CONVOLUTIONAL NEURAL NETWORKS FOR MODELING AND FORECASTING NONLINEAR NONSTATIONARY PROCESSES

Andrii Belas1, Petro Bidyuk2

1Department of Mathematical methods for System Analysis, Institute for applied system analysis, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic institute”, Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-7883-2489

2Department of Mathematical methods for System Analysis, Institute for applied system analysis, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic institute”, Kyiv, Ukraine
ORCID: https://orcid.org/0000-0002-7421-3565

Corresponding author: Andrii Belas, e-mail: andrii.belas@gmail.com

1. Introduction

1.1. The object of research

The object of research is modeling and forecasting nonlinear nonstationary processes presented in the form of time-series data.

1.2. Problem description

Forecasting based on models built on experimental (statistical) data is one of the most popular approaches to forecast the dynamics of such processes and has numerous applications in energy, network systems, trade, investment activities. It can be used for evaluating alternative economic strategies, forming budgets of enterprises, forecasting and managing the risks of arbitrary nature and solving other problems [1].

The problem of forecasting processes in technical systems is deeply analyzed using classical autoregressive approaches, which are quite simple to implement. This is a popular family of math-

ARTICLE INFO

Article history:
Received date 18.05.2021
Accepted date 24.04.2021
Published date 30.06.2021

Section:
Information Technology

DOI
10.21303/2313-8416.2021.001924

KEYWORDS
mathematical modeling
signal processing
time-series
nonstationary processes
convolutional neural networks
recurrent neural networks

ABSTRACT

The object of research. The object of research is modeling and forecasting nonlinear nonstationary processes presented in the form of time-series data.

Investigated problem. There are several popular approaches to solving the problems of adequate model constructing and forecasting nonlinear nonstationary processes, such as autoregressive models and recurrent neural networks. However, each of them has its advantages and drawbacks. Autoregressive models cannot deal with the nonlinear or combined influence of previous states or external factors. Recurrent neural networks are computationally expensive and cannot work with sequences of high length or frequency.

The main scientific result. The model for forecasting nonlinear nonstationary processes presented in the form of the time series data was built using convolutional neural networks. The current study shows results in which convolutional networks are superior to recurrent ones in terms of both accuracy and complexity. It was possible to build a more accurate model with a much fewer number of parameters. It indicates that one-dimensional convolutional neural networks can be a quite reasonable choice for solving time series forecasting problems.

The area of practical use of the research results. Forecasting dynamics of processes in economy, finances, ecology, healthcare, technical systems and other areas exhibiting the types of nonlinear nonstationary processes.

Innovative technological product. Methodology of using convolutional neural networks for modeling and forecasting nonlinear nonstationary processes presented in the form of time-series data.

Scope of the innovative technological product. Nonlinear nonstationary processes presented in the form of time-series data.
Innovative technologies in industry

Mathematical models based on linear self-dependence within time series (autocorrelation), which is able to explain future fluctuations [2, 3]. However, this approach is limited by the difficulty of taking into account a large number of external factors, due to the problem of multicollinearity; in addition, it also can’t model nonlinear interactions [4].

Therefore, it is proposed to consider the possibility of using neural networks, as they may take into account the nonlinear interactions or combined influence of external factors. The first thing that can be applied to any sequence analysis with neural networks is recurrent neural networks. They are created specifically for sequences with the ability to maintain a hidden state and learn time dependencies [5]. But, as recent research has shown, there is little use of these benefits in practice. Applying this approach requires a lot of computational costs, so this approach cannot be applied to very long sequences, which is a problem for solving modern problems using large amounts of data [4].

1. 3. Suggested solution to the problem

There is a need to develop a new approach that would allow efficient computational modeling of large sequences, taking into account the nonlinear or combined effects of external factors. To solve this problem, it is proposed to consider convolutional neural networks (CNN) [6, 7]. CNN is suitable for creating computer vision because it is able to capture the finest details (local patterns) in images or even 3D volumetric data. In addition, there are already a large number of modern architectures for convolutional neural networks, such as ResNet or DenseNet [8]. So it is possible to try to apply them to even simpler 1D data, in which it is possible to replace 2D convolutions with 1D. They are highly efficient, fast, can be optimized, and work well for both classification and regression analysis.

