Mapping GDP with Deep Feedforward Networks

Rafael Guimaraes*
00141311@ufrgs.br

This Version: May 13, 2019

PRELIMINARY AND INCOMPLETE - CHECK WITH AUTHOR FOR LATEST DRAFT.

Abstract

A relevant task in applied economics is finding metrics that represent the current state of the economy. There is vast information out there, but it is not usually straightforward to isolate the signal from the noise. The gross domestic product (GDP) is the most-watched current state indicator of the economy, but due to the intrinsic complexity of its calculation, the official GDP data are usually published quarterly and with lag that can take months. As a result, many coincident indicators with a wide range of techniques have been developed in economics. In this work, we adopted deep feedforward networks, known as universal approximators, as an approach for mapping the GDP. With the end-to-end strategy allowed by this technique, we propose to map a set of high-frequency variables to build indicators that fit the GDP on a cross-sectional approach. The performance of this strategy with Brazilian data indicates that this approach must belong to the economic toolset.

*Banco Central do Brasil and Universidade Federal do Rio Grande do Sul (UFRGS). The views expressed in this work are those of the author and do not necessarily reflect those of the Banco Central do Brasil and UFRGS or its members.

Contents

1 Introduction

Artificial Intelligence (AI) has gained considerable prominence due to performances in autonomous vehicles, intelligent robots, image and speech recognition, automatic translations, medical and law usage as well as beating champions in games like chess, Jeopardy, GO and poker (Makridakis, 2017). In Economics, the application of artificial neural networks, a Machine Learning (ML) method, is not new, and in a way, it has followed the phases of use in other areas: this has extended from the earliest attempts in the 1940s, followed by the rising expectations and the results in the 1960s, through the period of frustration in the 1970s, to the continuity of its use by a small group of researchers in the 1980s, and the resurgence in the 1990s (Stergiou and Siganos, 2011). Finally, from the beginning of the 21st century, significant progress has been observed in many areas, attracting attention and funding for research. This resurgence also occurs in economics, with researchers developing relevant works such as Varian and Choi (2009), Carriere-Swallow and Labbe (2010), Varian (2014), Ahonen (2015), Jean et al. (2016), Cook and Hall (2017), Hinds et al. (2017), and Gebru et al. (2017), to name a few.

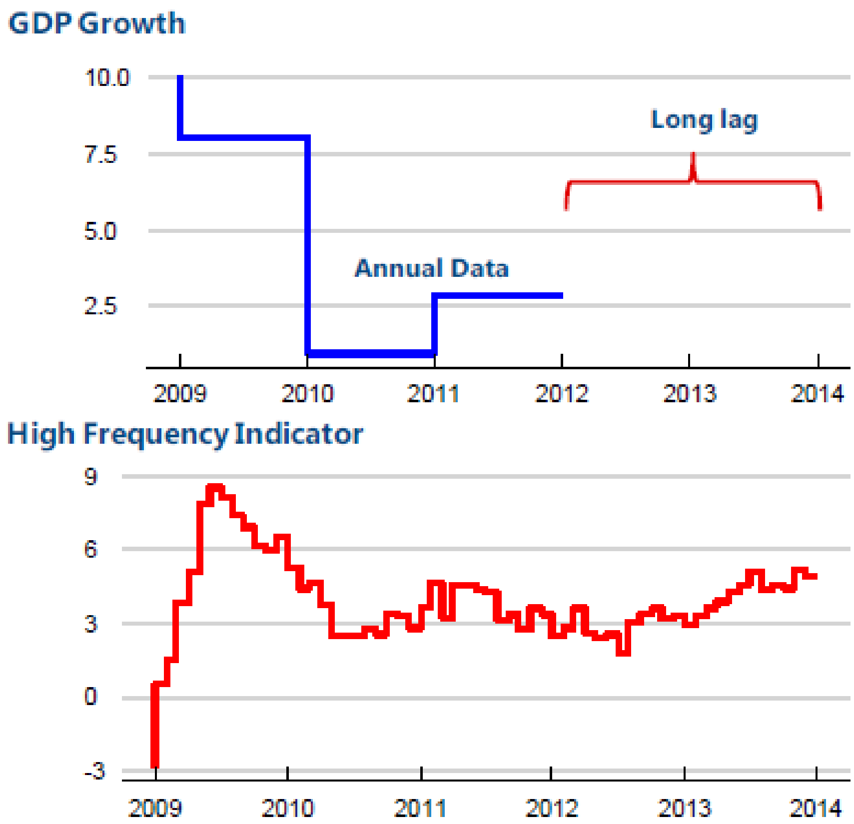


Figure 1: High frequency indicator - reproduced from (Tiffin, 2016, 4)

The official gross domestic product (GDP), this well-known metric customarily disclosed by government statistical institutes worldwide and accompanied by several segments, are usually published quarterly, sometimes yearly, and with lag that can reach several months, given the complexities involved in the calculations of national account systems. The relevance of anticipating this variable with high accuracy is indisputable, both for the monetary authority and for diverse economic agents, as asserted by (Hinds et al., 2017, 35): *What is the state of the [...] economy right now? Is it expanding or shrinking, and by how much? These are questions that official GDP statistics try to answer, but they take time to be published – can we obtain a quicker answer using other data sources?* For that purpose, economists have developed models to estimate economic activity in response to the regular flow of data. Since this process is an attempt to project the performance of the economy in near real time, these models are sometimes called *nowcasts*. Nowcasting as it is typically practiced is the process of using the latest economic information available to predict GDP growth, although, in principle, one could also produce nowcasts for inflation or other economically relevant series (Kliesen and McCracken, 2016). The term *nowcasting* is a contraction of *now* and *forecasting* and has become a standard activity for central banks (Tiffin, 2016). According to Giannone et al. (2010), nowcasting implies the prediction of the present, the very near future, and the very recent past. A successful nowcast will, thus, draw on real-time data to accurately forecast what *future* GDP releases will say about the *current* state of the economy (Tiffin, 2016).

The purpose of this article is to build specialized algorithms for mapping the low-frequency variable GDP from high-frequency variables, in a cross-sectional approach, as in Guimaraes et al. (????). Given enough labelled training datasets and suitable models, deep learning approaches can help humans establish mapping functions for operation convenience (Liu et al., 2017). For that end, we built deep feedforward networks (DFN) architectures with regression supervised learning algorithms. When evaluating an end-to-end strategy, we want to observe whether DFN models, known as universal approximators, are competitive when confronted with other models in the task of mapping a set of high-frequency variables to build a GDP indicator. However, two important distinctions must be made: i) we are not constructing a latent variable indicator to evaluate the state of the economy. Since the seminal article by Burns and Mitchell (1946), an extensive literature has developed along these lines, with emphasis on the dynamic factor models, e.g., Stock and Watson (2011) and Giannone et al. (2013); ii) we focus on the *quarterly* official data released by the government as a target, thus allowing us to apply an end-to-end DFN architecture with supervised learning to a high-frequency data already available as inputs, on a cross-sectional strategy to minimize the pitfalls related to time series approach[1].

The rest of the paper is organized as follows. In Section 2 we review some findings of the increase literature related to the application of machine learning in the field of economics. Section 3 provides a description of the dataset. In Section 4 we discuss the neural network models in more detail. Section 5 presents our empirical findings, and, in Section 6, we discuss some possible future directions for this research .

2 Literature review

Directing our attention to applied papers in economics, especially since the resurgence era, we begin with White (1988) which reports the results of a project using neural network modeling and learning techniques to search for and decode nonlinear regularities in asset price movements, based on the daily returns of the IBM common shares. In the same year, Dutta and Shekhar (1988) apply neural networks to predict the ratings of corporate bonds, finding evidence that it is a useful approach to generalization problems in such non-conservative domains. In 1991, Kuan and White (1991) published *Artificial Neural Networks: An Econometric Perspective*, a reference document for econometrics. According to Tkacz and Hu (1999), the use of neural networks in economics was still in its relative infancy, and the paper by Kuan and White (1991) was likely the definitive introduction of neural networks to the econometrics literature. In a business report, Schwartz (1992) states that in the world of finance, anything that provides even a slight edge over rivals can mean millions in extra profits. Thus, investment professionals are turning to gurus who offer exotic computer technologies such as neural networks and genetic algorithms. Forecasting corn futures, Kohzadi et al. (1995) found that the prediction error of

[1] In terms of time series forecasting, Makridakis et al. (2018) have shown that the accuracy of ML models is lower to that of statistical methods, claiming that more work is needed to improve such methods. According to them, we must realize that applying AI to forecasting is entirely different than doing so in games or in image and speech recognition and may require different, specialized algorithms to be successful.

a neural network model was between 18 and 40 percent lower than that of an ARIMA model, using different forecasting performance criteria. Tal and Nazareth (1995) reports that, in 1994, the Canadian Imperial Bank of Commerce replaced its index-based Leading Indicators with a neural network-based system and that the performance to that date had been very encouraging. Portugal (1995) provides an empirical comparative evaluation of the performance of the artificial neural network to the problem of economic time series forecast, performing exercises in the gross industrial output of the state of Rio Grande do Sul (Brazil) to find mixed results. Swanson and White (1997) applied neural networks to predict macroeconomic variables, contrasting different linear and nonlinear models using a wide array of out-of-sample forecasting performance indicators. Herbrich et al. (1998) provide an overview of existing economic applications of neural networks, distinguished in three types: classification of economic agents, time series prediction, and the modeling of boundedly rational agents. Tkacz and Hu (1999) pointed out that linear models are, in effect, constrained neural network models, forecast output growth using neural networks and compare the performance of such models with traditional linear specifications. They conclude that the best neural network models outperform the best linear models by between 15% and 19% of their data, implying that neural network models can be exploited for noticeable gains in forecast accuracy. According to them, the gains in forecast accuracy seem to originate from the ability of neural networks to capture asymmetric relationships. Blake (1999) makes straightforward analogies between artificial neural networks and models more familiar to economists. He also applies artificial neural network to model GDP growth using variables that could be expected to lead the growth cycle or predict likely future growth disturbances for six major European economies: France, Germany, Italy, the Netherlands, Spain and the UK.

