**Heating Load Predicting Using Logarithmic Regressions and a Feed Forward Neural Network with Back Propagation**

Joshua Lawson

**Abstract**

*Energy consumption has become a hot button subject in the United States of America. This encompasses a myriad of reasons such as environmental backlash, cost, and infrastructure. With that in mind, companies should do everything in their power to lower resource consumption to drive profits. To do this, I am looking at predicting heating load in buildings using a LASSO regression (LASSO), Ridge regression (Ridge), and feed forward neural network with backpropagation (FFNN).*

1. **Introduction**

Heating a building comes at a specific energy cost. Each room should be accounted for and a chosen temperature range accepted as suitable. The type of power used also comes into play in the problem as space heating takes up 86% of energy consumption when using natural gas and 5% for electric heating [1]. Lowering the heating load, energy required to heat a building, then allows a company to lower overhead.

So how does a company gauge heating load? There are environmental factors, climate factors, structural factors, and human factors. Pulling those apart I used data from the University of California Irvine (UCI) which used eight variables correlated to heating load [2]. I did this to allow for time constraints on a single semester project. This allowed me to focus on the dependency and build effective algorithms to predict a building’s requirements from:

* Relative Compactness (CMPT): The relative surface area to volume ratio.
* Surface Area (SA): The total area separating heated space from unheated space.
* Wall Area (WA): The total surface area not including floors and ceilings.
* Roof Area (RA): The total surface area not including walls and floors.
* Overall Height (OH): The height of the entire building.
* Orientation (O): The shape of the building.
* Glazing Area (GA): The total surface area for windows.
* Glazing Area Distribution (GAD): How windows are distributed throughout the building.
* Heating Load (Y1): The response value and the energy required to heat a building.

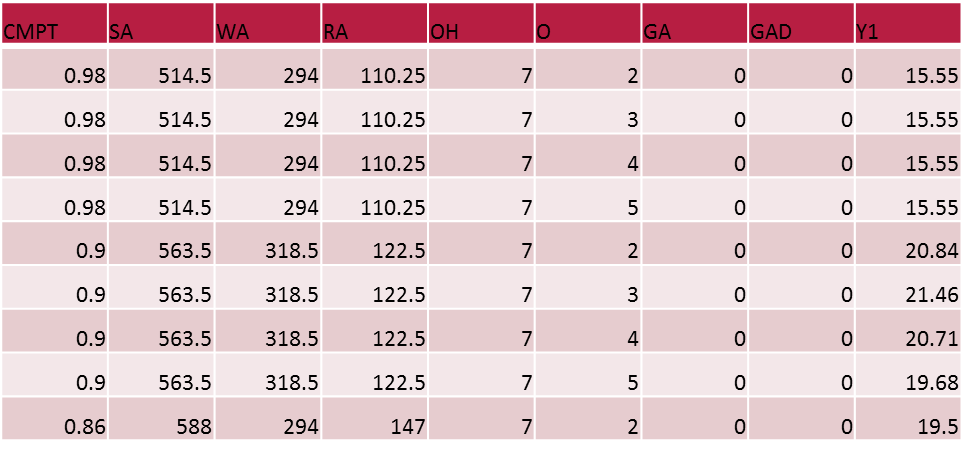
Each of these variables are correlated with heating load and are considered when running the algorithms and will henceforth be known as the predictors. Each model is trained and tested on the 768-observation dataset acquired from UCI to accurately predict heating load.

To correctly test my predictions, I compared each output per observation with the actual given heating load value from the dataset.

1. **Problem Statement**

So how do I correctly predict heating load using the given predictors? I used three different machine learning algorithms to reach a conclusion. I created a standard prediction procedure for each of the three algorithms.

Each predictor, e.g. relative compactness, and it’s given initial value are run through each model to fit the expected outcome. It became finding the correlation between all eight predictors and the response.



