KU Leuven

Data Mining and Neural Networks

Assignment 2

Zachary Jones
Master of Statistics

January 6, 2020

Contents

1 Santa Fe Dataset

1.1

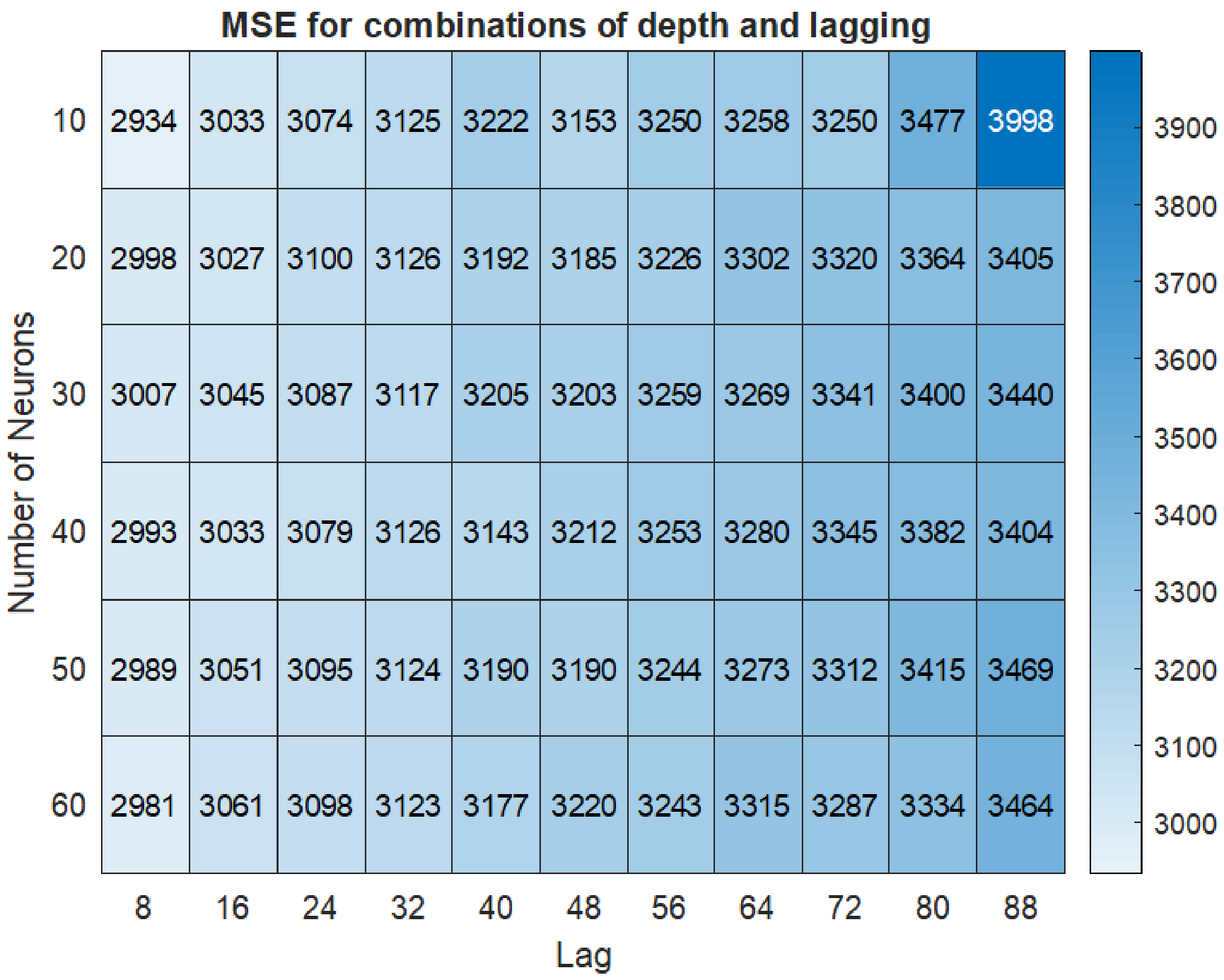


Figure 1: A heat map showing the combinations of neuron and lagging

Looking at the training set, we notice first and foremost that there is a periodicity of the peaks of approximately 8 time steps and a phase change occurs approximately every 600 time steps. So we primarily investigate the lag in increments of 8 and gradually increase the number of neurons. Care, however, must be taken, as random selection of data points for a validation set would violate the time ordering of the data and be a poor indicator of model fit. One strategy is cross validation, taking several disjoint "chunks" of the data in turn as validation sets and averageing over all the errors. However, this does present a complication as it ruins the time continuity of the training set, which may adverseley affect the hyperparameter optimization. Another simpler strategy, is to simply use only "chunks" from the final and/or initial timesteps, eschewing the time continuity problem alltogether. As we can see from 1, the error across all numbers of neurons increases