UNITED NATIONS CONFERENCE ON TRADE AND DEVELOPMENT

March 2021

UNCTAD Research Paper No. 62
UNCTAD/SER.RP/2021/5

Daniel Hopp
Associate Statistician
Division on Globalisation and Development Strategies, UNCTAD
daniel.hopp@unctad.org

Economic Nowcasting with Long Short-Term Memory Artificial Neural Networks (LSTM)

Abstract

Artificial neural networks (ANNs) have been the catalyst to numerous advances in a variety of fields and disciplines in recent years. Their impact on economics, however, has been comparatively muted. One type of ANN, the long short-term memory network (LSTM), is particularly well-suited to deal with economic time-series. Here, the architecture's performance and characteristics are evaluated in comparison with the dynamic factor model (DFM), currently a popular choice in the field of economic nowcasting. LSTMs are found to produce superior results to DFMs in the nowcasting of three separate variables; global merchandise export values and volumes, and global services exports. Further advantages include their ability to handle large numbers of input features in a variety of time frequencies. A disadvantage is the inability to ascribe contributions of input features to model outputs, common to all ANNs. In order to facilitate continued applied research of the methodology by avoiding the need for any knowledge of deep-learning libraries, an accompanying Python library was developed using PyTorch: https://pypi.org/project/nowcast-lstm/.

Key words: Nowcasting, Economic forecast, Neural networks, Machine learning, Python, nowcast_lstm

Contents

Acknowledgements

I would like to thank Fernando Cantú, Anu Peltola, Onno Hoffmeister, Henri Luomaranta and Stephen MacFeely for their valuable comments and support.

1. Introduction

A defining feature of the 21st century so far has been the explosion in both the volumes and varieties of data generated and stored (Domo, 2017). Almost every industry and aspect of life has been affected by this "data revolution" (Einav and Levin, 2014), (MacFeely, 2020). Simultaneously, rapid advancements in machine learning methods have been made, spurred on in part by the need for novel methods to analyze these new data quantities. Perhaps no methodology has gained greater prominence than the artificial neural network (ANN). ANNs are the engine behind tremendous leaps in fields as disparate as machine translation, image recognition, recommendation engines and even self-driving vehicles. Yet to date, their impact in the field of economic policy has been largely muted or exploratory in nature (Falat and Pancikova, 2015).

This is not to suggest that economic data have been immune to the transformative forces of the data revolution. Quite the opposite in fact, as classical economic data series from national statistical offices (NSO) and other organizations can now be fortified by alternative data sources like never before, helping to provide glimpses into the developments of the global economy with unparalleled granularity and timeliness (Glaeser et al., 2017). The COVID-19 pandemic and ensuing economic crisis showcased this, with analysts and policy-makers gaining insight to the rapidly evolving economic situation from such alternative data sources as Google mobility data (Yilmazkuday, 2021), booking information from dining apps (OpenTable, 2021) and transaction data from e-commerce sites (Statista, 2021), among many others.

The availability of a broad range of novel, timely indicators should ostensibly have led to significant advances in the field of economic nowcasting, where real-time macroeconomic variables that may be published with a significant lag are estimated based on an array of more timely indicators (Banbura et al., 2010), (Giannone et al., 2008). In reality, the field has not experienced the degree of progress seen in other fields, such as image recognition, in the past 10 years. A large factor in this relative stagnation is the fact that many of the issues facing nowcasting are not addressed by more data alone. Issues such as multicollinearity, missing data, mixed-frequency data and varying publication dates are sometimes even exacerbated by the addition of variables (Porshakov et al., 2016). As such, advancements in the field come from a combination of both new data and methodological developments. Dynamic factor models (DFM) in particular have been found to address many of the data issues inherent in nowcasting (Stock and Watson, 2002), and have been applied successfully in applications such as nowcasting economic growth in 32 countries (Matheson, 2011), nowcasting German economic activity (Marcellino and Schumacher, 2010) and nowcasting Canadian GDP growth (Chernis and Sekkel, 2017). The basic premise of DFMs is that one or more latent factors dictates the movement of many different variables, each with an idiosyncratic component in relation to the factor(s). With historical data, the factor(s) can be estimated from the variables. Subsequently, even in future periods where not all data are complete, the factor(s) can still be estimated and used to generate forecasts for variables that are not yet published, as each variable's relation to the factor(s) has already been estimated.

Despite DFMs' strengths in addressing a wide swath of nowcasting's data issues, the impressive performance of ANNs in other domains raises the question of their performance in nowcasting. ANNs have been applied to economic nowcasting in the past (Loermann and Maas, 2019). However, due to the time-series nature of many economic nowcasting applications, the long short-term memory (LSTM) architecture is better suited to the problem than the traditional feedforward architecture explored in Loermann and

Maas (2019). LSTMs are an extension of recurrent neural network (RNN) architecture, which introduces a temporal component to ANNs. LSTMs have been used to nowcast meteorological events (Shi et al., 2015) as well as GDP (Kurihara and Fukushima, 2019).

