НЕЙРОІНФОРМАТИКА ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

NEUROINFORMATICS AND INTELLIGENT SYSTEMS

UDC 004.852

NEURAL ORDINARY DIFFERENTIAL EQUATIONS FOR TIME SERIES RECONSTRUCTION

Androsov D. V. – Post-graduate student of the Institute for Applied System Analysis, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

ABSTRACT

Context. Neural Ordinary Differential Equations is a deep neural networks family that leverage numerical methods approaches for solving the problem of time series reconstruction, given small amount of unevenly distributed samples.

Objective. The goal of the following research is the synthesis of a deep neural network that is able to solve input signal reconstruction and time series extrapolation task.

Method. The proposed method exhibits the benefits of solving time series extrapolation task over forecasting one. A model that implements encoder-decoder architecture with differential equation solving in latent space, is proposed. The latter approach was proven to demonstrate outstanding performance in solving time series reconstruction task given a small percentage of noisy and uneven distributed input signals. The proposed Latent Ordinary Differential Equations Variational Autoencoder (LODE-VAE) model was benchmarked on synthetic non-stationary data with added white noise and randomly sampled with random intervals between each signal.

Results. The proposed method was implemented via deep neural network to solve time series extrapolation task.

Conclusions. The conducted experiments have confirmed that proposed model solves the given task effectively and is recommended to apply it to solving real-world problems that require reconstructing dynamics of non-stationary processes. The prospects for further research may include the process of computational optimization of proposed models, as well as conducting additional experiments involving different baselines, e. g. Generative Adversarial Networks or attention Networks.

KEYWORDS: neural ordinary differential equations, deep neural networks, variational autoencoders, recurrent neural networks, long term short memory networks

ABBREVIATIONS

NN is a neural network;

ODE is an ordinary differential equation;

LSTM is a long short-term memory network;

GRU is a gated recurrent unit network

ARMA is an autoregressive moving average model;

ARIMA is an autoregressive integrated moving average model;

GARCH is a generalized autoregressive conditional heteroskedasticity;

ELBO is an evidence lower bound function.

NOMENCLATURE

y is an unknown non-stationary non-negative continuous-time series;

t is a continuous time point, $t \in \mathrm{t}$;

$\lambda = \lambda(\tau)$ is an intensity parameter of time distribution, $\tau > 0$;

θ is a model parameter vector;

$\hat{y}$ is a observed sample from time series;

Y is a multivariate discrete-time or continuous-time process;

x_t is an input vector to neural network at time t, $x_t \in \mathrm{x}$;

E is a time series reconstruction loss;

N is a Normal distribution;

σ is a standard deviation of y;

α is a time series sampling percentage;

$y^{(rec)} = \hat{y} \bigcup f(x)$ is a reconstructed time series;

h_t is a hidden state of neural network at time t, $h_t \in \mathrm{h}$;

z_0 is a latent space initial parameter for decoder network;

ε is a moving average parameter for time series y;

b_h is a bias vector for hidden state of a recurrent neural network;

b_u is a bias vector for output state of a recurrent neural network;

DOI 10.15588/1607-3274-2023-4-7

b_r is a bias vector for reset state of a recurrent neural network;

W_h is a weight matrix for hidden state of a recurrent neural network;

$W_{\cdot u}$ is a weight matrix for output state of a recurrent neural network;

$W_{\cdot r}$ is a weight matrix for reset state of a recurrent neural network;

$h_t^{'}$ is a temporary hidden state at time t;

$h_t^{''}$ is a candidate hidden state at time t;

INTRODUCTION

Time series analysis nowadays is one of the most rapidly developing field of computational statistics. In recent years it gained a significant push towards applying machine learning methods to solve the problem of predicting real-world processes in various fields, e.g., physics or finances. The most popular approaches that are applied to overcome the given problems include autoregressive modeling via ARIMA and GARCH models [1–5]. These models completely rely on assumptions of autoregressive nature of given process, i.e., linear dependence between current state of process with previous ones, and stationarity, i.e., absence of mean/variance fluctuations through time of observation of given process. In order to process non-stationary data, ARIMA models leverage mechanisms of taking finite difference of given time series data to vanish trend curves and thus transform such data into stationary time series [6]. On the other hand, GARCH models perform conditional heteroskedasticity modeling via moving average estimation of variance [7]. The drawbacks of these approaches are implicated in assumption of linear dependence between lags of a given process.

