UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

GPD Growth Nowcast: A Neural Network Approach

Sebastián Gabriel Jiménez Montalvo

Matemáticas

Trabajo de titulación presentado como requisito

para la obtención del título de

Matemático

Quito, 12 de diciembre de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

GPD Growth Nowcast: A Neural Network Approach

Sebastián Gabriel Jiménez Montalvo

Nombre del profesor, Título académico: Santiago J. Gangotena, Ph.D.

Quito, 12 de diciembre de 2020

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Nombres y Apellidos: Sebastián Gabriel Jiménez Montalvo

Código: 00205503

Cédula de Identidad : 1720439999

Lugar y fecha: Quito, 12 de diciembre de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

AGRADECIMIENTOS

A mi viejo, por ser siempre el más firme, sin tu confianza no sería el mismo, a mi mamá por todo el aguante durante toda la vida, y a mis hermanos y mis tíos por ser siempre mi lugar de paz, amor y buen whisky.

Gracias también a todos los que siempre me inspiran con su trabajo, especialmente Santi y Carlitos.

RESUMEN

Este trabajo presenta un modelo de redes neurales para estimar el nowcast del crecimiento del PIB para el Ecuador a través del estudio de series de tiempo macroeconómicas, con observaciones mensuales desde enero del 2000 hasta junio del 2020. Se utiliza un mecanismo de re muestreo para obtener un estimador como resultado, desde el cual se pueden construir intervalos de confianza. Se obtiene una precisión del 90% cuando se analiza el crecimiento relativo al período anterior, y una del 30% respecto al valor puntual del crecimiento.

ABSTRACT

This work presents a neural network approach to estimate the GDP growth nowcast for Ecuador, through an analysis of macroeconomic time series, which have monthly observations from January 2000 to June 2020. A re-sampling approach is used to build an estimator as a result, from which we can build confidence intervals to make traditional statistics. A very precise estimation for the relative growth with respect to the last period is obtained, of about 90%, and just a 30% for the precise number of the GDP growth.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

It is on the interests of governments, central banks and the academic spectrum to get the most accurate predicted data for the economic development of countries. A huge range of approaches have been used for this respect. Giannone et al. (2008) use a two step estimator where they combine principal component analysis with Kalman filtering techniques that are used to calculate expected values of common factors that account for colinearity in macroeconomic data.

Another approach was presented by Kuzin et al. (2011) who compare the success of a mixed data sampling with the mixed frequency var, in order to account for mixed frequency data (in our data, for example, GDP is recorded quarterly and the rest variables monthly. But no conclusive statements, other than their use as complements where reached. Ferrara & Simoni (2019) and Aastveit et al. (2014) also present alternatives to nowcast GDP growth based on regression analysis with some other details.

In the other hand, neural networks has been increasing its attention to nowcast information for different kinds of data in the last years as Mengyin et al. (2019) and Pasero & Moniaci (2004) prove and state in their work. For this work we use a neural network approach to nowcast the real GDP growth rate for Ecuador as a counter part of the work presented by Gonzalez-Astudillo & Baquero (2018), which is a time series approach. This work presents the application of a time series data structure into a neural network that is analyzed over a statistical distribution of its outcomes.

Further than present a neural network to nowcast pointwise the GDP growth, we create a distribution of predicted values for every observation of the data set. In this way we test the accuracy of our estimations. Thus, the result is an estimator for the quarterly GDP growth with a normal distribution, and the conclusions are based on the confidence intervals build upon the distribution of all the possible values that the network can take to produce the prediction. Also

the accuracy of pointwise neural networks with some different approaches on their architecture are tested. These modifications are not presented as distributions because the computational cost of these realizations are very high, however some interesting insights are reached. Firstly, we present how data is obtained and treated for the interest of the networks. Secondly, a methodological section discusses the methods by which estimations are done and the details about the neural network architecture. Finally, results are presented for each model by a classical statistical analysis due to the characteristics of the estimations, and the neural networks are evaluated upon their accuracy.

2 Data

2.1 Data Set

The original data set used for this study comes from the data set that Gonzalez-Astudillo & Baquero (2018) used for their paper, which they kindly share with me. This is a data set of macroeconomic variables used to elaborate a nowcasting model for the growth rate of the real GDP of Ecuador by a time series analysis approach. From this data I take only the variables that has observations for every month from January- 2000 to June - 2020. These variables correspond to specific economic sectors as explained in Table 1.

