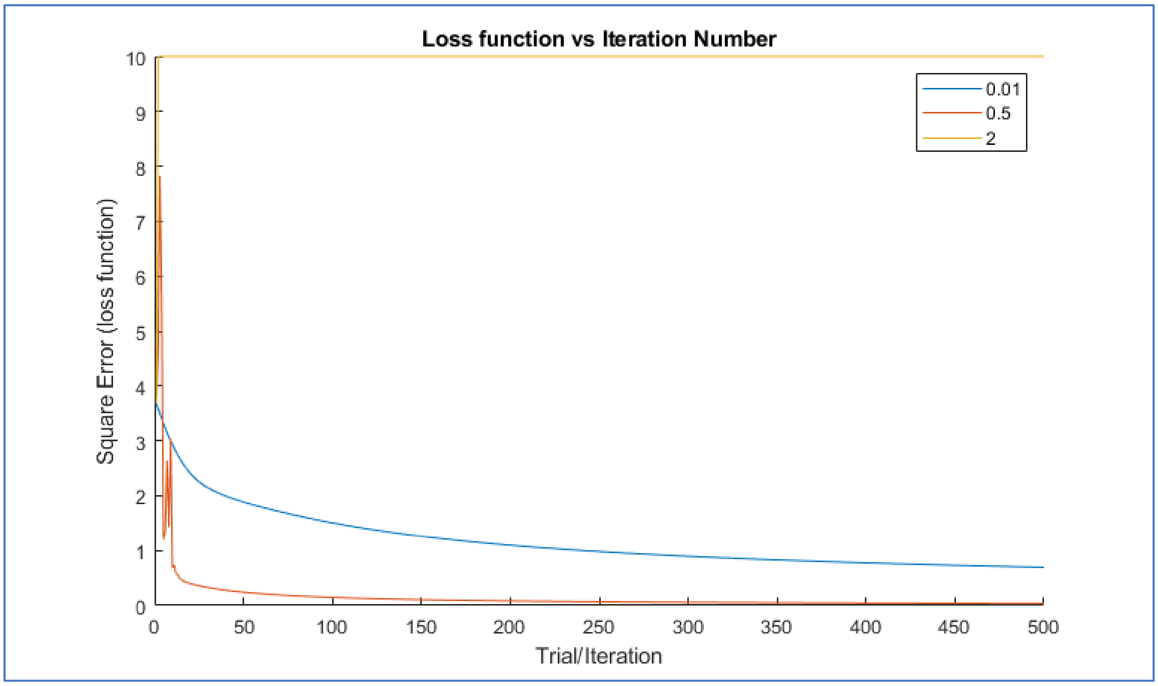
Assignment 5: Dr. Baker

Joon Hwan Hong; no collaborations to declare

Note on code organization: All figures are generated by running the MASTER.m script which uses modified versions of the provided codes (as well as my own functions) as a master script. Although I did package them individually as well for assignment submission

Task 1) Single-layer classifier

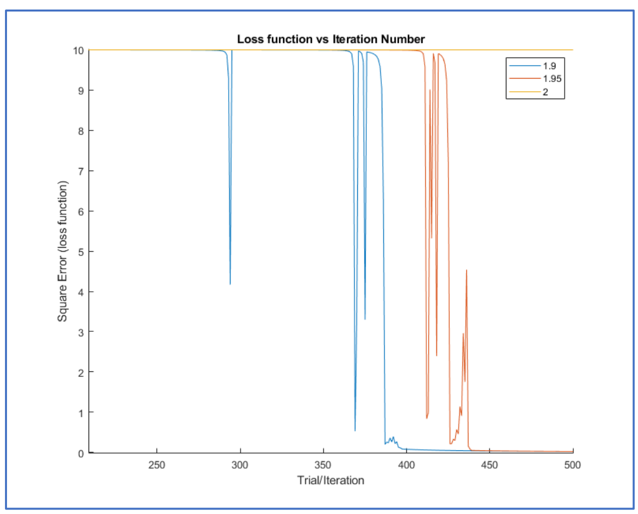
A) Loss function vs Iteration Number & B) Graph all 3 weights



The figure above shows 3 examples with different learning rate values/weights of $eta = \{0.01, 0.5, 2\}$. The figure shows the square error (loss) as a function of the trial #/iteration.

