Power Consumption Prediction Algorithm

Yash Gupta
0869229
Software Engineering
ygupta1@lakeheadu.ca

***Abstract*—The paper focuses on developing a product using different concepts of deep learning and data analysis. The paper explores concepts of Artificial Neural Network and Long-Short Term Memory. The paper uses these concepts to develop a prediction system for power consumption on a future day using data provided Synergy North.**

I. Introduction

Since 1956, AI has come a long way and now is a crucial part of our lifestyle. AI is now being used to solve world problems in many fields healthcare, Economy to name a few. The level at which we are operating is ANI or Artificial Narrow Intelligence. Artificial narrow intelligence (ANI), also referred to as weak AI or narrow AI, is the only type of artificial intelligence we have successfully realized to date. Narrow AI is goal-oriented, designed to perform singular tasks - i.e. facial recognition, speech recognition/voice assistants, driving a car, or searching the internet - and is very intelligent at completing the specific task it is programmed to do. These algorithms or models can be used to find patterns within data or long sequences of data.

Deep learning is one of the most exciting and powerful branch of Machine Learning. Deep Learning is the key behind most of the technology driven by AI. It is achieving results that were unimaginable before. Deep learning can be used to perform regression, classification or clustering. Deep learning models can sometimes achieve accuracy that can exceed human expectations. Deep Learning models are usually trained with a large dataset that can be either labelled (regression & classification) or unlabelled (clustering) and the neural network can be dense or shallow.

II. Artificial Neural Network

Artificial Neural Network or ANN is an information processing paradigm that is inspired by the way the biological nervous system such as brain process information. It is composed of a large number of highly interconnected processing elements or neurons working simultaneously to solve a specific problem. Figure 1 is an example of a dense ANN. From the figure we can see how multiple layers of neurons interact with each other and how information is passed from one layer to another.

The first layer of the ANN or any neural network is the Input layer and the number of neurons is determined by the number of features in the dataset. The input layer 'X' further passes the information the next layer by multiplying with a weighted synapses 'W' and undergoes some activation function or a mathematical function like sigmoid, tanh or Relu to name a few depending on the dataset and the model being used. After the activation function us applied to the input which results in an output. This process occurs with multiple neurons and over many layers to achieve the desired output. Figure 2 shows the mathematical representation of an ANN.

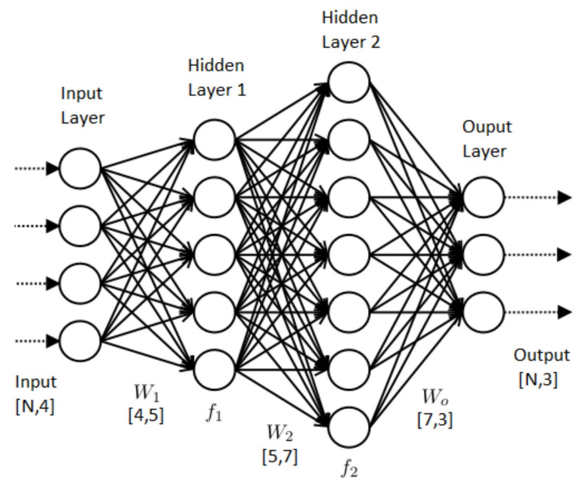

Fig. 1. Visualization of a Dense ANN

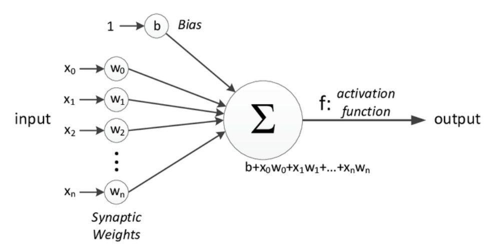

Fig. 2. Mathematical representation of ANN

In order to learn from the predicted output and the actual output, Neural Networks use an algorithm called Backpropogation. The algorithm is used to effectively train a neural network through a method called chain rule. In simple terms, after each forward pass through a network, backpropagation performs a backward pass while adjusting the model's parameters (weights and biases). During backpropogation each weight is updated through gradient descent. ANN learns through constant forward pass and backpropogation.

