Deep Learning-Based Approach for Time Series Forecasting with Application to Electricity Load

J.F. Torres, A.M. Fernández, A. Troncoso, and F. Martínez-Álvarez(✉)

Division of Computer Science, Universidad Pablo de Olavide, 41013 Seville, Spain
{jftormal,amfergom}@alu.upo.es, {ali,fmaralv}@upo.es

Abstract. This paper presents a novel method to predict times series using deep learning. In particular, the method can be used for arbitrary time horizons, dividing each predicted sample into a single problem. This fact allows easy parallelization and adaptation to the big data context. Deep learning implementation in H2O library is used for each subproblem. However, H2O does not permit multi-step regression, therefore the solution proposed consists in splitting into h forecasting subproblems, being h the number of samples to be predicted, and, each of one has been separately studied, getting the best prediction model for each subproblem. Additionally, Apache Spark is used to load in memory large datasets and speed up the execution time. This methodology has been tested on a real-world dataset composed of electricity consumption in Spain, with a ten minute frequency sampling rate, from 2007 to 2016. Reported results exhibit errors less than 2%.

Keywords: Deep learning · Time series · Forecasting · Apache spark

1 Introduction

Time series forecasting is a task of utmost relevance that can be found in almost any scientific discipline. Electricity is not an exception, and much work is devoted to predict both demand and prices [10]. Achieving accurate demand forecasts is critical since it can be used in production planning, inventory management, or even in evaluating capacity needs. In other words, it may lead to insufficient or excessive energy production, thus reducing profits.

A novel approach based on deep learning [5,12] is proposed in this article to forecast time series, with application to electricity demand. Deep learning is an emerging branch of machine learning that extends artificial neural networks. One of the main drawbacks that classical artificial neural networks exhibit is that, with many layers, its training typically becomes too complex [9]. In this sense, deep learning consists of a set of learning algorithms to train artificial neural networks with a large number of hidden layers. Deep learning models are also sensitive to initialization and much attention must be paid at this stage [13].

The main idea underlying the method is dividing the number of samples to be simultaneously predicted (horizon of prediction) into different subproblems.

J.M. Ferrández Vicente et al. (Eds.): IWINAC 2017, Part II, LNCS 10338, pp. 203–212, 2017.
DOI: 10.1007/978-3-319-59773-7_21

Every subproblem is independently solved making use of different pieces of the historical data. The implementation of the deep learning method used is that of the well-known H2O library, which is open source and designed for a distributed environment [2].

It is worth noting that this strategy is particularly suitable for parallel implementations and it is ready to be used for big data environments. Furthermore, in order to speed up the whole process, Apache Spark is used to load the data in memory.

The performance of the approach has been assessed in real-world datasets. Electricity consumption in Spain has been used as case study, by analyzing data from 2007 to 2016 in the usual 70–30% training-test sets structure.

The rest of the paper is structured as follows. Relevant related works are discussed in Sect. 2. The methodology proposed in this paper is introduced and described in Sect. 3. The results of applying the approach to Spanish electricity data are reported and discussed in Sect. 4. Finally, the conclusions drawn are summarized in Sect. 5.

2 Related Works

This section reviews relevant works in the context of time series forecasting and deep learning.

Some studies are currently applying deep learning to prediction problems. Ding et al. [4] proposed a method for event driven stock market prediction. They used a deep convolutional neural network, at a second stage, to model both short-term and long-term stock price fluctuations. Results were assessed on S&P 500 stock historical data.

A novel deep learning architecture for air quality prediction was first introduce in [8]. The authors evaluated spatio-temporal correlations by first applying a stacked autoencoder model for feature extraction. Comparisons to other models confirmed that the method achieved promising results.

A meaningful attempt to apply a data-driven approach to forecasting transportation demand can be found in [1]. In particular, a deep learning model to forecast bus ridership at the stop and stop-to-stop levels was there adopted. As main novelty, the authors claim that, for the first time, the method is only based on feature data.

Deep learning based studies can be found for classification as well. Image processing has been shown to be one of the most fruitful fields of deep learning application. A successful approach for image classification with deep convolutional neural networks was introduced in [7]. They classified 1.2 million high-resolution images achieving top errors in the ImageNet LSVRC-2010 contest.

The authors in [3] proposed a deep learning-based classifier for hyperspectral data. The hybrid method (it is also combined with principal component analysis and logistic regression) was applied to extract deep features for such kind of data, achieving competitive results.

Tabar and Halici [14] introduced an approach based on deep learning for classification of electroencephalography (EEG) motor imagery signals. In particular, the method combined convolutional neural networks and stacked autoencoders and showed to be competitive when compared to other existing techniques.