Table 1: Variables Description

Economic Sector	Variable	Description	Unit
External	exp_pet	Oil exports	Millions USD
External	exp_nopet	Non oil exports	Millions USD
External	imp_con	Consumption goods imports	Millions USD
External	imp_pri	Raw material imports	Millions USD
External	imp_cap	Capital goods imports	Millions USD

External	imp_combust	Oil imports	Millions USD
Fiscal	renta	Income tax	Millions USD
Fiscal	iva	Value added tax	Millions USD
Fiscal	precio_petro	Oil price	USD per barrel
Monetary	dep_vista	Sight deposits	Millions USD
Monetary	cuasidinero	Quasi-money	Millions USD
Monetary	cartera	Private portfolio to expire	Millions USD
Oil	exp_pet	Oil exports	Millions USD
Oil	imp_combust	Oil imports	Millions USD
Oil	prod_oil	National oil production	Thousands of barrels
Oil	precio_petro	Oil price	USD per barrel
Real	ipc	Consumer price index	2014=100
Real	ipc_nab	Consumer price index, no food or drinks	2014=100
Real	ipp	Productor price index	2015=100
Real	pib	GDP	Millions USD

Each one of these variables has monthly observations for the period stated above except for the GDP, which is recorded quarterly, in this case, we assign to every month in each quarter the GDP of the quarter.

To complete the original data set we add three lags of the GDP to each observation. In particular, for each month the first lag is the GDP of the last quarter, the second lag is the GDP of the second to last quarter before it and so on. In the same way, the rate of change of the GDP and three lags of it is assigned to each month. This is because the expected variation could be bias if we take into account only the level of the GDP and not it's variation. To complete our data set we add identifiers for each month, quarter and year, this is, twelve monthly identifiers per year, three for each quarter, and for each year, twenty one in total.

2.2 Data Transformations

It is clear that the structure of our data set consist on time series for each variable, which is the determinant factor to the application of neural networks. As Kim et al. (2004) explain, the use of neural networks over time series has gained attention in the analysis and prediction of financial indicators. But there exist one key characteristic in which this machine learning technique relies and it is the stationarity of the time series.

Stationarity is defined as a stochastic process $x_t : t = 1, 2, ...$ such that for every index set $t_1, ..., t_m : 1 \leq t_1 \leq ... \leq t_m$, the joint distribution of $x_{t_1}, .., x_{t_m}$ is the same as $x_{t_1+h}, .., x_{t_m+h}$ for any integer h. It is important when calculating estimators because it guarantee consistency of them, that is, as observations increase, the estimators approach to the real value. This is only obtained when the first and second momentum of the distribution of the estimators do not change along the time, which means estationarity.

On the other hand, the use of neural networks to predict values of a time series has stationarity as its main assumption as Kim et al. (2004) states, as it is the only way in which past observations can be related to present values. Also, as Butler & Kazakov (2011) show, error is minimized when stationarity is assumed.

But it is conceptually clear why these variables are non stationary, since macroeconomic indicators usually have trends and have some kind of seasonality, as well as the commodities prices. To make sure of the non stationarty of the variables, I run an Augmented Dickey Fuller Test as it is long stated in the literature as Mushtaq (2011) explains. The Augmented Dickey Fuller Test it the most popular unit root test, it regresses the observation over times on a lagged observation and a series of p lagged first differences, it tested the hypothesis that the coefficient of the lagged observation equals one, which means that it has a unit root.

After this test we failed to reject the null hypothesis of the Augmented Dickey Fuller Test, that is we do not have enough evidence to prove that the series are stationary. Furthermore, plots of the series make it clear. So we need to transform data to use a neural network approach. This test was carried out using python and looping for every series.

So the first thing we do to find a "more" stationary series is to take the moving average of variables over each quarter, which is a standard method to treat series that have a trend as Holt (2004) point out. Also , it accounts for the quarterly data of the GDP, that is, for mixed frequency in the data. This could be interpreted as if adding more information as the quarter occurs taking into account previous observations, modify the nowcasting process. That is, the data is transformed by the following equation

$$y_i = \frac{1}{i}\sum_{k=1}^{i} y_k$$

where $y_i \in X_j$ for $i = 1,2,3$ for some $j \in \{1,2,...,82\}$, and X is the set of all quarters of the data set indexed by j. After doing so, series are still stationary so we ended the process of transforming data by taking the difference of the logarithms of the observations over time, that is, for every y_i in the data set modified by the moving average we do the following transformation

$$y_i = \log y_i - \log y_{i-1}$$

We do this for every non stationary series, that is, the only series that we keep equal after the moving average are the lags of the variation of the GDP, since the variation is almost by definition a stationary time series. Note that we do take the difference of the logarithm for GDP level but not for the variations. After doing so we include the moving average of the percentage of the variables that are presented in millions of dollars with respect to the GDP level of the quarter, we include the dummy variables for the month, quarter and years identifiers and finally the oil price as its pure expression, since the level of this information is clearly a determinant for the growing rate of the GDP.

