Assignment for "Supervised Learning" - Curtis Baker

Task 1. Single-layer classifier - use provided code and data ("assign_classifier_gradDesc"), modify it 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 ?

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 ?

Task 2. Single-layer classifier with regularization

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. 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 in order 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 about right.

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*").

Modify the code to divide up the data into two sets, for Training (70%) and Validation (30%). Use the Training set for the iterative loop that learns the best estimate of the RF. Evaluate performance by the ability of the estimated RF to predict the Validation dataset, measured as mean-square-error.

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 place to stop, for "early stopping".

Now modify the code, to stop at the best place ("early stopping"), and plot:

- actual (model) receptive field profile, and the "learned" (estimated) receptive field, 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 error (loss).

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.

The provided code partitions the data into two sets, for training and validation. Modify the code, provide a third "Test" partition, for final evaluation of the trained model performance, using VAF (variance accounted for).

For scg-regression and white noise stimuli, modify the code to systematically search for the best alpha value to use for regularization. Graph loss or VAF against alpha (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 effect of the penalty value (alpha) being too small or too large.

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.

alternative options - If you are already very familiar with machine learning and would prefer something more challenging, for part or all of this Assignment, discuss with me to get prior approval. For example: using these or similar tasks with Matlab functions such as glmfit; similar tasks with more complex models and DNN learning using newer ML functions in Matlab (or, with Tensorflow / Python); or some other idea you would like to propose.

Assignment guidelines:

The above exercises should be done in standard Matlab. You must provide a written report, as a PDF file. The text of the report should be a maximum of 8 pages (including Figures), and easy to read (font size=12, margins = 1”). Illustrate key results such as graphs or plots with Figures. Label axes of each plot and give each Figure a short “legend”, describing what it shows and indicating which of your Matlab scripts produced it. For each part of the Assignment, describe in the text what you found, and discuss why (i.e., "what it means") - it is not sufficient to simply show Matlab Figures, with little or no explanatory text. The report should not only describe what you find, but very importantly, should discuss what you think the results mean or illustrate.

Also please provide the Matlab code in a form that I can run (i.e., *include any custom or non-standard functions*), so I can see what you have done. The Matlab code must be provided *in addition to* the above written report; comments in the M-files are very welcome, but they do not substitute for the report.

All of this should be packaged in a single archive file (.zip or .hqx). Please include your last name, as the prefix to the file name - e.g. Smith_603_Assign.zip