Finally, some works relating to electricity demand forecasting are also discussed. Talavera et al. [15] proposed a forecasting algorithm to deal with Spanish electricity data. The algorithm was developed under the Apache Spark which is an engine for large-scale data processing framework [16], and was applied to big data time series. Satisfactory results were reported.

Electricity demand profiles were discovered as initial step for forecasting purposes in [11]. Spanish data were also analyzed and, as happened in the afore discussed study, the method was designed to be able to evaluate big time series data. Relevant patterns were discovered, distinguishing between different seasons and days of the week.

Grolinger et at. [6] explored sensor-based forecasting in event venues, a scenario with typically large variations in consumption. They authors paid particular attention to the relevance of the size of the data and on the temporal granularity impact. Neural networks and support vector regression were applied to 15-minute frequency data for Ontario, Canada.

As it can been seen after the analysis of updated state-of-the-art, deep learning is being currently applied into a variety of problems. However, to the authors' knowledge, no method has been developed to forecast electricity-related time series and has been conceived for big data time series forecasting. Therefore, the conduction of this research is justified.

3 Methodology

This section describes the methodology proposed in order to forecast time series. Apache Spark has been used to load data in memory and a deep learning implementation in R language, within the H2O package, has been applied to forecast time series.

The objective of this study consists in predicting h next values for a time series, expressed as $[x_1,\ldots,x_t]$, being h the horizon of prediction, depending on a historical window composed of w values. This can be formulated as:

$$[x_{t+1}, x_{t+2}, \ldots, x_{t+h}] = f(x_t, x_{t-1}, \ldots, x_{t-w-1}) \tag{1}$$

where f is the model to be found in the training phase by the deep learning algorithm. However, the package chosen does not support the multivariate regression, therefore, multi-step forecasting is not supported either.

The solution for this is splitting the problem into h forecast subproblems, which can formulated as:

$$x_{t+1} = f_1(x_t, x_{t-1}, \ldots, x_{t-w-1}) \tag{2}$$

$$x_{t+2} = f_2(x_t, x_{t-1}, \ldots, x_{t-w-1}) \tag{3}$$

$$\ldots \tag{4}$$

$$x_{t+h} = f_h(x_t, x_{t-1}, \ldots, x_{t-w-1}) \tag{5}$$

That is, given w samples used as input for the deep learning algorithm, h values are simultaneously forecasted. Based on this formulation, each estimation is made separately, thus avoiding the consideration of previously predicted samples and, consequently, removing the error propagation. In other words, if the prediction of previous values would be used to predict the next value, the error would be higher because the error would be accumulated in each iteration of the prediction horizon. Also, to create a model for each h value could involve a higher computational cost than building just a model to predict all values.

The last step consists in obtaining the best model for each subproblem by applying deep learning and varying the number of hidden layers and neurons per layer. Once the training for each subproblem is complete, the test set is predicted.

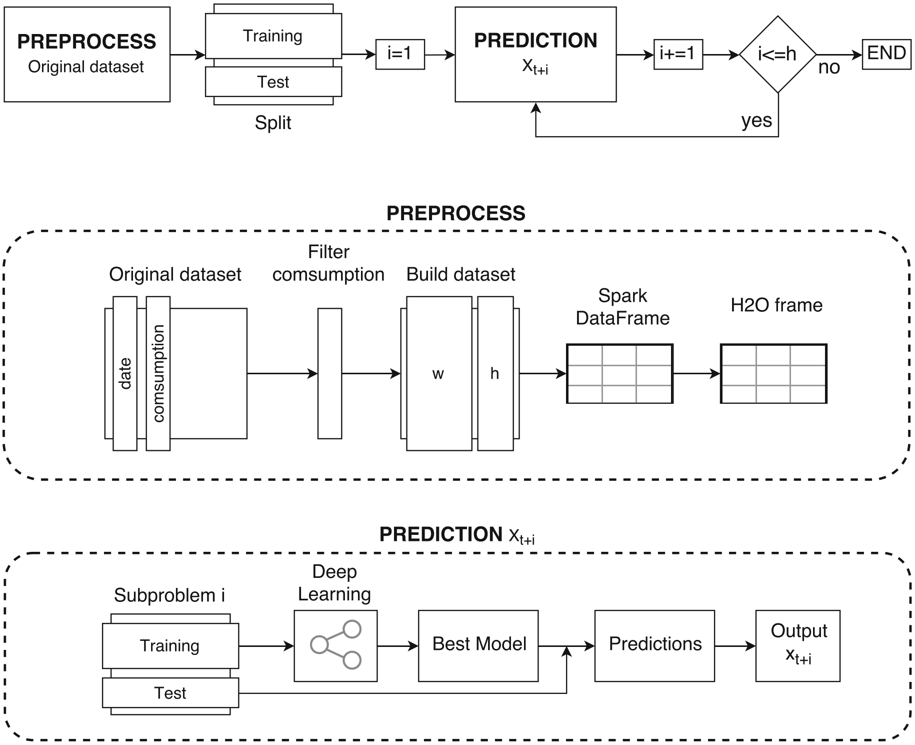

Fig. 1. Illustration of the proposed methodology.

