Thomas Christinck
260601793

NEUR 603 Assignment 6
Supervised Learning

1. Learning curves for 100 trials are shown in Figure 1 for various learning rates. A learning rate of 1.5 (Figure 1; left plots) was found to be too large a learning rate, as the learning is unstable, and the weights come close to diverging. A learning rate of 0.001 (Figure 1; center plots) was found to be too small a learning rate, as the optimization algorithm fails to converge in 100 trials. Generally, a learning rate that is too small will result in the optimizer failing to converge within a reasonable number of trials, and a learning rate that is too large will result in the optimizer exhibiting unstable behaviour, possibly failing to converge and diverging.

A learning rate of 0.2 (Figure 1; right plots) was found to be a good learning rate. The evolution of the weights during training is shown in Figure 2. Note that in the weight evolution curves, as the number of trials increases the weights continue to diverge away from 0. This indicates that we might be over-fitting, as large weight values are typical of over-fitting behaviour.

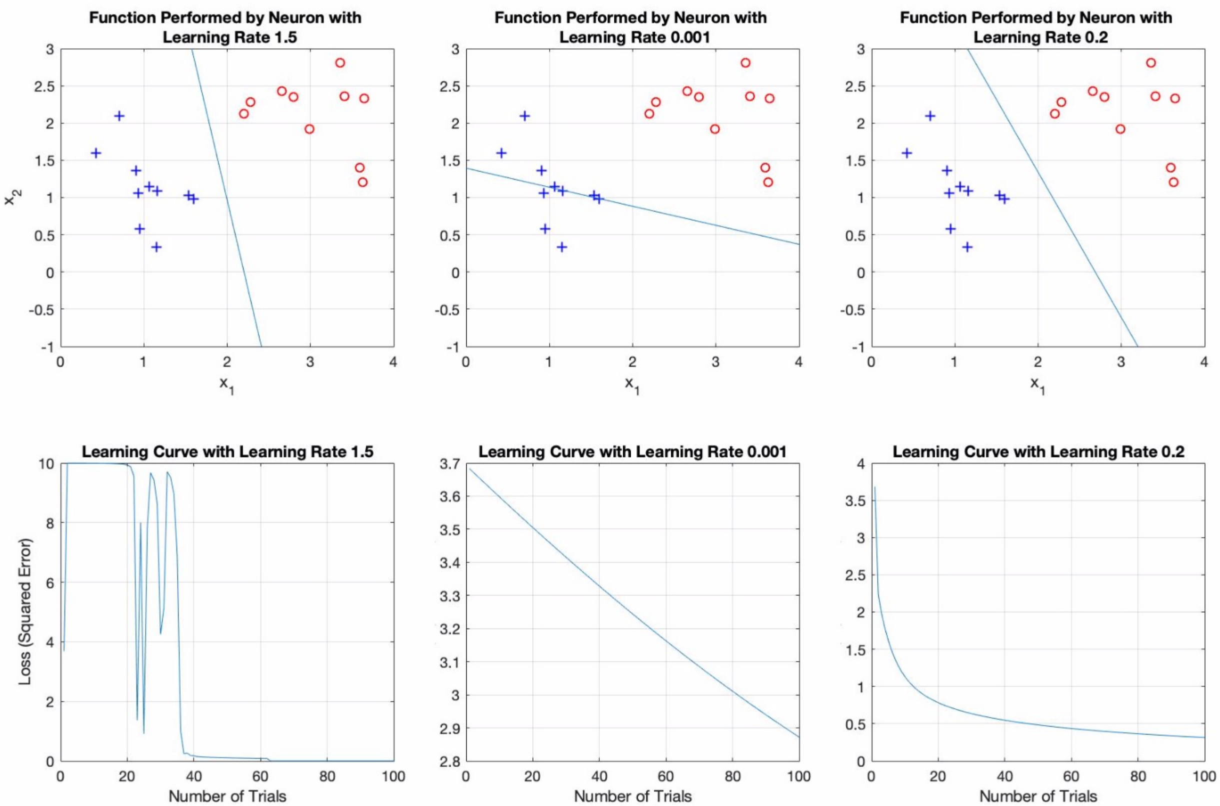

Figure 1 - Examples of learning curves where the learning rate is too large (left), too small (center) and a good value (right)

