ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Using Deep Learning and Evolutionary Algorithms for Time Series Forecasting

Rafael Thomazi Gonzalez and Dante Augusto Couto Barone

Institute of Informatics - Federal University of Rio Grande do Sul
Porto Alegre, RS - Brazil

Abstract. Deep Learning is one of the latest approaches in the field of artificial neural networks. Since they were first proposed, Deep Learning models have obtained state-of-art results in some problems related to classification and pattern recognition. However, such models have been little used in time series forecasting. This work aims to investigate the use of some of these architectures in this kind of problem. Another contribution is the use of one Evolutionary Algorithm to optimize the hyperparameters of these models. The advantage of the proposed method is shown on two artificial time series datasets and one electricity load demand dataset.

1 Introduction

A time series is an ordered sequence of data points, usually measured in uniform time intervals. An important property of time series is that data observations are interdependent and, thus, it is essential to maintain the order in which the data were generated. Time series forecasting attempts to understand the underlying context of the data points through the use of a model to forecast future values based on known past values [1]. Analysis of time series data has been the subject of active research for decades. Nowadays many researchers questioned the efficiency of statistical techniques in comparison with machine learning methods. Artificial Neural Networks (ANNs) have proved to be particularly good at capturing complex non-linear characteristics of time series, being widely studied at literature [2, 3]. However, more complex, high-dimensional, and noisy real-world time-series data make using shallow networks almost infeasible since the dynamics are either too complex or unknown. Deep Learning (DL) refers broadly to models that derive meaning out of data by using a hierarchy of multiple layers that progressively extracts higher-level features that could be more relevant for the final forecasting task [4]. DL models, that were originally used in classification and pattern recognition problems, started to be applied in various time series forecasting tasks [5, 6]. However, there are still no conclusive results about the role of pre-training on this model and the relationship between architecture and the dimensionality of the data.

A common trait in Machine Learning models is that they are parameterized by a set of hyperparameters, which can have wildly varying effects on the complexity and performance of the resulting model. The best set of hyperparameters is commonly defined manually by an “educated guess” or by applying a grid or random search over predefined search space [7]. Since the hyperparameter space is in general large and evaluating the objective function (i.e. the performance of the model with a given set of hyperparameters) is computationally expensive, the need of designing smarter methods for determining the hyperparameters is especially important for increasingly complex

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

architectures. Evolutionary Algorithms have been shown very efficient in solving this challenging optimization problem [8, 9].

In the context of time series forecasting, this work aims to provide a comparison of DL algorithms when applied on Time Series forecasting problems. Moreover, we tackle the problem of the automated hyperparameter selection using the Covariance Matrix Adaptation Evolution Strategy (CMAES) [10] algorithm. A comparative study of the performance of the proposed approach is conducted using synthetic and real data. The broader goal of this work is to further generate insights on how DL models perform on time series forecasting problems and how suited CMAES is for optimizing models with large number of hyperparameter.

The rest of the article is organized as follows. Next section presents basic concepts and principles related to Deep Learning algorithm and CMAES. Section 3 introduces the proposed methodology. The experimental results are presented in Section 4. Finally, the conclusions are drawn in Section 5.

2 Background

2.1 Deep Learning Algorithms

2.1.1 Stacked Autoencoder

Autoencoder (AE) [11] is an algorithm that learns an approximation to the identity function, so that it is trained to minimize the discrepancy between the input data and its reconstruction. However, by placing constraints on the network it can reveal interesting structure about the data. An example is changing the training reconstruction criterion for learning to clean partially corrupted input (i.e. with noise), or in short denoising. This leads to a variant of the basic AE called Denoising Autoencoder (DAE) [12].

Since Autoencoders are automatic features extractors, they can be stacked to create a deep structure to increase the level of abstraction of learned features. Thus, a Stacked Autoencoder (SAE) [13] is a neural network consisting of multiple layers of AEs. In this case, the network is pre-trained in a greedy layer-wise fashion, i.e. each layer is treated as a shallow AE, generating latent representations of the input data. These features are then used as input for the subsequent layers before the full network is fine-tuned using standard supervised learning algorithm.