Figure 1 shows the full study's flow, starting with input dataset and ending with aggregated output. It can be seen that, in its current implementation, an iterative strategy has been followed since each subproblem is solved after the previous one is done. However, it is easy to figure out that this strategy can be easily parallelized and adapted to a big data environment.

It is important to highlight that H2O frame can be created without Spark dataframe conversion, but this step allocates data in memory and makes the access more quickly. Also it is important to note that deep learning algorithm on H2O library has a lot of parameters to adjust the execution. In this study, some of this parameters have been used. They will be thoroughly discussed in Sect. 4.2

4 Results

As previously mentioned, a study to forecast a time series of electricity consumption has been conducted. This section presents the results obtained. First, Sect. 4.1 describes the dataset used for the study. Second, Sect. 4.2 provides the experimental setup carried out and, finally, Sect. 4.3 discusses results obtained.

4.1 Dataset Description

The dataset considered in this study provides electricity consumptions readings in Spain from January 2007 to June 2016 with a measure every 10 min, i.e., the time series is composed of 497832 measurements.

In study, the dataset was only filtered by consumption and redistributed in a matrix depending of the window size and prediction horizon. The values of these parameters were set to 168 and 24, respectively. After this preprocessing, the final dataset has 20736 rows and 192 columns into a 23.9 MB file which was recorded for further studies.

To perform the entire experimentation, the dataset has been split into 14515 instances for training (70%) and 6221 for test (30%).

4.2 Design of Experiments

In order to assess the performance of the algorithm, the well-known mean relative error (MRE) measure has been selected. For a matrix of data, the formula is:

$$MRE = \frac{1}{r * c} \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{|v_{pred} - v_{actual}|}{v_{actual}} \quad (6)$$

where r and c represents the number of rows and columns on the test set, v_{pred} stands for the predicted values and v_{actual} for the actual values.

As discussed in previous sections, it is necessary to define and initialize several variables. Variable values have been set to:

1. The size of the window (w) represents the length of the historical data considered to predict the target subsequence. It has been set to 168, which represents 7 blocks of 4 h (1 day and 4 h, in total). This parameter was set during the training phase with values 24, 48, 72, 96, 120, 144 and 168, and was found to be the one with minimum error.
2. As for the prediction horizon (h), it was set to $h = 24$ (4 h). Considering a higher h would turn the problem into a long-term forecasting one, and some others consideration should then be taken into consideration.
3. To apply deep learning, it is necessary to set the number of hidden layers and number of neurons. The number of hidden layer was set to 3 and the number of neurons for each one was set to an interval ranging from 10 to 100 with a step of 10, using a validation set composed of the 30% of the training set. Then, only the best value was chosen for the analysis.
4. λ was set to 0.001. This parameter is used for regularization of the dataset.
5. Also, two different parameters were set to describe the adaptive rate. These were ρ and ϵ, which were set to 0.99 and $1.0E - 9$, which are default values for those parameters, respectively.
6. The activation function chosen was the hyperbolic tangent function.
7. As for the distribution function, Poisson distribution was the one chosen.

These parameters were chosen based on several tests varying values. Some relevant results are shown in Table 1, in which it can be seen MRE values obtained for some parameters. For instance, Poisson distribution offers better results than other options.

Table 1. Errors varying deep learning parameters.

Lambda	Rho	Epsilon	Activation	Distribution	MRE (%)
1	0.9	1.0E-9	Tanh	Poisson	2.56
1	0.99	1.0E-9	Tanh	Poisson	2.43
1	0.999	1.0E-9	Tanh	Poisson	2.49
1	0.9	1.0E-9	Tanh	Gaussian	15.61
1	0.99	1.0E-9	Tanh	Gaussian	15.61
1	0.999	1.0E-9	Tanh	Gaussian	15.57
0.001	**0.99**	**1.0E-9**	**Tanh**	**Poisson**	**1.84**
1	0.99	1.0E-9	Tanh	Tweedie	4.21
10	0.99	1.0E-9	Tanh	Poisson	2.69
1	0.99	1.0E-9	Tanh	Huber	15.63
1	0.99	1.0E-9	Tanh	Laplace	15.63

The algorithm has been executed using the dataset described in Sect. 4.1. The computer used to complete this execution has been an Intel Core i7-5820K at 3.30 GHz, 15 MB cache, 12 cores and 16 GB of RAM memory working, under Ubuntu 16.04.