When the learning rate is too small (0.01 example): it will take a long time to train (similar to an exponential decay). It needs a greater number of iterations to be trained than $eta = 0.5$. In a situation where it requires more trials, this would be too computationally expensive and slow.

When the learning rate is too large (2 example): the model will simply fail to learn over the 500 steps. $eta = 1.95$ seems to be the "largest learning rate that works" given 500 iterations.



During learning, the new weight value for a label x_i is defined to be the current weight for value + the learning weight and the partial derivative of the error (correct label – network response) in respect to the current weight: $w_{new} = w - \eta \frac{\partial E}{\partial w}$ for each step.

An indication of overfitting would be the loss function only changing to give miniscule improvements to error while the iterating. An example of this can be seen in the $eta = 0.5$ curve, where after approximately 25 trials, the loss function value does not change a significant amount, and continuing to 500 trials would lead to overfitting.

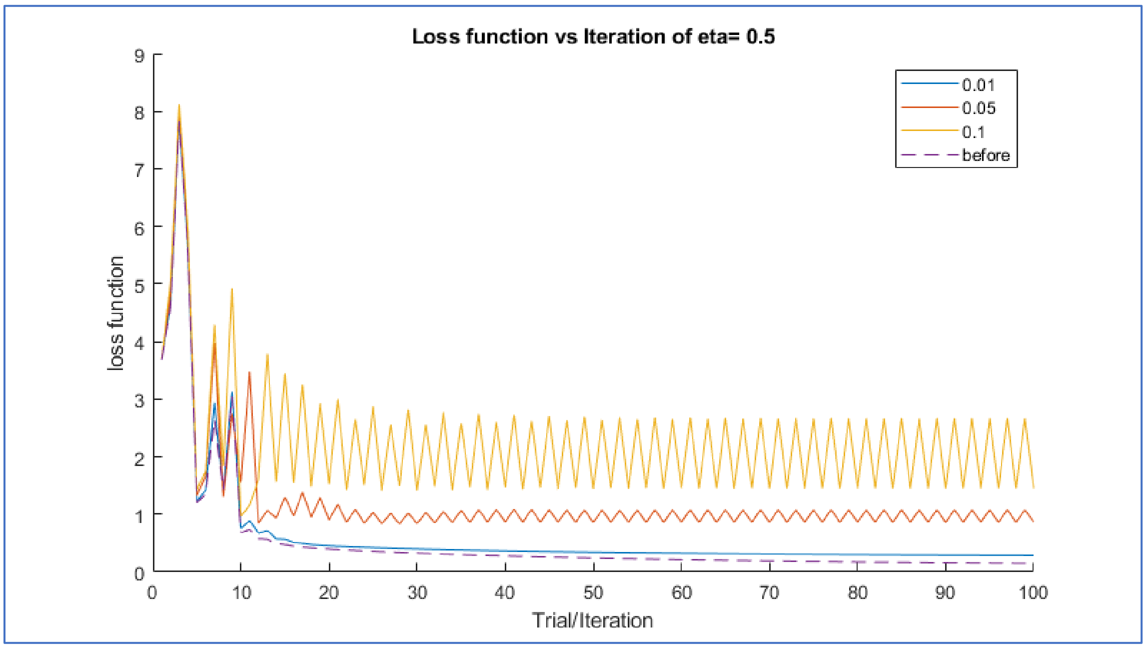
Task 2) Single-layer classifier with regression

A) Modify updating weights consistent with a penalty on the sum-squared weights

The *MASTER.m* script calls upon *classifier_reg*(eta, nTrials, alpha) which is a modified version of classifier_gradDesc(eta, nTrials).

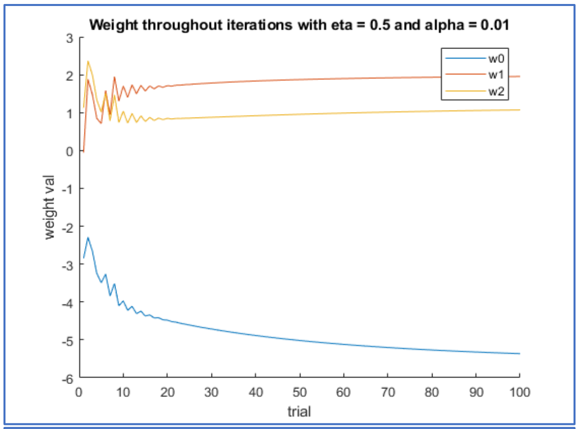
B) Experiment with different values of alpha