More recently, the continuity of data available for research and the reduction of data processing costs have provided significant advances in the application of artificial intelligence techniques in several areas of knowledge, especially image recognition, natural language processing, bioinformatics, robotics, and finance (Goodfellow et al., 2016), as in economics. Varian and Choi (2009) found that simple seasonal AR models and fixed-effects models that include relevant Google Trends variables tend to outperform models that exclude these predictors. Analyzing data from Chile, Carriere-Swallow and Labbe (2010) presents evidence that the inclusion of information on Google search queries improves both the in- and out-of-sample accuracy of car sales models. Varian (2014) states that econometricians, statisticians, and data-mining specialists often seek insights that can be extracted from the data, and while the most common tool used for summarization is (linear) regression analysis, machine learning offers a set of tools that can usefully summarize various sorts of nonlinear relationships in the data. To overcome the problem of reliable data on economic livelihoods in the developing world, Jean et al. (2016) developed a method of estimating consumption expenditure and asset wealth using high-resolution satellite imagery. With a similar goal to ours, but applying other techniques of machine-learning named elastic net and random forest, Tiffin (2016) built a nowcasting indicator for Lebanon's GDP and achieved good results within an ensemble model. Figure 13 in Appendix reproduces a resume provided by Hinds et al. (2017) of nowcasting models then in use by selected monetary authorities.

Lastly but not least, Makridakis et al. (2018) are extremely positive about the enormous potential of ML methods for forecasting, noting that these methods have been proposed in the academic literature as alternatives to statistical ones for time series forecasting with scant evidence of their relative performance in terms of accuracy and computational requirements.

3 Dataset

By definition, GDP is the sum of final goods and services produced in an economy over a period, but the access to its components and weights is not straightforward. From the perspective of an end-to-end approach, GDP is a composite index we want to map from their components, even if we have only indirect measures or so-called *proxy variables*. It should be noted, however, that these proxies are more related to the gross value of production, while we want the added value; for details, see World Bank (2009). In our strategy, the neural network must minimize this problem in its optimization process and also because of its non-linear structure, whose performance is measured by the resulting accuracy.

In the task of mapping the Brazilian GDP, we adopted monthly official data on industrial production, retail sales, services, households consumption, and others produced by the Brazilian Institute of Geography and Statistics (IBGE). The IBGE is also responsible for the official Brazilian GDP quarterly data. Other sources that provide monthly statistics for our task are the Foundation Center for Foreign Trade Studies (Funcex), the National Agency of Petroleum, Natural Gas and Biofuels (ANP), the Brazilian Association of Highway Concessionaires (ABCR), the Ministry of Labor, the Central Bank of Brazil (BCB) and the Brazilian hydroelectric power generation company (Eletrobras). The dataset contains 18 monthly series plus the quarterly GDP, from January 2002[2] until December 2018, as presented in the Appendix, Table 1. All data we deal with and its transformations are available at `https://github.com/rrsguim/PhD_Economics`.

3.1 Variable selection

The selection of variables for mapping Brazilian GDP was made considering two of the most followed nowcasting indexes for the Brazilian economy, namely, the GDP Monitor from Instituto Brasileiro de Economia (2015), and the Central Bank Economic Activity Index (IBC-Br) from Banco Central do Brasil (2016). These indicators, followed largely by agents interested in evaluating the current state of the Brazilian economy, adopt the strategy known as accounting indices, as mentioned in Section 4.1, that is, an index composed of many proxy variables. Our database contains fewer variables because we do not estimate any of them, but we want brevity, so we select those that are available up to approximately 45 days after the reference month and that are easily obtained from sources. In Section 5 we present our results and the comparison with those indexes, and in Section 6 we discuss other possible strategies for variable selection.

3.2 Data transformation

It is well-known that neural networks learn faster and present better performance when variables are preprocessed. In terms of scale, if one input variable ranges between 0.1 and 0.9, and another from 10,000 to 100,000, the network will learn to use huge weights values for the former and small ones for the second variable, compromising its performance, so this kind of data must be scaled, which is what we did[3].

Next, we construct three models: the monthly GDPBR19Level, in which the input variables, or features, do not receive additional transformation, but the quarterly GDP is repeated for each month in his quarter; the monthly GDPBR19M12 and the quarterly GDPBR19Q4, where we transformed each variable from GDPBR19Level with the log difference of 12 months and 4 quarters, respectively. See Figures 14, 15 and 16 in the Appendix.

[2]Limited by the first information available for industrial data series.

[3]We rescaled then for a fixed base of 100 points in a given year when the original data was not in this format. We also tested the strategy of normalizing the data, but the results performed poorly.

3.3 Other characteristics

Other characteristics, like trend, seasonality, structural breaks, may or may not be handled in the preprocessing phase depending on the problem and the approach. In our case, what makes more sense is to treat each point in time separately because we are mapping proxies to a composite index. For further discussion from an econometric perspective, see Kuan and White (1991), Maasoumi et al. (1994), Blake (1999), and Tkacz and Hu (1999).

4 Methodology and models

Due to the availability of various ML methods and that we are, especially in economics, in the explanatory era of its applications, works often apply several ML approaches to a specific dataset to compare their performances, as in Tiffin (2016), Cook and Hall (2017), Garcia et al. (2017) and Gu et al. (2018). Makridakis et al. (2018) go further to compare statistical and ML forecasting methods. Another possible approach, our choice, is to select in advance one that is suitable for the specific task.

4.1 Deep feedforward network (DFN)

A deep neural network is an artificial neural network with multiple layers hidden between the input and output layers (Bengio, 2009). Figure 2 elucidates this relevant aspect.

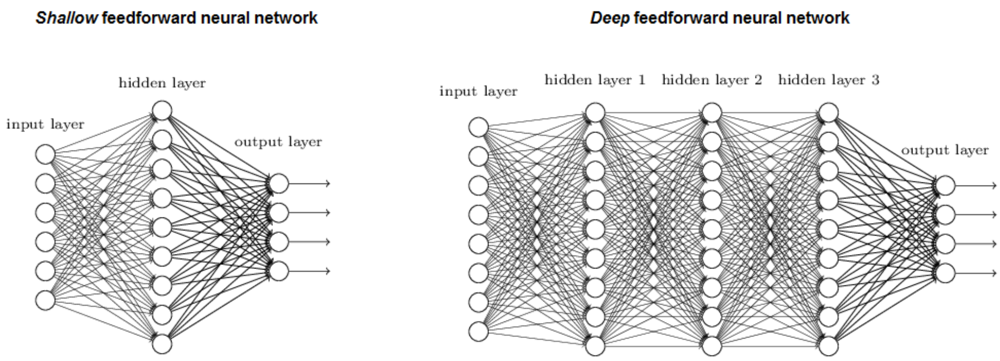


Figure 2: Shallow and deep neural networks - reproduced from Nielsen (2015)

There are variations of deep neural networks, such as convolutional, recurrent or recursive nets. We adopt *deep feedforward networks* (DFN), also called *feedforward neural networks* or *multilayer perceptrons* (Goodfellow et al., 2016, 164). These models are called feedforward because information flows through the function being evaluated from inputs, through the intermediate computations used to define the function, and finally to the output target. There are no feedback connections in which the outputs of the model are fed back into itself. A feedforward network with a single layer is sufficient to represent any function, but the layer might be infeasibly large and might fail to learn and generalize correctly. In many circumstances, using more in-depth models can reduce the number of units required to represent the desired function and can reduce the amount of generalization error.

Figure 3 is an example of a feedforward network having two layers. In this network there are d inputs, M hidden units and c output units. We can write down the correspondent analytic function as follows (Bishop, 1994, 118-9). The output of the jth hidden unit is obtained by

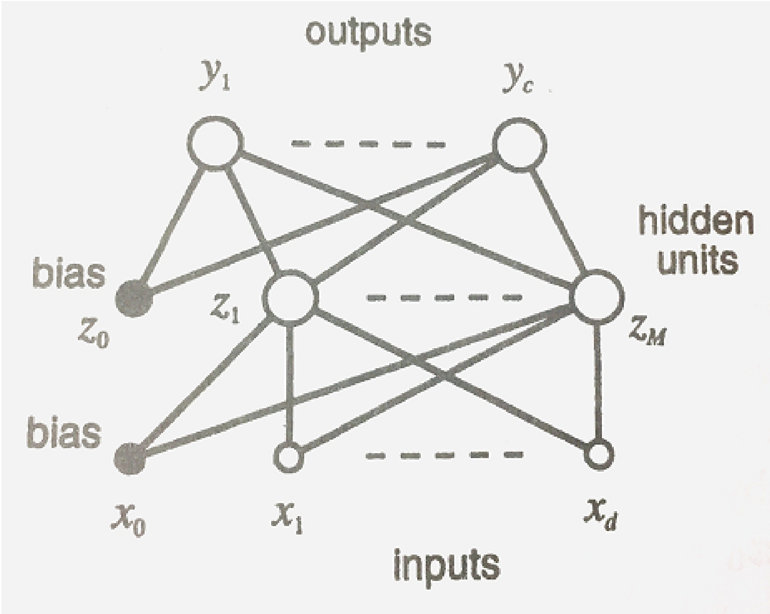


Figure 3: A feedforward network - reproduced from (Bishop, 1994, 117)

first forming a weighted linear combination of the d input values, and adding a bias, to give

$$a_j = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \; . \tag{1}$$

Here $w_{ji}^{(1)}$ denotes a weight in the first layer, going *from* input i *to* hidden unit j, and $w_{j0}^{(1)}$ denotes the bias for hidden unit j. The bias term for the hidden units is made explicit in Figure 3 by the inclusion of an extra input variable x_0 whose value is permanently set at $x_0 = 1$. This can be represented analytically by rewriting (1) in the form

$$a_j = \sum_{i=0}^{d} w_{ji}^{(1)} x_i \; . \tag{2}$$

The activation of hidden unit j is then obtained by transforming the linear sum in (2) using an activation function $g(\cdot)$ to give

$$z_j = g(a_j) \; . \tag{3}$$

The outputs of the network are obtained by transforming the activations of the hidden units using a second layer of processing elements. Thus, for each output unit k, we construct a linear combination of the outputs of the hidden units of the form

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \; . \tag{4}$$

Again, we can absorb the bias into the weights to give

$$a_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \; . \tag{5}$$

which can be represented diagrammatically by including an extra hidden unit with activation $z_0 = 1$ as shown in Figure 3. The activation of the kth output unit is then obtained by transforming this linear combination using a non-linear activation function, to give

$$y_k = \tilde{g}(a_k) \; . \tag{6}$$

Here we have used the notation $\tilde{g}(\cdot)$ for the activation function of the output units to emphasize that this need not be the same function as used for the hidden units. If we combine (2), (3), (5) and (6) we obtain an explicit expression for the complete function represented by the network diagram in Figure 3 in the form

$$y_k = \tilde{g}\left(\sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right) . \tag{7}$$

Therefore, the main reason we choose DFN is that we treat the GDP as a time-independent composite index: the GDP at time t is a weighted sum of what happens at time t. This approach, which is adopted by GDP coincident indicators like GDP Monitor from Instituto Brasileiro de Economia (2015) and Central Bank Economic Activity Index (IBC-Br) from Banco Central do Brasil (2016), is also known as *accounting indexes*[4]. This approach, broadly speaking, demand two main tasks: defining the *higher frequency*[5] proxy variables and defining the associated weights. So, the principal difference and also an innovation we are presenting in this work is applying a deep feedforward network to the second task. Figure 4, an adaptation of Figure 3, illustrates this point.