Finally, the dataset was loaded from Apache Spark to allocate it in memory instead of in disk, thus accessing to the data more efficiently and quickly.

4.3 Electricity Consumption Time Series Forecasting

This section describes the results obtained after applying the algorithm proposed to the dataset, which were described in Sect. 4.1 over the machine described in Sect. 4.2. This test provides a total of 20736 instances and 192 attributes, resulting in 149305 forecast values.

As forecasting are divided in h subproblems (in this case, h is 24), it is possible to use different neuron values in each subproblem to obtain smaller errors. In this study, it was decided to set the possible neurons combinations to 3 hidden layers, each one with a interval of neurons (10 to 100 with a step of 10), as discussed in the previous section. Table 2 shows the neuron configurations that are optimum for each subproblem:

Table 2. Optimum neurons configuration for each subproblem.

Subproblem	Hidden layers	Neurons	Error	Subproblem	Hidden layers	Neurons	Error
1	3	30	0.77	13	3	40	1.83
2	3	80	1.13	14	3	80	1.81
3	3	90	1.15	15	3	90	2.11
4	3	60	1.18	16	3	40	1.93
5	3	60	1.35	17	3	70	2.50
6	3	100	1.36	18	3	70	2.09
7	3	40	1.50	19	3	70	2.17
8	3	80	1.71	20	3	60	2.14
9	3	30	1.88	21	3	90	2.43
10	3	80	1.76	22	3	70	2.56
11	3	50	1.66	23	3	100	2.42
12	3	100	2.07	24	3	100	2.77

Table 2 summarizes the errors for each subproblem depending of the optimum number of neurons per layer. This error tends to increase as the subproblem increases because there exists a gap between the first sample in the historical data and the sample to be predicted, that is, there immediately after values of the target sample are missing and omitted during the forecasting process.

Using this configuration of neurons and the other deep learning parameter values mentioned in Sect. 4.2 the final value of MRE to predict the full data test has been 1.84%.

Figures 2 and 3 are depicted for illustrative purposes. They represent the best and the worst comparison between actual and predicted consumption on a

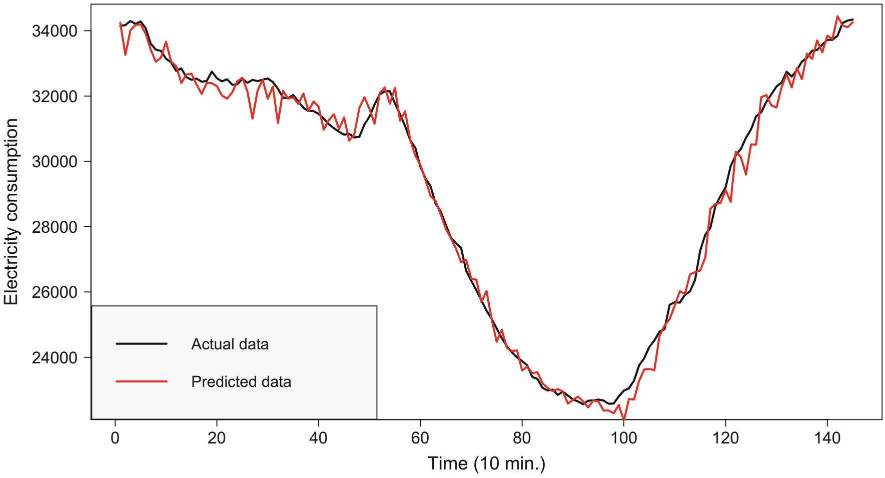

Fig. 2. The best forecast achieved for a full day.