However, use of LSTMs in nowcasting economic variables remains in its infancy, perhaps partly due to high barriers to their implementation. Many common deep learning frameworks, including Keras and PyTorch, include provisions for LSTMs. However, the implementations are general and require knowledge of the frameworks to successfully implement. As such, a Python library focused on economic nowcasting has been published alongside this paper, available for install on PyPi: https://pypi.org/project/nowcast-lstm/. Hopefully, a more accessible library will help stimulate interest and expand the applications of these powerful models.

The remainder of this paper is structured as follows: the next section will further explain nowcasting and its challenges; section three will explore LSTMs in more detail; section four will examine the LSTM's empirical performance compared with DFMs in nowcasting three series: global merchandise trade exports expressed in both values and volumes and global services exports; section five will introduce and explain the accompanying Python library; the final section will conclude and examine areas of future research.

2. Exposition of nowcasting problem

Nowcasting, a portmanteau of "now" and "forecast", is the estimation of the current, or near to it either forwards or backwards in time, state of a target variable using information that is available in a timelier manner. Keith Browning coined the term in 1981 (WMO, 2017) to describe forecasting the weather in the very near future based on its current state. The concept and term remained in the meteorological domain for years before being adopted into the economic literature in the 2000s. The concept of real-time estimates of the macroeconomic situation predates the adoption of the nowcasting terminology, as evidenced by Mariano and Murasawa (2003). However, Giannone et al. (2005) explicitly referenced the term "nowcasting" in its title and the term became commonplace in subsequent years, being applied for example to Portuguese GDP in 2007 (Morgado et al., 2007) and to Euro area economic activity in 2009 (Giannone et al., 2009). The 2010s saw a wealth of papers examining the topic both for a range of target variables as well as with a range of methodologies and models. Targets most often included GDP (Rossiter, 2010), (Bok et al., 2018), and trade (Cantú, 2018), (Guichard and Rusticelli, 2011). Common methodologies include dynamic factor models (DFM) (Guichard and Rusticelli, 2011), (Antolin-Diaz et al., 2020), mixed data sampling (MIDAS) (Kuzin et al., 2009), (Marcellino and Schumacher, 2010) and mixed-frequency vector autoregression (VAR) (Kuzin et al., 2009), among others. Nowcasting also has relevance in the context of the 2030 Agenda for Sustainable Development (UN, 2015). Many indicators face issues in terms of data quality, availability, timeliness, or all three. As such, nowcasting is being discussed as a possible method of ensuring maximum coverage in terms of indicators (UNSD, 2020).

Economic nowcasting is generally confronted with three main issues regarding data. The first is mixed frequency data, or when all independent variables and the dependent variable are not recorded with the same periodicity. This occurs frequently in economic data, for instance when trying to nowcast a quarterly target variable, such as GDP growth, using monthly indicators. Or estimating a yearly target variable with a mixture of monthly and quarterly variables. The second is the heterogeneous publication schedules of independent variables, frequently referred to as "ragged-edges". Any nowcasting methodology must provide provisions for incomplete or partially complete data, as varying availability of latest data is the reality of most datasets of economic series. Finally, there is the issue of the “curse of dimensionality”, which renders many classical econometric methods less effective in the nowcasting context and hinders the application of “big data” to the field (Buono et al., 2017). The problem stems from the nature of many economic variables, where they may have few observations relative to the potential pool of explanatory variables or features. The quarterly target series for UNCTAD’s own nowcasts for global merchandise trade, for instance, only began in 2005 (Cantú, 2018). That leaves only 60 observations for training a model at the end of 2020. Meanwhile, many more than 60 potential independent variables can be conceived of to estimate a model of global merchandise trade.

The nowcasting methodologies previously mentioned address these problems in varying ways to achieve better predictions, and LSTMs are no different. The following section will provide background information on their network architecture as well as the characteristics that allow them to address the aforementioned nowcasting data problems.

3. ANN and LSTM models

3.1 ANNs and RNNs

ANNs are made up of various inter-connected layers composed of groups of nodes or neurons. The structure's conceptual similarity to the biological sort is the source of their name. Each of these nodes receives inputs either from the external data source, the "input layer", or from previous layers, “hidden” and “output” layers, the latter if the final output of the model. The output of a node is found by taking the weighted sum of all its inputs, the connections between individual nodes being the weights, and then running it through a non-linear activation function. In training, these weights are initially randomized, and when the data has passed through all layers of the network, an output is obtained, which is then run through a predefined cost function to assess performance. Then, using calculated gradients, or derivatives of the cost function, the network adjusts its weights to obtain an output with a smaller error, and the process is repeated.

