Assignment for "Supervised Learning" - Curtis Baker

Task 1. Single-layer classifier - Using provided code and data ("assign_classifier_gradDesc"):

A. Modify the code to graph the learning curve - loss function vs. the iteration number. Experiment with different values of learning rate. Show 3 examples of learning curves, when the learning rate is: too small; a value that works well; and too large. Describe in words, what in general are the consequences of it being too small or too large ? For this example, what is the largest value of learning rate that works well ?

A learning rate that is too small will not approach the expected values fast enough while a learning rate that is too large will overshoot the expected values and will also not converge either.

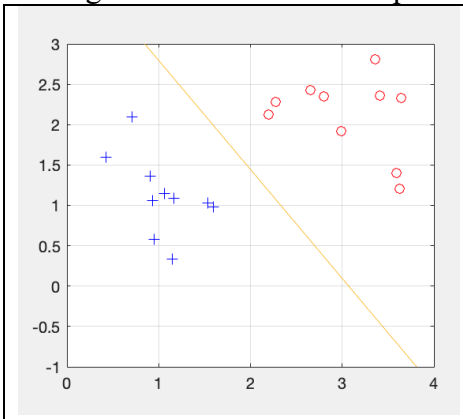

Figure 1 Eta=1 which results in an accurate classification

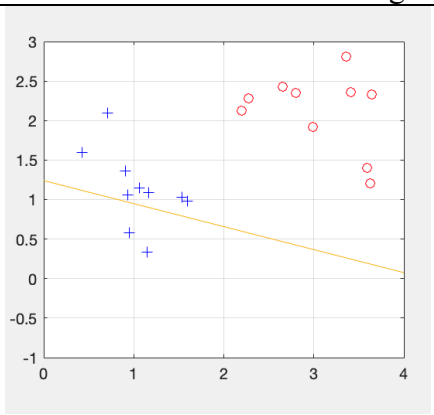

Figure 2 Eta=0.00001 which results in an insufficient calculation

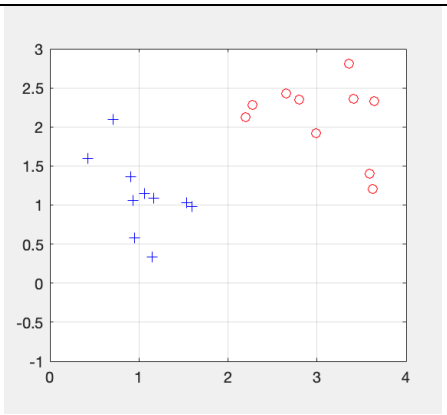

Figure 3 Eta=100 which results in no separation and failed estimation due to gradient overshooting.

B. Now modify the code to also graph all three weights vs. the iteration number and illustrate what happens with a good value of the learning rate. Comment on what happens to the weight values during learning - what can you see in these curves, that might indicate overfitting?

During learning, the weights become either larger or smaller over time. Weights that are too large or too small indicate overfitting as is seen here.

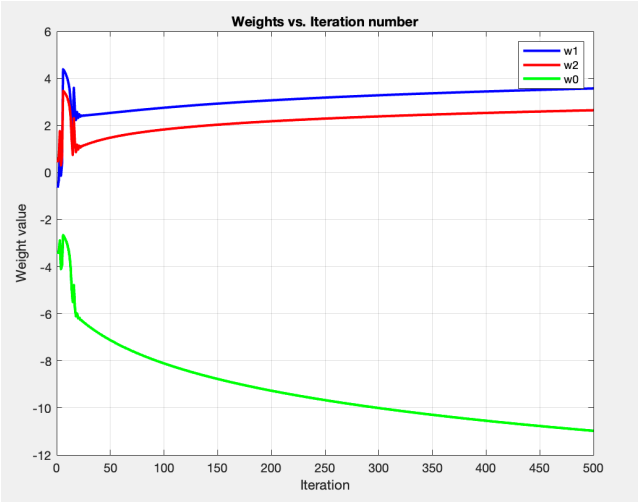

Figure 4 Eta=1

Task 2. Single-layer classifier with regularization

A. Using the code and data from Task 1, modify the updating of the weights, consistent with a penalty on the sum-squared weights. Take care to do this correctly - see MacKay, section 39.4, Figure 39.5.