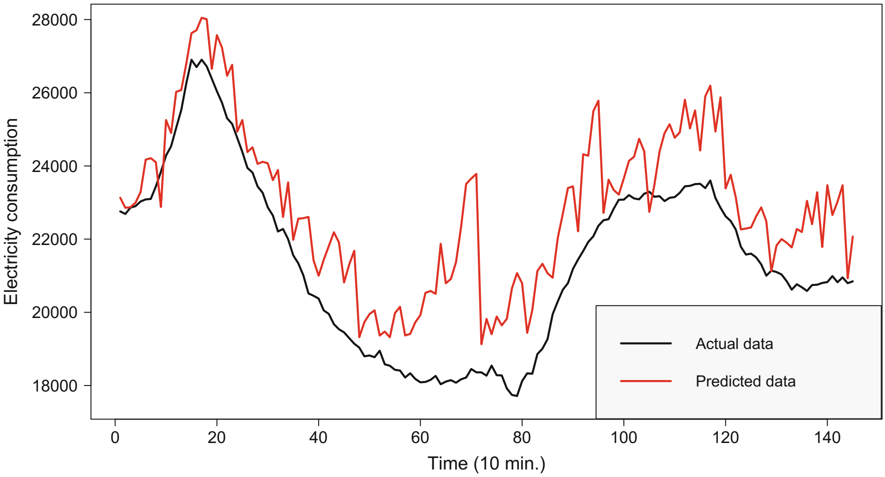

Fig. 3. The worst forecast achieved for a full day.

full day (144 measures) in the test set, respectively. It must be noted that some ripple in predicted data that is present not only in days depicted in the figures, but in almost the entire test set. This fact is justified because every sample is independently estimated. A feasible and successful post-processing could consist in the automatic application of any filter. In short, such a shape for the output must be further studied in future works.

5 Conclusions

This work describes a new approach to use deep learning methods as regressors and forecast the electricity consumption for the next twenty four values. It uses Apache Spark framework to load data in memory and the H2O library to apply the algorithm developed in R language. On this preliminary study, the results obtained can be considered satisfactory since errors are smaller than 2%. However, future works will be directed towards the improvement of the selection of the best parameters to forecast time series and to scale it to be applied to big data using a cluster of machines. Also, some post-processing seems to be necessary to reduce the ripple in forecasted values.

Acknowledgments. The authors would like to thank the Spanish Ministry of Economy and competitiveness and Junta de Andalucía for the support under projects TIN2014-55894-C2-R and P12-TIC-1728, respectively.

References

1. Baek, J., Sohn, K.: Deep-learning architectures to forecast bus ridership at the stop and stop-to-stop levels for dense and crowded bus networks. Appl. Artif. Intell. **30**(9), 861–885 (2016)
2. Candel, A., LeDell, E., Parmar, V., Arora, A.: Deep Learning with H2O. H2O.ai Inc., California (2017)
3. Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y.: Deep learning-based classification of hyperspectral data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **8**(6), 2094–2107 (2014)
4. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 2327–2334 (2015)
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
6. Grolinger, K., L'Heureux, A., Capretz, M.A.M., Seewald, L.: Energy forecasting for event venues: big data and prediction accuracy. Energy Buildings **112**, 222–233 (2016)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
8. Li, X., Peng, L., Hu, Y., Shao, J., Chi, T.: Deep learning architecture for air quality predictions. Environ. Sci. Pollut. Res. Int. **23**, 22408–22417 (2016)
9. Livingstone, D.J., Manallack, D.T., Tetko, I.V.: Data modelling with neural networks: advantages and limitations. J. Comput.-Aided Mol. Des. **11**, 135–142 (1997)
10. Martínez-Álvarez, F., Troncoso, A., Asencio-Cortés, G., Riquelme, J.C.: A survey on data mining techniques applied to energy time series forecasting. Energies **8**, 1–32 (2015)
11. Pérez-Chacón, R., Talavera-Llames, R.L., Troncoso, A., Martínez-Álvarez, F.: Finding electric energy consumption patterns in big time series data. In: Omatu, S., et al. (eds.) Proceedings of the International Conference on Distributed Computing and Artificial Intelligence. AISC, vol. 474, pp. 231–238. Springer, Cham (2016)

12. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
13. Sutskever, I., Martens, J., Dahl, G.E., Hinton, G.E.: On the importance of initialization and momentum in deep learning. In: Proceedings of the International Conference on Machine Learning, pp. 1139–1147 (2013)
14. Tabar, Y.R., Halici, U.: Deep learning-based classification of hyperspectral data. J. Neural Eng. **14**(1), 016003 (2016)
15. Talavera-Llames, R.L., Pérez-Chacón, R., Martínez-Ballesteros, M., Troncoso, A., Martínez-Álvarez, F.: A nearest neighbours-based algorithm for big time series data forecasting. In: Martínez-Álvarez, F., Troncoso, A., Quintián, H., Corchado, E. (eds.) HAIS 2016. LNCS, vol. 9648, pp. 174–185. Springer, Cham (2016). doi:10.1007/978-3-319-32034-2_15
16. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: Proceedings of the International Conference on Hot Topics in Cloud Computing, pp. 1–10 (2010)