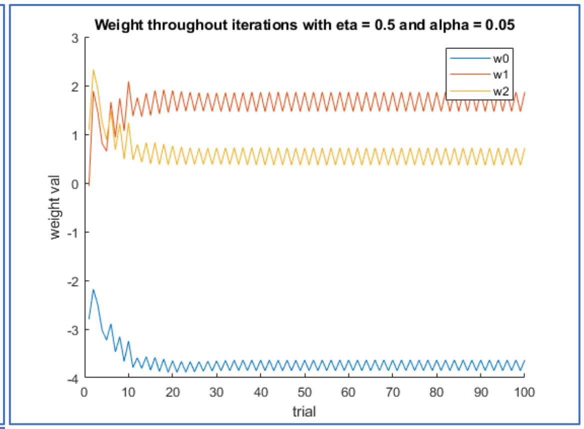
To note: generally, beyond ~100 trials only miniscule changes occurred. Thus, the figures for *Task 2* did not plot beyond iteration 100 to instead have a closer look between 0-100. The eta value of 0.5 used in *Task A* was selected to generate the figure. Alpha values of {0.01 0.05, 0.1} were chosen. The legend corresponds to the alpha values tested.

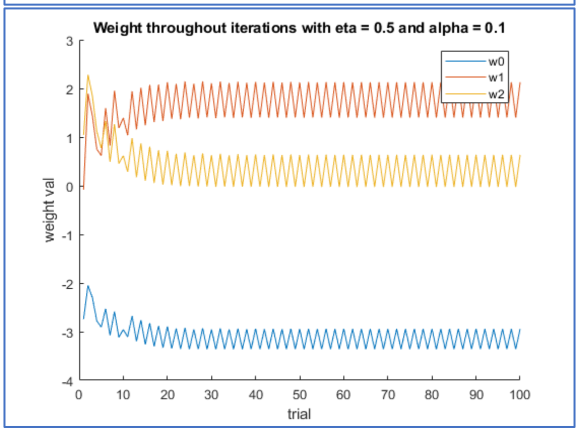


The figure above shows 3 different alpha values that was tested on eta=0.5. The function before regularization is labeled as 'before'. It shows the classifier behaviour for different regularization parameter alpha.

The graph indicates that larger alpha values induce greater fluctuations/ change per iteration; a trend of higher alpha values correlating to a larger MSE is observed. Alpha value of 0.01 would result in the least MSE, resulting in the "best" classifier behaviour out of the three alpha values tested.







The regularization hyperparameter (alpha) shapes the amount of decay; it can be used to against over-fitting and limit large weight values. To explore the effect of increasing the alpha value on the weight values (w0, w1 and w2), the three weights were recorded throughout each iteration. The figures above were generated in *classifier_reg()*.

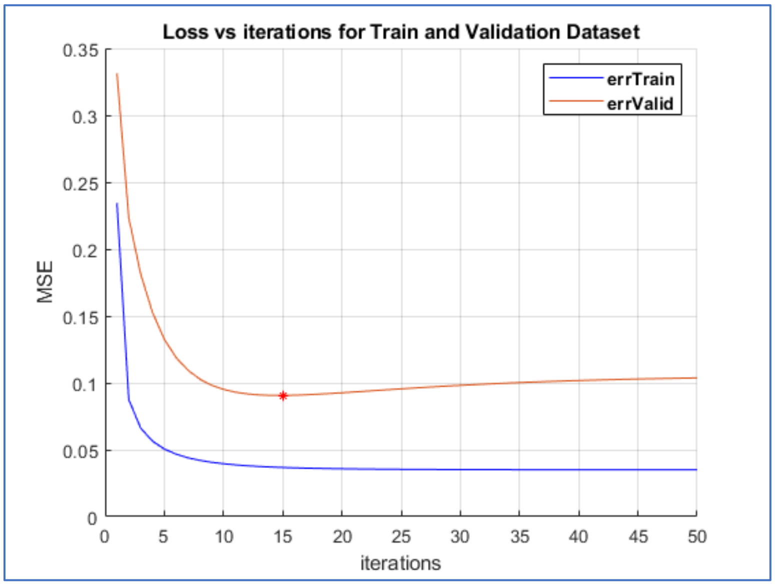
An overall trend of failing to converge (the zig-zag pattern from overshooting and undershooting) with higher alpha values is observed. With alpha values of 0.05 and 0.1, the weight values are unable to converge to a stable value as trials increase.