3 Methodology

To construct the nowcast model I build an artificial neural network, which is a statistical approach to processes of learning, that models posterior probabilities given a set of samples (Hristev, 1998). Thus, a neural network produces an outcome based on the input sample, this outcome is processed with weights that are obtained after the algorithm is carried out for the training sample. The training sample is a subset of the entire data set used for the learning process, from this subset the neural network receives the input and the output.

The neural network receive information from an input layer, that consist on all the variables that each observation has. After the input layer there exist a hidden layer, that is a layer where the neurons are located, each input neuron connects with every neuron in the hidden layer by the next equation

$$y_i = f(\sum_{j=0}^{n} x_j b_{ij} + \beta_i) \tag{1}$$

where y_i is the outcome that arrives to the neuron i of the hidden layer, f is the output function chosen by convenience, usually it is the sigmoidal function, x_j are the inputs neurons and b_{ij} are the weights associated the neuron i and β_i is the bias associated with the neuron i.

This process is repeated for every hidden layer, after the last one, the final output is calculated in the same way. The calculated output is compared with the output given, an error function is calculated, and based on this, weights are adjusted, in a process called back propagation. This is done several times until the desired outcome is reached.

Finally, the test data set is passed through the neural network and in the best case scenario, the outcomes should be the same as the real one.

3.1 First Approach

3.1.1 The algorithm

First of all, for the first neural network, the training set is chosen randomly. Since each observation has the lag of the GDP and its variations for three periods, as well as identifiers for the season of the year, then it is not necessary for the network to have short term memory. That is, to train the network, the first 75% of the data is chosen and the other 25% is assign as the test data.

The architecture of the neural network used for this model consisted on an input layer with the data described in the last section, one hidden layer and the output layer. The function that we choose, that is the function in 1, is the logistic signal function, which is sigmoidal and strictly increasing, it is given by

$$f(x) = \frac{1}{1+e^{-ax}} * b + c \tag{2}$$

where a is an scaling factor and we put $a = 1$, also, for us $b = 0.7$ and $c = -0.35$ so the function's range goes over the possible outcomes, that represent GDP variations. The first matrix of weights W_1 has the same number of rows as the amount of input neurons and the same number of columns as the number of neurons in the hidden layer, and the second matrix of weights W_2 has only one column and the same number of rows as the number of neurons in the hidden layer.

The first step, the dot product of the input neurons and the matrix of weights, for the first and second stage, is called feed forward. In this first attempt, our function does not receive any bias term. After doing this, the back propagation algorithm starts. It calculates the error through

the next function

$$E(W) = \sum_i (y_i - f(f(x_i W_1)W_2))^2 \tag{3}$$

To adjust the weights we take the derivative of the loss function with respect to the weights obtaining the next equation to adjust W_1

$$\frac{\partial E(W)}{\partial W_1} = 2(y_i - f(f(x_i W_1)W_2))f'(f(x_i W_1)W_2)W_2 f'(x_i W_1)X_i \tag{4}$$

and this one to adjust W_2

$$\frac{\partial E(W)}{\partial W_2} = 2(y_i - f(f(x_i W_1)W_2))f'(f(x_i W_1)W_2)f(x_i W_1) \tag{5}$$

We sum these derivatives evaluated to the weights and then we repeat the process. We repeat the process until the loss function reaches its minimum.

These whole algorithm was implemented in Python from the scratch, using only Numpy and Pandas libraries.

Predicting macroeconomic variables have the problem that there might be a lot of unobservable (or very difficult to measure) variables, such as political stability, worldwide economic status or people's expectations. This 'unobservables' are going to affect the nowcast of GDP growth, so we aim to capture them in a bias term included in the input of the function.