as the lagging increases, and the optimal neural network as trained on the training data using a validation set without recurrent prediction is a single layer network with a lag of 8, which produces an MSE of 2.9×0^3

1.2

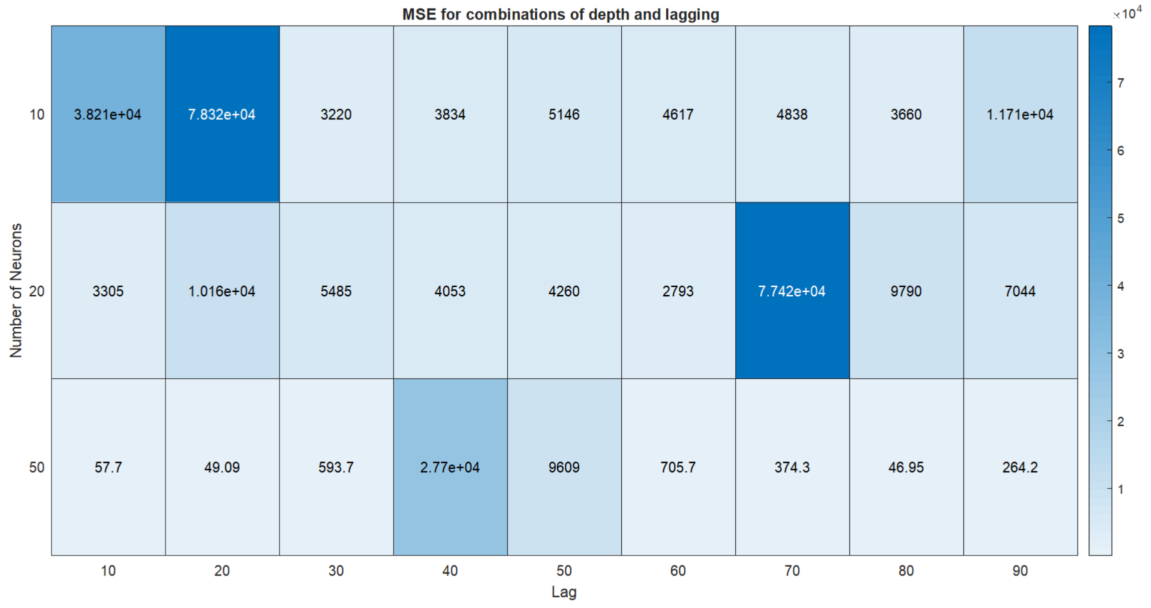


Figure 2: Heat map showing MSE for various neuron and lagging combinations

It is definitely sensible to perform a hyperparameter optimization on a validation set using the recurrent prediction method. After optimizting the hyperparameters using a validation set taken from the final 300 datapoints, we can see that a neural net with 50 neurons and a lag of 80 yields optimal results. When compared to the unoptomized network, using this architecture to recurrently predict the 100 datapoints of the test set produces an MSE of 6.4×10^2, an order of magnitude lower than the NN from the previous part. In 3 we notice that the MLP has difficulty approximating the function after the phase change, and it may be worth adding an additional layer to the neural network in order to "detect" this phase change and aid in classification. A naieve implementation of this strategy with two hidden layers of sizes 10 and 4 respectively and a lag of 16 is shown in figure 4.

