Digital Innovations & Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 6 No. 1 March, 2018

Article Citation Format
V.I.E. Anireh & E.N. Osegi (2018) The Effect of Variable Size Order Hierarchical Temporal Memory on Short-Term Load Forecasting. Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 6, No. 1. Pp 39-44

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 11[rd] December, 2017
Review Type: Blind
Final Acceptance:: 17[th] February, 2018
DOI Prefix: 10.22624

The Effect of Variable Size Order Hierarchical Temporal Memory on Short-Term Load Forecasting

V.I.E Anireh
Department of Computer Science
Rivers State University
Port-Harcourt, Nigeria
anireh.ike@ust.edu.ng

***E.N. Osegi**
Department of Information Technology
National Open University of Nigeria
Lagos State, Nigeria
emmaosegi@gmail.com

ABSTRACT

Hierarchical Temporal Memory (HTM) have been shown to be a promising machine learning approach to time series related prediction tasks. HTM which is based on the Cortical Learning Algorithm (CLA) is a biological inspired neural network that seeks to replicate some of the core activities of the neocortex – the seat of intelligence, in the brain. However, its potential as a technique for Short-Term Load Forecasting still remains to be demonstrated. In this paper we present in part Short-Term Load Forecasting (STLF) using variable sized HTM and on the other part, demonstrate that the use of a high Variable Size Order (VSO)[1] HTM for daily load forecasts can lead to improved error rates but at the price of longer processing time.

Keywords - Cortical Learning Algorithm, Hierarchical Temporal Memory, Neural Network, Short-Term Load Forecasting

1. INTRODUCTION

Short-term load forecasting (STLF) deals with the aspect of data forecasts in the power industry that involves making reliable estimates of the probable load demands from a few minutes, hours, to a few days. In a typical real world scenario, this will usually involve the specification of the load monitoring schedules and accompanying power demand for reliable power demand management.

However, due to the highly non-linear nature of the load demand profile, it is very difficult to perform reliable forecasts on load demands using conventional techniques such as regression analysis or by human expert insight. Thus, STLF still remains a perennial problem faced by many power system researchers worldwide. Nonetheless, the application of neural networks to STLF is an active area of research with promising results obtained by different researchers. Some interesting related areas of research include the use of Back Propagation Neural Networks [1] for day-ahead electricity price forecasts; load forecasting using an ensemble of Extreme Learning Machines (ELM) with wavelet decomposition and the use of recurrent ELMs [2, 3]; load forecasts using simple neural networks with local learning [4] and minute-by-hourly building load forecasts using variants of the Long Short-Term Memory (LSTM) deep neural networks [5].
While all these approaches are promising, they still fall short of the requirements for real time online prediction systems. As stated in Cui et al [6] such systems should possess the following desirable characteristics:

- continuous learning
- high order (Markov) predictions
- multiple simultaneous predictions
- no hyper-parameter tuning
- noise robustness and fault tolerance

In this regard, the authors in [6] recommend Hierarchical Temporal Memory (HTM) for online sequence/time-series related problems. HTM (including variants of HTM) has been applied in different domains including but not limited to weld anomaly detection [7], automated retina analysis [8] and sign language recognition [9] with good performances reported in these researches. However, it is yet to be seen if HTM is capable of competitive short-term load forecasts.

In this paper, the primary objective is to present the results of experiments performed using the HTM for day ahead load forecast modelling of two specific real world datasets. In particular, it is to demonstrate that the use of Variable Order Size (VSO) HTM cortical learning system can impact significantly on the performance of STLF.

2. METHODOLOGY

2.1 Hierarchical Temporal Memory

Hierarchical Temporal Memory (HTM) originally conceptualized in [10] is a suite of spatial-temporal algorithms that implement some key ideas of the neocortex – the seat of intelligence in the brain. The basic learning principles of HTM include [11, 12], the use of a hierarchy of Columnar Regions, use of regions (the memory elements themselves), the use of a Sparse-Distributed Representation (SDR) and the notion of time. In addition to the learning principles, the HTM also proposes some core functions which include [12]: learning, inference, prediction and behaviour. The HTM is implemented as a suite of algorithms popularly referred to as the Cortical Learning Algorithms (CLA). While most of the learning principles are necessary for a proper implementation of CLAs, time plays the most important role. In particular, time allows for specific motor functionality not inherent in most conventional neural network approaches. Full details of the HTM theory which is also based on CLA including mathematical formalisms can be found in [6, 11 and 13].

2.2 Proposed System Model

The proposed systems architecture for daily load predictions is shown in Figure 1. It consists of five key units (sub-systems) including the Load Data Unit (LDU), Encoder Unit (EU), Spatial Pooler Unit (SPU), Temporal Pooler Unit (TPU) and the Temporal Classifier Unit (TCU). These are described in the following sub-sections.

