Load Forecasting of Smart Meters Using LSTM

by

Shalini Manna

Enrolment No-510617014

B.Tech 4th Semester

A project report submitted for Summer Internship 2019

Under the guidance of ***Dr.Paramita Chattopadhyay***

Department of Electrical Engineering

Indian Institute of Engineering Science & Technology

Shibpur, Howrah-711103

May 2019-July 2019

INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR, HOWRAH - 711103.

FORWARD

I hereby forward the summer internship report entitled **Load Forecasting of Smart Meters using LSTM** prepared by **Shalini Manna** (Enrolment No.: **510617014**) under the guidance and supervision of **Dr. Paramita Chattopadhyay** for completion of the summer internship project 2019.

_____

Dr. Paramita Chattopadhyay (Supervisor)
Associate Professor,
Department of Electrical Engineering,
IIEST, Shibpur,Howrah - 711103.

Countersigned by

_____

Dr. Prasid Syam
Professor and Head,
Department of Electrical Engineering
IIEST, Shibpur,Howrah - 711103.

Introduction

Measuring electric energy consumption is essential for promoting economic growth and raising standards of living. In a country like India, where the population is vast and varied, meter readings are often found to be faulty, and denizens are wrongly charged.

The new “smart meters” aim to supplant old gas and electric meters and take automatic readings, which it communicates to the electricity supplier for monitoring and billing. This helps to curb incorrect readings and is found to be cost-effective. With the in-home display, the consumer will also be able to see how much energy is being consumed.

In this project, the aim is to develop a Machine Learning model using the Long Short Term Memory algorithm to try and predict the average household energy consumption. The significance of this would be exponential, as it could be applied to detect energy theft, curb the wasting of energy, among a few applications.

In this current project, the London Smart Meter dataset was used to model the algorithm, the data was explored following which a model and its results were observed.

Theory

1.Time Series Forecasting

Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. Predictions are made for new data when the actual outcome may not be known until some future date. The data itself has another dimension- the time dimension, which adds an order of dependence. Time series are widely used for non-stationary data, like economic, weather, stock price, and retail sales.

2.Long Short Term Memory

Recurrent Neural Networks are neural networks that can remember its past and its decisions are influenced by what it learns from the past. A network becomes “recurrent” when one repeatedly applies the transformations to a series of given input and produces a series of output vectors. There is no pre-set limitation to the size of the vector. And, in addition to generating the output which is a function of the input and hidden state, one updates the hidden sate itself based on the input and uses it in processing the next input.

Long short-term memory (**LSTM**) is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer".It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