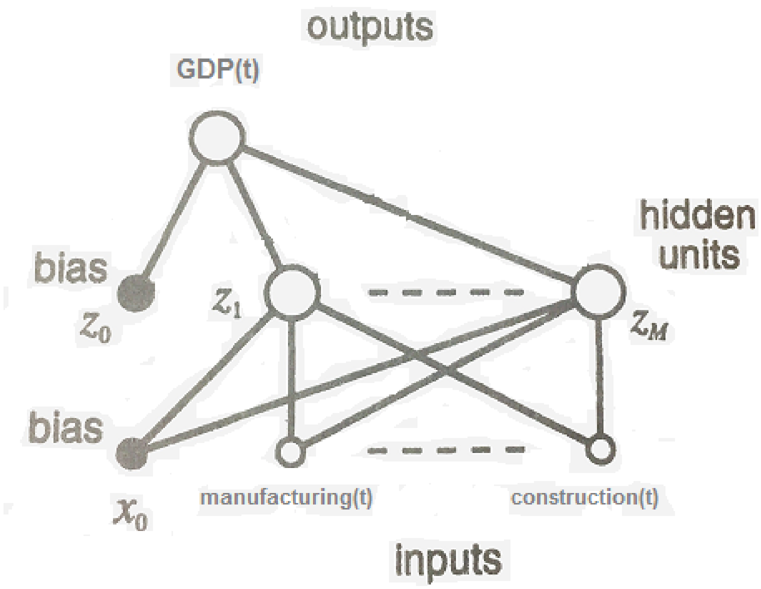


Figure 4: GDP as a composite index with DFN

There are other reasons, however. Neural networks: i) are robust to noise in input data and the mapping function and can even support learning and prediction in the presence of missing values; ii) do not make strong assumptions about the mapping function and readily learn linear and nonlinear relationships (Brownlee, 2018, 4).

According to Goodfellow et al. (2016), almost all deep-learning algorithms can be described as particular instances of a relatively simple recipe: they combine a specification of a dataset, described in Section 3, a cost function, an optimization procedure, and a model, detailed below.

4.2 Cost function

According to (Goodfellow et al., 2016, 173-4), the cost functions for neural networks are more or less the same as those for other parametric models, such as linear models. The total cost

[4]There are time series techniques treatments, especially in subcomponents, usually to solve specific issues.

[5]Generally, one looks for monthly frequency variables for the task of mapping the quarterly GDP.

function used to train a neural network will often combine cost functions with a regularization term in order to prevent overfitting[6]. Hence, we can think of three situations concerning the overfitting problem, where the model family being trained either 1) exclude the correct data-generating process − corresponding to underfitting and inducing bias, or 2) match the true data generating process, or 3) include the generating process but also many other possible generating processes − the overfitting regime where variance rather than bias dominates the estimation error. The goal of regularization is to take a model from the third regime into the second regime (Goodfellow et al., 2016, 224). Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error. Regularization is one of the central concerns of the field of machine learning, rivaled in its importance only by optimization (Goodfellow et al., 2016, 117).

4.3 Optimization procedure

When we use a machine learning algorithm, we sample the training set, and then use it to choose the parameters to reduce training set error. We then sample the test set. Under this process, the expected test error is greater than or equal to the expected value of training error. The factors that determine how well a machine learning algorithm will perform are its ability to: 1) make the training error small; 2) make the gap between training and test error small (Goodfellow et al., 2016, 109).

Back-propagation is a method to calculate a gradient that is needed in the calculation of the weights to be used when training the network. Back-propagation is a special case of an older and more general technique called automatic differentiation. In the context of learning, back-propagation is commonly used by the optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function. This technique is also sometimes called backward propagation of errors, because the error is calculated at the output and distributed back through the network layers. (Goodfellow et al., 2016, 213) describe the general back-propagation procedure.

The problem of determining the capacity of a deep learning model is especially difficult because the capabilities of the optimization algorithm limit the effective capacity, and we have little theoretical understanding of the general non-convex optimization problems involved in deep learning (Goodfellow et al., 2016, 112). Despite the fact that most practitioners, for many years, believed that local minima were a common problem plaguing neural network optimization, today that does not appear to be the case. The problem remains an active area of research, but experts now suspect that, for sufficiently large neural networks, most local minima have a low-cost function value, and that it is not essential to find the correct global minimum rather than to find a point in parameter space that has low but not minimal cost (Goodfellow et al., 2016, 282).

Following a trial-and-error approach, we obtained better results using the *Proximal Adagrad*[7] as the optimizer.

4.4 Models

The ideal model is an oracle that knows the true probability distribution that generates the data. Even such a model will still incur some error on many problems, because there may still be some noise in the distribution. In the case of supervised learning, the mapping from x to y may be inherently stochastic, or y may be a deterministic function that involves other variables besides those included in x (Goodfellow et al., 2016, 114).

[6] See Goodfellow et al. (2016), Chapters 6 and 7 for further discussion.

[7] Singer and Duchi (2009)

Most machine-learning algorithms have hyperparameters, settings that we can use to control the algorithm's behavior. The learning algorithm itself does not update the values of hyperparameters, though we can design a nested learning procedure in which one learning algorithm learns the best hyperparameters for another learning algorithm (Goodfellow et al., 2016, 118).

The primary architectural considerations are choosing the depth of the network and the width of each layer. Deeper networks are often able to use far fewer units per layer and far fewer parameters, as well as frequently generalizing to the test set, but they also tend to be harder to optimize. The ideal network architecture for a task must be found via experimentation guided by monitoring the validation set error (Goodfellow et al., 2016, 194).

There are many choices to make when structuring a neural network, including the number of hidden layers, the number of neurons in each layer, and which units are connected (Gu et al., 2018). We adopt a similar strategy to (Gu et al., 2018, 19-21), evaluating a variety of network architectures having up to six hidden layers[8]. Another consideration is that we are working with a small dataset, where simple networks with only a few layers and nodes often perform best, as it becomes difficult to support a rich parametrization without a very large number of observations (Gu et al., 2018, 19).

Among potential choices for the nonlinear activation function, such as sigmoid, hyperbolic, softmax, we choose, for all nodes, a popular functional form in recent literature known as the rectified linear unit (ReLU), defined as

$$ReLU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise.} \end{cases} \tag{8}$$

We split the dataset in 80% for training and 20% for testing, and, after some trial and error, the hyperparameters[9] were defined as: three hidden layers with 150, 100, and 50 neurons, respectively, learning rate = 0.001 and $L1$ regularization strength = 0.001.

In fact, we chose a similar configuration to a regression model developed by the *Tensorflow*[10] team, available at `https://www.tensorflow.org/tutorials/keras/basic_regression`. All our codes in *Python* language, using *Tensorflow* and *Keras*[11] as frameworks, are available at `https://github.com/rrsguim/PhD_Economics`. For linear regressions we adopted *EViews*.[12]

5 Empirical results

By now we may revisit our central question: *can a DFN achieves good performance in the task of mapping the GDP?*

5.1 Baseline model

The baseline model is a linear regression with quarterly data. As we mentioned in Subsection 4.1, we can assume that GDP and its proxies are correlated, but the relationship is probably not linear, nor constant over time neither restricted to these proxies. Thus, linear regression results are fair but require improvements (Figure 5). Among many possibilities in terms of more sophisticated models, we show below the results of our choice, the Deep Neural Network models.

[8]For further discussion about hyperparameters choice, see (Makridakis et al., 2018, 7)

[9]Due to random initialization of weights, it is not expected to observe the same results when running the model again but should be very similar.

[10] `https://www.tensorflow.org`

[11] `https://keras.io/`

[12] `https://www.eviews.com`

5.2 GDPBR19Level and GDPBR19M12 models

In the GDPBR19Level model, the target and all features are monthly and in level, base 100, while in GDPBR19M12 we apply a 12-month log difference transformation (See Appendix, Table 1 and Figures 14 and 15).

The learning curves of the training and validation set behave as expected for both models (Figure 6), although we may see less noise in the learning process of GDPBR19M12[13]. In terms of accuracy in the validation set, these models perform similarly, since we observed a Mean Absolute Percentage Error (MAPE) of 1.33 for the GDPBR19Level and Mean Absolute Error (MAE) of 0.0140 for the GDPBR19M12.

In terms of mapping, Figures 7 show that both performances are better than those of the baseline model, but still with room for improvement. See also Subsection 5.5.

5.3 GDPBR19Q4 model

In the GDPBR19Q4 model, the target and all features are quarterly, in a 4-quarter log difference transformation (See Appendix, Table 1 and Figure 16).

The learning curves of the training and validation set behave as expected. In terms of accuracy, this model performs better than the previous, since we see a MAE of 0.0107. We also observe better performance in terms of mapping (Figure 8 and Subsection 5.5).

5.4 Transfer learning

Transfer learning and domain adaptation refer to the situation where what has been learned in one setting, e.g., distribution P1, is exploited to improve generalization in another setting, say, distribution P2 (Goodfellow et al., 2016, 534). To test this possibility, we apply to the monthly features the parameters trained in quarterly data in the model GDPBR19Q4. The result can be seen in Figure 9, where we see that the performance is better than the one directly trained by the GDPBR19M12 model, suggesting that there is a way forward in this direction. See also Subsection 5.5.

5.5 Comparison with benchmarks

5.5.1 Monthly data

The good performance of the DFN models to generate monthly data is shown in Figures 10 and 11: the 12-months log-difference comparison of all models with benchmarks for monthly GDP data, namely the GDP Monitor from Instituto Brasileiro de Economia (2015) and the Central Bank Economic Activity Index (IBC-Br) from Banco Central do Brasil (2016). The highest correlation observed with IBC-Br is 0.953 (GDPBR19Q4-TL) and with GDP Monitor is 0.957 (GDPBR19M12).