The aim of the research is to develop a mathematical model based on convolutional neural networks for forecasting nonlinear nonstationary processes.

2. Materials and methods

2. 1. Neural networks of the LSTM type

One of the effective approaches for modeling and forecasting is the technology of artificial neural networks. For working with sequences (time series, signals, etc.), it is common to use recurrent neural networks.

Networks with long short-term memory – are usually simply called “LSTM” – a special kind of RNN, capable of learning long-term dependencies. They were proposed by [11] and were refined and popularized by many people in further work [12]. They provide the opportunity to get high quality results on a wide variety of problems and are currently widely used to simulate nonlinear processes.

LSTM have a chain structure, like the classic RNN, but the repeating module has a different structure. Instead of having a single neural network layer, there are four of them, and they interact in a special way (Fig. 1) [13].

![Fig. 1. Structure of repeating module LSTM](image)

Key to LSTM is the cell state – a horizontal line running through the top of the diagram. It runs straight down the entire chain, with only some minor linear interactions. Information can simply flow through it without change.
LSTM has the ability to remove or add information to the cell state, but this ability is carefully regulated by structures called gates. The first step in the LSTM is to decide what information let’s intend to throw out of the cell state. This decision is taken by a sigmoid layer (named after used activation function), also called the “forget gate layer”, which can be written as:

\[ f_i = \sigma(Wh_f [h_{t-1}, x_t] + b_f) \]

where \( h_{t-1} \) – previous hidden state; \( x_t \) – new input data; \( W_f, b_f \) – weight matrix and bias for this layer respectively.

The next step is to decide what new information are going to keep in a cell state. This step consists of two parts. Firstly, the another sigmoid layer, called the “input gate layer”, decides what values let’s update. Next, the layer of the hyperbolic tangent creates a vector of candidates for new values \( C(t) \), which can be added to the state. In the next step, let’s connect these two parts to create an update for the state. This step can be written as:

\[ i_t = \sigma(Wh_i [h_{t-1}, x_t] + b_i) \]

\[ \tilde{C}_t = \tanh(Wh_c [h_{t-1}, x_t] + b_c) \]

where \( h_{t-1} \) – previous hidden state; \( x_t \) – new input data; \( W_i, b_i, W_c, b_c \) – weight matrices and biases for this layer respectively.

Next, it is necessary to update the old cell state, \( C(t-1) \), with the new cell state \( C(t) \):

\[ C_t = f_t C_{t-1} + i_t \tilde{C}_t \]

Finally, it is necessary to decide what result are going to give way to the exit. This result will be based on the output result \( o_t \) and the cell state \( C_t \), but it will be a filtered version, using hyperbolic tangent activation function, to scale the values between –1 and 1:

\[ o_t = \sigma(Wh_o [h_{t-1}, x_t] + b_o) \]

\[ h_t = o_t \tanh(C_t) \]

However, the construction of such networks is associated with great computational difficulties. In addition, it is confronted with numerous problems [14] that will not allow them to work with too long sequences (for example, when processing a high-frequency sampling rate stream, for example, 500–100 Hz).

2. 2. Convolutional neural networks (CNN)

Convolutional neural networks (CNN) were introduced by LeCun in [6]. The network architecture got its name from the presence of a convolution operation, the essence of which is that each fragment of the image is multiplied by a convolution matrix element by element, and the result is summed and recorded in a similar position of the original image. In network architecture laid a priori knowledge of the subject area computer vision: pixel image strongly related to the neighboring (local correlation) and the object in the image can be found in any part of the image.

The basic idea of CNNs is that an image fed to the input of the neural network directly. And the network automatically learns and determines the hierarchy of necessary features. So it is possible to get a network that is more accurate than a network built on traditional approaches without difficult feature engineering [15]. Neural networks with convolution solve two problems at once. First, study nonlocal dependencies – they are able to find certain relationships and certain templates are not only linked to their local value. Second, here let’s study an incredible reduction in the number of parameters. CNN uses relatively little pre-processing and additional feature engineering and that makes them very efficient.

This idea was used to recognize symbols and numbers in [16]. But after a single successful use of convolutions neural networks they have not gained popularity. And only in [17] have revived
interest in convolutional neurons networks after showing impressively high classification accuracy images in the ImageNet Large Scale Visual Recognition Challenge. In this competition, neural networks were applied to a data set that had more than a million images from the Internet, which contained more than 1,000 different classes. This success revolutionized computer vision application of deep networks in various directions and in many different computer vision applications [18].