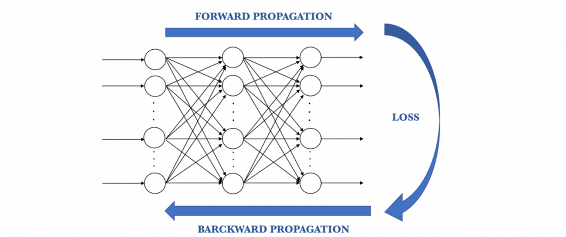

Fig. 3. Learning through Backpropogation

III. Recurrent Neural Network

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other. Like any other model RNN has its advantages and disadvantages. RNN can process input of any length without increasing the size of them model. A few drawbacks of using RNN are slow computation since long sequences are processed for every input, difficulty of accessing information from a long time, gradient vanishing and exploding problems.

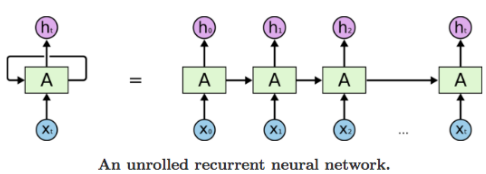

Fig. 4. Visualization of RNN

A. Long Short-Term Memory(LSTM)

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation through time as it is used to process sequences.

LSTM cell has three gates Input gate, Forget gate & Output gate. The input gate helps in discovering which values from the input should be used to update the memory. The sigmoid function of the gate makes the decision which values should go through since it results in either 1 or 0 and the tanh function helps in providing the some weight to the passed values ranging from -1 to 1 to signify their importance. The forget gate of the cell helps in discarding details from the block so that only important information is retained in the cell to make the computation efficient and effective. The decision is made by the sigmoid function which uses the previous state (h_{t-1}) and the current input provided (x_t). The sigmoid produces an output between 0(omit this) and 1(keep this) from the previous cell state C_{t-1}. The third and the final gate is the output gate which uses the results from the input and forget gate to decide the output of the current cellC_t. The sigmoid function of the cell helps in deciding which values should be passed and the tanh function provides a weight to the passed values to signify their importance multiplied by the value result with the output state.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Fig. 5. Input Gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Fig. 6. Forget Gate

$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Fig. 7. Output Gate

IV. Case Study

Electrical load forecasting is an important process that can increase the efficiency and revenues for the electrical generating and distribution companies. Understanding the future long-term load helps the company to plan and make economically viable decisions regarding future power generation and transmission investments. The forecasting can help the company in planning future power generation plant locations by knowing the areas of higher demands, they can most likely generate more power near the load. Forecasting can also help in maintenance of the power systems. The forecasting can also help in avoiding under or over generation. Forecasting can also help to determine the required resources such as fuels required to operate the generating plants as well as other resources. There are multiple challenges faced during power consumption forecasting. Power consumption forecasting is a difficult task due to its complex nature as it depends on many variables such as seasons where consumption in two seasons may vary, customer usage behaviour using different meters and tariffs to name a few.

Finding the peak load can be beneficial for a lot of reasons. It can help in reducing brownouts. A brownout is an intentional or unintentional drop in voltage in an electrical power supply system. Intentional brownouts are used for load reduction in an emergency. The reduction lasts for minutes or hours, as opposed to short-term voltage sag (or dip). Brownouts happen because the companies cannot keep up with high demands at certain times. This can happen when the company did not expect a high demand and just lacks the resources to keep up with it. In order to keep up with such high demands small scale power providers like Synergy North has to buy power from large companies for example Hydro One at a steep price. This amount might be small on a household stage but can be quite expensive at a large scale for example an entire city.

A possible solution to such a problem can be large 2 way batteries to help companies meet with the increasing demand. These batteries can be discharged to meet with the demand and can be charged when idle. But the issue at hand is knowing when to discharge the batteries. A possible solution can be provided by building a prediction model that can predict the power usage for the future day. Such models can be useful in helping the companies to discharge the batteries at the right time to make the process more efficient and meet with the growing demand. This can help in reducing brownouts and help companies to save money. The models have to have high accuracy in order to maximize efficiency and reduce power wastage. The problem can further be resolved with the involvement of electric cars.

The neural networks can be used for a variety of scenarios. In our case these neural networks are used to predict the power consumption of an area using historic data. The idea is to train the models on the recent data for example a week or a month before the prediction date in order to capture the most recent trends. This can help in eliminating long training time.

