TIME SERIES PREDICTION WITH MACHINE LEARNING TECHNIQUES. PART I – GENERAL INTRODUCTION TO TIME SERIES AND MACHINE LEARNING TECHNIQUES USED FOR FORECASTING

BOGDAN DIACONU, *University "Constantin Brâncuși" of Tg-Jiu*
LUCICA ANGHELESCU, *University "Constantin Brâncuși" of Tg-Jiu*
MIHAI CRUCERU, *University "Constantin Brâncuși" of Tg-Jiu*
CRISTINEL RACOCEANU, *University "Constantin Brâncuși" of Tg-Jiu*
ADRIANA TUDORACHE, *University "Constantin Brâncuși" of Tg-Jiu*
MARIUS EREMIA VLAICU POPA, *Complexul Energetic Oltenia*

ABSTRACT: Forecasting is a major area in economy, finances, engineering and many more. The paper introduces several machine learning models with applicability in time series forecasting. First, an introductory discussion on types of time series is given and in the second part of the paper Recurrent Neural Networks are presented with a focus on Long Short Term Memory RNNs, which are a recently developed, special type of recurrent neural networks, showing good performance in predicting sequences.

Keyords: - *Time series; Machine learning; Recurrent Neural Networks; Forecasting*

1. INTRODUCTION

Time series are discrete data sequences of the form taken at successive usually equally spaced points in time. Time series analysis is employed in applications such as economic forecasting, electricity consumption, stock market analysis, process and quality control, workload projections and many other areas. Time series can be classified as:

- Univariate, when one singular value corresponds to each time step. Example: electricity consumption during a given period of time
- Multivariate, when two or more values correspond to one time step. This kind of time series can be useful in revealing correlation in data. For example, plotting the average temperature against time over a long period shows an increasing trend. Adding the CO_2 concentration on the same plot demonstrates a certain degree of correlation

Analysis of time series is a complex field with the most interesting and widely used application being forecasting. Forecasting means predicting future values of a time series based on past values. Sometimes it is necessary to perform the inverse process – i.e. to forecast back in the past (process called imputation) to determine how the time series reached the present value.

Detection of patterns in signal processing is another interesting application of time series analysis. The classic example is analysis the spectrum of an audio signal in order to isolate the individual words.

1.1. Trends (Figure 1).

Time series come in many forms and sizes. There are a number of very common patterns, as follows:

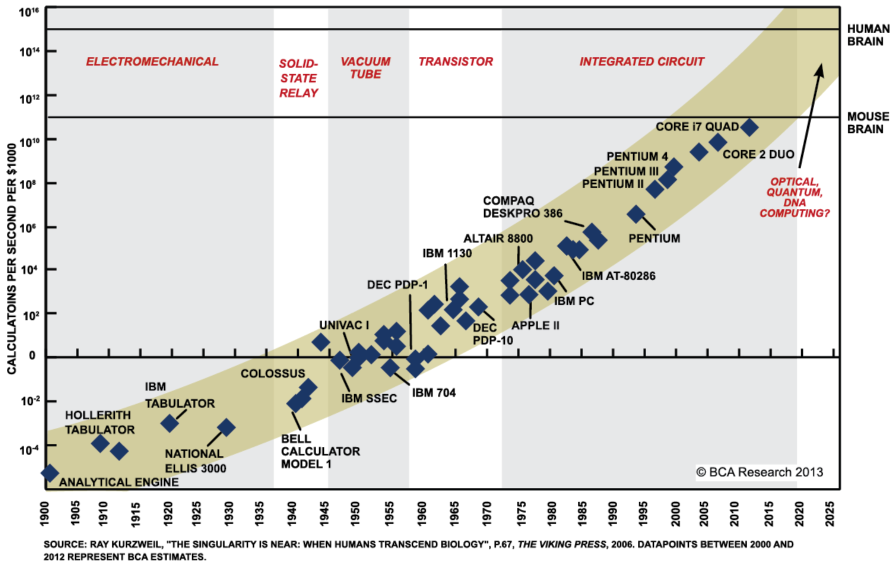

Figure 1. Typical example of time series exhibiting a trend [1]

A trend can be either positive or negative, linear or non-linear. In Figure 1, the well-known Moore law [1] (stating that the computing power approximately doubles every two years) is presented. Moore law is a typical example of time series with a positive trend.

