Energy Demand Forecasting Using Neural Networks: A First Step to Peak Shaving Cost Minimization

By Graham Booth, Jonah Simon, Trisha Singh

Github URL: https://github.com/gbooth27/ML_final_project

Introduction

The problem that we chose to investigate was how to choose the optimal battery size for a commercial building for the purpose of peak shaving. Utilities charge commercial buildings for their monthly electricity usage in two parts. One part is a fixed rate (approximately 20 cents/kWh) times the total amount of energy that the building consumed, and the other is a peak demand rate, where the utility charges a much higher rate (approximately $18/kW) times the peak power demand for a 15 minute period during that month. The peak power demand charge often makes up nearly half of a building's electricity bill, despite resulting from only one 15 minute period. In order to lower this part of the bill, batteries can be used, where they charge at times of low electricity usage and then discharge at high usage times so that the monthly peak is lowered. This practice is called peak shaving.

The problem of choosing the optimal battery size involves calculating the net profit, in present value, of a range of battery sizes, and then selecting the maximum profit as the optimal battery sizes. It is necessary to take the present value because the costs are the upfront price of installing that size battery, and the benefits, in the form of electricity bill savings, are realized over the lifespan of the battery.

In order to find the optimal battery size, we need to estimate how well the battery operator will be able to choose the level at which to discharge the battery. This process involves forecasting energy usage in the next day and the next month, which we attempt by training a neural network with historical energy consumption. The data that we used for this project was two years, 2014 and 2015, worth of energy data for two commercial building sites. For a given 15 minute period i, we first predict the energy usage in period i + 96, which is the same time on the following day. We then use this prediction as a feature to propagate our forecast forward, up to one month in the future with reasonable accuracy.

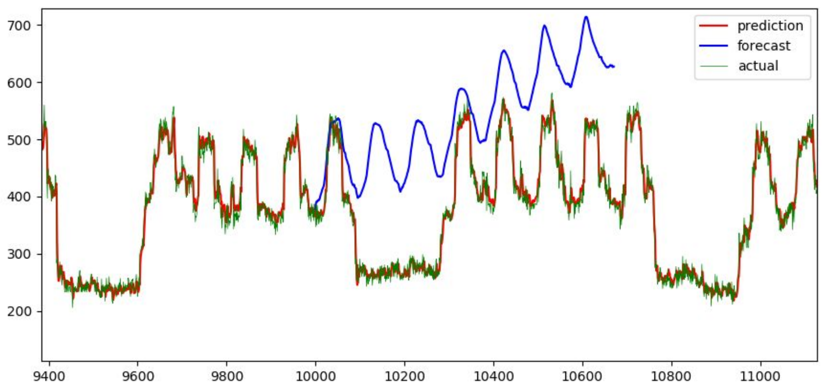
The percent accuracy of these estimates then informed us as to how much of the maximum possible peak shavings the battery operator would be able to achieve.

Experimental setup

We used data for two sites over 2 years for 15 minute periods, with 70,176 observations per site. First, we parsed the site 1 dataset to make it a list of numpy arrays.

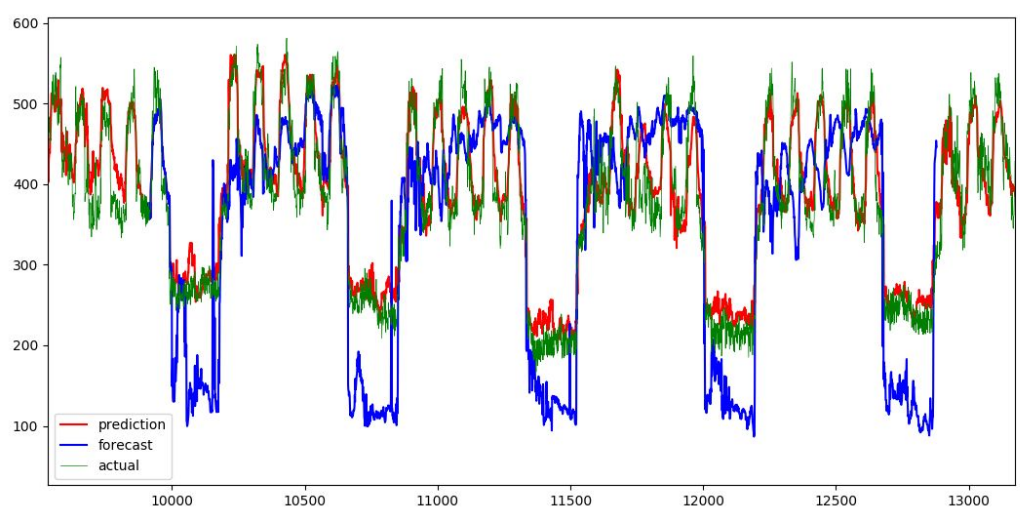
We then conducted some exploratory data analysis to aid in generating features (for example, see Figure 1). We generated around 200 informative features, such as the following: the minute, hour, month, a dummy for whether the day is a weekend or weekday, whether the day is a US holiday, energy usage over the past 96 time periods, the minimum energy load over the past 3, 6, 12, and 24 hours, the maximum energy load over the aforementioned periods, average load of the same hour in all days of the previous week (also the same only for weekdays), the maximum, minimum, and average consumption for that day in the 4 previous weeks. This feature generated is adapted from the paper by Gajowniczek (2017).