Weight stability is somewhat achieved with the introduction of the regularization term, but there are still instabilities present.

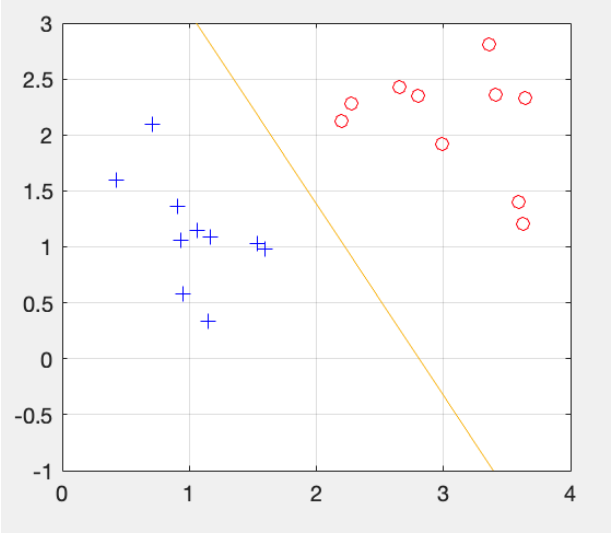

Figure 5 Convergence is achieved. Eta=1

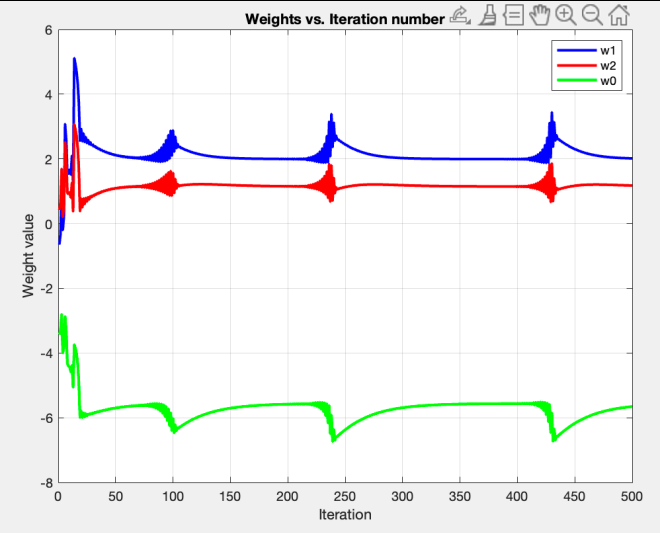

Figure 6 Weight stability is somewhat achieved with alpha=0.001.

B. Experiment with different values of the hyperparameter (alpha) - I suggest you try values of alpha between 0.0 and 1.0. Beware that for larger values of alpha, you will need smaller values of the learning rate to get sensible behaviour (also values between 0 and 1 might be a good range to explore). What is the effect of increasing the alpha value on the weight values, and on the overall results? Show relevant Figures to illustrate what happens for an alpha value that is effective against overfitting.

I increased alpha to 0.01, but I needed to decrease the learning rate to 0.001, and I increased the batch count to 2000. I achieved stability and convergence while avoiding exploding and vanishing gradients.