Let’s examine how CNNs applied to the image data so successful and then think how it is possible to transfer this into time series data.

The input data is defined as a tensor of 3rd or 2nd rank, depending on the number of channels present in the data. Channels mean the number of variables which describes the color of the pixel. That is, in the case of RGB encoding, let’s have 3 channels. However, to reduce the amount of data, quite often the input data is translated into 1 black and white channel. Resulting value of the convolutional operation for each pixel of the filtered image is calculated based on the square area around it using the convolution core (kernel). Fig. 2 demonstrates this transformation:

\[
(I * K)(r,s) = \sum_{i=-u}^{u} \sum_{j=-v}^{v} K(i,j) I(r+i,s+j),
\]

where \((I*K)(r,s)\) – value of the resulting matrix at the point \((r,s)\); \(K\) – convolutional kernel, matrix size of \((2u+1, 2v+1)\); \(I\) – input matrix, \(I(i,j)\) – pixel value of the input image.

As it is possible to see from the Fig. 2, and the formula above, in convolutional operation there is elementwise matrix multiplication, sliding kernel matrix across the input matrix. Kernel matrix is the matrix with model parameters, which have to be learned. It initialized at random and then changed during training as a result of optimization procedure such as stochastic gradient descent.

It is also worth noting that usually when building a convolutional layer at the same time several filters are used, as a result of which let’s get new ones at the output images, commonly referred to as feature maps. That is, in total the features can change as shown in Fig. 3.

It is possible to write this as:

\[
y'_j = b'_j + \sum_{i=1}^{n} K'_{i,j} y''_{j-i},
\]
where $y_i^s$ – values in the $s$ layer; $b_i^s$ – bias for the $i$ value in the $s$ layer; $m_{s-1}$ – size of the $(s-1)$th layer; $K_{i,j}^s$ – kernal value at the given point.

So, it is possible to see how convolutional neural networks can be applied on computer vision problems and perform particularly well, due to their ability to operate convolutionally, extracting features from local input patches and allowing for representation modularity and data efficiency. The same properties that make CNNs excel at computer vision also make them highly relevant to sequence processing.

It is possible to try to use CNNs to the problem of forecasting time series data in which it is possible to replace 2D convolutions with 1D. Therefore, it is possible to try to use all of the advantages described above and achieve great performance, nonlinear and multivariable interactions with reasonable speed and complexity. Time can be treated as a spatial dimension, like the height or width of a 2D image. Such 1D convnets can be competitive with RNNs on certain sequence-processing problems, usually at a considerably cheaper computational cost. Small 1D CNNs can offer a fast alternative to RNNs for tasks such as time series forecasting. The convolution layers introduced previously were 2D convolutions, extracting 2D patches from image tensors and applying an identical transformation to every patch. In the same way, it is possible to use 1D convolutions, extracting local 1D patches (subsequences) from sequences (Fig. 4).

![Fig. 4. How 1D convolution works: each output time-step is obtained from a temporal patch in the input sequence.](image)

Such 1D convolution layers can recognize local patterns in a sequence. Because the same input transformation is performed on every patch, a pattern learned at a certain position in can later be recognized at a different position, making 1D CNNs translation invariant (for temporal transformations). For instance, a 1D CNN processing time series using convolution windows of size 5 should be able to learn pattern fragments of length 5 or less, and it should be able to recognize them in any other context in an input series.

2. 3. Experimental procedures

For practical examples, let’s use a sample of real historical sales data for 1 of the 45 Walmart stores located in different regions [9]. The problem is forecasting weekly sales in various stores and departments for retail trade. The sample size is 138 weeks. This paper uses the last 28 weeks to test and evaluate the quality of forecasts.
Data loaders for the neural network training created using sliding window approach with 1 week step size, 1st week forecast horizon, and 4 weeks lookback period (Fig. 5). Such approach is very popular to prepare data in form that is used for neural networks training and described in many works [19]. Data was normalized before training.