Thus, we add a bias term to the input for the activation function, recall that in the original model the outputs for each neuron h_i of the hidden layer are defined by

$$h_i = f(xW_{1i})$$

where f is the function defined in the equation 1. and the final output is given by

$$y_i = f(hW_2)$$

For this time we include a bias term that accounts for unobserved changes, so then the output in each neuron of the hidden layer is given by

$$h_i = f(xW_{1i} + b_i)$$

and the final output by

$$y_i = f(hW_2 + c)$$

. We also do this process for a neural network that has $5 * |X|$ neurons in the hidden layer, where $|X|$ is the size of the input layer, more precisely, the number of input variables.

3.1.2 The application

We get our results by repeating the process described above, one thousand of times. This approaches is taken with the aim of producing an estimator for the GDP nowcast, rather than a pointwise prediction. For every iteration we save the same test set, but we choose randomly 80% of the training set to train the network. After each training process, test data is evaluated in the final network. Thus, the result is the is a distribution of one thousand observations for the predicted value of the test set. Optimization with neural networks attempt to find a local minima of the loss function in the training set that can be generalized to any set of information that is given over all the possible solutions, that is the learning process. Thus, the random choice of the training set accounts for the possibility of finding an spurious minima if we take always the same training set, which could not be generalized to any set of input information (Aggarwalet al. , 2018).

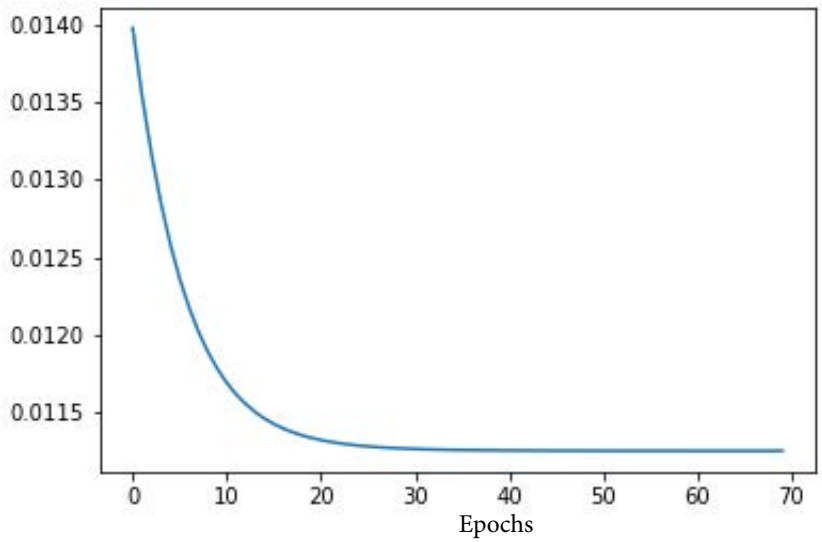
This whole process is iterated thirteen times, each one for different number of neurons in the hidden layers. The total number of neurons for each time is defined by

$$n_i x$$

for $n = 2, 2.5, 3, ..., 8$, by this process we can test if different amount of neurons produce better predictions. Thus, to evaluate the predicted values of the network we compare the actual value with the confidence interval of the distribution of predicted values for each observation.

To evaluate the evolution of the loss function over the epochs, random realizations of the training process are selected, obtaining the function described in Figure 1, where it can be seen that it reaches a minimum before the one hundred epochs. Although the loss function is strictly decreasing, a huge limitation of the architecture presented here for the neural network is that the loss function is bounded below at a level of 20% less than the initial loss value. Further modifications are required in order to decrease the loss function closer to zero.

Figure 1: Loss Function



3.2 Modifications

3.2.1 Not random training set

The first modification to the presented model is to take the same outcome distribution production by one thousand repetitions but for no random selected training set. As the result, we get one thousand predicted values for each observation on the last 25% of the observations. We do this process for a neural network that has $5 * |X|$ neurons in the hidden layer, where $|X|$ is the size of the input layer, more precisely, the number of input variables. This is to evaluate if the random assumption actually drives to better outcomes in the network.

3.2.2 Original data set

In order to prove if the literature agreements to smooth the data are accurate enough to generalize to this problem, we compare our result found in the initial application and in the first modification with the same model applied to the original data set. That is, without taking time average and differences of logarithms to the time series. As neural networks are also understand as a learning process that receives raw data from the environment (in this case the macroeconomic situation of the country) and predicts an output, this is the only way to prove if it is necessary to preprocess data to get better outcomes.