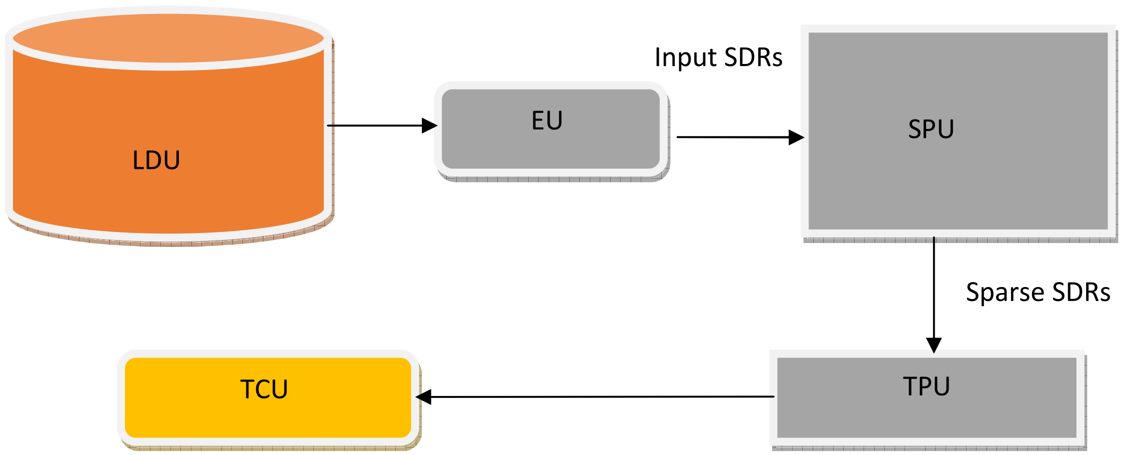

Fig 1. Proposed HTM Forecasting System Architecture

Load Data Unit

The Load Data Unit (LDU) serves as container for accessing hourly load dataset. In the present design, the LDU uses a text file for easy retrieval and data storage.

Encoder Unit

An Encoder Unit (EU) is used for transforming the load dataset into a sparse distributed representation (SDR). This is typically a scalar encoder; though other type of encoders are possible, a scalar encoder is more appropriate for the time series, due to lower data processing cost. The EU specifically encodes the inputs into binary data.

Spatial Pooler Unit

A Spatial Pooler Unit (SPU) generates a sequence of SDRs (based on the cellular generation of cortical synapses) in the memory space. For the HTM, this is based primarily on the overlap principle; at each time step a minimum duty cycle and an overlap score is typically assigned to a spatial-pooler training program in this unit.

Temporal Pooler Unit

The Temporal Pooler Unit (TPU) performs temporal processing on the SDR sequence generated in the SPU. In the HTM, this process is handled using the "union principle" described in Ahmad and Hawkins [13]. This principle allows a previous representation to be compared with a current one in order to determine the next set of predictions; if a match is found to be above a certain threshold, the current prediction becomes active and is predicted at the next time step; otherwise it remains inactive.

The predictive state in a HTM learning cortical network at time step t is typically computed as [6]:

$$\pi_{ij}^{t} = \begin{cases} 1 & if \ \exists_d \parallel \tilde{D}_{i\,j}^{d} \circ \mathrm{A}^{t} \parallel_1 > \theta \\ 0 & otherwise \end{cases} \tag{1}$$

where,

$\tilde{D}_{i\,j}^{d}$ = an M×N binary matrix representing the permanence of a connected synapse

d = a cortical segment

i, j = cell and column states

A^{t} = an M×N binary matrix denoting the activation state of the cortical network

θ = the cortical segment activation threshold

N = number of cortical columns

M = number of neurons (cortical cells) per column

The Temporal Classifier Unit

The Temporal Classifier Unit (TCU), performs error estimates in accordance with known practices. In the present case, the Mean Absolute Percentage Error (MAPE) was used as the evaluation metric. The temporal classifier for evaluating the predictive classification performance of the entire system based on MAPE is expressed as:

$$MAPE = \left(\frac{\sum_{i}^{n} (\mid y_{(i)} - \hat{y}_{(i)} \mid}{n_z} \right) * 100, \qquad i = 1,2,3\ldots,n \quad n \in \mathrm{Z} \tag{2}$$

where,

MAPE = mean absolute percentage error

y = the observed load data

$\hat{y}$ = the model's predictions of y

n_z = size of the observation matrix i.e. rows multiplied by columns, and

n = number of the observations.

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

The HTM algorithm was tested on a daily load time series dataset available in [14]. This dataset can also be obtained from http://www.gdudek.el.pcz.pl/varia/stlf-data.The dataset is divided into two parts: STL1 and STL2 for short-term load forecast datasets 1 and 2 respectively. The MATLAB programming system was used for developing the HTM program codes and for the simulation experiments.

In order to make the dataset compatible with the HTM program, it was ported to a text file. This has the advantage of fast data retrieval in the programming system. The key HTM parameters used for the experiments are given in Table 1. A total of 5 trials (program runs) for each VSO configuration were performed and the corresponding MAPE values recorded. The run-times taken for each trial were also recorded. The VSO configurations are of order: 10x50, 50x50 and 100x50 for VSO1, VSO2 and VSO 3 respectively.