Fig. 5. Example of using the sliding window approach for building the data loaders for neural network training

The methodology proposed in [10] is used as a general approach. Naïve forecast with previous observation were used as a benchmark baseline. For comparison let’s train RNN and CNN models. For RNN let’s use 1 RNN layer with hidden size of 32 and linear layer to compute the output prediction (Fig. 6).

Fig. 6. Recurrent neural network architecture

For CNN let’s use 1D convolution with 1 channel, kernel size 2 and stride 1. For computing output prediction linear layer was used (Fig. 7).
3. Results

For evaluation of the results let’s use MAE as our metric here. Comparison in terms of accuracy and complexity of the models (number of parameters) is given in the following Table 1.

| Model        | MAE            | Number of parameters |
|--------------|----------------|----------------------|
| Naive baseline | 2374.9903571428576 | 1                    |
| RNN          | 1769.4269642857143 | 1153                 |
| CNN          | 1618.9840513392858 | 7                    |

From this results it is possible to see, that, first of all, our models are better than naive baseline, so results are at least meaningful. Secondly, it is possible to see that convolutional networks are superior to recurrent ones in terms of both accuracy and complexity. It was able to build more accurate model with a much less number of parameters. It clearly indicates that 1D CNNs can be quite reasonable choice for solving time series forecasting problems.

4. Discussion

The task of forecasting processes in technical systems is very deeply analyzed in literature using classical regression approaches, which are quite simply used both from a theoretical and a computational point of view [2, 3, 20, 21]. However, this approach has drawbacks, because it cannot take into account a large number of external factors, due to the problem of multicollinearity, moreover, if they show nonlinear interaction [4].

Another popular approach is to consider neural networks of the LSTM type [5], which also solves the problem of modeling sequences, in addition to taking into account the nonlinear or combined effects of external factors. This is the most common and accepted approach for the task of modeling and forecasting nonlinear nonstationary processes using time series data [7]. There are a lot of modern studies using this approach to solve the problem mentioned above [22, 23]. But they are focusing only on recurrent neural network architectures, such as LSTM.

However, the application of this approach requires large computational costs, and this approach cannot be applied to very long sequences, what creates the problem for modern studies with the use of big data [4].

In this study it was able to develop a mathematical model for forecasting nonlinear nonstationary processes, based on novel architecture for this kind of data - convolutional neural networks, so it is possible to take into account nonlinear influence of the previous observations or the external factors without creating very complicated (in terms of number of parameters and computational architecture) model.
This particular dataset was studied in [24, 25]. In both works authors used classical machine learning approaches. The results of their research indicate that the Random Forest is the best algorithm which has scored the minimum amount in MAE evaluation of 1979.4. However, in our work, using novel CNN approach let’s achieve MAE of 1618.9, which is much better result.

Limitation of the current approach is that kernel size is limited and it is not possible to handle the very start and the very end of the sequence in one kernel.

However in the further research it is possible to explore deeper or more complicated architectures of the convolutional neural networks such as ResNet and try to translate it to the time series domain; or it is possible to try combination of the CNNs and LSTMs or combinations with classical regression approaches to include more diverse information into the model.

5. Conclusions

In this study a comparative analysis of recurrent and convolutional neural networks for modeling and prediction of nonlinear nonstationary processes was performed both in theory and on practical examples based on real world data.

The model for forecasting nonlinear nonstationary processes presented in the form of the time series data was built using convolutional neural networks. Current study shows results in which convolutional networks are superior to recurrent ones in terms of both accuracy and complexity. It was able to build more accurate model with MAE of 1618.9, while in RNN approach MAE was 1769.4. Also CNN model has a much less number of parameters – CNN model has only 7 parameters, while RNN 1153, which is much harder from computational point of view and has tendency to overfitting. It clearly indicates that 1D CNNs can be quite reasonable choice for solving time series forecasting problems.

Because RNNs are extremely expensive for processing very long sequences, but 1D CNNs are cheap, it can be a good idea to use a 1D CNNs as a preprocessing step before an RNN, shortening the sequence and extracting useful representations for the RNN to perform further processing. For the future research the idea of combining convolutional and recurrent networks in one model or combining neural networks with classical approaches for better detection of time dependences also looks quite promising.

In conclusion, one-dimensional convolutional networks or a combination of convolutional networks with recurrent networks or regression models can solve the problem of modeling nonlinear nonstationary processes presented in the form of long sequences in which there is a nonlinear or combined influence of external factors. Thus, this approach can be a powerful tool for creating adequate models and acceptable quality forecasts of selected processes.