To overcome these challenges recurrent neural networks (RNN) [8] were introduced. RNN rely on the same concept of "dependency" between time series states as in autoregressive models, except that it applies nonlinear transformations of input states and hidden states. These chains of operations allow RNN to model nonstationary series without performing reduction to stationarity. LSTM models are the most widely used RNN application, since they achieve ability to capture patterns in time-dependent data at large scale of observation [8, 9]. Since 2014, new family of recurrent neural networks was introduced – gated recurrent unit network (GRU) [8, 9]. GRU is, in essence, a lightweighted version of LSTM, offering reduced complexity whilst learning, both time and space. However, the weakest point of given approaches is that they are invariant to occurrence gaps, i.e., they built regarding the assumption that intervals between each sample are equal. However, for various cases that assumption is not a valid one, e.g., for tracking real-time financial data.

The object of study is the process of time series reconstruction from samples with uneven distribution regarding time. This data is difficult to predict since both autoregressive models, and recurrent neural networks are invariant to intra-sample gaps, Therefore, it is proposed to construct a new model, called latent ordinary differential equations variational autoencoders to improve the quality of predictions of given data and to solve process recreation task.

The subject of study is methods for time series prediction and recreation.

The purpose of the work is to create a machine learning model to solve time series reconstruction from small and unevenly distributed data samples.

1 PROBLEM STATEMENT

For a given sample $\hat{y}$ of time series y it is desired to create a reconstruction $y^{(rec)}$, such that $\forall \varepsilon > 0, d(y^{(rec)}, y) \le \varepsilon$, where $d(\cdot,\cdot)$ is some distance metric.

2 REVIEW OF THE LITERATURE

Autoregressive models, such as ARMA [9], presented in 1951, are the most used ones for the time series modeling task. They consider the time series to be in the form $y_t = \theta_0 + \sum_{i=1}^{p} \theta_i y_{t-i} + \sum_{j=1}^{q} \theta_{p+j} \varepsilon_j$ [1]. ARMA models are suitable for learning behavior of stationary processes and hence, are widely and successfully used applied to this task [1–5, 9, 10].

In order to cope with non-stationary processes, ARIMA models were introduced [1, 11]. They perform numerical differentiation techniques, i.e. applying finite differences operator $\nabla_d(y_t) = \nabla_{d-1}(y_t \cdot - y_{t-1}), d \in \mathrm{N}$ to a given time series to vanish it's non-stationarity [1].

The other autoregressive approach is to model stationary processes in the form $y_t = \varepsilon \sqrt{\theta_0 + \sum_{i=1}^{p} \theta_i y_{t-i}^2 + \sum_{j=1}^{q} \theta_{p+j} \sigma_{t-j}^2}$ [1].

Since the breakthrough in the development of computational capacities, it became feasible to examine machine learning algorithms, and, in particular, artificial neural networks, on time series modeling and predictions tasks. One of the most successful approaches is to apply recurrent neural networks to discrete-time time series.

Consider sample of time series y_t of the form $D = \langle \mathrm{x}_t, y_t \rangle$. RNN in this case is a mapping function $f: \mathrm{x}_t \to y_t$, and that function is essentially a chain of non-linear transformations over affine transformations that are provided by state-space modeling of y_t [8]. Classic RNN models these chains in a following way:

$$h_t = \sigma(W_{xh} x_t + W_{hh} h_t + b_h),$$

DOI 10.15588/1607-3274-2023-4-7

$$y_t = u_t = \sigma(W_{hu} h_t + b_u).$$

RNN is trained to maximize logarithmic likelihood $\log p(y_t \mid \mathrm{x}, U, W, V, c)$ [8]. However, despite the ability to model non-stationary time series, RNN has a severe drawback – gradient vanishing – that is caused by it's architecture [8].

To overcome this challenge, LSTM model was introduced in 1997 [12]. LSTM offers more complex architecture yet greater precision of forecasts, introducing 3 "gates" – input, output and forget. The latter one implements the process of capturing short-term dependencies at a long scale, thus giving it's name to the method [8, 12].

In 2014, GRU model was firstly described [8, 13]. GRU model aims to achieve the same quality of forecasts but at a lower computational cost, reducing number of parameters to train. Despite LSTM-based models have become de facto standard in recent years, GRU models are also widely used [8, 13].