2. The code from Task 1 was modified to penalize the sum-squared weights with a regularization parameter α. Various α-values were explored, and results are shown in Figure 3. As we increase α, large weights are penalized more. Thus, choosing a small α (see Figure 3, left) we will see insufficient regularization (the same diverging weight issue seen in Figure 2). Choosing a large α (see Figure 3, right) we will see that the objective function has been excessively regularized and so weights are kept too small to effectively separate the data. An α that is just right should prevent the weights from diverging, without compromising the ability of the model to effectively separate the data (see Figure 3, center).

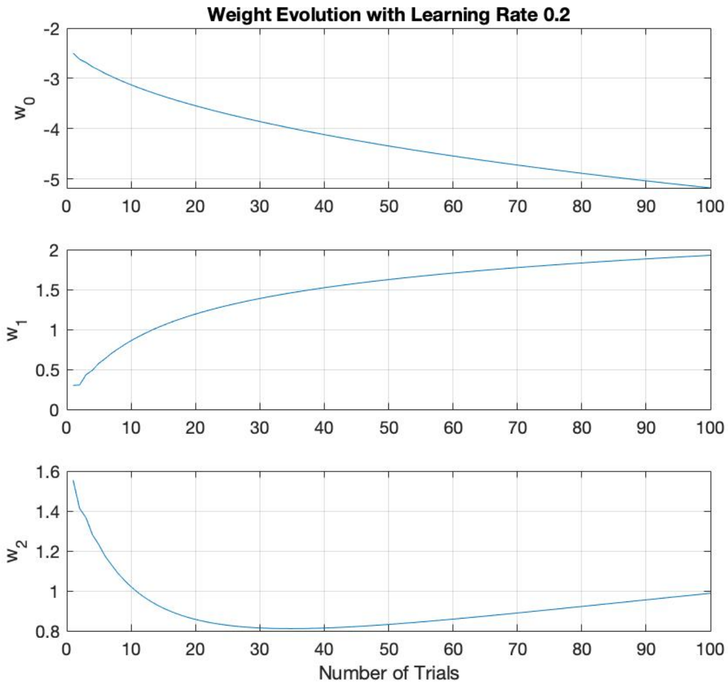

Figure 2 - The weight evolution (for w0, w1, and w2) for a learning rate of 0.2, found to a good learning rate