We trained a neural network on this dataset to predict the energy consumption for the 15 minute period, initially we were predicting the usage for the next 15 minutes, but soon realized that if we were aiming to forecast any further than about a day the compounding error meant that our forecast was drastically over predicting. We were forecasting by propagating the network's predictions by appending them back as features and then predicting the next 15 minutes until we got a whole week. However, this resulted in sub-optimal results, shown below:



The blue line is the forecast data, which deviates more the further out we get

The mean square error for this weeklong forecast was 27573 kW which is not very good. We realized that the algorithm was likely too dependant on more recent usage data instead of taking historical trends like weekend usage into account. We decided the best option would be to instead predict usage for that 15 minute period 24 hours in the future as opposed to the 15 minutes in the future we were initially predicting. Because we were now predicting the value 24 hours into the future, the neural net did not rely as heavily on the more recent data, and took historical trends into account more. This allowed us to forecast with reasonable accuracy a full month in advance, propagating the predictions forwards as shown below:



This is a sample forecast for 1 month forwards in time, represented by the blue line. The weekends are under forecasted, but the algorithm does a good job of guessing the data for the weekdays.

As you can see, the issue where the algorithm was predicting higher the further into the future we went was alleviated, and it was better able to interpret historical trends. The mean square error for the month long forecast was 7847 kW which is much better than the 27573 kW error we were getting for just one week previously.

After tuning the network architecture on a cross-validation set, our network architecture was set to [m, 400, 200, 200, 200, 200, 400, 1000, 200, 200, 400, 200, 200, 400], where m is the number of training examples. Thus, there are 13 hidden layers, each with a ReLU activation function. We used the keras ADAM optimizer and ran our training sessions on a GTX 1060 GPU on a gaming laptop. We had to transition to the GPU after it started to take multiple hours to train our net on a CPU. We split the dataset into 80% training and 20% and trained the model. We used the mean squared error for evaluating the model, and tuned it such that it would have a low error on both the training and cross validation sets. We then used site 2 data as the test set.

After deciding on a model, we forecasted the energy consumption for a month using the forecasting technique mentioned previously. The operator of our the building's batteries would then use this to calculate a threshold to try to shave the peaks above for a given month. We used the percent accuracy to calculate how close the operator would come to the lowest threshold they would set if they new exactly what the energy usage would be. We made a series of assumptions about the batteries and the inflation of rates in order to solve for an optimal size. One assumption about the batteries was that we plan for them to last 10 years, and to not have any loss in capacity over those 10 years. Since the standard Tesla Powerpack has a ratio of kWh capacity to peak kW discharge of 2:1, we assumed this to be a constant ratio for all batteries that the building owner would purchase. The price of a Tesla Powerpack is also $500 per kWh, so we used this as our price. We also assumed electricity demand prices to go up with inflation at a constant 2% annually. Lastly, we assumed that the energy usage for the next 10 years would be the same as in each of the two years of our data set. Since we do not have more years of data to generate a trend nor do we know why other sorts of energy efficiency technology will be installed, we figured this to be the best assumption.

Results

We trained and tuned the model on site 1 data by splitting it into a training and cross validation set. We then evaluated the performance of this model on site 2 data by training it on a training set, and then evaluating it on a test set.

On site 1 data, we had a mean squared error (mse) of 3135.52 on the training set (mean absolute error of 37.564 kW), and 5200.01 for the cross validation set (mean absolute error of 60.0972kW) after 50 epochs. After a more epochs, the cross validation error decreased. When used for forecasting on the test set, we generated an mse of 7000.21. This higher error is expected since the error in predicted values will compound as we forecast for next time periods.

The performance of the model can also be visualized in Figure 2. When we used this tuned model on the site 2 dataset, we had an mse of 1734.52 on the training set, and an mse of 2844.85 on the test set.