5.5.2 Quarterly data

Again, due to our cross-sectional approach, we are limited to cross-sectional forecast. In the last subsections, we show the capacity of DFNs to generate predictions of missing points (quarter to monthly data). Another possibility is to use the mapped GDP to produce a nowcast, in the sense of forecasting the near past, of the official quarterly GDP. There are many issues related to this area of research, while one of the most relevant is about near-real-time versus

[13] According to Ng (2015), trying a smaller set of features usually reduces output variance.

real-time proxies[14]. That said, Figure 12 shows the out-of-sample forecast from 2015Q1 to 2018Q4. For each quarterly estimates, the GDPBR19Q4 model is trained with data available up to approximately 45 days after the end of the respective quarter.

6 Discussion

In this paper, we have constructed specialized algorithms for mapping the low-frequency gross domestic product from high-frequency variables, in a cross-sectional approach. The deep feed-forward network architectures with regression supervised learning algorithms demonstrated good performance when applied to Brazilian data, with IBC-Br and GDP Monitor as benchmarks. Given the low computational requirements and the simplicity of the algorithms, this strategy can be valuable as a complementary approach to mapping high-frequency variables from low-frequency variables. The next step of this research is toward more sophisticated models, including, but not limited to, a grid search for hyperparameters and big data as inputs, such as road traffic, satellite images or financial data.

References

Ahonen, P., 2015. Institutionalizing big data methods in social and political research. Big Data & Society 2 (2).

Banco Central do Brasil, 2016. Indice de Atividade Economica do Banco Central (IBC-Br) - revisao metodologica. Relatorio de Inflacao 18 (1), 23–27.

Bengio, Y., 2009. Learning Deep Architectures for AI. Foundations and Trends in Machine Learning 2 (1), 1–127.

Bishop, C. M., 1994. Neural Networks for Pattern Recognition. Oxford University Press.

Blake, A., 1999. An artificial neural network system of leading indicators. National Institute of Economic and Social Research (NIESR) Discussion Papers 144, National Institute of Economic and Social Research.

Brownlee, J., 2018. Predict the future with mlps, cnns and lstms in python. In: Deep Learning for Time Series Forecasting. eBook.

Burns, A. F., Mitchell, W. C., 1946. Measuring Business Cycles. National Bureau of Economic Research, Inc.

Carriere-Swallow, Y., Labbe, F., July 2010. Nowcasting with google trends in an emerging market. Working Papers of the Central Bank of Chile (588).

Cook, T. R., Hall, A., Sep. 2017. Macroeconomic Indicator Forecasting with Deep Neural Networks. Research Working Paper RWP 17-11, Federal Reserve Bank of Kansas City.

Dutta, S., Shekhar, S., 1988. Bond rating: A non-conservative application of neural networks. In: IEEE Int Conf on Neural Networks. IEEE, pp. 443–450.

[14] Further advice from Kliesen and McCracken (2016) from his work about tracking the US economy is that users of nowcasting models should be aware that most of the monthly source data are initially sample-based estimates. This means, in effect, that the initial estimates are subject to repeated revision as new information becomes available.

Garcia, M. G. P., Medeiros, M. C., Vasconcelos, G. F. R., 2017. Real-time inflation forecasting with high-dimensional models: The case of brazil. International Journal of Forecasting 33, 679–693.

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Lieberman Aiden, E., Fei-Fei, L., Feb. 2017. Using Deep Learning and Google Street View to Estimate the Demographic Makeup of the US. ArXiv e-prints.

Giannone, D., Banbura, M., Modugno, M., 2013. Now-casting and the real-time data flow. ECB Working Paper (1564).

Giannone, D., Banbura, M., Reichlin, L., May 2010. Nowcasting. ECARES Working Paper (21).

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Gu, S., Kelly, B., Xiu, D., 2018. Empirical asset pricing via machine learning. Chicago Booth Research Paper 18 (4).

Guimaraes, R., Monteiro, S., Ziegelmann, F., ???? Mapping high- to low-frequency variables with deep feedforward networks, in progress.

Herbrich, R., Keilbach, M., Graepel, T., Bollmann-Sdorra, P., Obermayer, K., 1998. Neural networks in economics: Background, applications and new developments. In: Advances in Computational Economics: Computational Techniques for Modelling Learning in Economics. Kluwer Academics, pp. 169–196.

Hinds, S., Rimmington, L., Dance, H., Gillham, J., Sentance, A., Hawksworth, J., 2017. A machine learning approach to estimating current gdp growth. Report, PriceWaterhouseCoopers.

Instituto Brasileiro de Economia, November 2015. Metodologia do indicador - preliminar. In: http://portalibre.fgv.br/lumis/portal/file/fileDownload.jsp?fileId=8A7C82C54F7DAFDC01515E19A30A394E. Fundacao Getulio Vargas.

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., Ermon, S., 2016. Combining satellite imagery and machine learning to predict poverty. Science 353 (6301), 790–794.

Kliesen, K. L., McCracken, M. W., 2016. Tracking the u.s. economy with nowcasts. The regional economist, Federal Reserve Bank of St. Louis.

Kohzadi, N., Boyd, M. S., Kaastra, I., Kermanshahi, B. S., Scuse, D., 1995. Neural networks for forecasting: An introduction. Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie 43 (3), 463–474.

Kuan, C.-M., White, H., 1991. Artificial neural networks: an econometric perspective. Econometric Reviews 13 (1), 1–91.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F. E., 2017. A survey of deep neural networks architectures and their applications. Neurocomputing 234 (C), 11–26.

Maasoumi, E., Khotanzed, A., Abaye, A., 1994. Artificial neural networks for some macroeconomic series: A first report. Econometric Reviews 13 (1), 105–122.

Makridakis, S., 2017. The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms. Futures 90, 46–60.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. Statistical and machine learning forecasting methods: Concerns and ways forward. PLoS ONE 13 (3).

Ng, A., 2015. Advice for applying machine learning. In: https://see.stanford.edu/materials/aimlcs229/ML-advice.pdf.

Nielsen, M. A., 2015. Neural networks and deep learning. In: http://neuralnetworksanddeeplearning.com/chap5.html. Determination Press.

Portugal, M. S., 1995. Neural networks versus time series methods: a forecasting exercise. Revista Brasileira de Economia 49 (4), 611–629.

Schwartz, I. E., November 1992. Where neural networks are already at work. Business Week, pp. 136–137.

Singer, Y., Duchi, J. C., 2009. Efficient learning using forward-backward splitting. In: Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., Culotta, A. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 495–503.

Stergiou, C., Siganos, D., 2011. Neural networks. Tech. rep., Imperial College London.

Stock, J., Watson, M., 2011. Dynamic Factor Models. Oxford University Press, Oxford.

Swanson, N., White, H., 1997. Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models. International Journal of Forecasting 13 (4), 439–461.

Tal, B., Nazareth, L., 1995. Artificial intelligence and economic forecasting. Canadian business economics, Canadian Imperial Bank of Commerce.

Tiffin, A., 2016. Seeing in the dark: a machine-learning approach to nowcasting in lebanon. IMF Working Paper (56).

Tkacz, G., Hu, S., 1999. Forecasting gdp growth using artificial neural networks. Staff Working Papers 99-3, Bank of Canada.

Varian, H. R., 2014. Big data: New tricks for econometrics. Journal of Economic Perspectives 28 (2), 3–28.

Varian, H. R., Choi, H., April 2009. Predicting the present with google trends. In: http://googleresearch.blogspot.com/2009/04/predicting-present-with-google-trends.html.

White, H., 1988. Economic prediction using neural networks: the case of ibm daily stock returns. In: IEEE 1988 International Conference on Neural Networks. Vol. 2. pp. 451–458.

World Bank, 2009. System of national accounts 2008 (english). In: http://documents.worldbank.org/curated/en/417501468164641001/System-of-national-accounts-2008.

7 Appendix

Dependent Variable: GDP
Method: Least Squares
Date: 04/01/19 Time: 10:46
Sample: 2003Q1 2018Q4
Included observations: 64

Variable	Coefficient	Std. Error	t-Statistic	Prob.
MINING	0.021518	0.015539	1.384774	0.1729
IND	0.266309	0.039244	6.785948	0.0000
ELETCO	0.016221	0.060739	0.267063	0.7906
ELETIN	0.088590	0.028515	3.106831	0.0033
ELETOT	-0.068679	0.034891	-1.968399	0.0552
ELETHO	0.150078	0.054003	2.779068	0.0079
CONSTR	-0.099852	0.038978	-2.561754	0.0138
RETAIL	0.086325	0.034002	2.538809	0.0147
EXPORT	0.001566	0.012525	0.125044	0.9010
IMPORT	0.007444	0.014025	0.530790	0.5982
DIESEL	0.045317	0.040912	1.107650	0.2739
TRUCKT	0.105216	0.049349	2.132051	0.0385
JOBRTL	0.470936	0.162625	2.895838	0.0058
JOBTRS	-0.327508	0.094955	-3.449100	0.0012
JOBACC	0.087259	0.238807	0.365395	0.7165
JOBHLT	-0.034221	0.143145	-0.239068	0.8121
JOBEDC	0.163442	0.122202	1.337474	0.1878
CREDIT	-0.028633	0.020038	-1.428950	0.1599
C	-0.001425	0.005372	-0.265282	0.7920

R-squared	0.984559	Mean dependent var	0.022668
Adjusted R-squared	0.978382	S.D. dependent var	0.033076
S.E. of regression	0.004863	Akaike info criterion	-7.572709
Sum squared resid	0.001064	Schwarz criterion	-6.931791
Log likelihood	261.3267	Hannan-Quinn criter.	-7.320219
F-statistic	159.4044	Durbin-Watson stat	1.909084
Prob(F-statistic)	0.000000		