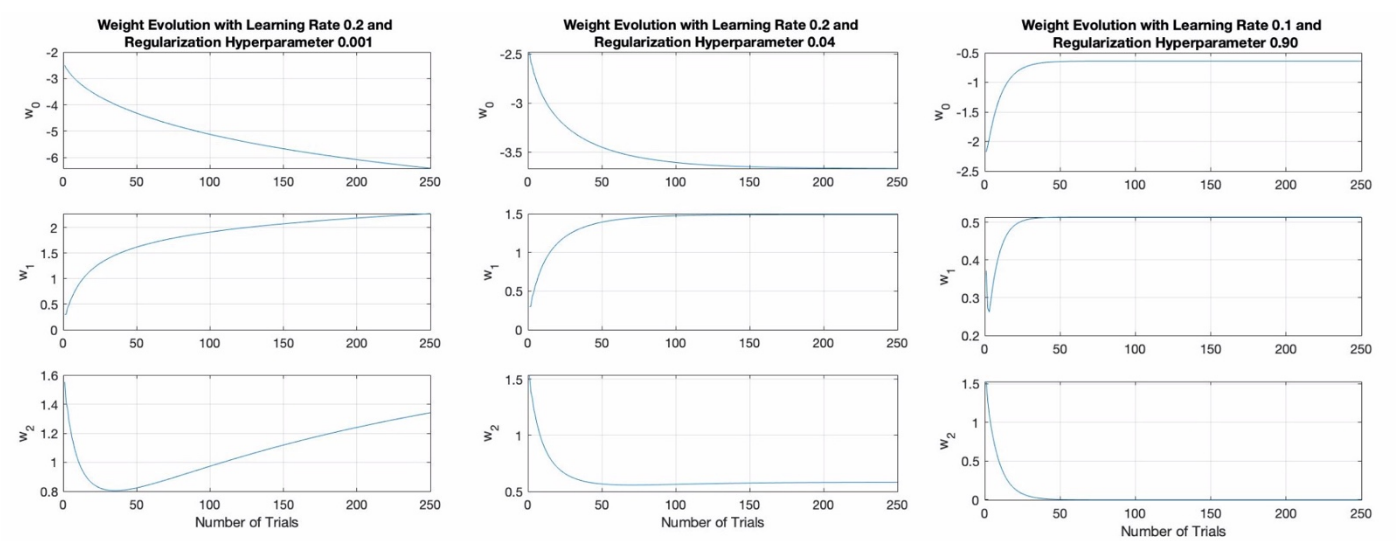

Figure 3 - The weight evolution (for w0, w1, and w2) with varying regularization hyperparameters. The neuron was trained for 250 trials; the learning rate was decreased to 0.1 in the right-most case to stabilize learning behaviour

3. The code was modified to divide data into a training set and a validation set (a 70/30 training/test split was used). The learning curves for both the training and validation sets are plotted in Figure 4. The best place to stop (approximately) is where the validation error is minimal and is denoted by a vertical purple line.

The code was then modified to stop at the iteration where validation error starts to increase. The actual receptive field profile and the learned receptive field, with and without stopping, are shown in Figure 5. A learning rate of 0.1 and 15 iterations were used. The resulting training and validation losses are 0.037 and 0.091, respectively.

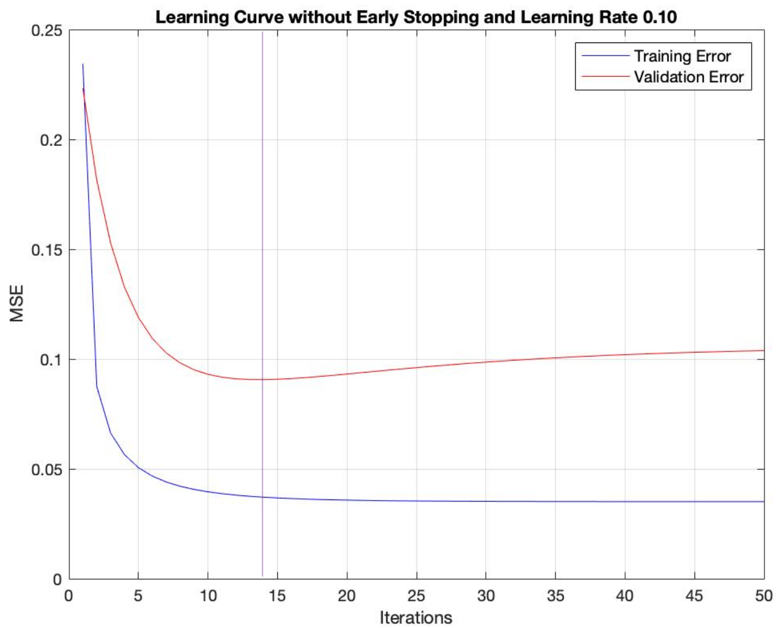

Figure 4 - The learning curves for validation (red) and training (blue) sets. The best stopping place is where validation error is minimum and is (approximately) denoted by the vertical purple line