In recent years, new family of neural networks was introduced, called neural ordinary differential equations [14]. Neural ODE model interprets time series as a continuous process with unknown dynamics and thus solving a differential equation with respect to the hidden state and time: $\frac{d\mathrm{h}}{dt} = f(\mathrm{h}(t), t, \theta)$. Neural ODEs are proven to be effective for survival analysis [15] and weather data prediction [16].

3 MATERIALS AND METHODS

It is proposed to recreate time series structure from the latent space (i.e., some mapping of feature space) process dynamics. Let's add a mapping $f : \mathrm{P}^m \to \mathrm{P}^n$, such that:

$$\mathrm{H} = f(\mathrm{Y}), \tag{1}$$

where H represents hidden dynamics in latent space and thus is proposed to be modeled via Neural ODE:

$$\frac{d\mathrm{h}}{dt} = g(\mathrm{h}(t), t, \theta_{\mathrm{h}}). \tag{2}$$

Integrating (2) with respect to time allows forecasting multivariate continuous-time process (1) in latent space. Then, to retrieve forecast of Y it is necessary to add an inverse mapping $f^{-1} : \mathrm{P}^n \to \mathrm{P}^m$. However, the inverse mapping could not exist under certain conditions, e.g., f is not bijective. In that case, to overcome this restriction, it is proposed to use encoder-decoder approach, i.e., defining another mapping $g : \mathrm{P}^n \to \mathrm{P}^m$, such that:

$$\hat{\mathrm{Y}} = g(\mathrm{H}). \tag{3}$$

By substituting (1) and (4) as a parameters to the given reconstruction loss, an optimization problem is defined:

$$E\left(\mathrm{Y}, \hat{\mathrm{Y}}\right) \to \min. \tag{4}$$

There are multiple approaches to define exact task in the form of problem (4). Let's consider that process Y is drawn from unknown random vector of distribution $\rho(\mathrm{x})$. Then chain of transformations defined by (1) and (4) produce the process $\hat{\mathrm{Y}}$, drawn from random vector of distribution $\hat{\rho}(\mathrm{h})$. Then to measure difference between these two distributions it is proposed to use Kullback-Leibler divergence, defined as:

$$KL(\rho \mid \hat{\rho}) = \int_{-\infty}^{+\infty} \rho(\mathrm{x}) \log \frac{\rho(\mathrm{x})}{\hat{\rho}(\mathrm{x})} d\mathrm{x}. \tag{5}$$

However, since $\rho(\mathrm{x})$ is an unknown distribution, mentioning that distribution proposed approach should define both mappings (1) and (3), it is proposed to define a joint probability distribution $p(\mathrm{x}, \mathrm{h})$, and applying defining it as:

$$p(\mathrm{x}, \mathrm{h}) = p(\mathrm{h} \mid \mathrm{x}) p(\mathrm{x}) = p(\mathrm{h}) p(\mathrm{x} \mid \mathrm{h}). \tag{6}$$

Taking a logarithm of (7):

$$\begin{aligned} &\log p(\mathrm{x}, \mathrm{h}) = \log p(\mathrm{h} \mid \mathrm{x}) + \log p(\mathrm{x}) = \\ &= \log p(\mathrm{h}) + \log p(\mathrm{x} \mid \mathrm{h}). \end{aligned} \tag{7}$$

To achieve deterministic measure of "fitness" of distributions $p(\mathrm{h}) = p(\mathrm{h} \mid \theta)$ and $p(\mathrm{x}) = p(\mathrm{x} \mid \theta)$, where θ is a parameters vector of the desired model, let's apply a mathematical expectation operator with respect to latent state h to (8):

$$\begin{aligned} &\int_{\mathrm{h}} q(\mathrm{h}) \log p(\mathrm{x} \mid \theta) d\mathrm{h} = \\ &= \int_{\mathrm{h}} q(\mathrm{h}) \log p(\mathrm{x}, \mathrm{h} \mid \theta) d\mathrm{h} - \\ &- \int_{\mathrm{h}} q(\mathrm{h}) \log p(\mathrm{h} \mid \mathrm{x}, \theta) d\mathrm{h} \end{aligned} \tag{8}$$

Left-hand side of (8) is simply equals to $p(\mathrm{x} \mid \theta)$. Then let's add and subtract mathematical expectation of logarithmic probability of hidden state:

DOI 10.15588/1607-3274-2023-4-7

$$p(\mathrm{x} \mid \theta) = \\ = \int_{\mathrm{h}} q(\mathrm{h})(\log p(\mathrm{x},\mathrm{h} \mid \theta) - \log p(\mathrm{h}))d\mathrm{h} - \\ - \int_{\mathrm{h}} q(\mathrm{h})(\log p(\mathrm{h} \mid \mathrm{x},\theta) - \log p(\mathrm{h}))d\mathrm{h}. \quad (9)$$

Last component of right-hand side of the equation (9) is Kullback-Leibler divergence of $q(\mathrm{h})$ and $p(\mathrm{h} \mid \mathrm{x},\theta)$ distributions, and the first one is ELBO. ELBO is proposed to use as a reconstruction loss of a proposed model.