1.2. Seasonality. A seasonal pattern occurs when the time series values are influenced by factors such as time of the year or day of the week. A typical example of seasonal time series is the household electricity consumption over a year. Higher values can be observed during summer due to air conditioning appliances.

1.3. Cyclic. When a time series values exhibit data which increase or decrease with variable frequency it can be qualified as cyclic. Such fluctuations are usually caused by economic conditions and are sometimes related to a business cycle [2]. It is important to distinguish between seasonal and cyclic time series. If the fluctuations frequency is variable then they are cyclic; if the frequency does not vary and it is associated with some aspect of the calendar, then the time series pattern is seasonal.

1.4. Irregularity. Most time series exhibit to different degrees trends, seasonality and cyclicity. However, a significant share of the time series are completely random (Figure 2 – Google . For such time series it is almost impossible to make accurate predictions.

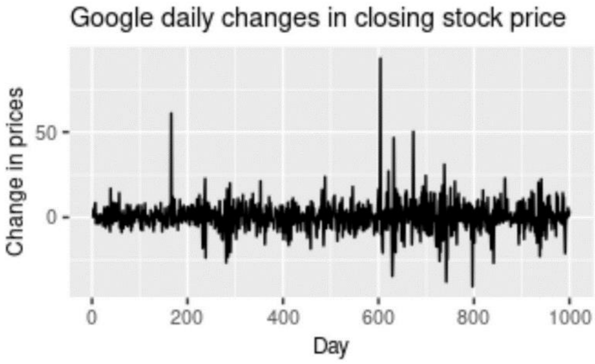

Figure 2. Irregular time series [3]

2. MACHINE LEARNING TECHNIQUES IN TIME SERIES FORECASTING

Artificial Neural Networks are efficient learning models that are capable of achieving state-of-the-art results in a wide range of supervised and unsupervised machine learning tasks.

Recurrent neural networks (RNNs) are connectionist models with the ability to selectively pass information across sequence steps, while processing sequential data one element at a time [4]. In 1982, Hopfield introduced a family of recurrent neural networks that have pattern recognition capabilities [5]

2.1. Recurrent Neural Networks

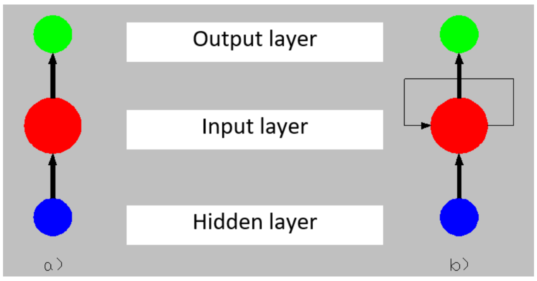

Figure 3. Feed-forward Artificial Neural Networks (a) and Recurrent Neural Networks

Standard feed-forward artificial neural networks (Figure 3.a) do not have a memory feature, hence they cannot process sequential data, as it is the case of time series. Recurrent Neural Networks (Figure 3.b) add a memory mechanism in the form of a loop. This loop allows information to pass from one time step to the next. This piece of information is contained in the hidden state, which is a representation of previous inputs.

The operation principle of a RNN can be described as follows: (1) the network layers weights and biases and the initial hidden state are initialized using random values or special algorithms that ensure convergence. The shape and dimension of the hidden state depend on the shape and dimension of the RNN. (2) The input data is passed to the RNN in a sequential manner and the RNN returns the output and a modified hidden state. The input data continues to be passed to the RNN until the input dataset is fed completely to the RNN. Last layer – the output layer – of the network is usually a standard feed-forward layer. The output of the recurrent layers is eventually fed to the output layer, which delivers the actual output of the network. The key difference between a standard artificial neural network and a RNN is the way the input data is fed to the network.