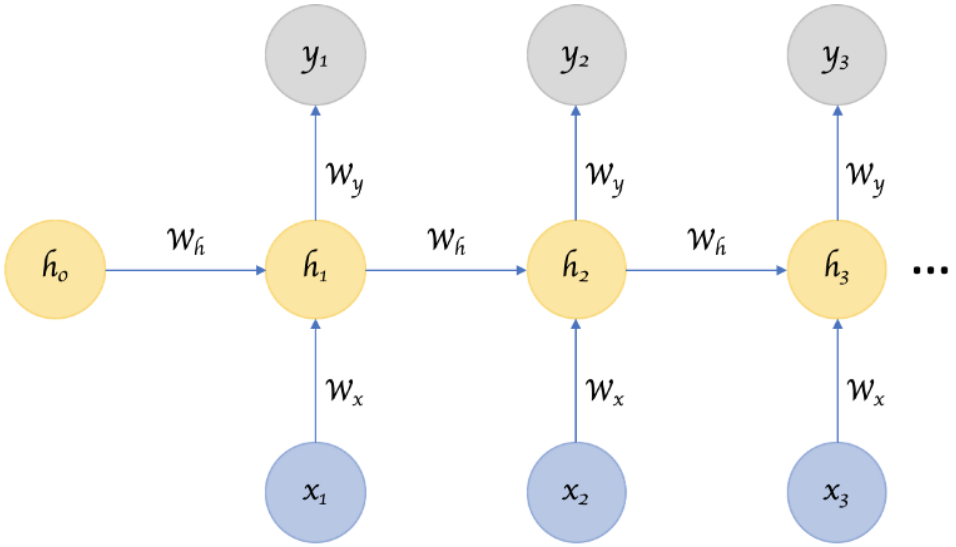


Figure 1Recurrent Neural Network

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

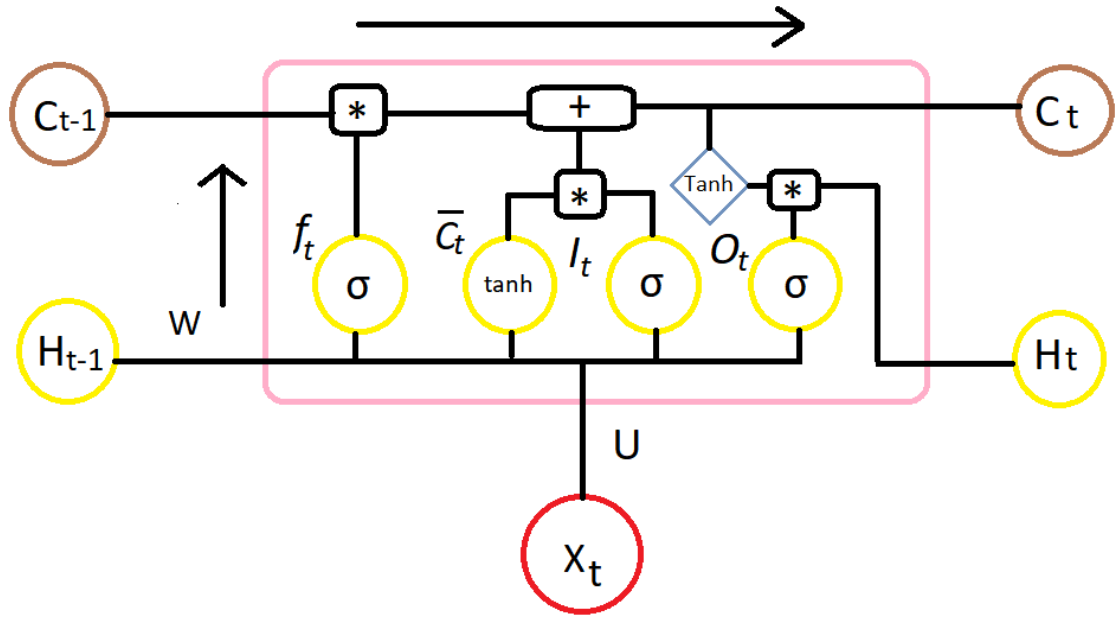


Figure 2:LSTM Architecture

2.1 Architecture

A typical LSTM network is comprised of different memory blocks called cells. There are two states that are being transferred to the next cell- the **cell state** and the **hidden state**. The memory blocks are responsible for remembering things and manipuations to this memory is done through three major mechanisms, called **gates.**

2.1.1 Forget Gate

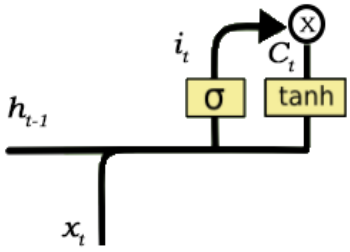
It is responsible for removing information from the cell state. The information is no longer required for the LSTM to understand information. Unwanted parts are removed via multiplication of a filter.This is required for optimizing the performance of the LSTM network. This gate takes two inputs—**H_t-1** and **X_t.**

H_t-1 : The hidden state from the previous cell**, X_t** : Input at the particular time step.

The given inputs are multiplied by the **weight matrices** and a bias is added, following which a sigmoid function is applied to this value. The sigmoid function outputs a value from 0 to 1. A ‘0’ indicates information to forget, ‘1’ means that it has to be remembered.

2.1.2 Input Gate

The process of adding information can be done via the input gate. The addition process is a three-step process:-



1.Reguating the values that need to be added to the cell state by involving a sigmoid function. This is similar to the forget gate and acts as a filter for all the information from h_t-1 and X_t.

2. Creating a vector containing possible values that can be added to the cell state, via "**tanh"** function, that gives values from -1 to +1.

3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created gate vector (the tanh function), then adding this useful information to the cell state via addition operation.

2.1.3 Output Gate

The job of selecting useful information from the current cell state and showing it out as an output is done via the output gate.

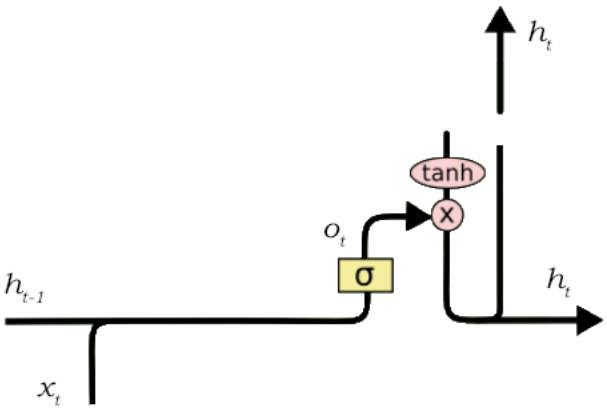


Figure 3:Output gate

The functioning is a three-part process: -

1. Creating a vector after applying tanh function to the cell state.
2. Making a filter using values of h_t-1, x_t. This filter again employs a sigmoid function.
3. Multiplying the value of this regulatory filter to the vector created in step 1, and sending this as output and also to the hidden state of the next cell.

Problem Formulation and Implementation

1. Dataset and Preprocessing

The dataset used for the project is the London smart meter dataset. It contains the energy consumption readings for a sample of 5,567 London Households that took part in the UK Power Networks led Low Carbon London Project between November 2011 and February 2014. The data is only on electrical consumption. It contains ACORN classification details of consumer classification by which the UK segments population.