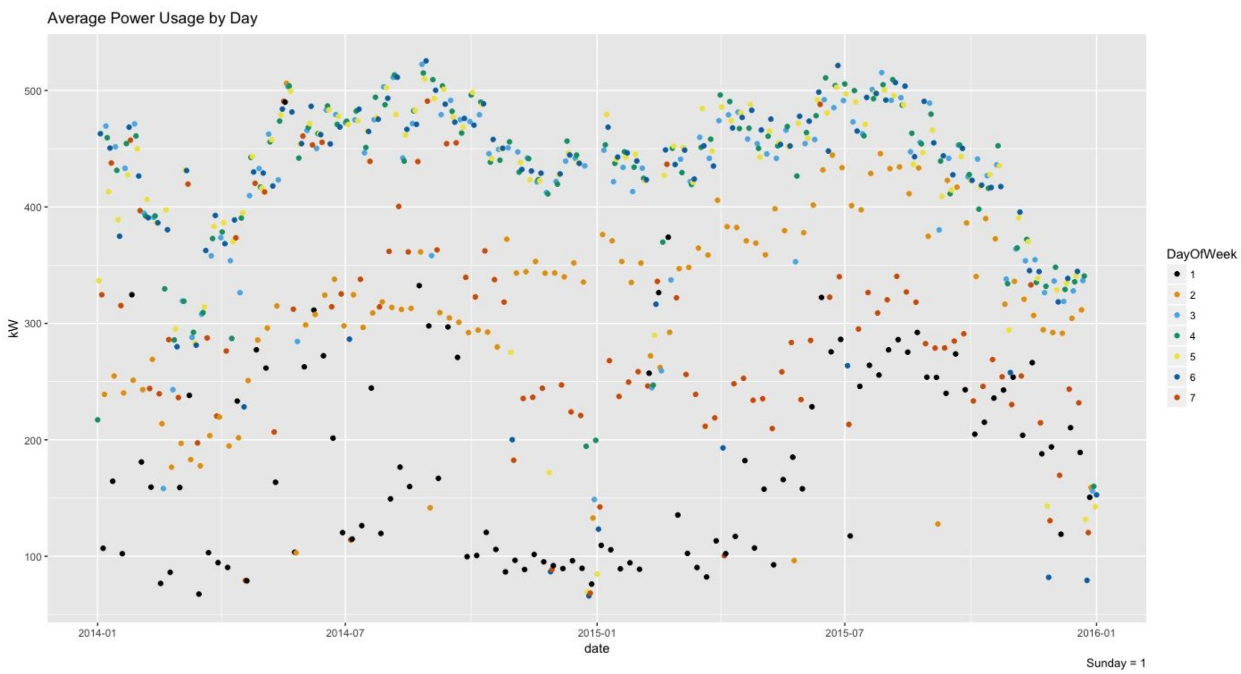
For our end result of optimal battery size for each building site, we calculated the optimal size for each site based on each year of data. For building site 1, we found that based off both 2014 and 2015 usage, a battery of 150 kWh capacity and 75 kW peak discharge to be most cost effective. The battery would cost $75,000 to install, and over the lifespan of the battery, save $166,700. For building site 2, we found different sizes based on each year. Assuming the next 10 years are all like 2014, the optimal size is 200 kWh, 100 kW peak discharge. It would cost $100,000 to install and save $209669 over 10 years. Using 2015 data, the optimal size is 150 kWh capacity and 75 kW peak discharge, costing $75,000 and saving $164068.

Conclusion

Based on our results, we thought that neural network were effective for predicting energy consumption. With more data and a deeper neural network, we could reduce this error and arrive at tighter forecasts. Currently, our forecasts become further from the actual value since each error in prediction compounds when the consumption for the next period is predicted. Thus, it is important to reduce the bias in our model.

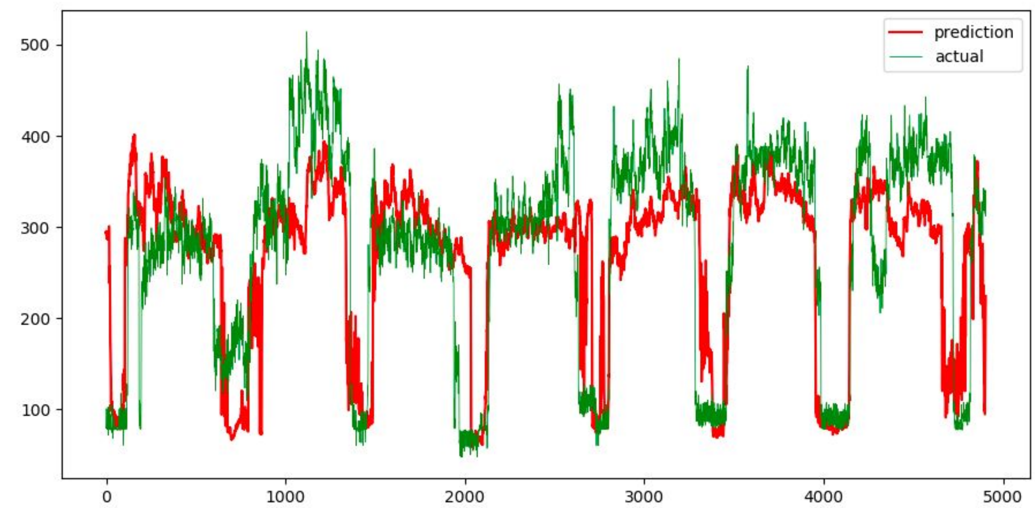
Figures

Figure 3: Exploratory Analysis of Power Consumption by Day



Note: The variation in energy consumption in this plot shows the need to have dummy for each day of the week in order to capture this variation.

Figure 4: Performance of the Model after 50 Epochs



Note: The red line represents our predicted values on the test set.

<u>Sources</u>

Gajowniczek K, Ząbkowski T (2017). Electricity forecasting on the individual household level enhanced based on activity patterns. PLoS ONE 12(4): e0174098. https://doi.org/10.1371/journal.pone.0174098