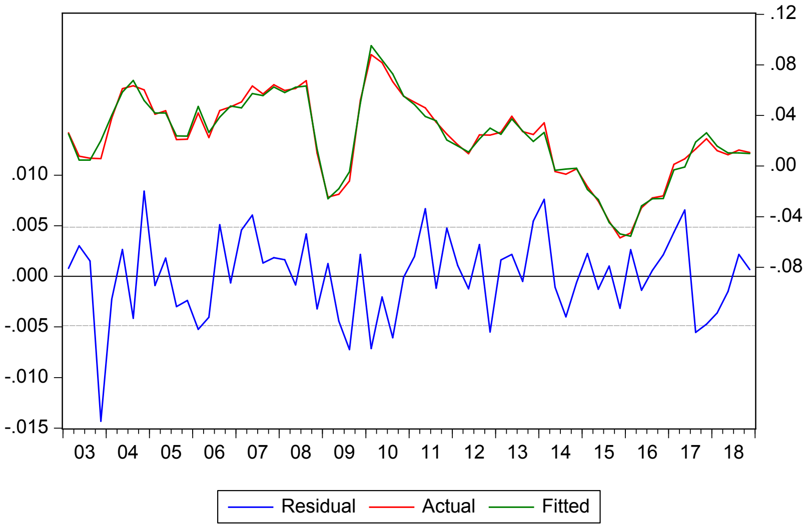
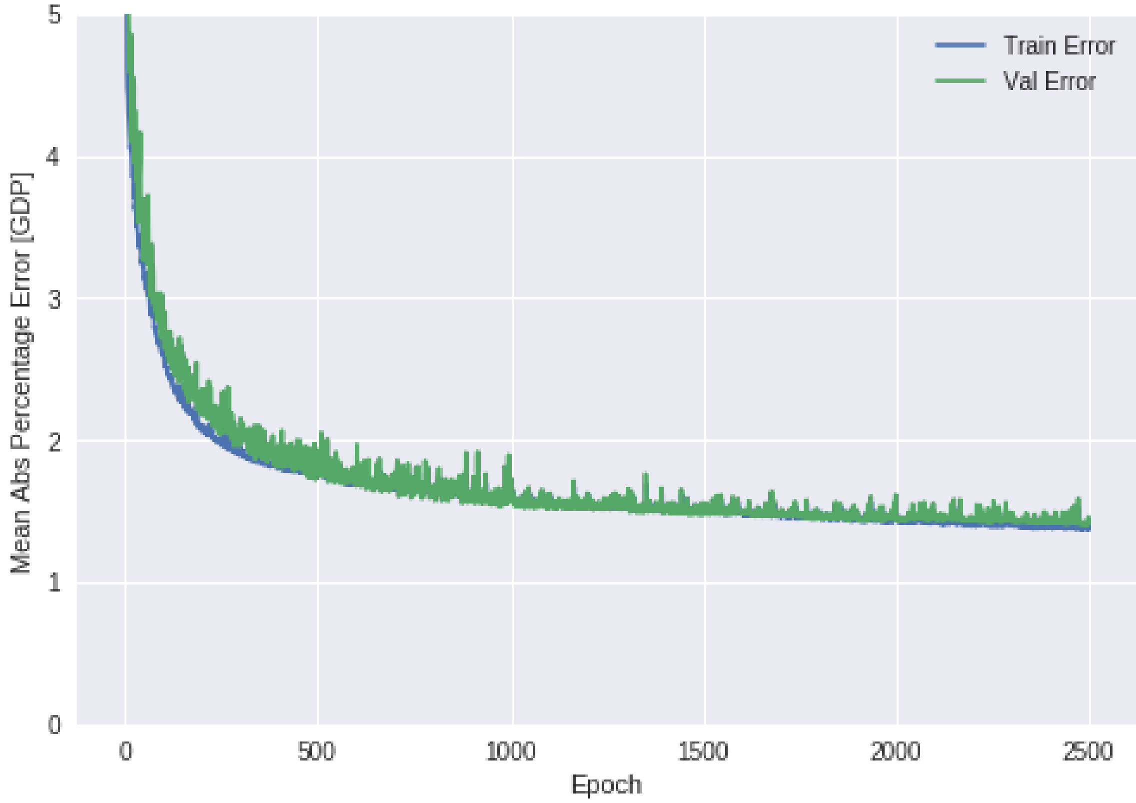