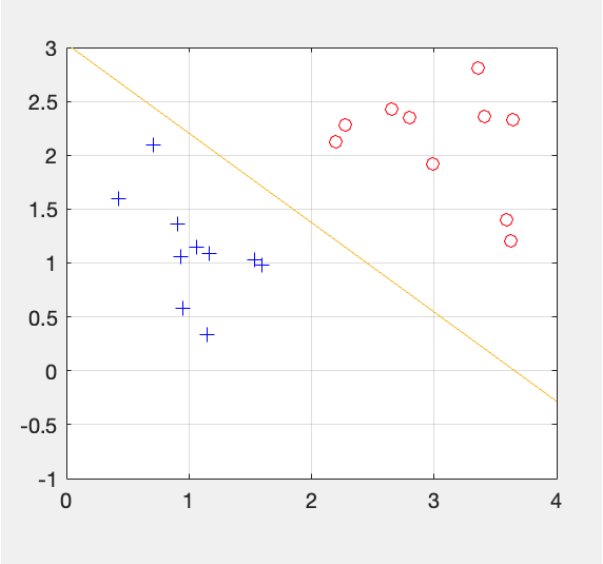

Figure 7 Separation of Data. Alpha = 0.01, eta = 0.001, and batch count = 2000

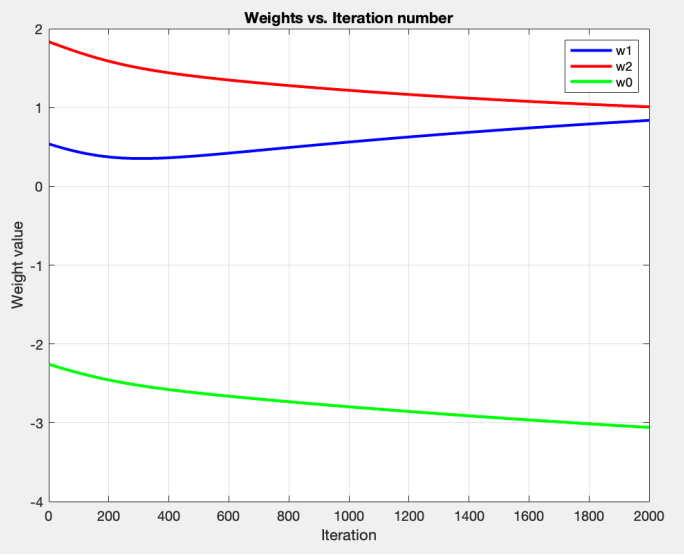

Figure 8 Weights vs Iteration

Task 3. Receptive field (RF) estimation using regression with early stopping - use provided code to do this for a simulated model of a 1-d receptive field profile ("*assign_1d_RF_sysIdent_overfit.m*").

A. Modify the code to generate an additional simulated Validation dataset with 40 measurements (in addition to the Training set of 60). Use the Training set for the iterative loop that learns the best estimate of the RF, and on each iteration, evaluate performance for the Validation set. Plot the learning curves, i.e. error (loss) vs. iterations, for both Training and Validation datasets (superimposed, with different line types). On this plot, indicate the best iteration to stop, for "early stopping". Show the full learning curve, well beyond the optimal place to stop.

The existence of the validation set helps to determine when to stop ie when error begins increasing in the validation set.