The data was sampled every 5 minutes of the day i.e. most days had 288 points. The dataset had 17 features ranging from date, weather variables and days of the week. The time variable was represented in terms of Sine and Cosine in order to capture all the time points in a day and making them numerical for easy processing. Basic EDA was performed on the data to find correlations between variables and to observe the nature of the dataset. The data was seasonal and therefore different time series models could be used in order to build a prediction system.

A. *Implementation*

For the project two different neural networks were implemented FFN and LSTM. Both neural networks required different preprocessing of data since the input is respectively different.

1) FFN: The FFN is one of the most important models to implement while dealing with prediction systems since it can provide a good understanding of the dataset being used. The FFN provided with some good results but failed to deliver on odd days.

TABLE I
RESULTS OBTAINED FROM FFN MODEL

Date	*Layers*	*M.S.E.*
2017-07-12	1	0.002978
2017-07-12	3	0.001138
2017-07-12	5	0.002656
2018-01-12	1	0.002884
2018-01-12	3	0.003158
2018-01-12	5	0.002168

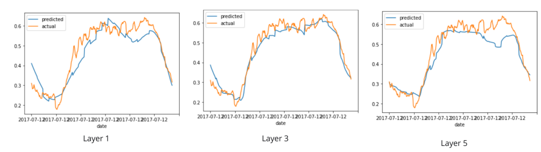

Fig. 8. Resulting Graphs from Feed Forward Neural Network on a random Summer day

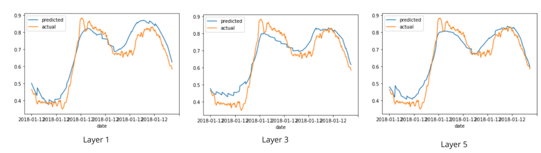

Fig. 9. Resulting Graphs from Feed Forward Neural Network on a random Winter day

2) LSTM: LSTM neural networks are know for learning long sequences and hence were beneficial in our project. Multiple variants were used with variable window size i.e. number of past instances, hidden layers and size of each layer. The LSTM network though powerful wasn't able to predict accurately on odd days since we were training the network with short windows and recent data from the prediction date.

TABLE II
RESULTS OBTAINED FROM LSTM MODEL

Date	*Layers*	*M.S.E.*
2017-07-12	1	0.004531
2017-07-12	3	0.001483
2017-07-12	5	0.003898
2018-01-12	1	0.003826
2018-01-12	3	0.002644
2018-01-12	5	0.003008

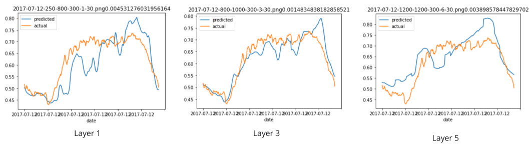

Fig. 10. Resulting Graphs from Long Short-Term Memory Neural Network on a random Summer day

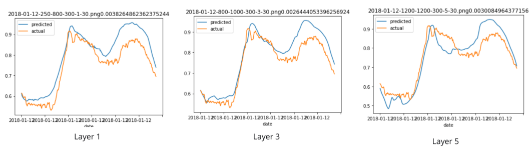

Fig. 11. Resulting Graphs from Long Short-Term Memory Neural Network on a random Winter day

V. Analysis

A. Feed Forward Neural Network

From the results mentioned above it can be observed that the simplest of neural network can provide really good results. The model was implemented with variation in layers, neurons and epochs. It can be observed that increasing the number of hidden layers combined with a longer training time and neurons can provide really good results. A possible explanation can be that the neural network has more layers to perform feature extraction and has longer time to learn and extract the pattern in the training set.

B. Long Short-Term Memory(LSTM)

From the results it can be observed that LSTM models did a great job in analysing the trends as well as tried to mimic the peaks of the data. Different level of layers provided different results and it is safe to say that the model with 3 layers had the least amount of error when compared to the others. A possible idea behind the results can be the time taken by the model, since it had just the right data to learn from but not over-fit it and since the window was small it could retain most of the recent memory which helped in prediction.

VI. Future Work

The system can be improved by performing in depth EDA and feature engineering, i.e. finding more important variables and reducing some. The system can be improved by training on a bigger dataset instead ff small subsets in order to capture the different trends in the data. Access to faster hardware can play a crucial role in providing quick results and hence providing more time for improving the models.