Let's define the mapping functions in the scope of problem (1)-(9). It is proposed to use a recurrent neural network, in particular, GRU as an encoder mapping, i.e. mapping (1) is defined as:

$$h_t^{''} = \hat{f}(y_\theta, h_{t-1}, t), \quad (10)$$

$$u_t = \sigma(W_{xu}x_t + W_{hu}h_t^{''} + b_u), \quad (11)$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_t^{''} + b_r), \quad (12)$$

$$h_t^{'} = \sigma(W_{xh'}x_t + W_{hh'}h_t^{''} + b_{h'}), \quad (13)$$

$$h_t = u_t h_t^{''} + (1-u_t)h_t^{'}, \quad (14)$$

Hence it is proposed to interpret latent space features as the dynamics of given process, defined by (1), (10) is defined as:

$$h_t^{''} = Sol(f_\theta, h_{t-1}, t), \quad (15)$$

where Sol is a numeric ODE solver, e.g., Runge-Kutta method.

Let's add a layer that produces parameter for latent parameter z_0 and define mapping (4) as:

$$\forall t \in \mathrm{t}, z_t = Sol(z_0, \theta_f, t), \quad (16)$$

$$x \sim p(\mathrm{x} \mid z_t, \theta). \quad (17)$$

For achieving time sensitivity, it is feasible to model probability distributions of time spots using non-stationary Poisson processes. By adding and modeling intensity parameter λ, (16)–(17) can be augmented in the following way:

$$\forall t \in \mathrm{t}, t \sim \mathrm{PoissProcess}(\lambda(\tau)). \quad (18)$$

Then (10) could be augmented by adding:

$$\log(\mathrm{t} \mid t_{\min}, t_{\max}, \lambda)) = \sum_{t \in \mathrm{t}} \log \lambda(t) - \int_{t_{\min}}^{t_{\max}} \lambda(t)dt. \quad (19)$$

By applying task (10)–(17) to input samples, drawn from Y and minimizing loss, defined in (9) and (19), task of time series reconstruction was achieved.

4 EXPERIMENTS

To solve previously defined task and measure the effectiveness of proposed approach the computer program that implements time series reconstruction was developed.

For time series reconstruction experiment a following synthetic dataset was chosen:

$$y = \sin(25\pi t) + \varepsilon, \varepsilon \sim \mathrm{N}(0,\sigma). \quad (20)$$

Multiple samples were drawn from (20) with the following setups:

1. $\sigma = [0.1, 0.5, 1]$;
2. $\alpha = [0.15, 0.35, 0.55]$.

For model there were chosen 2 options – LODE-VAE without modeling distribution of time points and LODE-VAE with modeling distribution of time points using (18). For reference, first model is called LODE-VAE-N and LODE-VAE-P.

For both models the next parameters were chosen:

1. Dimension size of latent state dynamics process is 6.
2. Dimension size of integrated state vector is 6.
3. Dimension size of decoded vector is 1.
4. Number of epochs is 200.
5. Learning rate is adaptive with exponential decay with start rate at 0.01.
6. ODEs are solved using Dormand-Prince method.

Metrics for benchmarking are mean squared error (MSE) and coefficient of determination R^2.

Results of LODE-VAE-N model benchmarking by MSE metric are shown in Table 1.

5 RESULTS

In the following Tables 1 – 4 results of benchmarking of LODE-VAE-N and LODE-VAE-P models by MSE and R^2 metrics are provided. Since MSE and R^2 metrics are both used for validating model adequacy for forecasting time series, their optimization objectives are opposite – MSE needs to be minimized and R^2 needs to be maximized.

6 DISCUSSION

As follows from Tables 1–4, MSE and R^2 metric values of benchmarking of LODE-VAE-N and LODE-VAE-P differs slightly. Despite the difference, both models are well suitable for time series reconstruction and forecasting from obtained unevenly distributed samples.