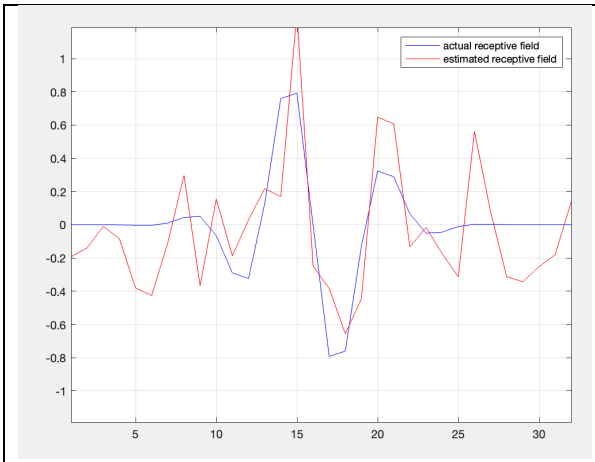

Figure 9 Eta=0.1 and 1000 batches

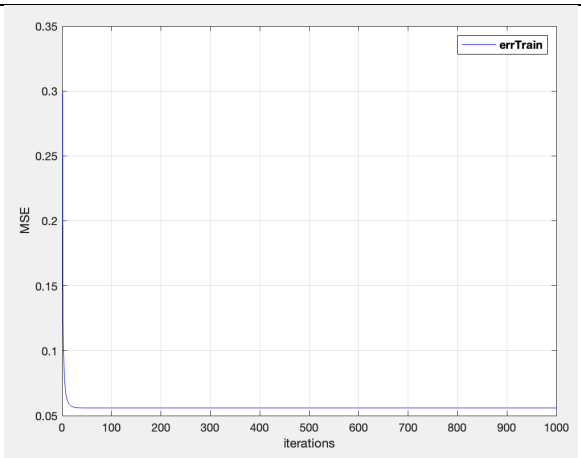

Figure 10 Error of the overfit

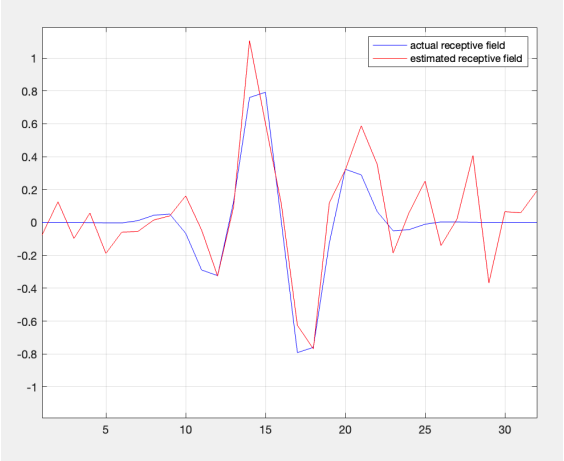

Figure 11 Eta=0.001 and 2000 batches

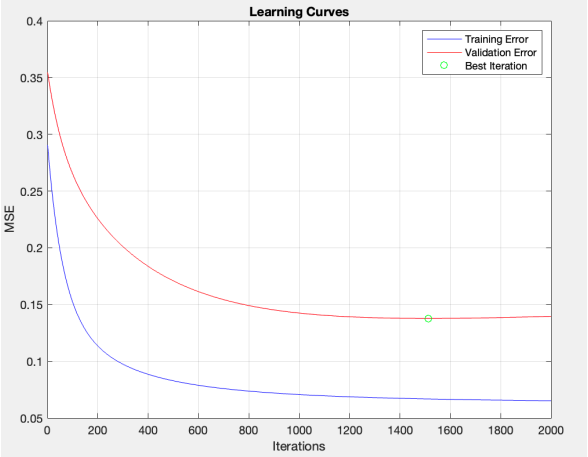

Figure 12 Error of training and validation sets. The best iteration was found at 1513 with a MSE of 0.1378 in the validation set.

B. Now at this optimal place to stop, plot the "learned" (trained) receptive field, and the actual (model) receptive field profile, superimposed using different line and/or symbol types.

Be sure to indicate in the report, the learning rate and number of iterations that you used, and the resulting mean-square error (loss).

This experiment shows the best estimation of the Gabor cell. The overfit is still present as shown by the spikes in the part of the figure that should be flat, but the early stopping has helped to reduce the overfit compared to the previous experiment.