3.2.3 Learning rate

Every model above is computed with a learning rate of 0.01, for this modification we compute our network with a learning rate of 0.001. Clearly, this implies that more epochs to train the network are necessary, but this computational cost could compensate the bias in the precision

of the network generated by this rate. As we got better results with a non random choice of the training set we apply this modification to a non random selected training set.

3.2.4 Economic sector

The last modification to our model is motivated by the economic structure. As described in Table 1 the variables of each observation correspond to different economic sectors: real, monetary, fiscal, external and oil. Each sector has its own dynamics, but they also influence each other. In this section we assign input neurons to neurons in the hidden layer divided by their economic sector. Thus, we 'cut' the connections between input neurons and those assigned only to each sector. The amount of neurons assigned to each sector is divided equally, that is total number of neurons$/5$ for each sector.

3.3 Pointwise models

To compare results with a more traditional approach, we also produce models to have pointwise prediction for GDP growth nowcast. This models have the architecture explained above as no random, for different learning rates: 0.1, 0.01, 0.001, 0.0001 and 0.000001. Also we analyse the architecture of the network divided by economic sector for the same learning rates. In this case we evaluate the precision of the network by the root mean square error.

4 Results

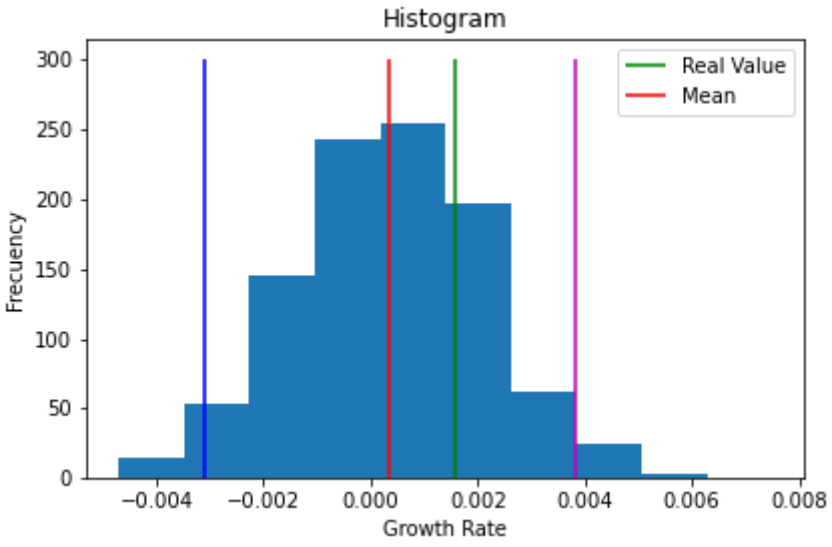
Results are presented as follows. Firstly, the number of times that the real value of the GDP growth relies between the confidence interval of our estimations for each model presented in the

last section. The a comparison between distributions of the predicted values of our models is presented. At the end, a ranking of the pointwise models is presented based on their root mean square error.

4.1 First Results

Recall that the first model presented as the neural network described in the methodology section, where the 25% of the data set is chosen as the test set, but each time that the network is trained only the 80% of the remaining 75% is chosen as the training set. The training set is chosen randomly from the 75% each time. When we apply the test set to the neural network we find that in the 30% of the time our network predicts with a 95% of confidence the real value of the GDP. To build the confidence intervals of our estimations we use 1.96 standard deviations away from the mean. To check if this is appropriate I ran a Shapiro-Wilk test to ensure normality. In every distribution of our results we fail to reject the null hypothesis at a 95% confidence level, thus we prove that distributions are normal. Distributions of estimated values look like Figure 2.

