Stats315B – Homework 3
Spring 2018

Due: June 8, 2018 (11:59pm)

Submit via Gradescope

1. (15) Consider a multi-hidden layer neural network trained by sequential steepest-descent using the weight updating formula

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta\ \mathbf{G}(\mathbf{w}_{t-1}).$$

Here t labels the observations presented in sequence (time) and $G(\mathbf{w})$ is the gradient of the squared-error criterion evaluated at $\mathbf{w}$. Derive a recursive "back-propagation" algorithm for updating all of the network weights at each step. With this algorithm the update for an input weight to a particular hidden node is computed using only the value of its corresponding input (that it weights), the value of the output of the hidden node to which it is input, and an "error signal" from each of the nodes in the next higher layer to which this node is connected. Thus, each node in the network can update its input weights using information provided only by the nodes to which it is connected.

2. (10) Consider a radial basis function network with spherical Gaussian basis of the form

$$B(\mathbf{x}\,|\,\mu_m, \sigma_m) = \exp\left[-\frac{1}{2\sigma_m^2}\sum_{j=1}^{n}(x_j - \mu_{jm})^2\right],$$

with the function approximation given by

$$\hat{F}(\mathbf{x}) = \sum_{m=1}^{M} a_m B(\mathbf{x}\,|\,\mu_m, \sigma_m)$$

and sum-of-squares error criterion. Derive expressions for the gradient $\mathbf{G}(\mathbf{x})$ with respect to all (types of) parameters in the network.

3. (10) Consider a ("elliptical") radial basis function network with elliptically symmetric Gaussian basis

$$B(\mathbf{x}\,|\,\mu_m, \mathbf{\Sigma}) = \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_m)^t\,\mathbf{\Sigma}\,(\mathbf{x} - \mu_m)\right].$$

where $\mathbf{\Sigma}$ is a positive definite matrix. Show that the output of such a network is equivalent to that of one composed of spherically symmetric Gaussian basis functions (Problem 2) with $\sigma_m = 1$, provided the input vector is first transformed by an appropriate linear transformation. Find expressions relating the transformed input vector $\bar{\mathbf{x}}$ and the transformed basis function centers $\bar{\mu}_m$ to the corresponding original vectors $\mathbf{x}$ and μ_m.

4. (10) Now consider a more general "elliptical" radial basis function network with Gaussian basis functions

$$B(\mathbf{x}\,|\,\mu_m, \mathbf{\Sigma}_m) = \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_m)^t\,\mathbf{\Sigma}_m(\mathbf{x} - \mu_m)\right].$$

Here the matrix $\mathbf{\Sigma}_m$ is allowed to be different for each basis function. In standard feed–forward neural networks the "hidden units" compute (transfer) functions that vary in only one direction

in the input space. Characterize the type of matrices $\mathbf{\Sigma}_m$ that would cause radial basis functions to have this property also. In this sense general (elliptical) radial basis functions networks can be viewed as a generalization of standard feed-forward networks as well.

5. (10) Describe K–fold cross-validation. What is it used for. What are the advantages/disadvantages of using more folds (increasing K). When does cross–validation estimate the performance of the actual predicting function being used.

6. (10) Suppose there are several outcome variables $\{y_1, y_2, \cdots, y_M\}$ associated with a common set of predictor variables $\mathbf{x} = \{x_1, x_2, \cdot \cdot \cdot, x_n\}$. One could train separate single output neural networks for each outcome y_m or train a single network with multiple outputs, one for each y_m. What are the relative advantages/disadvantages of these two respective approaches. In what situations would one expect each to be better than the other.

7. (25) **Spam Email**. The data sets *spam_stats315B_train.csv, spam_stats315B_test.csv* and documentation for this problem are the same as in Homework 2 and can be found in the class web page. You need first to standardize predictors and choose all the weights starting values at random in the interval [-0.5, 0.5].

(a) Fit on the training set one hidden layer neural networks with 1, 2,..., 10 hidden units and different sets of starting values for the predictors (obtain in this way one model for each number of units). Which structural model performs best at classifying on the test set?

(b) Choose the optimal regularization (weight decay for parameters 0,0.1,...,1) for the structural model found above by averaging your estimators of the misclassification error on the test set. The average should be over 10 runs with different starting values. Describe your final best model obtained from the tuning process: number of hidden units and the corresponding value of the regularization parameter. What is an estimation of the misclassification error of your model?

(c) As in the previous homework the goal now is to obtain a spam filter. Repeat the previous point requiring this time the proportion of misclassified good emails to be less than 1%.

HINTS: (a) For the data problems you need to first attach the neural networks library. This can be done by executing the command **library(nnet)**. Examples using the SPlus functions **nnet** and **predict.nnet** are in the book "Modern Applied Statistics with SPlus by Venables and Ripley and the online R help system. Chapter 11 of the text may be also useful for a quick overview of the methodology. (b) For classification use the default logistic output units. A way to choose the percentage of misclassified emails to be less than 1% is to threshold the probability of being a good email at a appropriate value. Using the option **type="raw"** in **predict.nnet**, the output consists of vectors of probabilities. (c) Make sure that you start early. The data problem (tuning neural networks), although not difficult from the conceptual point of view will be time consuming.