This is of course an oversimplification of the process, however, there exists a vast literature outlining and explaining the methodology of ANNs for those desiring a deeper examination of their mathematics. Those interested can see Sazli (2006), Singh and Prajneshu (2008), or even Loermann and Maas (2019) for an explanation in the nowcasting context.

Traditional feedforward ANNs have a long history of use in time series forecasting (Kohzadi et al., 1996). These models, however, lack an explicit temporal aspect. This can be introduced to their architecture, resulting in recurrent neural networks (RNN) (Amidi and Amidi, 2019). As opposed to the unidirectional relationship between inputs and outputs in feedforward networks, RNNs introduce a feedback loop, where layer outputs can be fed back into the network (Stratos, 2020), (Dematos et al., 1996). This architecture makes RNNs well-suited to applications with a temporal aspect or flow, such as natural language processing or speech processing. However, due to vanishing gradients, RNNs tend to have a very “short” memory, limiting their usefulness in the nowcasting application (Grosse, 2017).

3.2 LSTMs

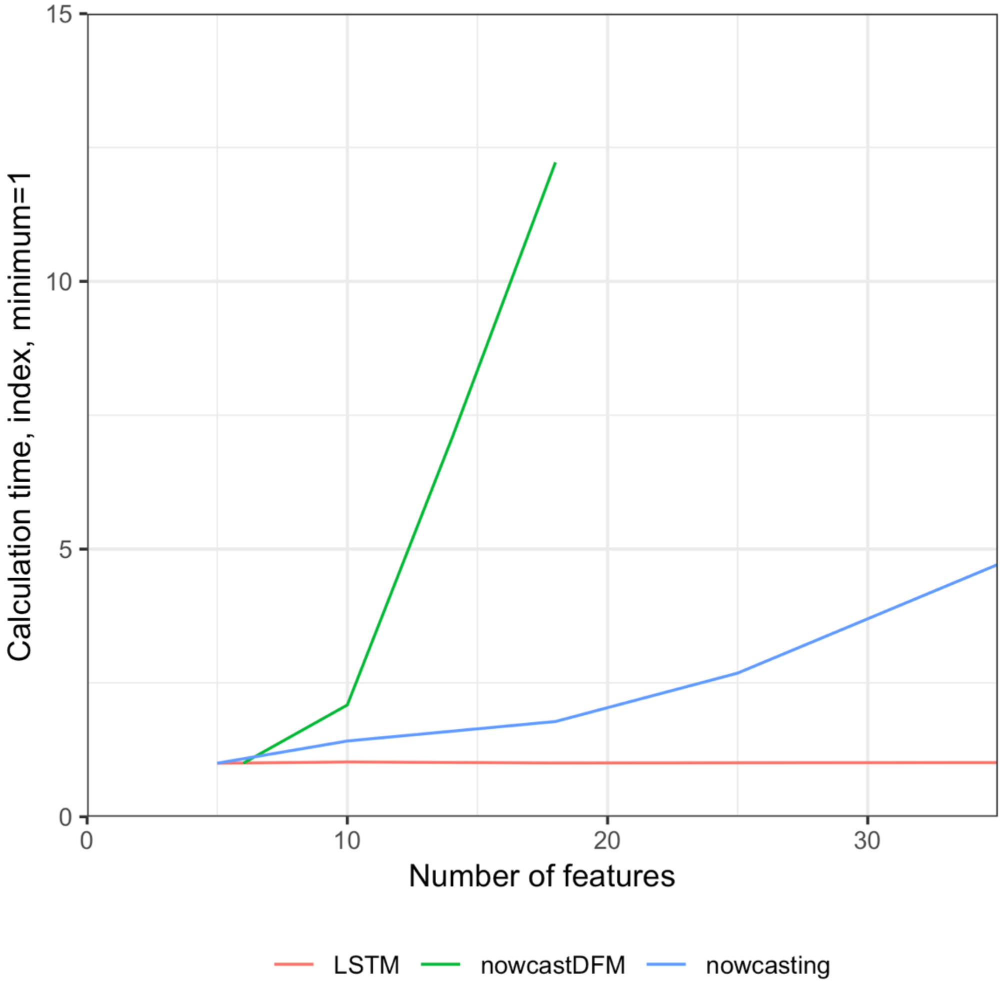
LSTMs introduce a memory cell and three gates: an input, output and forget gate (Chung et al., 2014). Crucially, this architecture then allows gradients to flow unchanged through the network, mitigating the vanishing gradient problem of RNNs and rendering them more suitable for application to the nowcasting problem. Data input to the LSTM network has the shape of *number of observations x number of timesteps x number of features*. The addition of the timesteps dimension allows the model to be trained on multiple lags of each variable, rather than just cotemporaneous observations.