2.1.2 Long Short-Term Memory Network (LSTM)

Multilayer Perceptron (MLP) architecture assumes that all inputs and outputs are independent of each other. Therefore, in order for a MLP to model a time series, it is necessary to include some temporal information in the input data. Recurrent Neural Networks (RNNs) are neural networks specifically designed to solve this problem, making use of a recurrent connection in every unit that feed the activation of a neuron back to itself [14].

In some cases, RNNs are hard to train properly because the gradient of some of the weights starts to become too small or too large if the network is unfolded for too many time steps. These issues are called the vanishing or exploding gradients problems. Long Short-Term Memory (LSTM) [15] networks aim to solve these problems by using memory cells, instead of recurrent units, to store and output information. The behavior

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

of each memory cell is defined by components called gates. Depending on the states of these gates, LSTM can represent long-term or short-term dependency of sequential data.

2.2 Covariance Matrix Adaptation Evolution Strategy (CMAES)

The Covariance Matrix Adaptation Evolution Strategy (CMAES) [10] is an Evolutionary Algorithm for difficult non-linear non-convex derivative-free optimization problems in continuous domain. CMAES is considered as state-of-the-art and one of the most powerful EAs for real-valued optimization.

In CMAES, a covariance matrix describing correlations between decision variables is learned and adapted during the search to maximize the likelihood of generating successful solutions. The state of CMAES is given by the parameters $m \in \mathbb{R}^d$, σ>0 and $C \in \mathbb{R}^{dxd}$ of its multivariate normal search distribution $N(m, \sigma^2 C)$. CMAES is an iterative algorithm, that, in each of its iterations, samples λ candidate solutions from a multivariate normal distribution, evaluates these and then adjusts the sampling distribution used for the next iteration. Here we apply what can be considered a baseline version, featuring non-elitist (μ, λ) selection and cumulative step-size adaptation (CSA). All tuning constants are set to their default values [16].

3 Methodology

Data are divided into three sets: training set, validation set and testing set. The forecasting models are generated using the training data, but the error is calculated for the validation set. Training is performed up to the moment when the error for validation set starts to increase; this is a regularization technique called early stopping. In order to evaluate any overfitting and generalization issues, the performance on unseen data (i.e. test data) is also considered during model comparison.

The DL models were evaluated using two synthetic series and one real data series. The synthetic series were generated using the Mackey-Glass [17] and Lorenz Attractor System [18] functions. The Mackey-Glass dataset is comprised of 5000 data points, and the Lorenz System dataset contains 8000 data points from the x component of the equation. For these two datasets, the testing set represents the last 20% points. The real dataset consists of hourly electricity load data obtained from the 2017 Global Energy Forecasting Competition (GEFCom 2017) [19]. The training set ranges from 04 January 2015 to 30 November 2016, and the testing set ranges from 01 December 2016 to 31 December 2016. The data was normalized calculating the hourly log return.

Across all experiments a fixed parametrization for the internals parameters of CMAES was used. The population size (λ) is set to 24. The termination criteria are defined by a maximum number of generations ($g_{max} = 150$) and by a minimum fitness function target that depends on the dataset. MSE over the validation set is used as fitness function.

A 5-fold rolling window cross-validation (CV) [20] is performed using the training and validation set combined in order to access the performance of each optimal model found by CMAES. The forecast accuracy is computed by averaging over the 5 test sets. Finally, it is performed a one-step-ahead prediction for the testing set in order to estimate the performance in unseen data. Four evaluation measures are used in this

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

study: RMSE, MSE, MAE, and the Coefficient of Determination (R^2). The first three metrics are estimates of variance of residuals, or non-fit, in the population; while R^2 is a measure of how well the regression line represents the data [21].

4 Experimental Results

In the first case of study, the proposed approach is evaluated using the Mackey-Glass dataset. Table 1 shows the mean and standard deviation of the training performance obtained from the CV process. By analyzing this table, is possible to state that LSTM obtained the best performance taking into consideration all the performance metrics and also proved to be the most stable model (i.e. smallest CV standard deviations). LSTM also outperformed all other models in unseen data, as shown in Table 2 that presents the results for the testing set.