*Figure 2.1 A small sample of the dataset.*

1. **Algorithms**

I attacked this problem using three different algorithms designed for regression. I implemented a LASSO regression algorithm, Ridge regression algorithm, and a feed forward neural network using back propagation.2

1. **LASSO**

LASSO, also known as the Least Absolute Shrinkage and Selection Operator, is a penalized regression method. The L1, or the shrinkage of the difference of absolute values, is what the LASSO method uses [3].

2

* G = The function for the LASSO model.
* = The coefficients for the predictors.
* X = The predictor variable matrix.
* Y = The target matrix, the response column in the dataset.
* = Lambda constant.
* N = The number of observations, rows in the dataset.
* p = The number of predictors, columns for predictors.

The minimization of G allows for the best estimate of . In matrix form:

This allows us to turn it into a square matrix. The s are multiplied with the , which is the identity matrix with rows and columns, which then turns it into a identity matrix with on the diagonal [3].

Taking the partial differential of G in relation to :

This gives us a rate of change for and a sign for the direction of change. As this regression is a penalized ordinary least squares (OLS):

Inserting the OLS equation into the differential you can get the , for the LASSO coefficients. Also, if is greater than or equal to zero it is a negative in the equation, less than zero means it is a positive.

The divided by two becomes a threshold for the LASSO coefficients. If the is less than the over two, the will be dropped to zero. This drops uncorrelated predictors from the model to lower complexity. The variable t in the equation is the time variable, the steps your equation takes. Each value of corresponds to a unique t value.

Knowing this, the variable can then be subtracted from the summation of the OLS to give you .

The whole idea behind the LASSO is to shrink the coefficients and to only use the predictors that are correlated with the response [3]. This allows for any observation to have the coefficients applied to it to predict the outcome.

1. **Ridge**

Ridge is another penalized regression method, but uses the L2 penalty rather than the L1 used by LASSO. Therefore, Ridge is known as a sum of squares regression [4]. The penalty for L2 means the coefficient matrix is the squared and summed rather than the absolute value taken.

Just as in the LASSO method you can break this down to matrix form and have:

The partial differential of G taken in relation to :

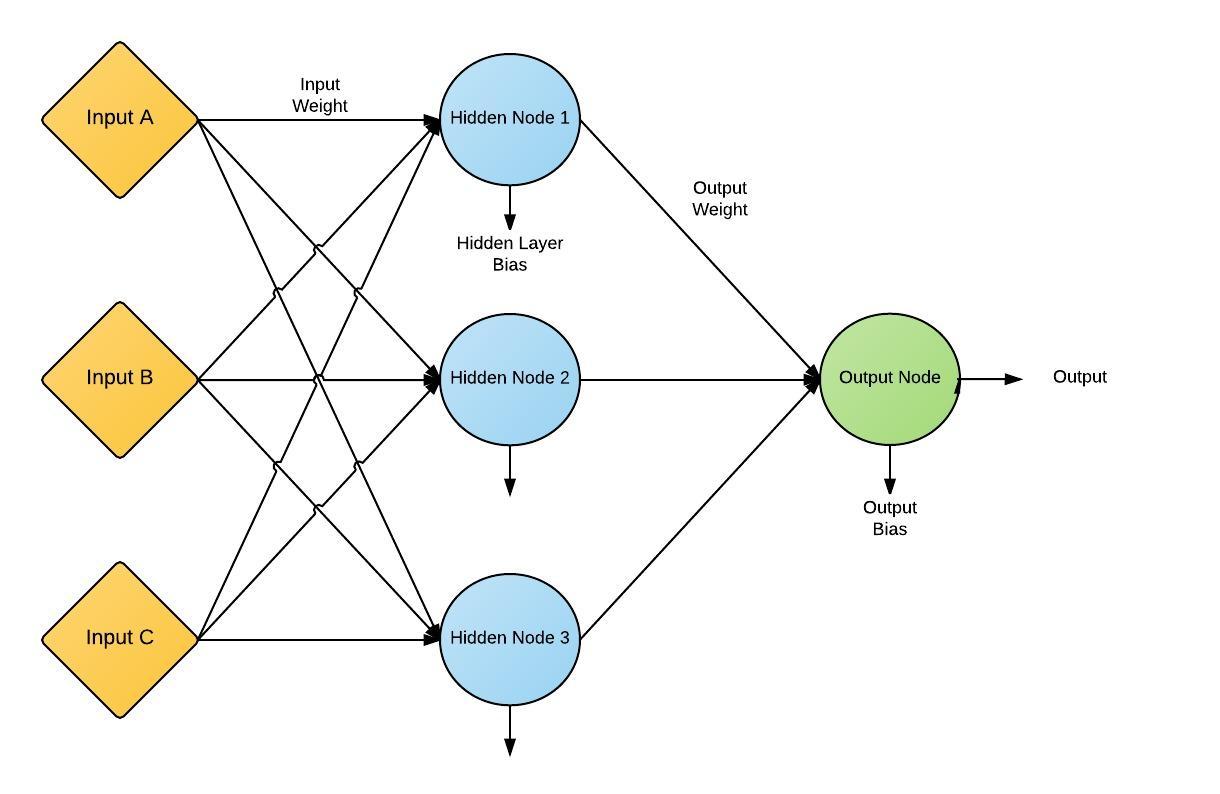
The solution for Ridge can then be put in:

The fact that, unlike LASSO, Ridge does not zero out coefficients and uses either all the predictors or none of them makes it ideal for datasets with non-zero values.

Shrinking the correlated coefficients as low as possible uniformly as well. As the goes up the coefficients go down.

1. **FFNN**

Lastly, the FFNN takes in the predictors as inputs and feeds them through a single hidden layer of equal to the inputs and outputs the predicted value. The architecture for a FFNN lends to regression problems with supervised learning if the response is known for the training set. FFNNs are modeled after the biological brain synapse [5].



*Figure 3.1: Details the Architecture for a Feed Forward Neural Network.*

As shown in figure 3.1, the three input FFNN is made up of inputs (our predictors in the problem), a hidden node layer, and an output node layer. Each connecting line has a randomly assigned, to start off with, weight that will adjust the inputs as they filter through the hidden layer to the output.

The output for each of the nodes is given as:

The constant is left at one and is left out of the equation. The variable x is the , which is given by [6]:

This shows the equals the bias plus the summation of the output times the input weight. The output error signal and the hidden error signal respectively:

For the output error signal, is the target value of the output (what the output should be), and is the actual output. The hidden error signal uses again and is the output error signal. is the weight between the hidden layer and the output node [6]. Once you’ve moved forward through the network, backpropagation takes over and the weight change begins:

The step size chosen for the network is given by . Each of the weights and biases are changed to run through the network again and get an output closer to the target.

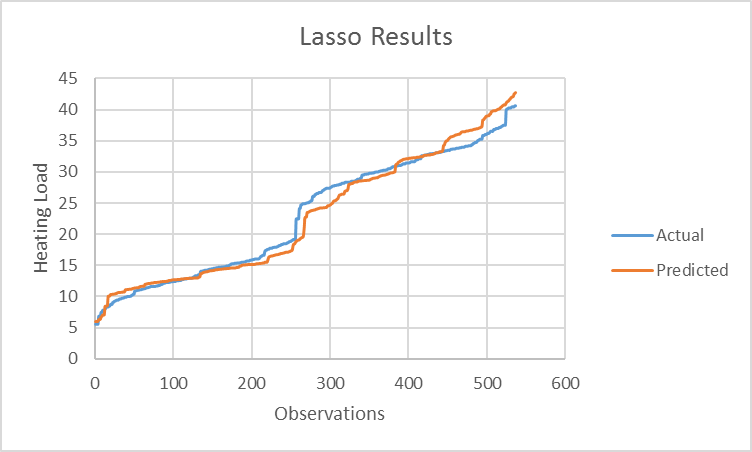
To halt the algorithm from forming an infinite loop you need to solve for the Total-Sum-Squared-Error (TSSE) and the Root-Mean-Squared-Error (RMSE) [6].