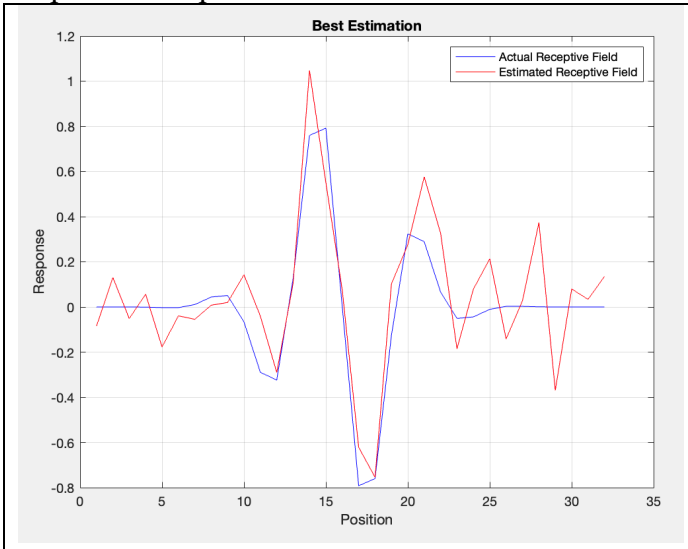

Figure 13 Best Estimation: eta=0.001, batches=2000, best iteration=1513

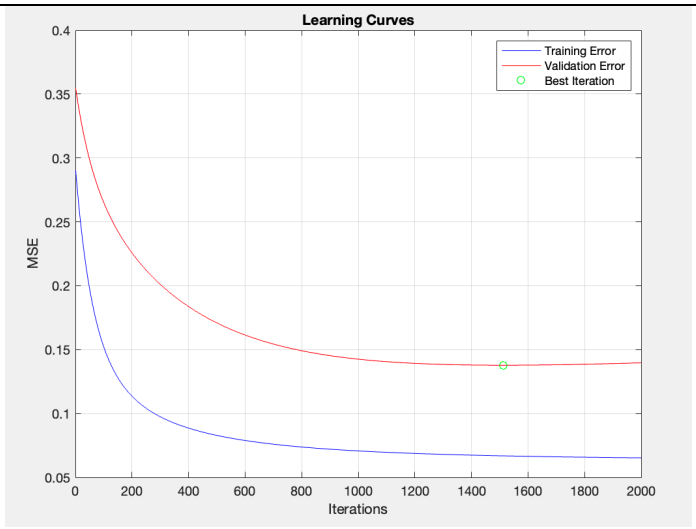

Figure 14 Error of training and validation sets. The best iteration was found at 1513 with a MSE of 0.1378 in the validation set.

Task 4. RF estimation, comparing regression vs. correlation for different stimuli - use provided code ("assign_2d_RF_sysIdent") which uses a scaled conjugate gradient ridge-regression algorithm ('scg', from the provided *netlab* toolbox) that automatically optimizes the learning rate. The program provides different options for two types of visual stimuli and two types of analysis algorithm, for estimation of a 2d spatial receptive field.

A. The provided code partitions the data into two sets, for training and validation. Modify the code to provide a third "Test" partition, for final evaluation of the trained model performance, using VAF (variance accounted for). Be sure to use each of these datasets at the correct places in your code.

The validation set helps for a more accurate and precise estimation of the test set. VAF is used to determine the capability of the model to capture the variability in the data where a higher VAF demonstrates an increased ability for the model to capture the variability but may also indict overfitting.