Figure 5: Baseline model: linear regression



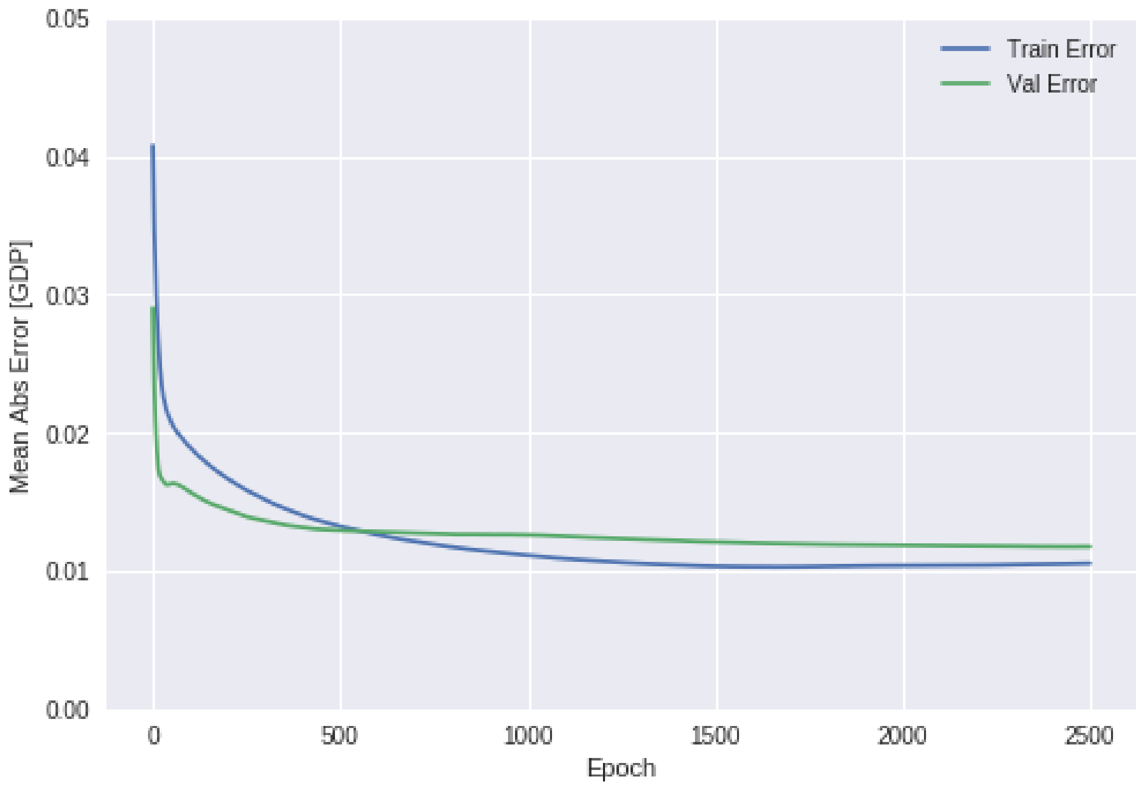


Figure 6: Learning curves: GDPBR19Level and GDPBR19M12

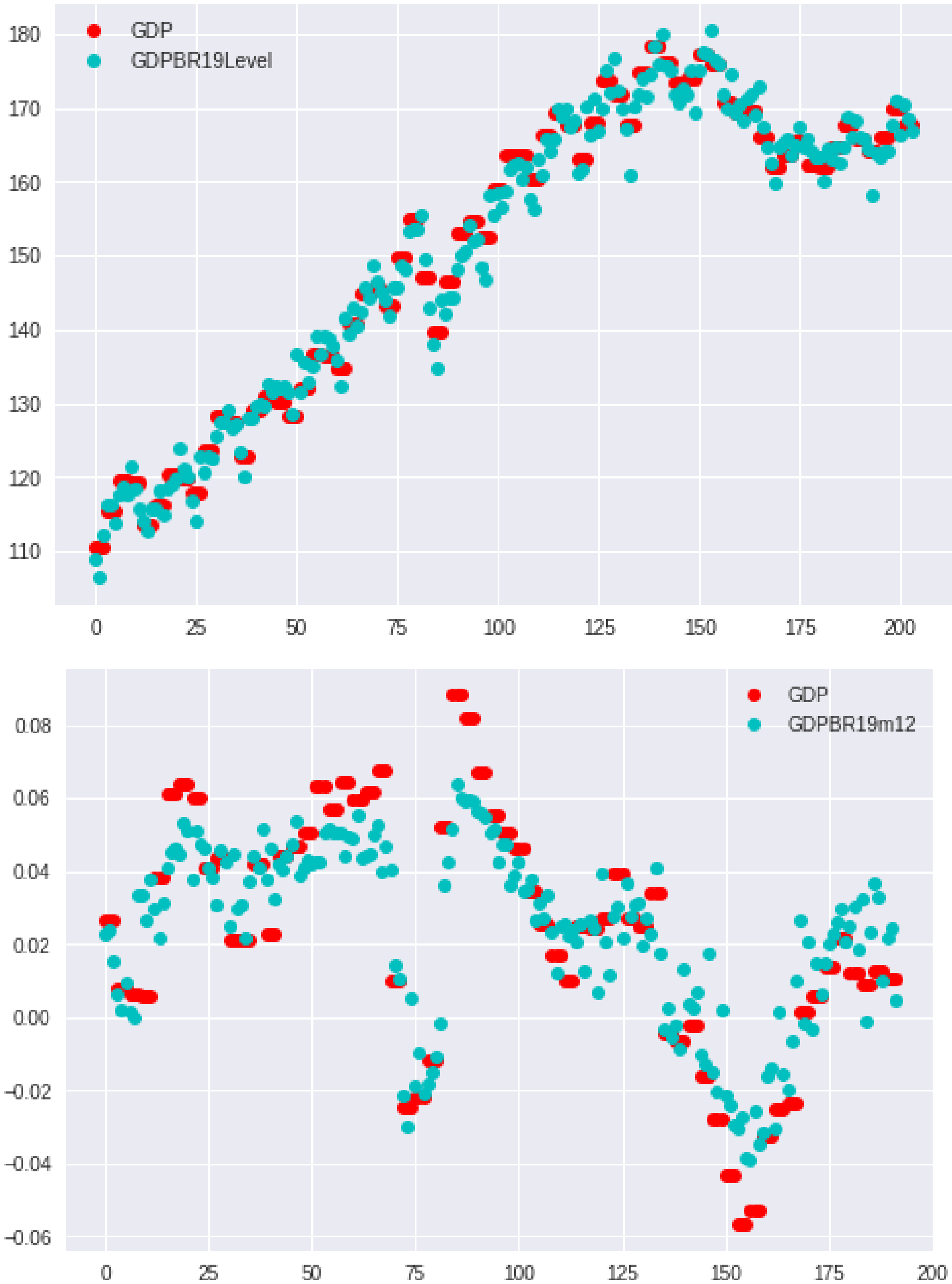


Figure 7: Mapping: GDPBR19Level and GDPBR19M12

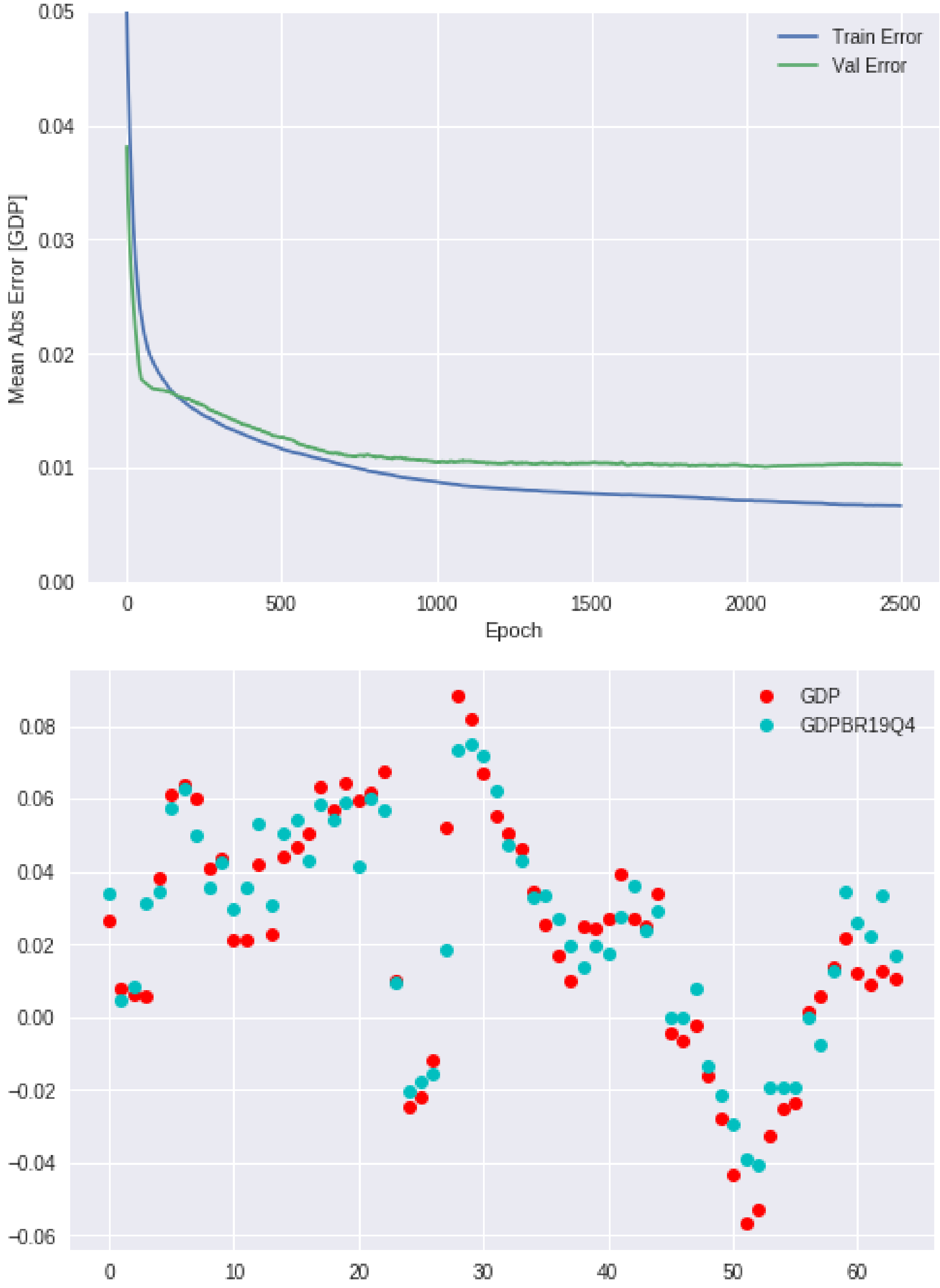


Figure 8: Learning curves and mapping: GDPBR19Q4

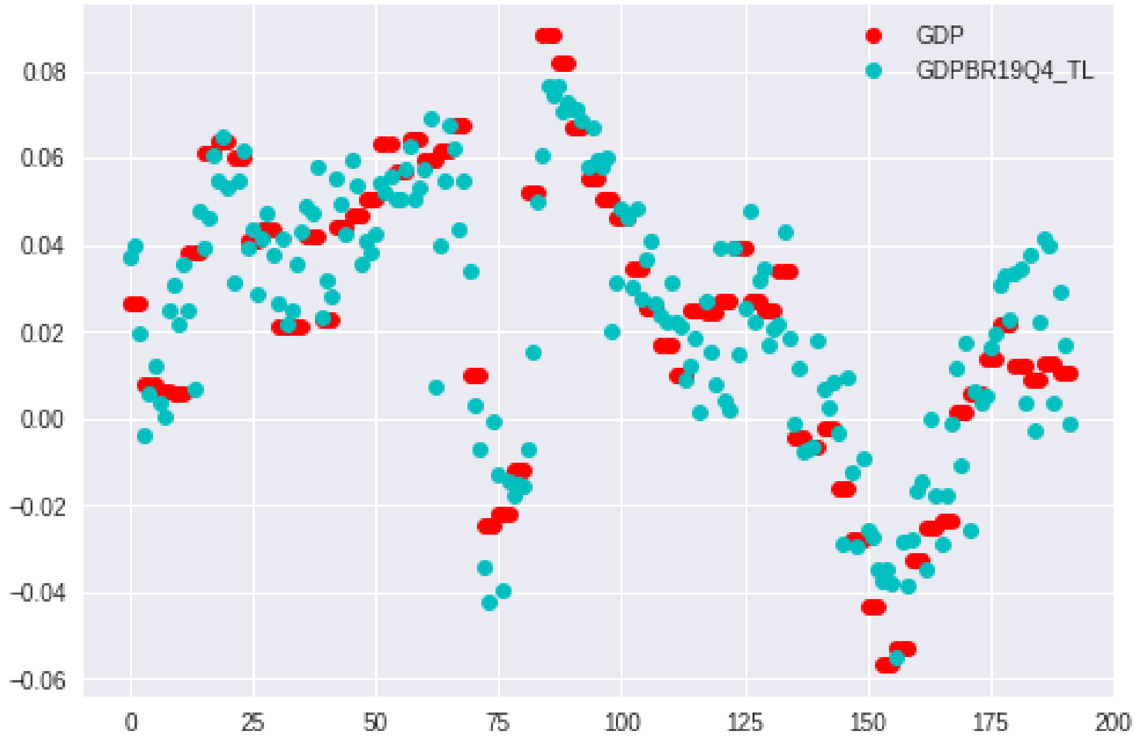


Figure 9: Transfer learning

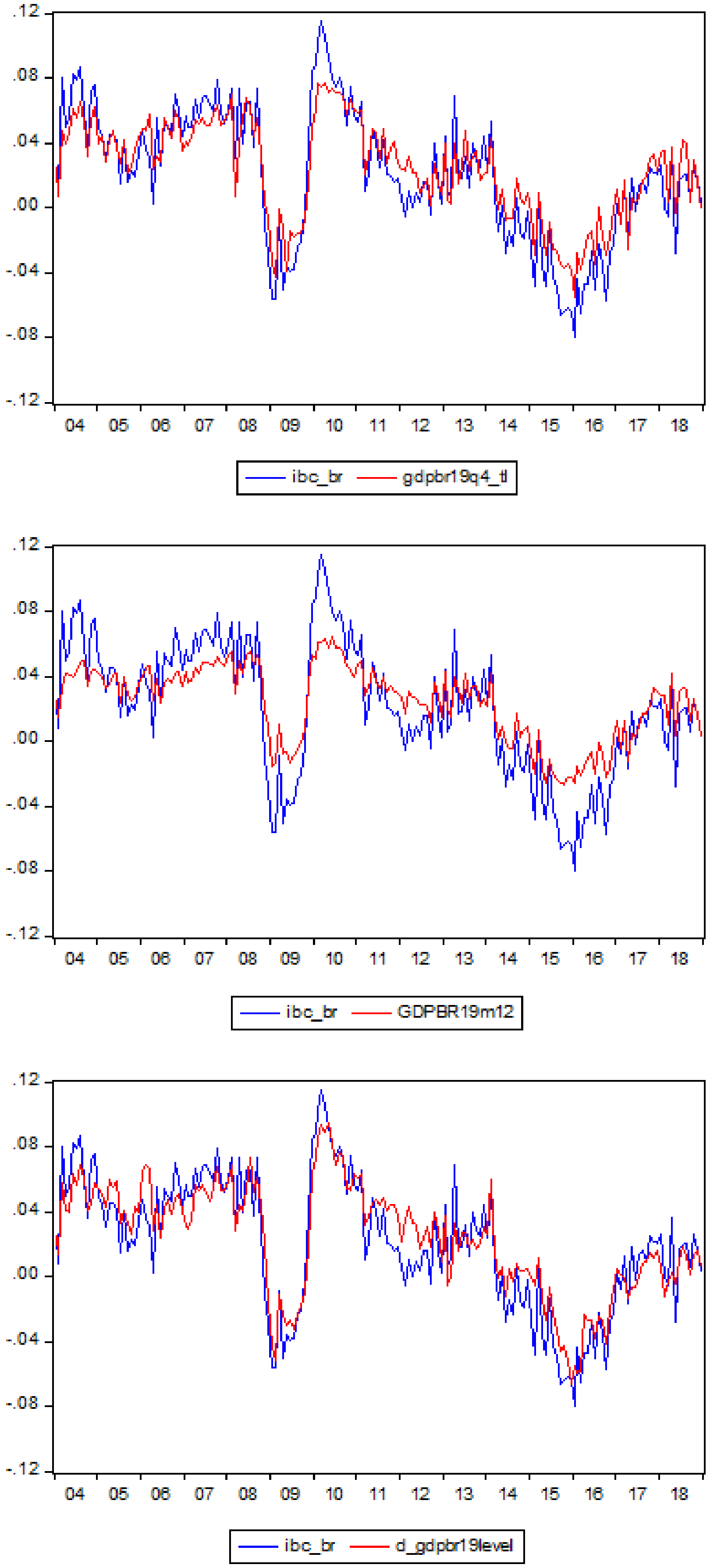


Figure 10: Twelve-months log-difference comparison with IBC-Br

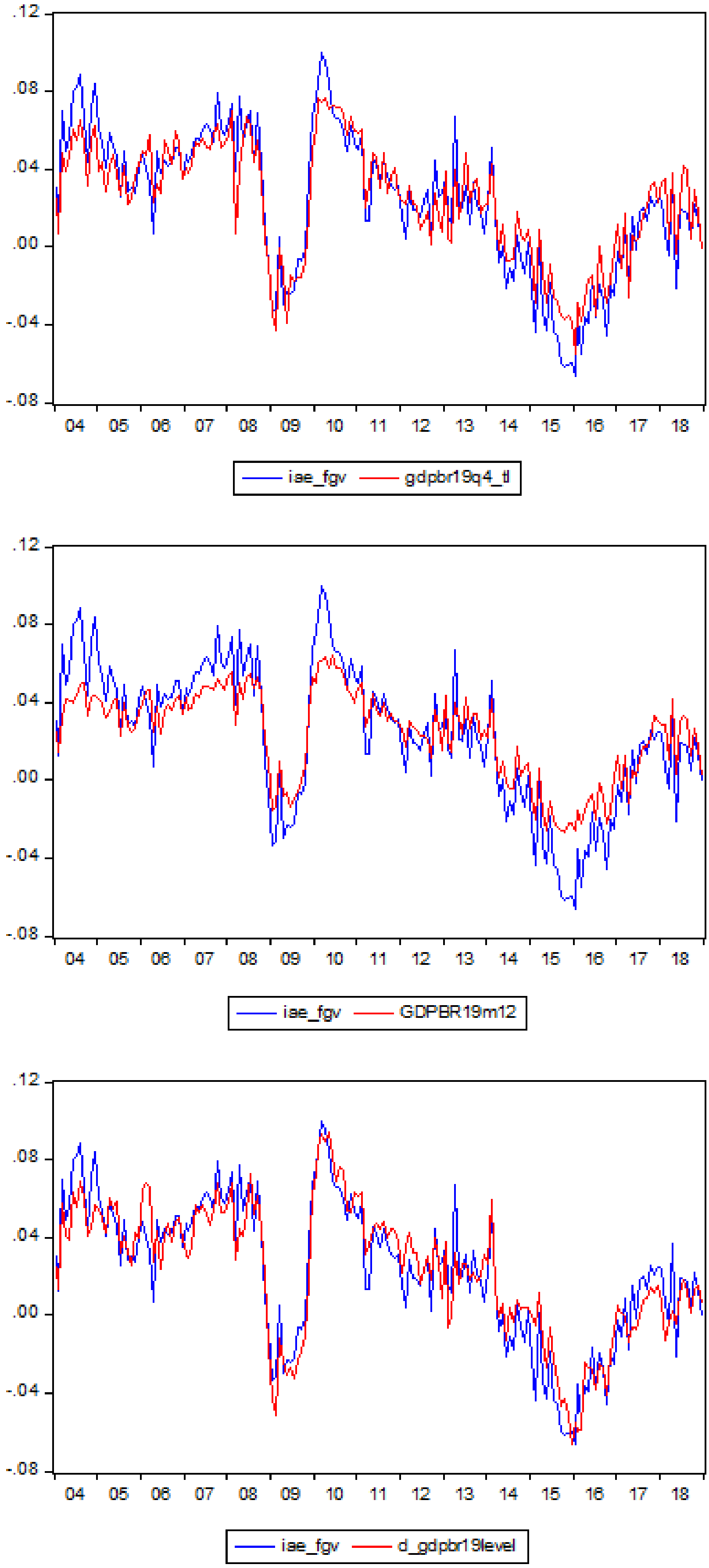


Figure 11: Twelve-months log-difference comparison with GDP Monitor

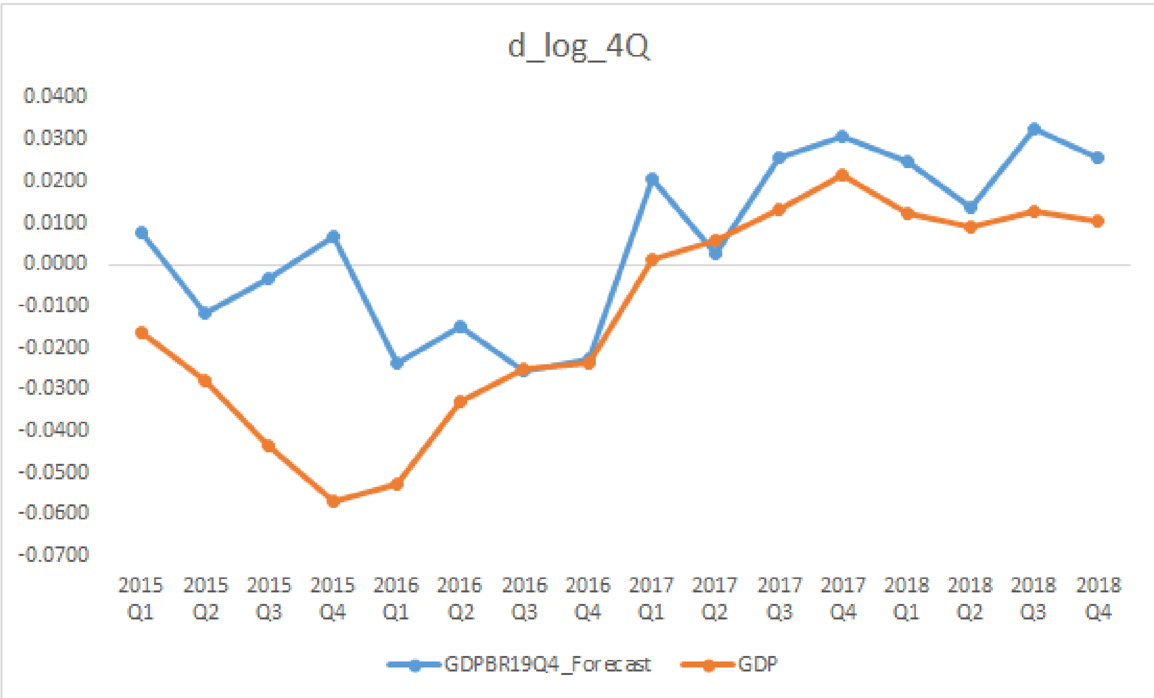


Figure 12: Out-of-sample GDP forecast

Table 1: Dataset

Code	Series	Start	Characteristics and transformations	Source
gdp	Gross domestic product - **Target**	2002:Q1	1995=100, log-diff. 12 months, log-diff. 4 quarters	IBGE
mining	Mining and Oil	2002:M1	2012=100, log-diff. 12 months, log-diff. 4 quarters	IBGE
ind	Industrial production	2002:M1	2012=100, log-diff. 12 months, log-diff. 4 quarters	IBGE
eletco	Electricity consumption: commerce	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Eletrobras
eletin	Electricity consumption: industry	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Eletrobras
eletho	Electricity consumption: households	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Eletrobras
eletot	Electricity consumption: other	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Eletrobras
constr	Civil construction inputs	2002:M1	2012=100, log-diff. 12 months, log-diff. 4 quarters	IBGE
retail	Retail sales: retail sector	2002:M1	2014=100, log-diff. 12 months, log-diff. 4 quarters	IBGE
export	Export goods	2002:M1	2006=100, log-diff. 12 months, log-diff. 4 quarters	Funcex
import	Import goods	2002:M1	2006=100, log-diff. 12 months, log-diff. 4 quarters	Funcex
diesel	Diesel consumption	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	ANP
truckt	Truck traffic	2002:M1	1999=100, log-diff. 12 months, log-diff. 4 quarters	ABCR
jobrtl	Formal jobs: rentals	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Ministry of Labor
jobtrs	Formal jobs: transportation	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Ministry of Labor
jobacc	Formal jobs: accommodation	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Ministry of Labor
jobhlt	Formal jobs: health	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Ministry of Labor