References

[1] Palit, A., Popovic, D. (2005). Computational intelligence in time series forecasting: theory and engineering applications. Springer Science & Business Media, 372. doi: http://doi.org/10.1007/1-84628-184-9
[2] Bidyuk P., Romanenko V., Timoschuk O. (2010). Analysis of time series. Kyiv: NTUU «KPI».
[3] Hyndman R., Athanasopoulos G. (2013). Forecasting: Principles and Practice. OTexts.
[4] Belas, O., Bidiuk, P. Belas, A. (2019). Comparative analysis of autoregressive approaches and recurrent neural networks for modeling and forecasting nonlinear nonstationary processes. Information Technology and Security, 7 (1), 91–99. doi: http://doi.org/10.20535/2411-1031.2019.7.1.184395
[5] Gers, F. A., Eck, D., Schmidhuber, J. (2001). Applying LSTM to Time Series Predictable through Time-Window Approaches. Proceedings of International Conference on Artificial Neural Networks, 669–676. doi: http://doi.org/10.1007/3-540-44668-0_93
[6] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1 (4), 541–551. doi: http://doi.org/10.1162/neco.1989.1.4.541
[7] Goodfellow I., Bengio, J., Courville, A. (2018). Deep learning. MIT Press.
[8] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: http://doi.org/10.1109/cvpr.2016.90
[9] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9 (8), 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735
[10] Hochreiter S., Bengio Y., Schmidhuber, J. (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. IEEE Press.
[11] Nikolenko, S., Kadurin, A., Arkhangelskaya, E. (2018). Deep learning. Saint Petersburg: Peter, 479.
[12] Chollet, F. (2017). Deep Learning with R. Manning. Black & White, 384.
[13] Zeiler, M. D., Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. ECCV Press, 818–833. doi: http://doi.org/10.1007/978-3-319-10590-1_53
[14] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86 (11), 2278–2324. doi: http://doi.org/10.1109/5.726791
[15] Krizhevsky, A., Sutskever, I., Hinton, J. (2012). Imagenet classification with deep convolutional neural networks. NIPS, 1106–1114.
[16] LeCun, Y., Kavukcuoglu, K., Farabet, C. (2010). Convolutional networks and applications in vision. Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2 (4), 253–256. doi: http://doi.org/10.1109/iscas.2010.5537907
[17] Walmart. (2014). Walmart Recruiting - Store Sales Forecasting [Dataset]. https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data.
[18] Laptev, N., Smyl, S., Shanmugam, S. (2017). Engineering Extreme Event Forecasting at Uber with Recurrent Neural Networks. Uber Engineering. Available at: http://roseyu.com/time-series-workshop/submissions/TSW2017_paper_3.pdf
[19] Belas, O., Belas, A. (2021). General methods of forecasting nonlinear nonstationary processes based on mathematical models using statistical data. System research and information technologies, 1 (1), 79–86.
[20] Bergmeir, C., Hyndman, R. J., Benítez, J. M. (2016). Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. International Journal of Forecasting, 32 (2), 303–312. doi: http://doi.org/10.1016/j.ijforecast.2015.07.002
[21] De Livera, A. M., Hyndman, R. J., Snyder, R. D. (2011). Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing. Journal of the American Statistical Association, 106 (496), 1513–1527. doi: http://doi.org/10.1198/jasa.2011.tm09771
[22] Sagheer, A., Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing, 323, 203–213. doi: http://doi.org/10.1016/j.neucom.2018.09.082
[23] Chimmula, V. K. R., Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos, Solitons & Fractals, 135, 109864. doi: http://doi.org/10.1016/j.chaos.2020.109864
[24] Colon, S., Gil, J. (2019). Data Mining Techniques and Machine Learning Model for Walmart Weekly Sales Forecast. Puerto Rico. Available at: https://prcrepository.org/xmlui/bitstream/handle/20.500.12475/174/FA-19_Articulo %20Final_Jose %20Santaella.pdf
[25] Elias, N., Singh, S. (2018). Forecasting of Walmart sales using machine learning algorithms. Available at: https://api.semanticscholar.org/CorpusID:209465807