LSTMs' ability to address the first common nowcasting data issue, mixed frequency data, stems from ANNs' ability to learn complex, non-linear relationships in data, a product of multiple neuron layers coupled with non-linear activation functions. More information on activation functions and their role in ANNs can be found in Sharma et al. (2020). As such, mixed frequency data can be fed to the network in the highest frequency available, with lower frequency data having missings at time periods where data are not published. These missing data can then be filled using a variety of approaches, including with the mean, the median, with values sampled from a distribution (Ennett et al., 2001), or with other more complex methods (Smieja et al., 2019). In the analysis performed in this paper, mean replacement was chosen and implemented in the accompanying Python library due to simplicity and empirical performance, as the network learns to recognize these in-between values as containing no novel information.

LSTMs are able to address the ragged-edges problem through no special mechanism other than standard missing-filling methods. These include using ARMA or VAR models to fill in ragged-edges (Kozlov et al., 2018), as well as using the mean or Kalman filters (Doz et al., 2011). The method chosen in the context of LSTM nowcasting can be considered a hyper-parameter to be tuned and tested empirically. At the time of writing, the Python library supports ARMA filling and any n-to-1 series transformation, e.g., mean, median, etc. ARMA filling was used in the analysis performed in this paper due to superior empirical performance compared with other methods.

The last major problem of nowcasting, the curse of dimensionality, is partially addressed by LSTMs' efficiency compared with other methods, i.e. their computation time scales very slowly with the number of variables (Hochreiter and Schmidhuber, 1997), as evidenced by Figure 1.

Figure 1. Development of model calculation time depending on number of features



The figure illustrates the development in computation time as more features are added for the LSTM model and two R implementations of the DFM; *nowcastDFM* (Hopp and Cantú, 2020) and *nowcasting* (Marcolino de Mattos, 2019). The DFMs' computation time scales exponentially while the LSTM's time remains nearly constant. The DFM and LSTM both scale linearly with the number of observations, however. The target variable was global merchandise exports in values and the various independent variables were a sample from the same pool presented in section 4. However, for this illustrative case, it is the computation times that are of interest, which display the same general patterns independent of the specific values of the input data.

As a result of this efficiency, a functional model can be trained with many more features than a DFM. Additionally, while standard methods of feature reduction such as principal component analysis (PCA) or Lasso can still be used on a data set intended for use with LSTM networks, their necessity is reduced due to ANNs' robustness to multicollinearity (De Veaux and Ungar, 1994).

Within the LSTM architecture, as in any ANN, there are many choices to be made regarding network architecture and hyper parameters. Some examples include the number of hidden states, the number of layers, the loss function and the optimizer, among many others. The logic for the defaults chosen for the Python library will be discussed in section 5.

4. Empirical analysis

4.1 Description of data and models

In order to assess the relative performance of LSTMs vs DFMs, three target variables were used: global merchandise exports in both value (WTO, 2020) and volume (UNCTAD, 2020a), and global services trade (UNCTAD, 2020a). These are the same series UNCTAD currently produces nowcasts for using DFMs (UNCTAD, 2020b: 20) and which were examined in a previous UNCTAD research paper (Cantú, 2018). The target series are all quarterly. A large pool of 116 mixed-frequency monthly and quarterly independent series was used to estimate each of the target series. These series are listed in Appendix 1, while more information on any individual series is available upon request. All series were converted to seasonally adjusted growth rates using the US Census Bureau's X13-ARIMA-SEATS methodology (USCB, 2017).

The DFM model used was the same examined in Cantú (2018) and currently in use by UNCTAD. In this model, the DFM is modeled in a state-space representation where it is assumed that the target and independent variables share a common factor as well as individual idiosyncratic components. The Kalman filter is then applied and maximum likelihood estimates of the parameters obtained. This is a common method of estimating DFMs and is explained in further detail in Bańbura and Rünstler (2011). The LSTM model used was that present in the *nowcast_lstm* Python library, which is further explained in section 5, using the average of 10 networks' output with basic hyper-parameter tuning of the number of training episodes or epochs, batch size, number of hidden states, and number of layers. The logic of averaging the output of more than one network to obtain predictions is discussed further in section 5, but see Stock and Watson (2004) for a discussion of forecast combination.

4.2 Modelling steps

Hyper-parameter tuning of the LSTM and model performance was evaluated using a training set dating from the second quarter of 2005 to the third quarter of 2016. The test set dated from the fourth quarter of 2016 to the fourth quarter of 2019.

A pool of independent variables was used to ensure the robustness of results, as either model could perform better on a single set of features due to chance. As such, the models' performance was evaluated by taking random samples of between five and 20 features, then fitting both an LSTM and DFM model on this same sample. Both methods'