RMSE =

1. **Results**

Using these three algorithms, each one predicted the heating load to a varying degree. Starting with the LASSO I trained the algorithm on 40% of the data and tested it on the remaining 60%. The average error for the prediction was 1.16. I had to take into account that the LASSO does drop what it considers uncorrelated predictors out of the model. Both Wall Area and Glazing Area Distribution were pushed to zero.

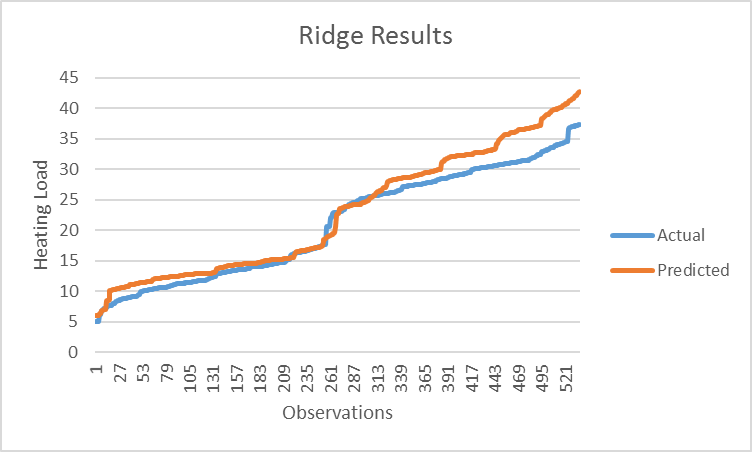
The problem with this technique is that if two predictors are of equal weight one will be zeroed out which can cause issues if the predictors are correlated themselves. Figure 5.1 shows the actual and predicted heating loads for the LASSO.



*Figure 5.1: The results for LASSO regression.*

Ridge followed the same pattern as LASSO and was trained with 40% of the data and tested with 60%. The average error for the prediction using Ridge was 1.86. Both

Wall Area and Glazing Area Distribution were contained in the model but the coefficients were incredibly low. As the heating load grew the prediction became farther off and overfit the data.



*Figure 6.1: The results for Ridge regression.*

Lastly, the FFNN had predictions that beat out both other algorithms. Trained on 50% and tested the other 50%. The average error of the prediction for the FFNN came out to be 0.48, which is less than half that of LASSO. Though the efficiency could be

considered if used with larger datasets. The FFNN took almost ten times as long to run with both LASSO and Ridge taking under a second and the FFNN taking just over ten seconds.

*Figure 6.2: The results for the FFNN.*

With each method, heating load can accurately be predicted and allow for companies to make intelligent decisions based upon construction and purchases of new buildings. However, I would argue that if a company needs to base financial stability on energy consumption, the FFNN should be used to predict the heating load.

1. **References**
2. "Energy Use in Office Buildings." (1980): n. pag. Madison Gas and Electric. E Source Companies LLC., 2010. Web. 18 Oct. 2016.
3. A. Tsanas, A. Xifara: “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools', Energy and Buildings, Vol. 49, pp. 560-567, 2012
4. Zhao, Xi. "Lasso Regression Thesis." (n.d.): n. pag. Http://www.d.umn.edu/. University of Minnesota Duluth. Web. 10 Oct. 2016.
5. Marquardt, Donald W., and Ronald D. Snee. "Ridge Regression in Practice." The American Statistician 29.1 (1975): 3-20. Web. 20 Oct. 2016.
6. Rosa, João Luís Garcia. "An Artificial Neural Network Model Based on Neuroscience: Looking Closely at the Brain." Artificial Neural Nets and Genetic Algorithms (2001): 138-41. Web. 1 Nov. 2016.
7. Tagliarini, G. (2010). An Introduction To The Backpropagation Algorithm [Powerpoint slides]. Retrieved from http://people.uncw.edu/tagliarinig/Courses/415/Lectures/An%20Introduction%20To%20The%20Backpropagation%20Algorithm.ppt