This instability of weight fluctuations is also observed in the prior figure, where the loss function for alpha = 0.05 and 0.1 alternate between two loss values as trial increases.

It can be concluded that alpha shapes the magnitude of change in weight after each iteration. Interestingly, increasing alpha value correlates with increased average value for w0 (less negative).

Task 3) RF estimation using regression with early stopping

A) Error for both Training and Validation datasets

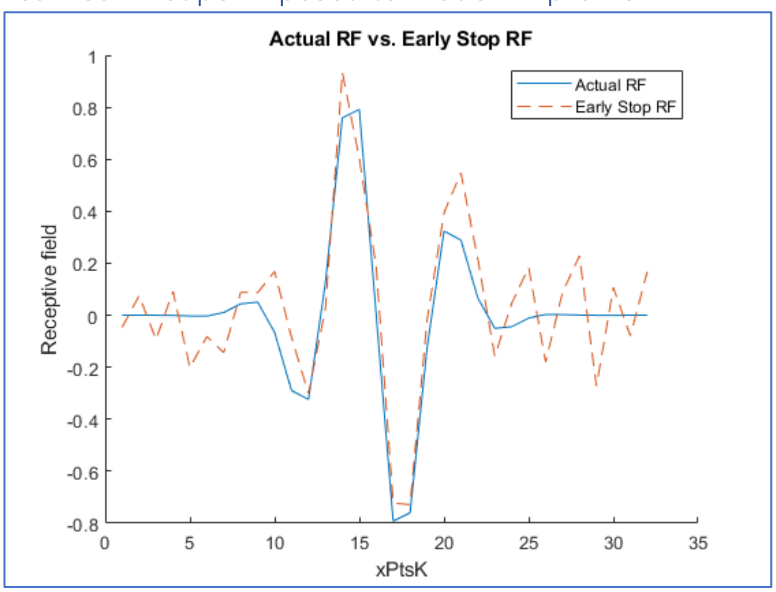


The following figure shows the error of the Training dataset and the error of the Validation dataset.

"Early stopping" iteration is defined by: *when the next validation error is greater than the current validation error (red star)*.

For this instance, it was calculated to be at the 15 iteration.

B) Learned RF superimposed to model RF profile.



The following figure is the learned RF superimposed to the model RF profile.

While the estimated model cannot accurately predict relatively stable changes to RF (ex. from 25 to 32), it is able to fit drastic changes well (from 14 to 20)

Reported Values:
Learning rate = 0.1 (default)
Iteration = 15
Loss = 0.090621

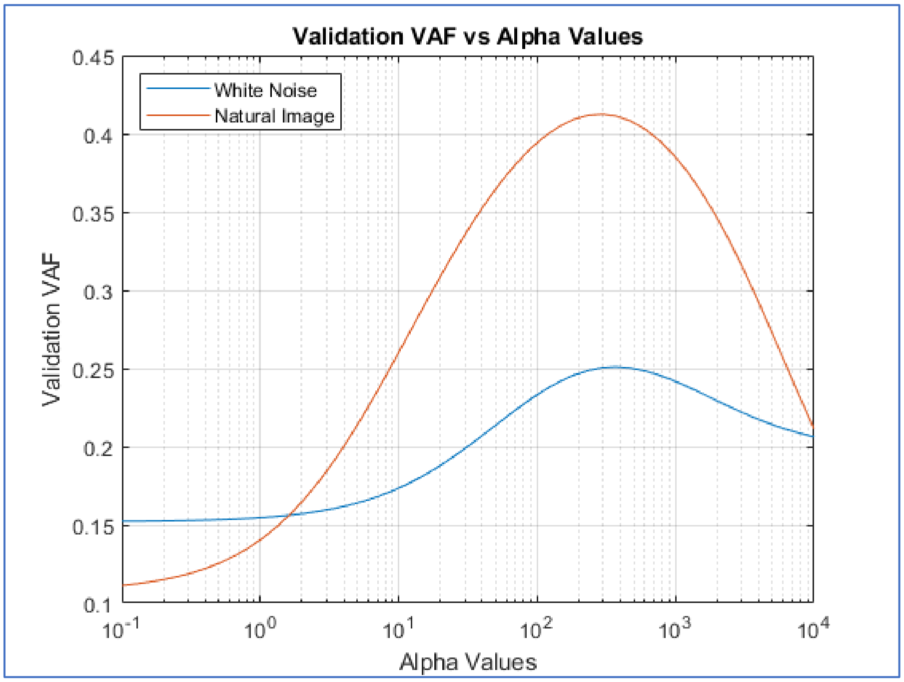
Both figures A) and B) were obtained by *MASTER.m* calling *assign_1d_RF_sysident_overfit.m* script.