The training process of a RNN follows the same general principles as for standard neural networks:

1. A forward pass is executed and a prediction is made with the network layer weights/biases from the initialization stage.
2. The predicted values are compared to the ground truth using a loss function, which can be mean arithmetic error, mean squared error, etc. The loss function generates an error value which is an estimate of how close the prediction of the RNN was compared to the real values.
3. The error value is used to perform a back propagation, which calculates the gradients for each weight and bias in the network layers. The gradient is the value used to adjust the networks internal weights, allowing the network to learn. The higher the gradient, the larger the adjustments and the opposite.

This training process has an intrinsic problem. During the back propagation process, each node weight gradient depends on the gradients from the layer before it. The back propagation starts from the output -loss function value – and progresses to the input layers (hence the name "back propagation"). If the adjustments to the layers before it is small, then adjustments to the current layer will be even smaller.

That causes gradients to decrease much faster than linear as it back propagates towards the input layer. The earlier layers fail to update the weights and biases, eventually being unable to learn. This phenomenon is known as the vanishing gradient problem. The vanishing gradient problem renders the RNN unable to learn the long-range dependencies across time steps. For this reason, the RNNs are called short-term memory neural networks.

2.2. Long Short-Term Memory Neural Networks

In order to compensate for the short-term memory problem, a specialized RNN was proposed by Hochreiter and Schmidhuber [6]. This special RNN is called Long Short-Term Memory (LSTM). LSTMs essentially function just like RNN's but they're capable of learning long-term dependencies using internal mechanisms called "gates." These gates consist of tensor operations that can learn what information to add or remove to the hidden state. Because of this ability, short-term memory effect is circumvented.

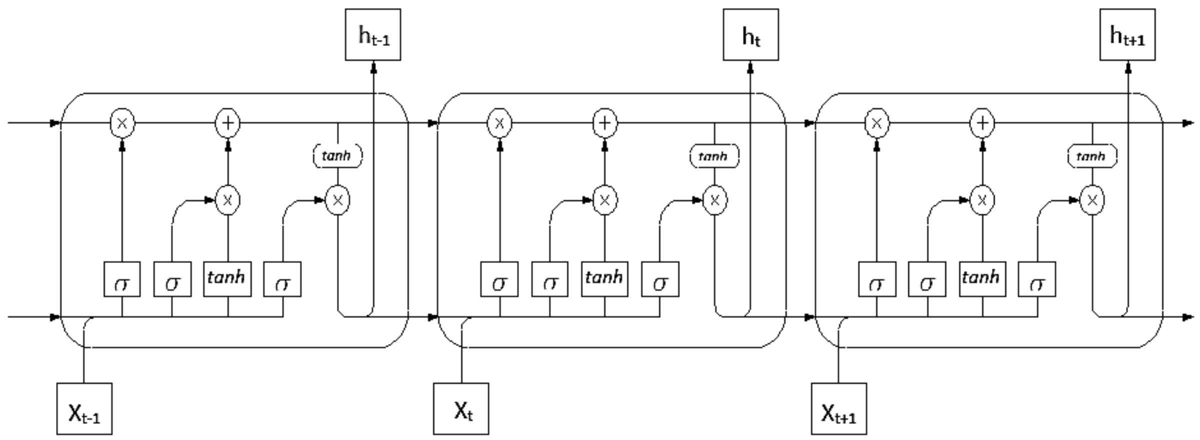

Figure 4. The structure of a LSTM RNN

LSTMs have a chain-like structure; instead of a single neural network layer, LSTMs consist of four layers, interacting in a special way, as depicted in Figure 4. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain through time, with only some minor linear interactions. It's very easy for information to just flow along it unchanged. In this way, a long-term memory effect is achieved. The LSTM is able to remove or add information to the cell state, in a way controlled by internal structures called gates.