	MLP	SAE	SDAE	LSTM
R^2	0.9980 ± 8.8e-4	0.9972 ± 1.2e-3	0.9979 ±4.9e-4	0.9987 ± 4.8e-4
MSE	9.3e-5 ± 3.8e-5	1.2e-4 ± 5.7e-5	9.8e-5 ± 2.3e-5	5.9e-5 ± 2.2e-5
RMSE	9.4e-3 ± 1.8e-3	1.1e-2 ± 2.5e-3	9.8e-3 ± 1.7e-3	7.5e-3 ± 1.4e-3
MAE	7.8e-3 ± 1.2e-3	9.2e-3 ± 2.2e-3	8.1e-3 ± 9.7e-4	6.2e-3 ± 1.1e-3

Table 1: CV results for Mackey-Glass training dataset.

	MLP	SAE	SDAE	LSTM
R^2	0.9988	0.9983	0.9988	0.9995
MSE	5.4e-5	7.9e-5	5.6e-5	2.2e-5
RMSE	7.3e-3	8.9e-3	7.4e-3	4.7e-3
MAE	5.9e-3	7.2e-3	6.1e-3	3.9e-3

Table 2: Forecasting performances for Mackey-Glass test set.

In the second numerical example, the models are tested using the Lorenz System dataset. The standard MLP outperformed all the other models and presented the smallest standard deviations values as shown in Table 3. Moreover, as can be seen in Table 4, MLP also presented the best overall forecasting performance over the test set. SAE achieved a very similar performance. Furthermore, differently from the training results, LSTM presented the worst performance in the test set, which may indicate overfitting.

	MLP	SAE	SDAE	LSTM
R^2	0.9996 ± 1.5e-4	0.9934 ± 6.8e-3	0.9985 ± 1.0e-3	0.9994 ± 3.0e-4
MSE	1.5e-5 ± 7.4e-6	2.8e-4 ± 3.0e-4	5.6e-5 ± 4.1e-5	2.4e-5 ± 1.4e-5
RMSE	3.8e-3 ± 1.0e-3	1.3e-2 ± 9.5e-3	7.0e-3 ± 2.5e-3	4.7e-3 ± 1.3e-3
MAE	2.8e-3 ± 8.6e-4	1.2e-2 ± 9.4e-3	5.4e-3 ± 2.6e-3	3.3e-3 ± 9.2e-4

Table 3: CV results for Lorenz System training dataset.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

	MLP	SAE	SDAE	LSTM
R^2	0.9997	0.9997	0.9992	0.9988
MSE	9.1e-6	1.0e-5	3.2e-5	4.9e-5
RMSE	3.0e-3	3.2e-3	5.7e-3	7.0e-3
MAE	2.3e-3	2.2e-3	4.9e-3	5.7e-3

Table 4: Forecasting performances for Lorenz System test set.

The third use case involves the hourly electricity load data obtained from GEFCOM 2017. As can be seen in Table 5, LSTM achieved the best average performance across all the four metrics. Regarding the generalization performance, as shown in Table 6, LSTM also outperformed all other models on the test set.

	MLP	SAE	SDAE	LSTM
R^2	0.9512 ± 6.0e-3	0.8949 ± 2.6e-2	0.9508 ± 9.4e-3	0.9553 ± 1.6e-2
MSE	1.3e-4 ± 1.3e-5	2.8e-4 ± 8.0e-5	1.3e-4 ± 2.5e-5	1.1e-4 ± 3.7e-5
RMSE	1.1e-2 ± 6.0e-4	1.6e-2 ± 2.2e-3	1.1e-2 ± 1.0e-3	1.0e-2 ± 1.7e-3
MAE	8.1e-3 ± 2.8e-4	1.2e-2 ± 2.3e-3	8.3e-3 ± 7.2e-4	7.9e-3 ± 1.4e-3

Table 5: CV results for the hourly energy demand training dataset.