1.1 Exploratory Data Analysis

- The data analysis and modelling is done using the Jupyter Notebook under the Anaconda IDE, in which mainly SciPy, NumPy, and pandas libraries are used. Different functions and modules of these libraries are imported. For the LSTM modelling, keras and its associated features are used.
- All blocks have been combined into a single dataframe. Day-level energy consumption data per household is used to normalize the inconsistency in household count.
- Possible relationships between energy consumption and weather is identified by correlation matrix.
- The UK holidays dataset is included to predict more accurately.
- Only total energy sum per day for a given household has been predicted into the future.
- ACORN details are avoided since household level is removed and instead, is normalized.

1) It is seen that the number of households sampled every day is inconsistent, as shown: -

```
In [60]: housecount = energy.groupby('day')[['LCLid']].nunique()
         housecount.head(4)
```

Out[60]:

day	LCLid
2011-11-23	13
2011-11-24	25
2011-11-25	32
2011-11-26	41

Thus the data is normalized using average energy per household.

Finally after combining relevant columns of all individual blocks into the “energy.csv” file, a summary is shown: -

```
In [64]: energy.describe()
```

Out[64]:

	energy_sum	LCLid	avg_energy
count	829.000000	829.000000	829.000000
mean	42870.715689	4234.539204	10.358458
std	20141.286953	1789.994799	1.886206
min	90.385000	13.000000	0.208997
25%	34421.895002	4084.000000	8.565752
50%	45846.575997	5138.000000	10.372293
75%	58795.512000	5369.000000	11.832222
max	82650.492003	5541.000000	15.940238

2) Weather information is taken from the darksky api in the dataset which includes daily as well as hourly data. In this case, daily dataset has been used.
   The missing data has been dropped, only the numerical variables have been selected.
   The relationships of different weather conditions with energy consumption has been explored by plotting them individually, and finally making a correlation matrix.
3) Energy consumption vs Different weather variables: -

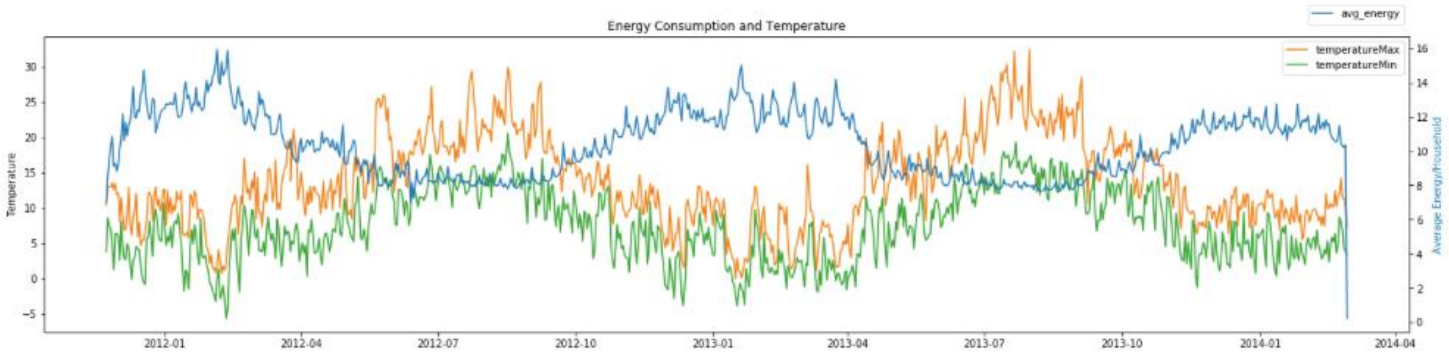


Figure 4Energy Consumption & Temperature

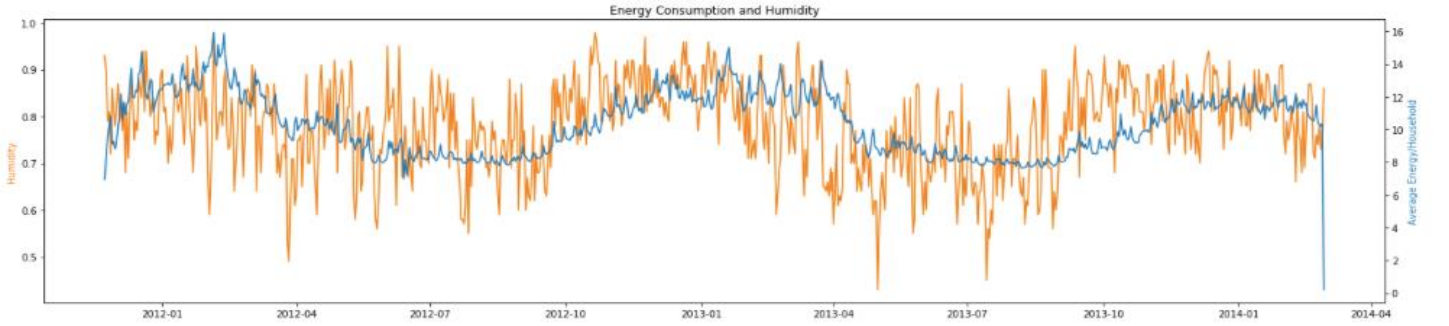


Figure 5 Energy Consumption & Humidity