Gates are a way to conditionally allow the information to pass through. They consist of a sigmoid neural net layer and a element-wise multiplication operation (dot product). The sigmoid layer outputs real values between zero and one, describing how much of each component should pass. A value of zero means "let nothing through," while a value of one means "let everything through". An LSTM has three sigma layers to protect and control the cell state.

LSTM operation principle

The first step in the LSTM operation is to decide what information will be eliminated from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at and and outputs a number between 0 and 1 (sigmoid function) for each number in the cell state . The value 1 represents "completely keep this" while 0 represents "completely eliminate."

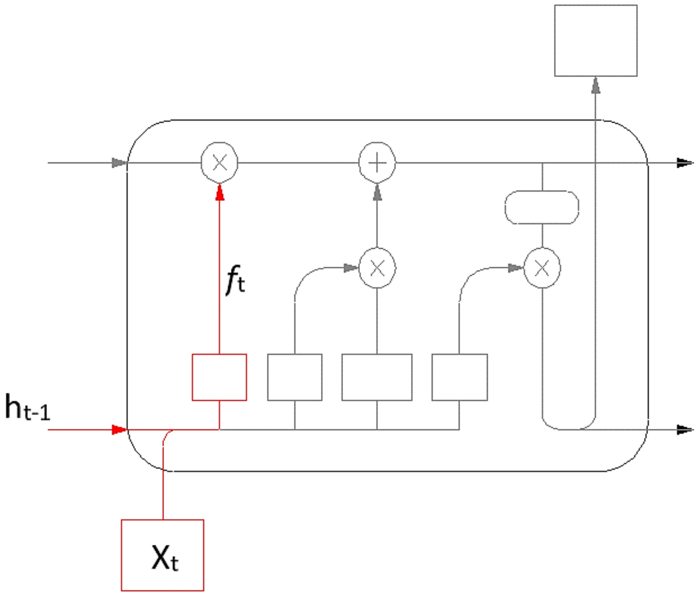

Figure 5. Operation of the first sigmoid layer of the LSTM

The output of the first sigmoid layer is given by the following equation:

where:

- is the weight matrix for the first sigmoid layer
- is the bias vector for the same layer

The next step is to decide what new information will be stored in the cell state. This step has two parts. First, a sigmoid layer (Figure 6) called the "input gate layer" decides what values will be updated. Next, a *tanh* layer creates a vector of new candidate values, , that could be added to the state. An intermediate cell state is thus generated. In the next step, these two values will be used to create an update to the state. The equations for this step are as follows:

Next, the old cell state, , is updated into the new cell state . The old state value is multiplied by , forgetting the information selected in the previous step earlier. Then it is added to . This is the new candidate values, scaled by how much it was decided to update each state value.

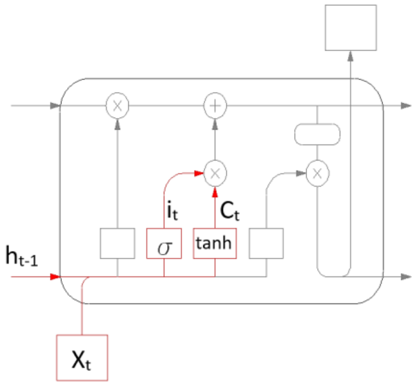

Figure 6. Generating an intermediate cell state

Updating the cell state (Figure 7) is achieved through the following equation:

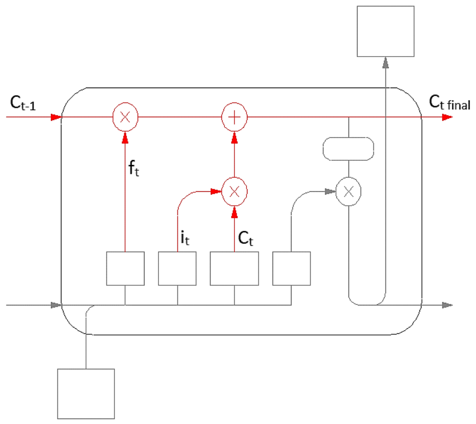

Figure 7. Updating the cell state

In the final step, the output of the cell is decided. This output will be based on the cell state, but it will be a filtered version. First, a sigmoid layer is applied which decides what parts of the cell state will pass through the output. Then, the cell state will be passed through a tanh layer (to force the values to be between -1 and 1) and multiply it by the output of the sigmoid gate.