Tables 1–4 show the same tendencies for both metrics and both models – the more data is available the better the quality of predictions.

DOI 10.15588/1607-3274-2023-4-7

Table 1 – Results of LODE-VAE-N model benchmarking by MSE metric

α\σ	0.1	0.5	1
0.15	0.4309	0.7043	1.4491
0.35	0.4081	0.5921	1.4104
0.55	0.3802	0.6757	1.2195

As expected, model performed reconstruction task well, and results the better the noise level is lower

Results of LODE-VAE-P model benchmarking by MSE metric are shown in Table 2.

Table 2 – Results of LODE-VAE-P model benchmarking by MSE metric

α\σ	0.1	0.5	1
0.15	0.4011	0.583	1.4583
0.35	0.3301	0.6047	1.3876
0.55	0.1065	0.429	1.0456

As is shown above, learning time distribution better model forecasts, and the gap between two models is increasing with more dense samples.

Results of LODE-VAE-N model benchmarking by R^2 metric are shown in Table 3. Coefficient of determination is stable and increasing slowly with increasing density of sampling and decreasing noise level. The same is true for LODE-VAE-P.

Results of LODE-VAE-P model benchmarking by R^2 metric are shown in Table 4.

Table 3 – Results of LODE-VAE-N model benchmarking by R^2 metric

α\σ	0.1	0.5	1
0.15	0.7343	0.5943	0.5523
0.35	0.7529	0.6303	0.571
0.55	0.827	0.6988	0.61

Table 4 – Results of LODE-VAE-P model benchmarking by R^2 metric

α\σ	0.1	0.5	1
0.15	0.843	0.5413	0.5612
0.35	0.8501	0.6171	0.6001
0.55	0.9146	0.7307	0.658

LODE-VAE-P model is demonstrating better results for all the metrics and all the experiment setups. By leveraging separate model for learning the distribution of time points in the sample, the latter model can better approximate the ground truth distribution of the sample.

CONCLUSIONS

The problem of continuous-time processes reconstruction from noised and unevenly distributed samples is solved in this work.

The scientific novelty of obtained results shows that neural ordinary differential equations models could be embedded into variational autoencoders framework for reconstructing dynamic of given unknown but observed process. Combining numerical integration techniques with stochastic generative models is a valid and effective approach for modeling and forecasting non-stationary time series.

The practical significance of current work and its' results is that implemented models could be applied to forecast non-stationary processes from real world, such as climate-related processes or simplifying simulations of physical processes.

Prospects for further research are to study different approaches to use as a decoder network, replacing variational autoencoders with different stochastic generative models.

ACKNOWLEDGEMENTS

The author of the following research wants to appreciate his scientific advisor, Nadezhda I. Nedashkovskaya, – Dr. Tech. Sc., Associate Professor at the Department of Mathematical Methods of System Analysis, Institute for Applied Systems Analysis at National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

REFERENCES

1. Bidyuk P. I., Romanenko V. D., Timoshchuk O. L. Time series analysis. Kyiv, Polytechnika, 2013, 230 p. (In Ukrainian)
2. Parfenenko Y. V., Shendryk V. V., Kholiavka Y. P., Pavlenko P. M. Comparison of short-term forecasting methods of electricity consumption in microgrids, *Radio Electronics, Computer Science, Control*, 2023, № 1, pp. 14–23. DOI: https://doi.org/10.15588/1607-3274-2023-1-2
3. Terence C. M. Chapter 3 – ARMA Models for Stationary Time Series, *Applied Time Series Analysis. A Practical Guide to Modeling and Forecasting*. Cambridge, Academic Press, 2019, Ch. 3, pp. 31–56. DOI: https://doi.org/10.1016/B978-0-12-813117-6.00003-X.
4. Terence. C. M. Chapter 4 – ARIMA Models for Nonstationary Time Series, *Applied Time Series Analysis. A Practical Guide to Modeling and Forecasting*. Cambridge, Academic Press, 2019, Ch. 4, pp. 57–69. DOI: https://doi.org/10.1016/B978-0-12-813117-6.00004-1.
5. Charles A., Darné O. The accuracy of asymmetric GARCH model estimation, *International Economics*, 2019, Vol. 157, pp. 179–202. DOI: https://doi.org/10.1016/j.inteco.2018.11.001
6. Bidyuk P., Prosyankina-Zharova T., Terentiev O. Modeling nonlinear nonstationary processes in macroeconomy and finances, *Advances in Computer Science for Engineering and Education*, 2019, Vol. 754, pp. 735–745. DOI: https://doi.org/10.1007/978-3-319-91008-6_72.