	MLP	SAE	SDAE	LSTM
R^2	0.9508	0.9162	0.9512	0.9724
MSE	1.1e-4	2.0e-4	1.1e-4	6.7e-5
RMSE	1.0e-2	1.4e-2	1.0e-2	8.1e-3
MAE	8.6e-3	1.0e-2	7.6e-3	5.8e-3

Table 6: Forecasting performances for the hourly energy demand test set.

5 Conclusions

This work sought to investigate the effectiveness of a hybrid approach based on Deep Learning and Evolutionary Algorithm as a time series forecasting method. Three different deep learning models have been selected and tested: SAE, SDAE, and LSTM. A standard MLP is used as baseline. Furthermore, in the proposed methodology, the CMAES algorithm is used for hyperparameter optimization.

Through the experiments and the analysis of the results, it was found that all proposed DL methods presented relevant results and obtained significantly good performance on all studied cases. LSTM presented the best performance in both datasets with seasonal components (i.e., Mackey-Glass and hourly energy demand) and obtained relevant results on the Lorenz System set, proving to be the model best suited for learning temporal dynamics. As explained before, models that apply unsupervised pre-training are not able to capture very well the time dependency in the data. This can explain why both SAE and SDAE obtained poor results on all test cases. The results also revealed that CMAES effectively traverses the solution space and delivers consistent and high-quality results. Therefore, it has been concluded that augmenting

minimal deep neural network and optimizing them using Evolutionary Algorithm are a promising alternative for time series forecasting problems.

References

[1] W. Wei. Time Series Analysis: Univariate and Multivariate Methods. Pearson Addison Wesley, 2006.

[2] E. M. Azoff. Neural Network Time Series Forecasting of Financial Markets. 1st. ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[3] K. Chakraborty et al. Forecasting the behavior of multivariate time series using neural networks. Neural networks: the official journal of the International Neural Network Society, v. 5, n. 6, p. 961–970, 1992.

[4] Y. Bengio. Deep Learning of Representations: Looking Forward. Lecture Notes in Computer Science. p. 1–37. 2013.

[5] P. Romeu et al. Time-Series Forecasting of Indoor Temperature Using Pre-Trained Deep Neural Networks. *In: Proc. of the International Conference on Artificial Neural Networks (ICANN)*, 2013.

[6] T. Kuremoto et al. Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing*, v. 137, p. 47–56, 2014.

[7] J. Bergstra et al. Algorithms for Hyper-Parameter Optimization. In: Proc of the 24th International Conference on Neural Information Processing Systems, p. 2546–2554, 2011.

[8] R. T. Gonzalez, C. A. Padilha and D. Barone. Ensemble system based on genetic algorithm for stock market forecasting. *Evolutionary Computation*, p. 3102-3108, 2015.

[9] I. Loshchilov and F. Hutter. CMA-ES for Hyperparameter Optimization of Deep Neural Networks. *In: Proc. of the International Conference on Learning Representations*, 2016.

[10] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, v. 9, n. 2, p. 159–195, 2001.

[11] A. Ng et al. Autoencoders and Sparsity. Available in: <http://deeplearning.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity>. Visited on: 04/2018.

[12] P. Vincent et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research (JMLR)*, v. 11, n. Dec, p. 3371–3408, 2010.

[13] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, v. 2, n. 1, p. 1–127, 2009.

[14] M. Hüsken and P. Stagge. Recurrent neural networks for time series classification. *Neurocomputing*, v. 50, p. 223–235, 2003.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, v. 9, n. 8, p. 1735–1780, 1997.

[16] N. Hansen. The CMA Evolution Strategy: A Tutorial. Machine Learning, 2016.

[17] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[18] E. N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, v. 20, n. 2, p. 130–141, 1963.

[19] T. Hong. GEFCom2017. Available in: <http://www.drhongtao.com/gefcom/2017>. Visited on: 09/2018.

[20] E. Zivot and J. Wang. Modeling Financial Time Series with S-PLUS. *Springer Science & Business Media*, 2013.

[21] R. A. Fox. *The Incorporated Statistician*, v. 11, n. 3, p. 170–171, 1961.