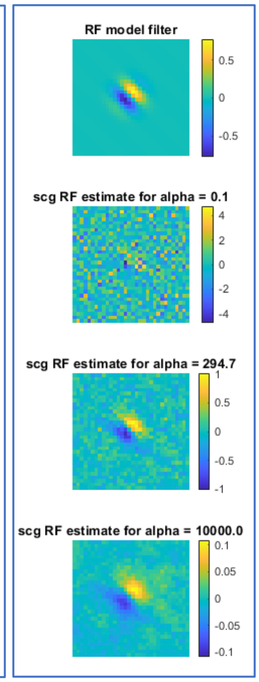
Task 4) RF estimation, regression vs correlation for different stimuli

A) Test partition and evaluate performance using VAF

Nothing to report for this section.

B) For scg-regression and white noise stimuli



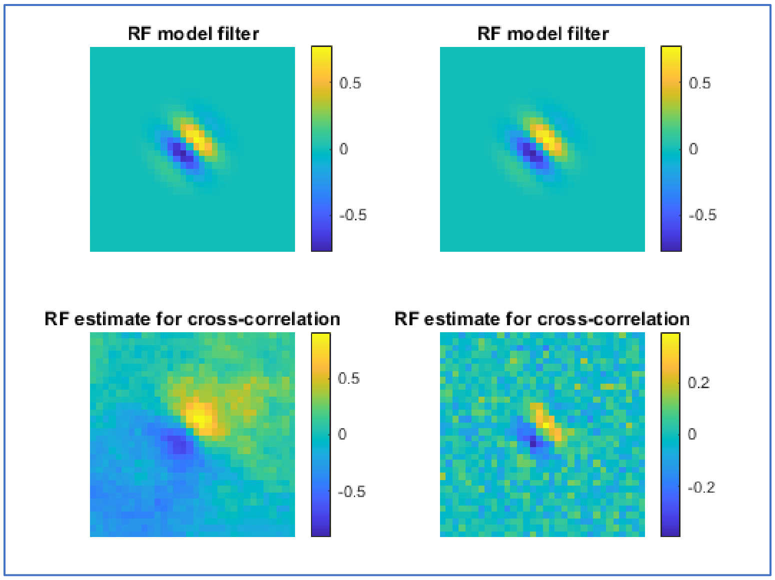


I have decided to graph the validation VAF as the optimal VAF for training (as mentioned in the assignment) would just result in the lowest value of alpha (which will allow overfitting). The VAF throughout log-spaced values between 0.1 and 10000 was graphed for both white noise and natural image stimuli.

Three alpha values were chosen (0.1, 294.7, 10000) as examples of alpha being too small, optimal, or large. 294.7 was chosen as it was the alpha value for the peak VAF of natural images. As observed, the estimated RF for alpha = 0.1 results in noise overpowering the overall estimated receptive field, while the estimated RF for alpha = 10000 resulted in too high of contrasts in the image (the negative and positive regions have expended too much, well beyond the model RF's size)

C) Evaluate results using cross-correlation instead of regression

When evaluating the results using cross-correlation instead of regression, the following results were obtained for VAC calculation: white noise resulted in VAF = 0.078366; Natural images resulted in VAF = 0.19614.



And produced the following estimated RFs. The right RF estimation is from natural images, while the left RF estimation is from white noise.

An advantage of regression would be the ability to modify its RF estimation using a hyperparameter value alpha. It can optimize its estimated RF to emulate the RF model more accurately. Although this in itself can be a disadvantage as this requires more computational power for larger model and dataset compared to cross-correlation. In addition, a suboptimal hyperparameter value can lead to worse results, seen in the scg RF model using alpha = 0.1.