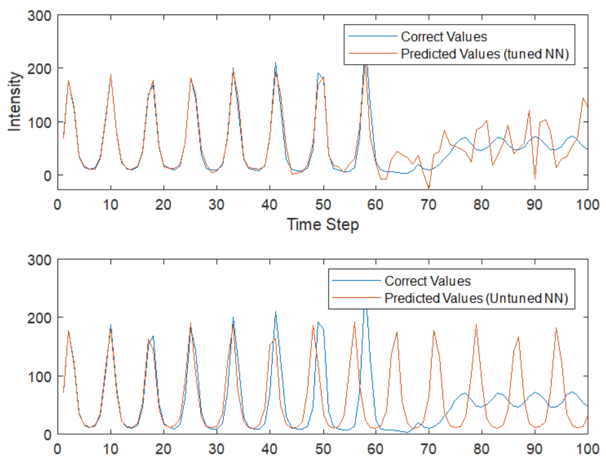


Figure 3: comparison of recurrent predictions generated by tuned NN with 50 neurons and lag of 80 (top) and a neural network with 10 neurons and a lag of 8

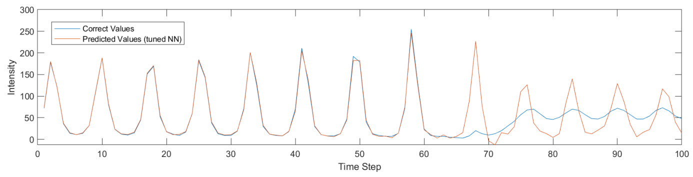
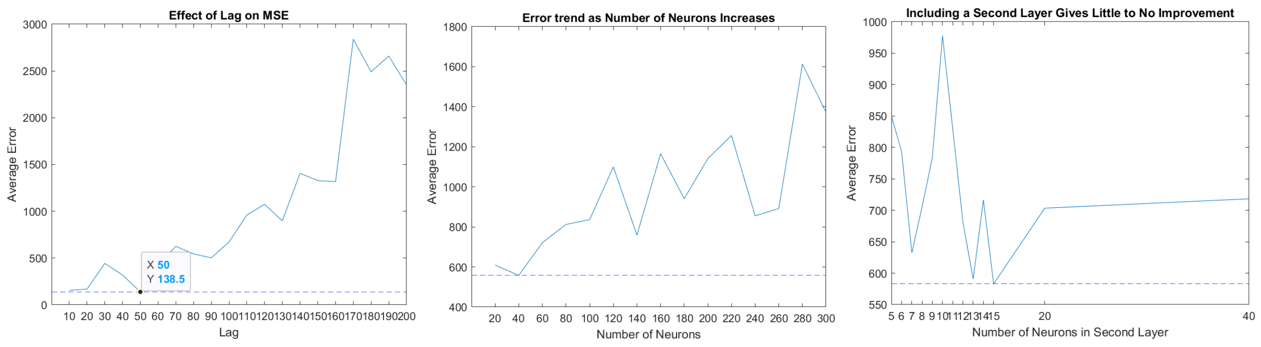


Figure 4: [10 2] Neural Network producing smoother curve closer to actual laser output

2 PM 2.5 Concentration Analysis

2.1



(a) Plot showing a minimum MSE with a lag of 50

(b) MSE minimized when number of neurons in the first layer is 40

(c) Minimum Error is similar to that of the single layer model

Figure 5: Trends for the Lag, Number of Neurons in 1 Layer, and Number of Neurons in the Second Layer

Unfortunately, the only way to choose an optimal model in such a case is trial and error on a validation set. Using the validation set we can adjust the size of each layer, the number of layers, and the time laggings in a systematic way and evaluate their performance using the mean squared error such that we are able to determine the structure of a more optimal model. With an eye towards larger datasets it is important to find "representative" pieces of the data to perform both training and validation on, and with an exceedingly large dataset it may be worthwhile to investigate different architectures on small partitions of the data before applying it to the whole set. To fully specify this model, we test one hyperparameter at a time, choosing a multilayer perceptron with 40 neurons, a lag of 50, and no second layer.