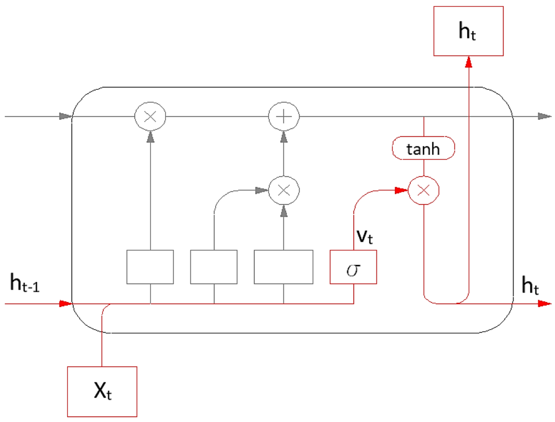

Figure 8. The calculation of the output

The output of the cell is determined based on the schematic diagram in Figure 8. The equations corresponding to the diagram represented in Figure 8 are as follows:

CONCLUSIONS

Forecasting is required in many situations and many business decisions are nowadays based on forecasting. Whatever the applications and the time horizon required, it is important to consider several key points:

- Some things are easier to forecast than others
- There are things that cannot be forecasted
- The accuracy of any forecast depends mainly on the following factors: how well the influencing factors are understood; the amount of data available; whether the forecasting process can influence the future evolution of the things being forecasted.

RNNs are a subtype of ANNs used for time series processing and especially for forecasting. A more advanced RNN subtype is represented by the LSTM, which are capable of circumventing the problem of vanishing gradient. An application of RNNs and LSTMs to a forecasting problem will be presented in the second part of this paper.

REFERENCES

[1] https://www.extremetech.com/extreme/210872-extremetech-explains-what-is-moores-law. Accessed on October 10th, 2021

[2] Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on October 10th, 2021

[3] https://blog.paperspace.com/forecasting-stock-prices-using-deep-learning. Accessed on October 10th, 2021

[4] Lipton, Z.C., Berkovitz, J., A Critical Review of Recurrent Neural Networks for Sequence Learning, arXiv:1506.00019v4 [cs.LG] 17 Oct 2015

[5] Hopfield, J., Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79 (8):2554–2558, 1982

[6] Hochreiter, S., Schmidhuber, J., Long short-term memory. Neural Computation, 9(8):1735–1780, 1997

[7] Racoceanu Cristinel, Exergy analysis of power equipment – Part I: General exergy balance, Annals of the" Contantin Brancusi " University of Târgu Jiu, Engineering Series, Vol3/2020. pag.61-64, 2537-530X (online), ISSN-L 1842-4856.

[8] Racoceanu Cristinel, Exergy analysis of power equipment – Part II: exergy efficiency of the combustion process, Annals of the" Contantin Brancusi" University of Târgu Jiu, Engineering Series, Vol3/2020. pag.65-68, 2537-530X (online), ISSN-L 1842-4856.

[9] Racoceanu Cristinel, Study on the impact of lignite combustion on air quality in the turceni thermoelectric power plant, Annals of the „Constantin Brancusi" University of Targu Jiu, Engineering Series , No. 1/2017, pag.25-29, ISSN 1842 – 4856.

[10] Racoceanu., C. Study concerning the application of the nonpolluting ash exhaust technology to the Benson steam generators, Annals of the "Constantin Brâncuși" University of Târgu-Jiu. Engineering Series, 4 (2019), 100-104 [ISSN: 1842-4856].