAF as a function of the number of principal components to see at what point we have enough principal components to make a good model. (see Figure 4) With only one principal component, the prediction is extremely poor for both training and testing:

Training Fraction of Variance Accounted For: 1.0e-03 *
0.3303 0.0174
Test Fraction of Variance Accounted For: 1.0e-03 *
0.0747 -0.7005

We first observe that the performance increases with the number of components. At about 200 principal components, we reach the maxima with a FVAF of about 0.79 for shoulder and 0.48 for elbow test prediction. There is a maxima because at some point, other dimensions will just include irrelevant information in the model and thus overfeeding, this is why with no compression (960 principal components), the results are poorer:
Training Fraction of Variance Accounted For: 0.9260 0.8038
Test Fraction of Variance Accounted For: 0.7583 0.4114

while the performances for 200 principal components are clearly better:
Training Fraction of Variance Accounted For: 0.8880 0.6856
Test Fraction of Variance Accounted For: 0.7944 0.4817

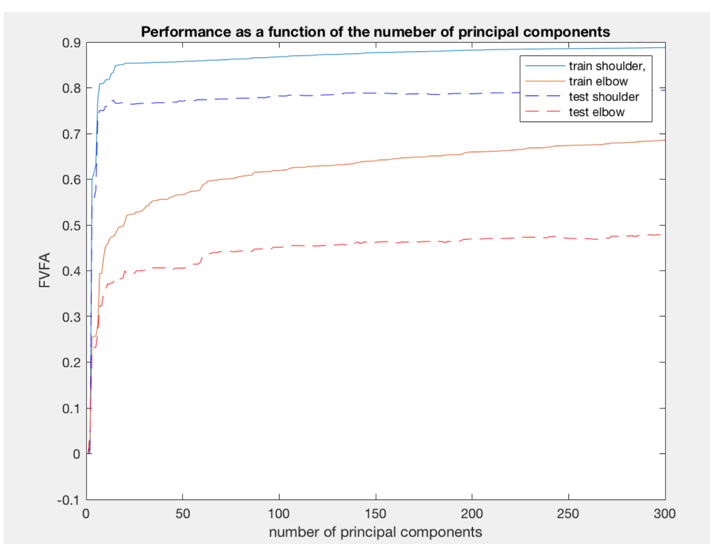
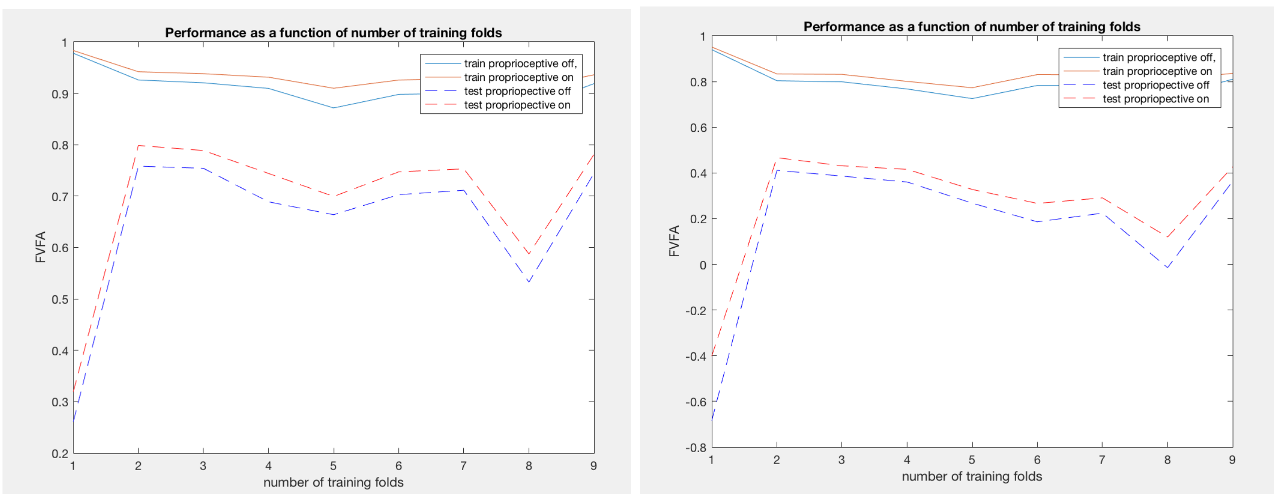


Figure 4: Performances for shoulder torque and elbow torque prediction with PCA.

Task 5 - Proprioceptive Inputs

For torque or acceleration prediction, we can add four additional inputs which reflect the proprioception. In fact, the brain-arm system is a closed-loop and thus we analyze in figure 5 the performances with or without those additional inputs. As we can see on both plots, for any number of training sets, the performance is increased for both training and testing FVAF. Thus we conclude that this feedback is not negligible for the model and that it is a benefit to include the proprioceptive inputs.



(a) Performances of the shoulder

(b) Performances of the elbow

Figure 5: Widows of training and testing perfomances

Task 6 - Varying Neural Delays

In this task, we analysed the monitoring time needed to get sufficient information of the firing rate of the neuron for accurate predictions. By looking at the test graphs of figure 6, we see a maxima at 9 periods which give the best results for both elbow and shoulder. Each periode is 50ms so 450 ms of monitoring seems like the optimal choice. The performance is poor for few periods and we observe that the performance also decreases when the monitoring time is too long.

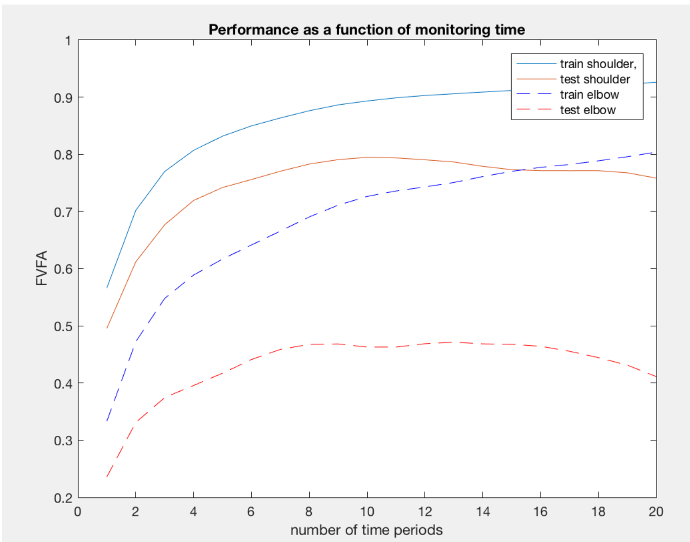


Figure 6: Performance as a function of the number of time periods taken into account

Task 7 - Optimizing the model

Setting all the HPs to the optimal values we got when fixing all the other HPs do not necessarily get us the best solution because those values have dependencies with each other. Thus we can not say that the optimal HP for certain fixed values is the same optimal HP for other fixed values. A possible way to get the best values of HPs to optimize the system as a whole is to use an artificial neural network. The back-propagation of this technique is an efficient way to find the best HP values.