2.2

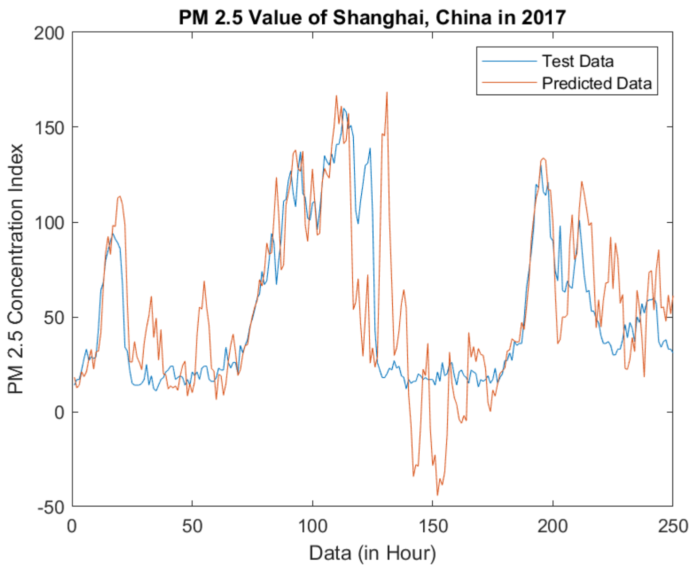


Figure 6: Predictions found using NN specified above with MSE of 950.76

We may be able to increase our accuracy by respecifying our model from an autoregressive model to one that incorporates a "moving average" model. These models, of the form:

$$X_t = \mu + A_t - \theta_1 A_{t-1} ... - \theta_q A_{t-q}$$

Average the data at each step to make a new prediction, before performing a linear regression (which we are essentially doing in the above case). We can perform a similar transformation on the data and outputs of our nonlinear model, which may be more insulated from random shocks.

In addition we can further test our hyperparameters for interaction effects, and we can validate off of recurrent predictions as opposed to prepared test data for a more accurate model.