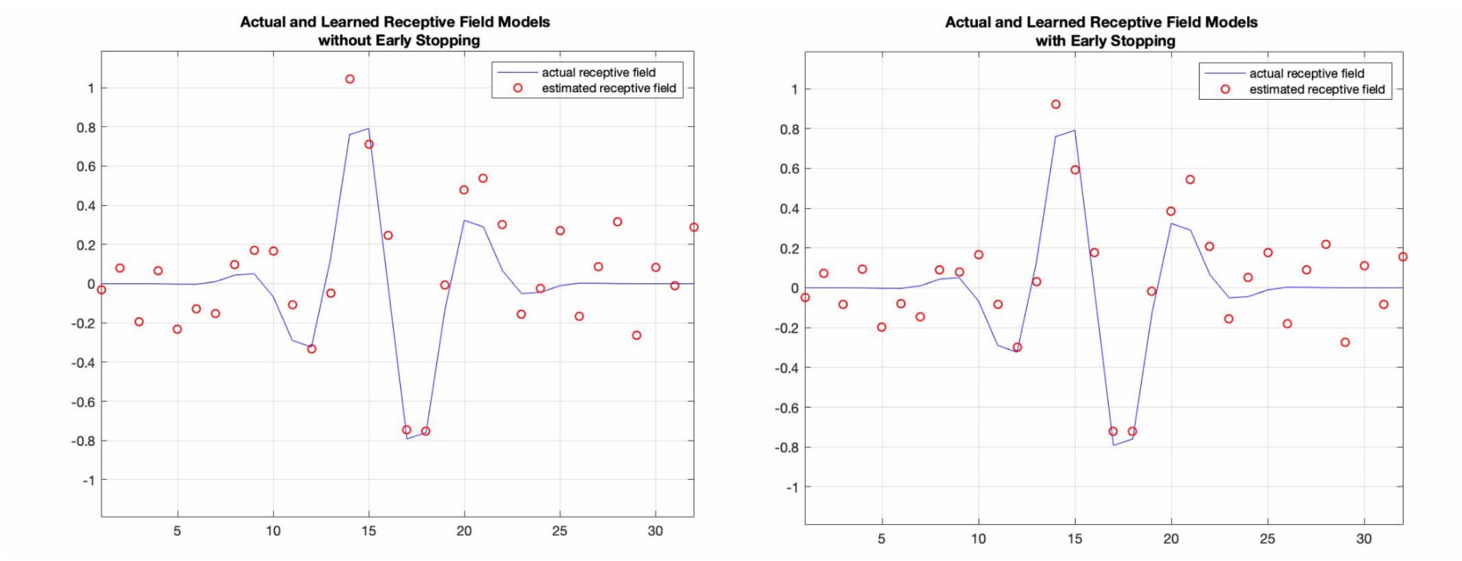

Figure 5 - The actual and learned receptive fields without early stopping (left) and with early stopping (right) after 15 iterations

4. The code was modified so that a third "test" partition of the dataset was kept aside for final evaluation of the trained model. The code was modified to systematically search for the best α value to use for regularization. The loss is plotted as a function of the regularization parameter α for both white noise stimuli and natural image stimuli in Figure 6. Figure 7 shows the actual simulated receptive field as well as estimated receptive fields for several values of alpha given a white noise stimulus. Figure 8 shows similar receptive fields given a natural image stimulus. Note that for large values of α, the weights are penalized too heavily, and we see that the receptive field tends to take near-zero values everywhere. For small values of α, the weights are allowed to grow arbitrarily large and the model begins to over-fit the training data (i.e. the model is allowed to fit the noise in the data). For an appropriately chosen α, the model is biased so as not to overfit the data but has sufficient capacity (i.e. its weights are allowed to grow sufficiently large) to successfully model the training set and generalize to new data in the validation set.