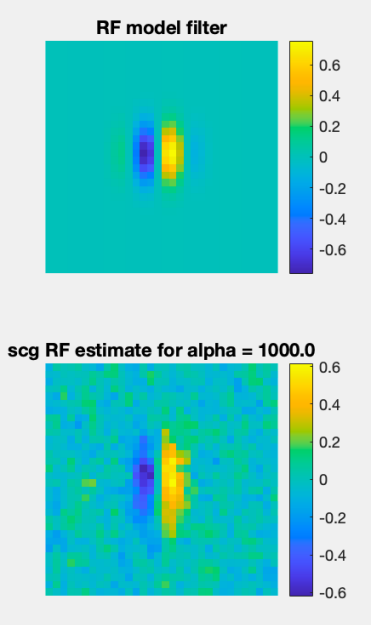

Figure 15 Natural Image Estimation

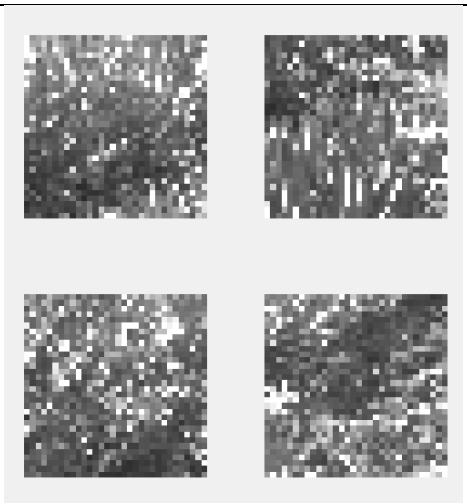

Figure 16 Natural Image Stimulus

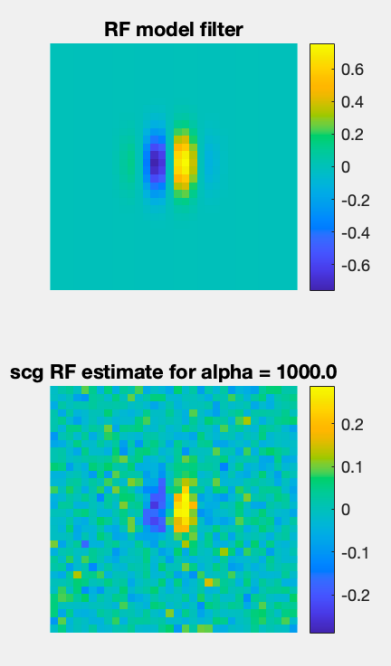

Figure 17 White Noise Estimation

Figure 18 White Noise Stimulus

B. For scg-regression and white noise stimuli, modify the code to systematically search for the best alpha value to use for regularization. Graph VAF against alpha for the training dataset, on a semilogx plot (I suggest testing log-spaced values of alpha between 0.1 and 10000). Do this also for natural image stimuli. Show a figure with 4 subplots: the actual simulated model RF, and the estimated RF for three values of alpha (too small, optimal, much too large) - plot them all on the same z-scale, so they can be compared. Discuss the effects of the penalty value (alpha) being too small or too large.

Alpha determines the relevance of the weights in the error calculations. When alpha is too small, there can be too much fitting which leads to overfitting. When alpha is too large, the fitting will not be as aggressive as it needs to be so there will be underfitting. According to this experiment, 10000 was found to be the optimal value according to VAF meaning that an alpha of 10000 best captured the variability in the data.