Table 2 shows the performance of the HTM using the different VSO configurations including average MAPE values and running times for STL1 and STL2 datasets respectively. As shown in Table 2, the HTM performance is improved for higher VSO for STL1 dataset. However, we can see that the increase in VSO is accompanied with a corresponding increase in program run-times. Also notice that the STL2 dataset gave a much lower MAPE estimate at VSO1 which was unexpected. Progressing from VSO2 to VSO3, the MAPE estimates of VSO3 was expectedly lower than that of VSO2 though the margin was minimal (about 0.1 unit). This behaviour may be attributed to the effect of seasonality variations in the datasets under study. However, the run times followed a similar pattern as in the first dataset. In general, STL2 gave a much lower MAPE estimate than STL1.

Table 1
HTM parameters

Parameter	Desired local activity	Minimum overlap	Initial permanence value	Number of Monte-Carlo Runs
Values	3	59	0.21	10

Table 2

Estimated MAPE values and run-times at different VSO. Best values are in bold.

	STL1			STL2		
	VSO1	VSO2	VSO3	VS01	VSO2	VS03
MAPE (%)	8.09	7.69	**7.08**	**5.76**	5.99	5.98
RUN-TIME (s)	12.60	21.80	56.2	11.80	12.00	23.4

4. CONCLUSION

This research has investigated the potentials of HTM based on Cortical Learning Algorithms (CLA) for STLF tasks. A learning systems model based on a HTM network of Variable Size Order (VSO) was investigated. From the experimental results, it was found that increasing the HTM network size leads to improved MAPE estimates. However, this comes at the price of higher computational effort by the HTM network as evidenced in the run-time report. Thus, a compromise has to be made between the desired MAPE values and the computational run-time of the HTM network.

Acknowledgement
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. Source codes for the experiments are available at the MATLAB central website: www.matlabcentral.com.

REFERENCES

1. Reddy, S. S., Jung, C. M., & Seog, K. J. (2016). Day-ahead electricity price forecasting using back propagation neural networks and weighted least square technique. *Frontiers in Energy*, *10*(1), 105-113.
2. Li, S., Goel, L., & Wang, P. (2016). An ensemble approach for short-term load forecasting by extreme learning machine. *Applied Energy*, *170*, 22-29.
3. Ertugrul, Ö. F. (2016). Forecasting electricity load by a novel recurrent extreme learning machines approach. *International Journal of Electrical Power & Energy Systems*, *78*, 429-435.
4. Dudek, G. (2013, June). Forecasting time series with multiple seasonal cycles using neural networks with local learning. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 52-63). Springer Berlin Heidelberg.
5. Marino, D. L., Amarasinghe, K., & Manic, M. (2016, October). Building energy load forecasting using deep neural networks. In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE* (pp. 7046-7051). IEEE.
6. Cui, Y., Surpur, C., Ahmad, S., & Hawkins, J. (2015). Continuous online sequence learning with an unsupervised neural network model. *arXiv preprint arXiv:1512.05463*.
7. Rodriguez-Cobo, L., Ruiz-Lombera, R., Conde, O. M., López-Higuera, J. M., Cobo, A., & Mirapeix, J. (2013). Feasibility study of Hierarchical Temporal Memories applied to welding diagnostics. *Sensors and Actuators A: Physical*, *204*, 58-66.
8. Boone, A. R., Karnowski, T. P., Chaum, E., Giancardo, L., Li, Y., & Tobin, K. W. (2010, May). Image processing and hierarchical temporal memories for automated retina analysis. In *Biomedical Sciences and Engineering Conference (BSEC), 2010*. IEEE.
9. Rozado, D., Rodriguez, F. B., & Varona, P. (2010, September). Optimizing hierarchical temporal memory for multivariable time series. In *International Conference on Artificial Neural Networks* (pp. 506-518). Springer Berlin Heidelberg.
10. Hawkins, J., & Blakeslee, S. (2007). *On intelligence*. Macmillan.
11. Hawkins, J., Ahmad, S., & Dubinsky, D. (2010). Hierarchical temporal memory including HTM cortical learning algorithms. Technical report, Numenta, Inc, Palto Alto http://www.numenta.com/htmoverview/education/HTM_CorticalLearningAlgoritithms.pdf
12. Osegi, N. E. and P.Enyindah (2016). An Improved Intelligent Agent for Mining Real-Time Databases Using Modified Cortical Learning Algorithms. *Advances in Multidisciplinary & Scientific Research Journal, 2*(3), 47-58.
13. Ahmad, S., & Hawkins, J. (2015). Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory. arXiv preprint. *arXiv*, *1503*.
14. [dataset] Dudek, G. (2015). Short-term load forecasting using random forests. In *Intelligent Systems' 2014* (pp. 821-828). Springer International Publishing.