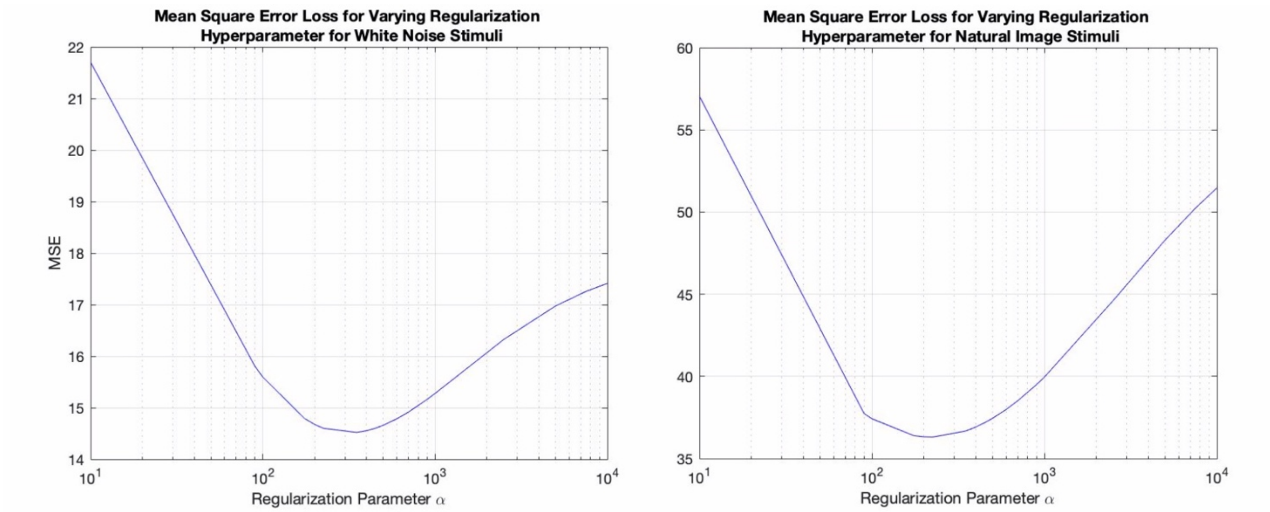

Figure 6 - Receptive field validation loss for varying regularization hyperparameter alpha with white noise stimuli (left) and natural image stimuli (right).

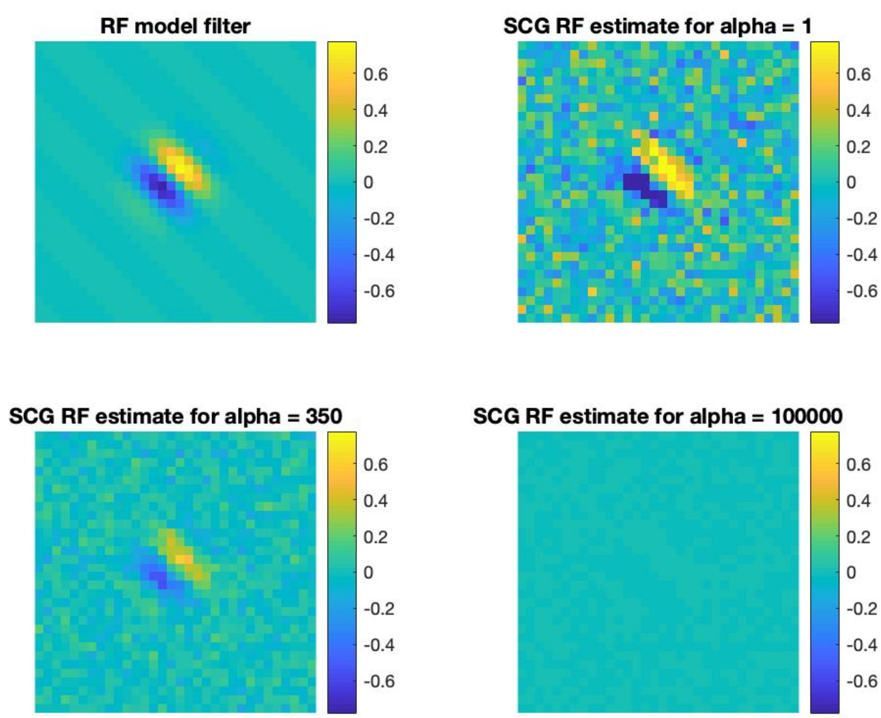

Figure 7 -The actual simulated receptive field as well as estimated receptive fields for several values of alpha given a white noise stimulus. An appropriate alpha was found to be 350; alpha-values significantly larger than this will penalize weights excessively and overly bias the model; alpha-values significantly lower than this will be able to grow arbitrarily large and over-fit the training data