DOI 10.15588/1607-3274-2023-4-7

7. Kumar A. S., Anandarao S. Volatility spillover in crypto-currency markets: Some evidences from GARCH and wavelet analysis, *Physica A: Statistical Mechanics and its Applications*, 2019, Vol. 524, pp. 448–458. DOI: https://doi.org/10.1016/j.physa.2019.04.154.
8. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol: O'Reilly Media Inc., 2017, 760 p.
9. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, The MIT Press, 2016, 802 p.
10. Whittle P. Hypothesis Testing in Time Series Analysis. Stockholm, Almquist and Wicksell, 1951, 187 p.
11. Box G. E.P., Jenkins G. M. Time Series Analysis: Forecasting and Control. San-Francisco, Holden-Day, 1976, 575 p.
12. Hochreiter S., Schmidhuber J. Long short-term memory, Neural computation, 1997, Vol. 9, № 8, pp. 1735–1780.
13. Cho K., Merrienboer B. van, Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP 2014: Conference on Empirical Methods in Natural Language Processing, Doha, 25–29 October 2014: proceedings.* Doha, Association for Computational Linguistics, 2014, pp. 1724–1734. DOI: https://doi.org/10.48550/arXiv.1406.1078
14. Chen R. T.Q., Rubanova Y., Bettencourt J., Duvenaud D. Neural ordinary differential equations [Electronic resource]. *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, Vancouver, 3–8 December 2018: proceedings*. Access mode: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf
15. Lu J., Deng K., Zhang X., Liu G., Guan Y. Neural-ODE for pharmacokinetics modeling and its advantage to alternative machine learning models in predicting new dosing regimens, *Science*, 2021, Vol. 24, Issue 7, pp. 1–13. DOI: https://doi.org/10.1016/j.isci.2021.102804.
16. De Brouwer E., Simm J., Arany A., Moreau Y. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series [Electronic resource], *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, Vancouver, 8–14 December 2019: proceedings*. Access mode: https://proceedings.neurips.cc/paper_files/paper/2019/file/455cb2657aaa59e32fad80cb0b65b9dc-Paper.pdf

Received 06.09.2023.
Accepted 15.11.2023.

УДК 004.93

НЕЙРОННІ ЗВИЧАЙНІ ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ ДЛЯ РЕКОНСТРУКЦІЇ ЧАСОВИХ РЯДІВ

Андросов Д. В. – аспірант Інституту прикладного системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу реконструкції нестаціонарних часових рядів на основі моделей кодувальник-декодувальник за допомогою нейронних звичайних диференціальних рівнянь. Об'єктом дослідження є задача відновлення та прогнозування нестаціонарних часовиї рядів та процесів в неперевному часі. Мета роботи – синтез моделі на основі архітектури кодувальник-декодувальник та з використанням моделей типу нейронних звичайних диференційних рівнянь для реконструкції часових рядів по зашумленими, нерівномірно розподіленими у час, вхідними сигналами.

Метод. Запропоновано метод, що реалізує архітектуру кодувальника-декодувальника та аппарат штучних нейронних мереж з розв'язанням диференціальних рівнянь у латентному просторі. Було встановлено, що даний підхід демонструє високу ефективність та якість прогнозів при вирішенні задачі реконструкції часових рядів по зашумленим вхідним сигналам з випадковими інтервалами між сигналами. Запропонована модель варіаційного автокодувальника на з використанням апарату нейронних мереж була протестована на синтетичних нестаціонарних даних з додаваням білим шумом і семплінгом з випадковими інтервалами між кожним сигналом.

Результати. Розроблені показники реалізовані програмно і досліджені при вирішенні задачі реконструкції нестацонарного ряду з сезонністю.

Висновки. Проведені експерименти підтвердили, що запропонована модель ефективно вирішує задану задачу і рекомендується застосовувати її для вирішення реальних завдань, що вимагають реконструкції динаміки нестаціонарних процесів. Перспективи включають в себе подальші дослідження різних архітектур нейронних мереж, окрім рекурентних нейронних мереж та архітектур автокодувальників. Зокрема пропонується використовувати інші підходи генеративного нейромережевого моделювання, як генеративно-змагальні мережі у контексті відновлення структури часового ряду

КЛЮЧОВІ СЛОВА: нейронні звичайні диференціальні рівняння, глибокі нейронні мережі, варіаційні автокодувальники, рекурентні нейронні мережі, мережі довгострокової короткої пам'яті.