3 Classification

3.1

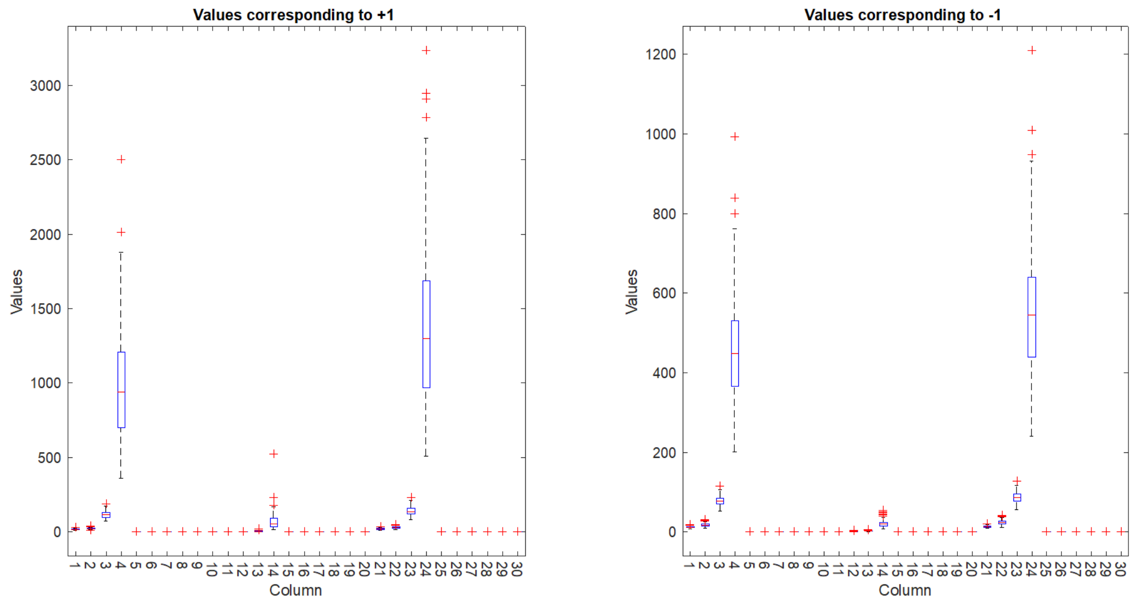


Figure 7: Box Plot showing clear separability of data points

The first step is to visualize the breast cancer dataset, do do this we first look at boxplots of each of the 30 variables. The key takeaway from 7 is that the primary differnce between the two different labels occurs on column 4 and column 24 where the two classes are clearly distinguishable from one another. Another way to represent the data would be through a dimensionality reduction technique such as PCA or a self organizing map.

3.2

In order to build and optimize a neural network to classify these two datasets, as can be seen there is little advantage to a second layer, as it would not serve a classification purpose in distinguishing between a pair of variables. A neural network with two neurons in one hidden layer trained using the trainbr algorithm produces the least error with an average of 1. Since the number of epochs is sufficient for the algorithm to converge, we compare the neural network classifier of all data with a linear classifier trained using stochastic gradient descent on data from columns 4 and 24, with results shown in figure 8

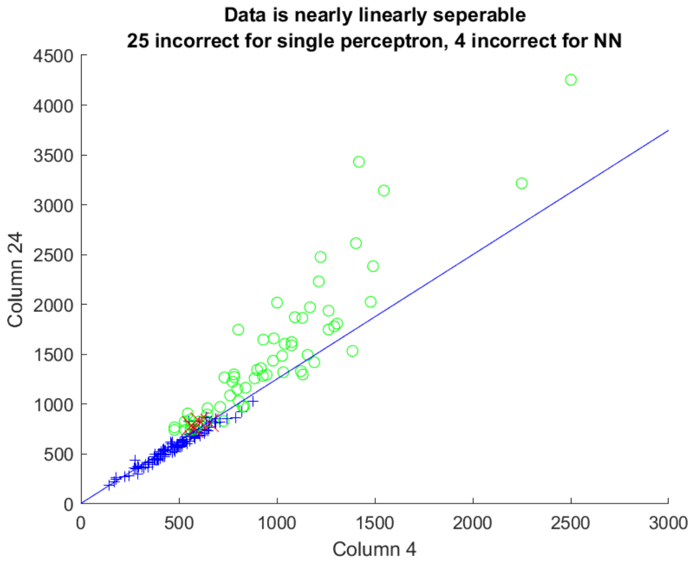


Figure 8: Data presented on columns 4 and 24 with datapoints misclassified by neural network marked in red