jobedc	Formal jobs: education	2002:M1	1996=100, log-diff. 12 months, log-diff. 4 quarters	Ministry of Labor
credit	Nominal credit: non-earmarked new operations	2002:M1	2001=100, log-diff. 12 months, log-diff. 4 quarters	BCB

Country	Central Bank	Use of nowcasting
United Kingdom	Bank of England	The Bank of England's Monetary Policy Committee (MPC) uses a compilation of nowcasts from three different models to form its initial view on the current state of the economy. In particular: (i) based on different industries (e.g. retail services, manufacturing, construction etc.) to mimic the production approach to calculating GDP, (ii) a mixed-data sampling model and (iii) a dynamic factor model. These are then taken together to form a judgement-based nowcast which is used by the MPC to inform its monetary policy decisions from month-to-month.
United States	Federal Reserve Bank of Atlanta (FRBA)	FRBA's Centre for Quantitative Economic Research produces frequent publicly available nowcasts of US GDP in advance of and following the US Bureau of Economic Analysis's advance estimate. It nowcasts thirteen separate expenditure components of GDP (e.g. consumer spending, investment, etc.) to mimic the expenditure approach to calculating GDP using dynamic factor modelling.
United States	Federal Reserve Bank of New York (FRBNY)	FRBNY publishes its own publicly available nowcast of US GDP growth to provide "a model-based counterpart to the more routine forecasts produced at the bank, which have traditionally been based on experience knowledge". Similar to FRBA, it also uses a dynamic factor approach, but does not mimic a particular approach to calculating GDP.
Eurozone	European Central Bank (ECB)	The ECB also uses dynamic factor-based nowcasting models to inform its policy decisions. Its staff have released a number of working papers which form cornerstones of the nowcasting literature, such as "Now-casting and the real-time data flow" (2013), by Marta Bańbura, Domenico Giannone, Michele Modugno and Lucrezia Reichlin.
Norway	Norges Bank	Norges Bank uses a variety of statistical nowcasting and short-term forecasting models of GDP and inflation to inform its policy rate decisions. Using several models it compiles a composite nowcast using a technique it calls SAM (System for Averaging Models), which produces a weighted average of the results of different models.