ЛІТЕРАТУРА

1. Bidyuk P. I. Time series analysis / P. I. Bidyuk, V. D. Romanenko, O. L. Timoshchuk. – Kyiv : Polytechnika, 2013. – 230 p. (In Ukrainian)
2. Comparison of short-term forecasting methods of electricity consumption in microgrids / [Y. V. Parfenenko, V. V. Shendryk, Y. P. Kholiavka, P. M. Pavlenko] // Radio Electronics, Computer Science, Control. – 2023. – № 1. – P. 14–23. DOI: https://doi.org/10.15588/1607-3274-2023-1-2
3. Terence. C. M. Chapter 3 - ARMA Models for Stationary Time Series / C. M. Terence // Applied Time Series Analysis. A Practical Guide to Modeling and Forecasting. – Cambridge: Academic Press, 2019. – Ch. 3. – P. 31–56. DOI: https://doi.org/10.1016/B978-0-12-813117-6.00003-X.
4. Terence. C. M. Chapter 4 – ARIMA Models for Nonstationary Time Series / C. M. Terence // Applied Time Series Analysis. A Practical Guide to Modeling and Forecasting. – Cambridge : Academic Press, 2019. – Ch. 4. – P. 57–69. DOI: https://doi.org/10.1016/B978-0-12-813117-6.00004-1.

DOI 10.15588/1607-3274-2023-4-7

5. Charles A. The accuracy of asymmetric GARCH model estimation. / A. Charles, O. Darné // International Economics. – 2019. – Vol. 157. – P. 179–202. DOI: https://doi.org/10.1016/j.inteco.2018.11.001
6. Bidyuk P. Modeling nonlinear nonstationary processes in macroeconomy and finances. / P. Bidyuk, T. Prosyankina-Zharova, O. Terentiev // Advances in Computer Science for Engineering and Education. – 2019. – Vol. 754. – P. 735–745. DOI: https://doi.org/10.1007/978-3-319-91008-6_72.
7. Kumar A. S. Volatility spillover in crypto-currency markets: Some evidences from GARCH and wavelet analysis. / A. S. Kumar, S. Anandarao // Physica A: Statistical Mechanics and its Applications. – 2019. – Vol. 524. – P. 448–458. DOI: https://doi.org/10.1016/j.physa.2019.04.154.
8. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow / A. Geron. – Sebastopol : O'Reilly Media Inc., 2017. – 760 p.
9. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge : The MIT Press, 2016. – 802 p.
10. Whittle P. Hypothesis Testing in Time Series Analysis / P. Whittle. – Stockholm : Almquist and Wicksell, 1951. – 187 p.
11. Box G. E.P., Jenkins G. M. Time Series Analysis: Forecasting and Control / G. E. P. Box, G. M. Jenkins. – San-Francisco: Holden-Day, 1976. – 575 p.
12. Hochreiter S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computation. – 1997. – Vol. 9, № 8. – P. 1735–1780.
13. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. / [K. Cho, B. van Merrienboer, C. Gulcehre et al.] // EMNLP 2014: Conference on Empirical Methods in Natural Language Processing, Doha, 25–29 October 2014: proceedings. – Doha: Association for Computational Linguistics, 2014. – P. 1724–1734. DOI: https://doi.org/10.48550/arXiv.1406.1078
14. Neural ordinary differential equations [Electronic resource] / [R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, // Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, Vancouver, 3–8 December 2018: proceedings. Access mode: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf
15. Neural-ODE for pharmacokinetics modeling and its advantage to alternative machine learning models in predicting new dosing regimens / [J. Lu, K. Deng, X. Zhang et al.] // Science. – 2021. – Vol. 24, Issue 7. – P. 1–13. DOI: https://doi.org/10.1016/j.isci.2021.102804.
16. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series [Electronic resource] / [E. De Brouwer, J. Simm, A. Arany, Y. Moreau] // Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, Vancouver, 8–14 December 2019: proceedings. Access mode: https://proceedings.neurips.cc/paper_files/paper/2019/file/455cb2657aaa59e32fad80cb0b65b9dc-Paper.pdf

DOI 10.15588/1607-3274-2023-4-7