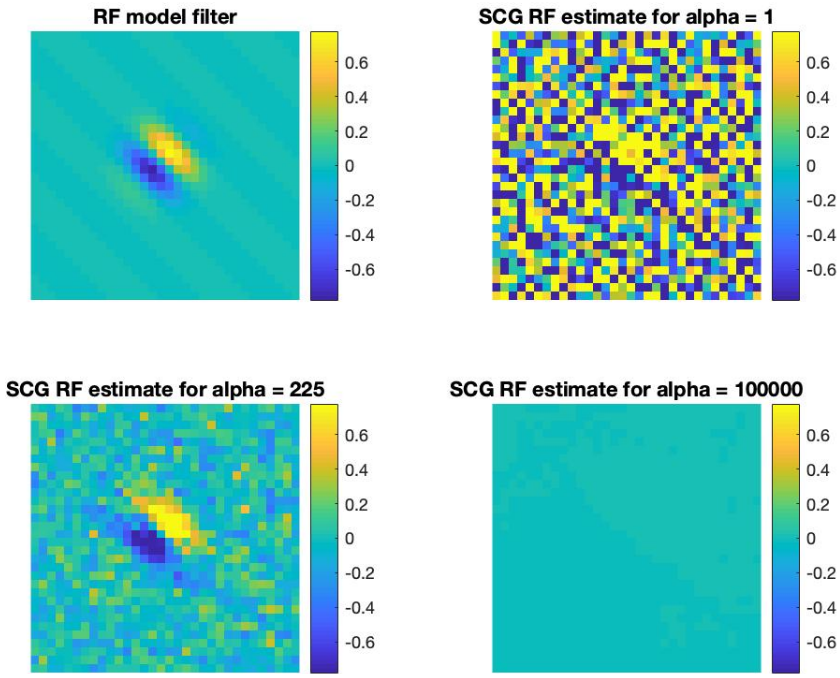

Figure 8 - The actual simulated receptive field as well as estimated receptive fields for several values of alpha given a natural image stimulus

Finally, the results are evaluated using cross-correlation instead of regression. The cross-correlation results are plotted in Figure 9. A comparison across regression and correlation methods of system identification of receptive fields is provided in Table 1. Table 1 compares the variance accounted for (VAF) using regression and cross-correlation receptive field identification with both white noise and natural image stimuli. Stimulus-response cross-correlation will produce a reasonable RF if the stimulus is white noise; otherwise, it is not an appropriate method of estimating a receptive field. The regression model is capable of finding the RF even when the inputs are correlated. For example, Table 1 shows that the regression model with SCG achieves a test set VAF of 39.0% when a natural image is used. Regression is a useful tool because it is highly flexible and is compatible with a number of optimization algorithms and regularization methods.

Method	Stimulus	Variance Accounted For (VAF) (%)
Cross-Correlation	White Noise	17.8
	Natural Image	7.4
SCG Regression	White Noise	24.6
	Natural Image	39.0

Table 1 - Cross-correlation and SCG regression model VAF results for white noise and natural image inputs

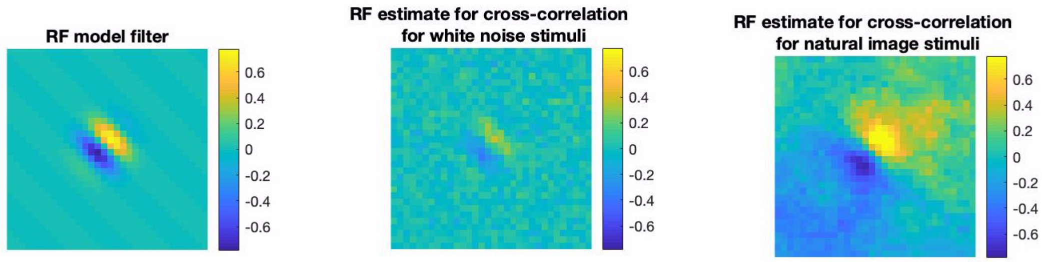

Figure 9 - Cross correlation results for white noise stimuli (center) and for natural image stimuli (right). The actual receptive field is shown on the left