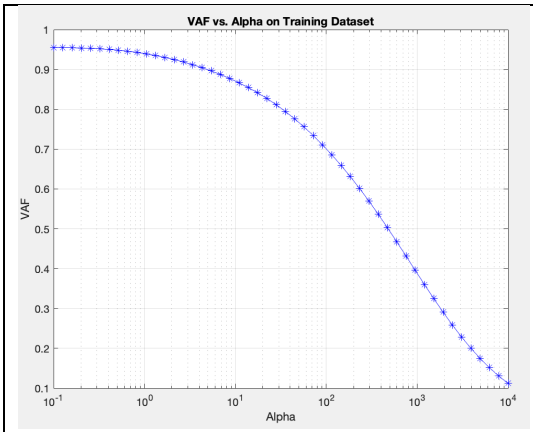

Figure 19 VAF vs Alpha on Training Dataset

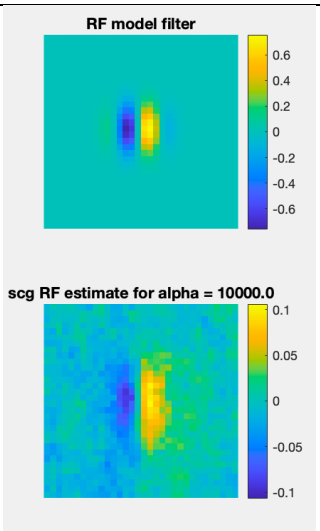

Figure 20 Filter (target values) vs Estimated Values

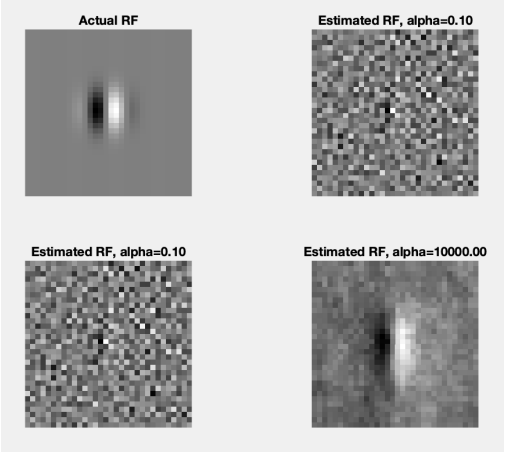

Figure 21 Target and Estimations

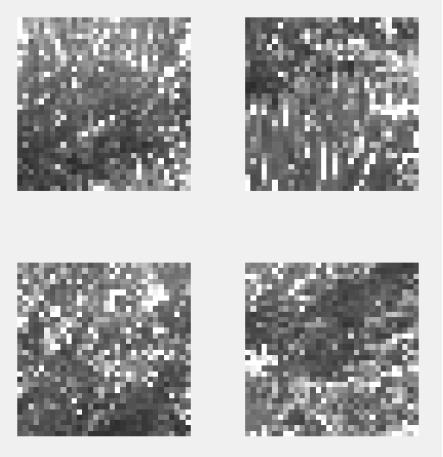

Figure 22 Stimulus

C. Now evaluate the results using cross-correlation instead of regression - in this case, there is no ridge regression and therefore no alpha value - just run it and compare the results (estimated RF and VAF), for both white noise and natural image stimuli. Discuss relative advantages and disadvantages of regression vs. correlation methods for system identification of neuronal receptive fields.

Regression is more complicated but can uncover non-linear patterns, causality, and can generalize to new patterns. However, it is susceptible to over or under fitting and takes a longer time to run. Correlation is less complicated and is much quicker to run but can not uncover non-linear patterns, can not uncover causal patterns, and can not generalize to new patterns.

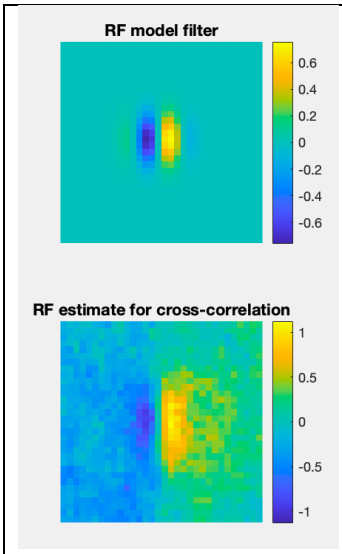

Figure 23 RF estimate for cross-correlation

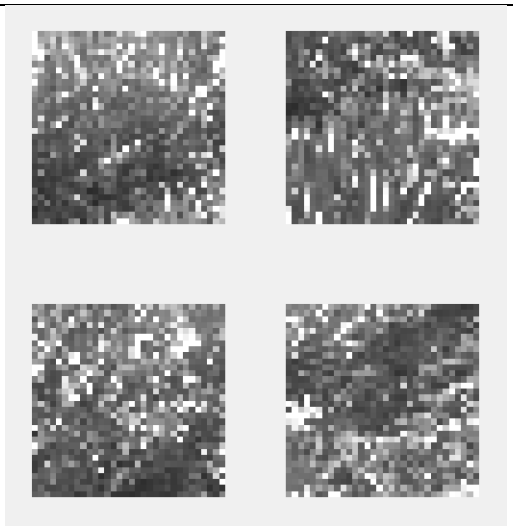

Figure 24 Natural Image Stimulus

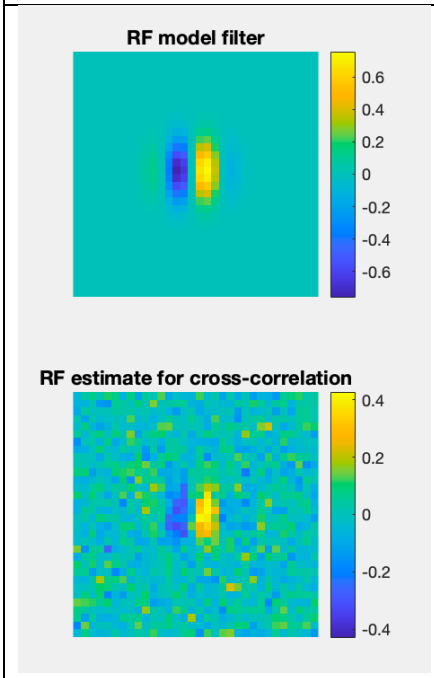

Figure 25 RF estimate for cross-correlation

Figure 26 White Noise Stimulus