4 Automatic Relevance Determination

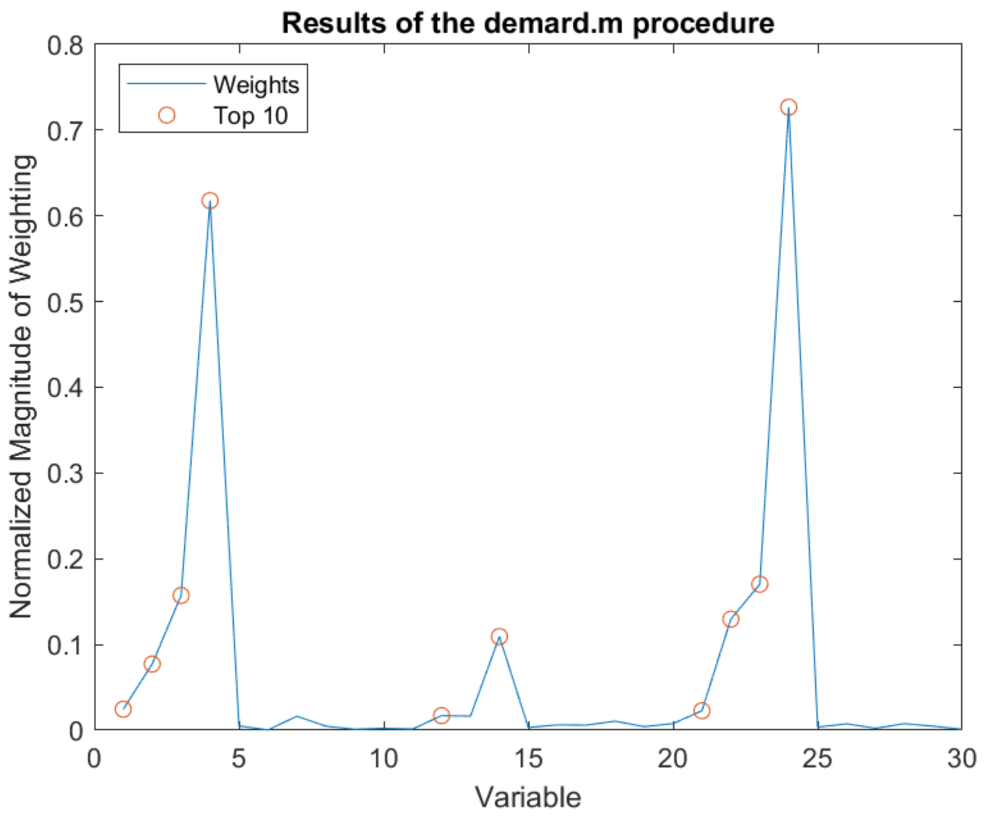


Figure 9: Figure Illustrating the 10 highest weightings from the ARD procedure

We begin this section noting that we can fit a nonlinear regression to the dataset, extract the weight vectors, and determine which factors are most important. As can be seen in 9, a few values "stand out" as clear contributors to the classifier, implying that the data may be learnable from fewer dimensions than previously thought. The most appropriate strategy for including weights is to establish a cutoff point or a maximum number to include. For purely aesthetic reasons, and a desire to include the "peak" value at variable 14, the top six values were chosen, so variables 24,4,23, 3, 22, and 14 were used to train and classify a neural network with the same archetecture as the previous section. It was found to have an MSE of 1.68 and misclassified only 9 points on the test set, giving it 93% accuracy.

What is most interesting is that an almost equivalent interpretation of this result can be found using the simple boxplot procedure. Note the similarity between figures 9 and 7. $X1$ and $X2$ show high variance on variables 3, 4, 14, 23, and 24. An almost 1:1 comparison can be drawn between the two. This is likely because the coefficents of the demard procedure are in fact directly related to the variance of the variables, which contribute more to the final classification if they vary more widely.
Another point of note is how many of the variables are necessary to achieve the same accuracy, a simple plot in figure 10 shows a downward trend in the number incorrect with the number of variables included in the classifier, and illustrates the 15 variables are sufficient to get similar results to a the full dataset.

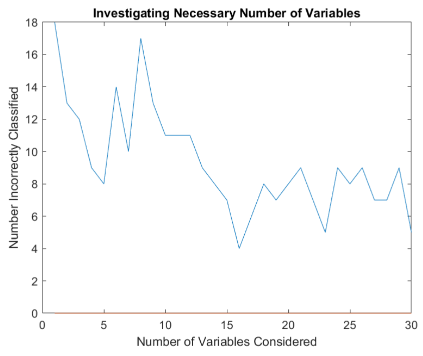


Figure 10