CS4100/CS5100 Complement to Assignment 1

October 24, 2019

This complement to Assignment 1 is strictly optional (you cannot earn even extra marks by doing it). It is for students who do not plan to take CS4950/CS5950 (Deep Learning) but are interested in neural networks.

Tasks

1. Implement the neural network for regression (as described in the lectures, Chapter 4, or [1], Sections 11.3–11.5) for p attributes and M neurons in the hidden layer, where $M \in \{1, 2, \ldots\}$ is a parameter of the algorithm. You are allowed to set p to a specific value that allows you to do the following tasks.

2. Apply your program to the `Auto` data set to predict `mpg` given `horsepower`, `weight`, `year`, and `origin`.

3. Split the data set randomly into two equal parts, which will serve as the training set and the test set.

4. Train your neural network on the training set using independent random numbers in the range $[-0.7, 0.7]$ as the initial weights. Find the MSE on the test set. Try different values of the number of hidden neurons M, of the learning rate η, and of the number of training steps.

5. **Even more optional:** Try different stopping rules, such as: stop when the value of the objective function (the training MSE) does not change by more than 1% of its initial value over the last 10 training steps.

6. Fix a value for the learning rate (a reasonably small positive constant) and a stopping rule. Find the best value of M (the number of neurons in the hidden layer) from the training set using 4-fold cross-validation. What is the test MSE for the optimal value of M? **Remark:** You should apply the cross-validation procedure only to the training set. Therefore, it is the training set of size 196 that is split into 4 folds of size 49. You do cross-validation on the training set and compute the test error on the test set.

7. Train linear regression in R on the training set for predicting `mpg` given `horsepower`, `weight`, `year`, and `origin`. (You are allowed to use the function `lm()`.) What is its MSE on the test set? How does it compare to the test MSE for the neural network?

8. Run the neural net for fixed values of M and η and for a fixed stopping rule (producing reasonable results in your experiments so far) 100 times, for different values of the initial weights (produced as above, as independent

random numbers in $[-0.7, 0.7]$). In each of the 100 cases compute the test MSE and visualize it as a boxplot.

9. **Even more optional:** Redo the experiments in items 4–6 modifying the training procedure as follows. Instead of training the neural network once using back-propagation, train it 4 times using back-propagation with different values of the initial weights and then choose the network with the best training (items 4 and 5) or cross-validation (item 6) MSE.

10. **Extremely optional:** Add regularization (as described in the lectures or [1], Section 11.5.2) to your neural network. How does it affect the network's performance?

Remark

It is very easy to make a mistake and implement the following modification of the algorithm on slides 86–87 instead of the algorithm itself:

The algorithm Start from random weights β_m and α_{ml}. Repeat the following for a number of epochs (or until a stopping condition is satisfied). For each training observation ($i = 1, \dots, n$) do the following.

1. **Forward pass:** Compute the variables Z_m and Y for the ith training object x_i following slide 78.

2. **Backward pass:** Compute the "error"

$$\delta := 2(y_i - Y)$$

and back-propagate it to

$$s_m := \delta \beta_m \sigma'(\alpha_{m0} + \alpha_m' x_i).$$

Update the weights:

$$\beta_m := \beta_m + \frac{\eta}{n} \delta Z_m \qquad \alpha_{ml} := \alpha_{ml} + \frac{\eta}{n} s_m x_{il}.$$

This modification gives reasonable results and is even easier to implement than the version given in the lectures.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, second edition, 2009.