Sources: Bank of England, FRBA, FRBNY, ECB, Norges Bank, PwC

Figure 13: Nowcasting models in use by selected monetary authorities - reproduced from (Hinds et al., 2017, 38)

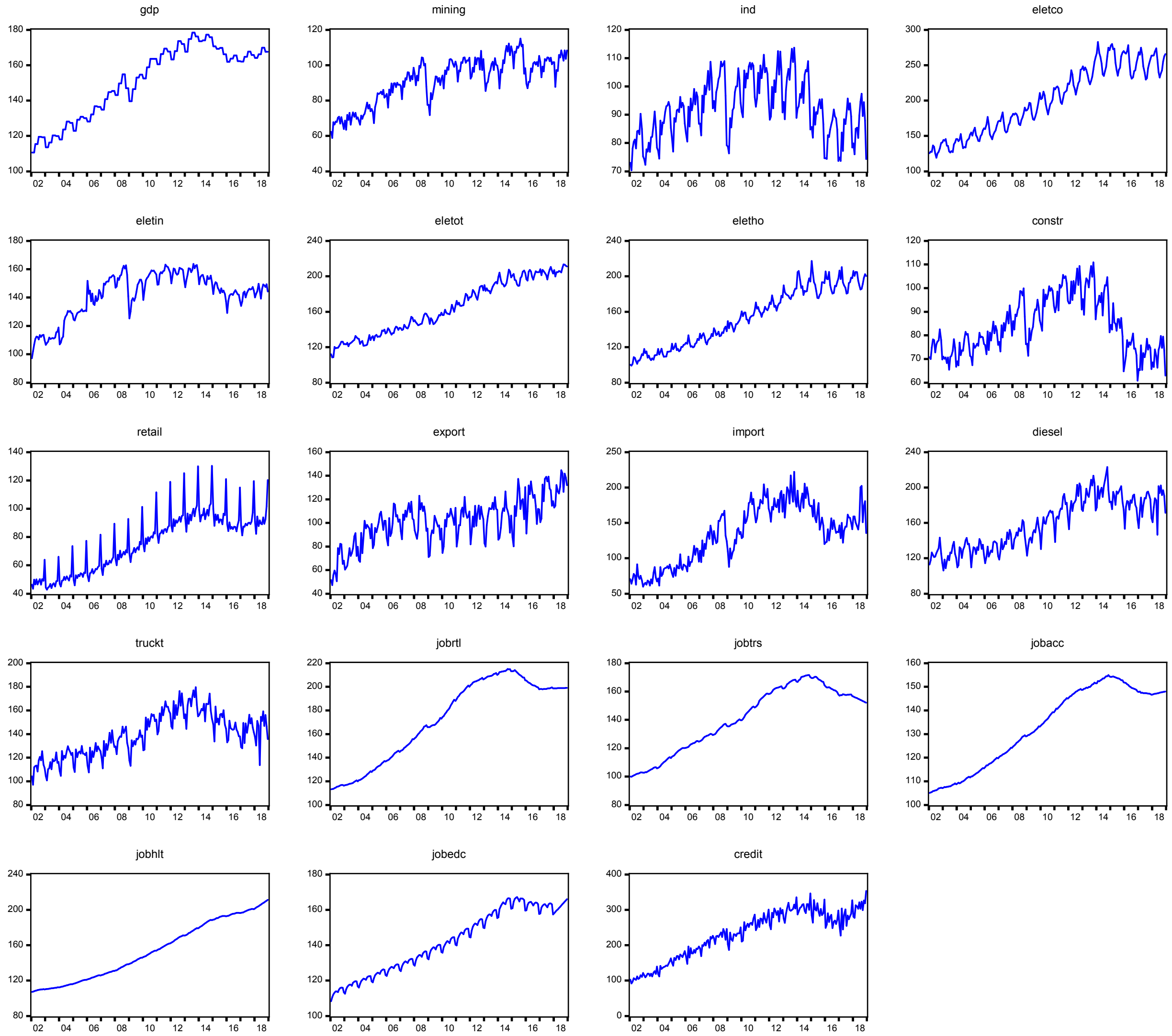


Figure 14: Series for the GDPBR19Level model

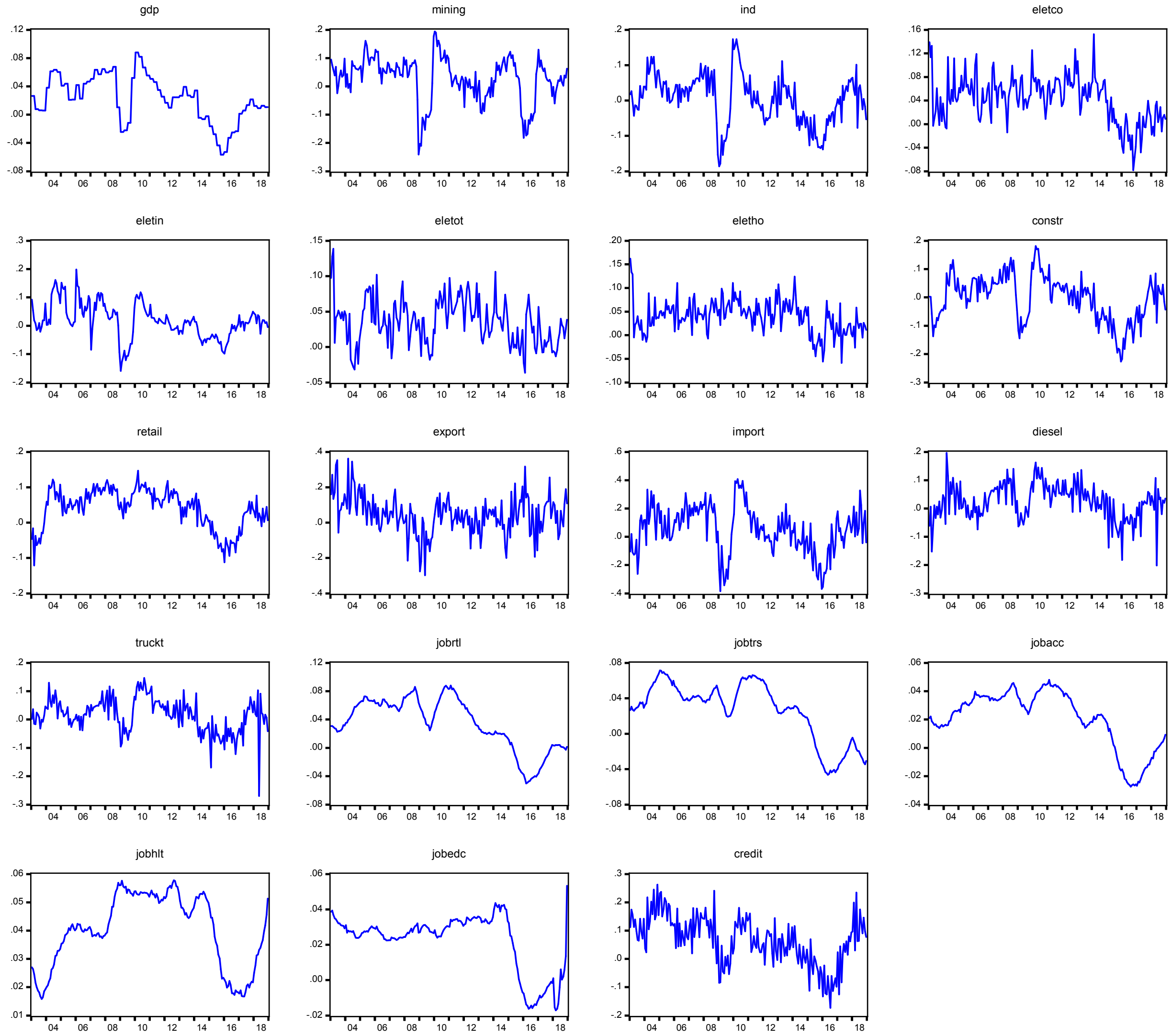


Figure 15: Series for the GDPBR19M12 model

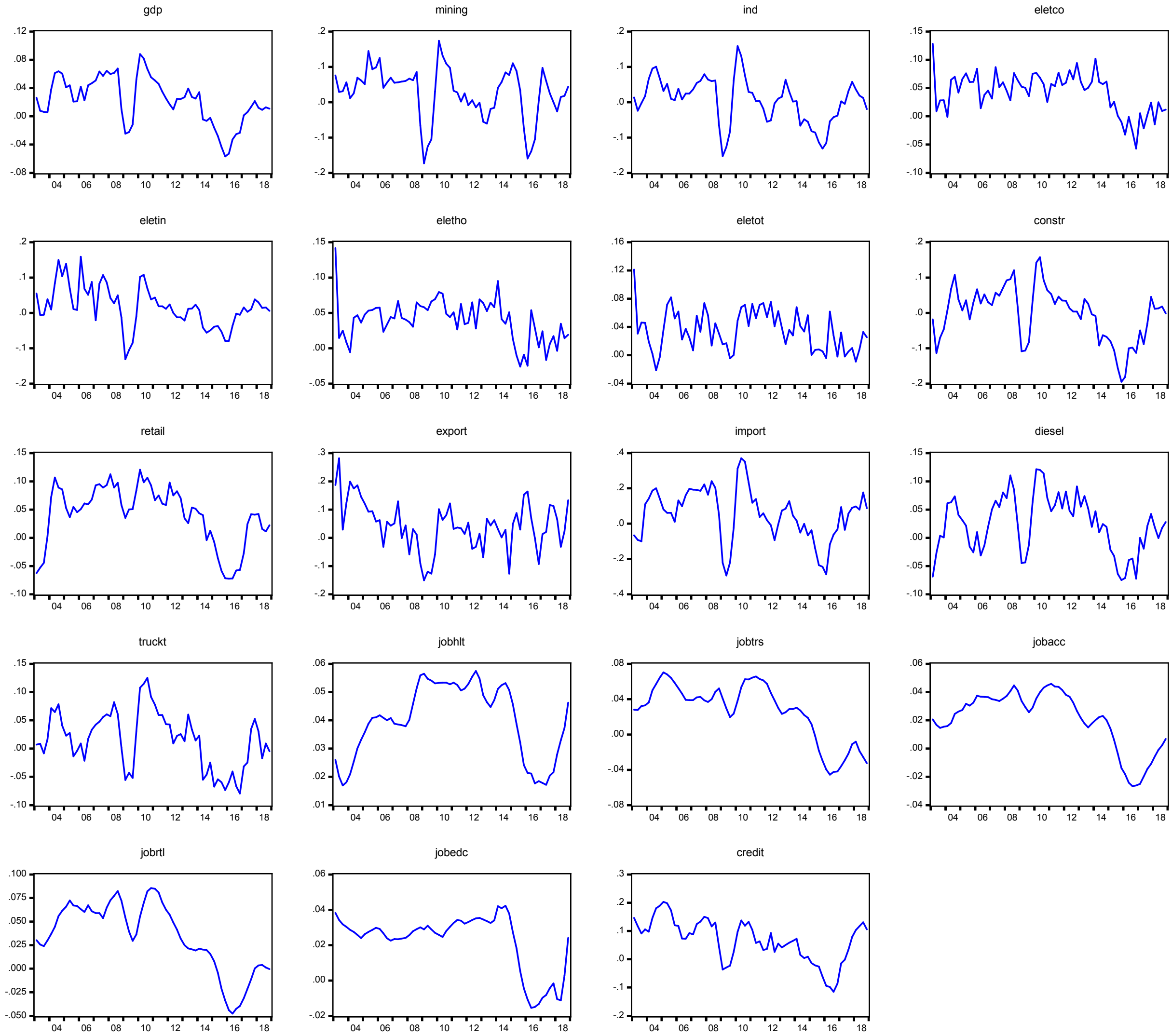


Figure 16: Series for the GDPBR19Q4 model