Figure 2: Random Sample Result Distribution

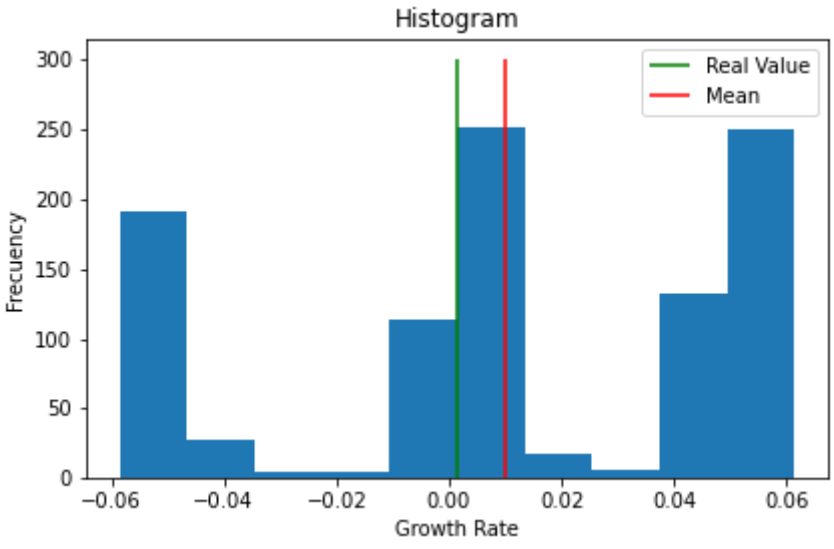


In the other hand, results given by the no random training set model give the exact same result, so no significant change is reached. While using a ten times smaller learning rate increases the predicted values to the 32%. Furthermore, the model that divides the input neurons with respect to their economic sector decreases the predicted values to the 24%.

4.2 Original data set

Now we compare these results with the network tested and trained with the original data set. That is, no moving average, logarithms or first differences are taken. For the first attempt, same architecture as the base model is taken, and randomly selected training sets are used to train the network each iteration. Same process is repeated, but no normal distribution is reached, every sample looks like the one shown in Figure 3. As it is clear by its distribution in the histogram, no normality can be assumed. Therefore, in order to test how accurate predictions are, two non parametric tests are performed: Kruskal Wallis and Wilcoxon tests. Both gives very different results. Kruskal Wallis performed to every distribution failed to reject the the null hypotheis that means are equal, giving a 100% accuracy. while Wilcoxon test gives 30%.

Figure 3: Original Data Trained



None of these test gives conclusive results given that no ergodicity is presented in time series that are used to train the network. Thus, different approaches are needed to compare the performance of the network with raw data. As Figure 3 shows, results suggest non linear relations. To eliminate linearity from the model, one more hidden layer is added to the network. Although normality is reached with this new architecture, we get no better predictions, furthermore, all the predictive power is lost. Under this approach it is easily seen that smooth data produced better outcomes than row data.

4.3 Pointwise predictions

For our pointwise prediction the results correspond to a no random training set model with different learning rates, and the model that divides neurons of the input layer according to their economic sector. The Table 2 ranks these models in this sense. It shows that the optimal learning rate is 0.0001 for both types of network. Greater learning rates do not produce comparable outcomes since RMSE produced there are more than one hundred times bigger.

Table 2: Root Mean Square Error

Type	Learning Rate	RMSE
No sector	0.0001	0.0087
Sector	0.0001	0.0087
No sector	0.000001	0.0088
Sector	0.000001	0.0091
No sector	0.001	0.0103
Sector	0.001	0.0099

5 Conclusion

Learning rates in our model, as well as the amount of neurons and layers do not present a linear relation with respect to the accuracy of predictions. Furthermore, treated data produce not only more comparable outcomes with the real values of the predicted data, but more accurate predictions with respect to the distribution of the estimators. Normality is given as a consequence of data treatment, however the estimators distribution suggest that a more non linear approach may be taken in order to analyse outcomes produced by the raw data.

Using neural networks to predict time series data present a lot of particularities that depend on the hyper parameters and on the data structure itself. Further investigation is needed in order to improve the accuracy of the model. As it is taken as an estimator, computational cost may result high when recurrent methods are taken. Also more statistical analysis is required in order to compare results produced by original data set with the modified data structure, since approaching normality take a very high opportunity cost in the accuracy of the model.

6 References

Aastveit, K. A., Gerdrup, K. R., Jore, A. S., & Thorsrud, L. A. (2014). Nowcasting gdp in real time: A density combination approach. *Journal of Business & Economic Statistics*, *32*(1), 48–68.

Aggarwal, C. C. et al. (2018). *Neural networks and deep learning*. Springer.

Butler, M. & Kazakov, D. (2011). The effects of variable stationarity in a financial time-series on artificial neural networks. In *2011 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, (pp. 1–8). IEEE.

Ferrara, L. & Simoni, A. (2019). When are google data useful to nowcast gdp? an approach via pre-selection and shrinkage.

Giannone, D., Reichlin, L., & Small, D. (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, *55*(4), 665–676.

Gonzalez-Astudillo, M. & Baquero, D. (2018). A nowcasting model for the growth rate of real gdp of ecuador: Implementing a time-varying intercept.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, *20*(1), 5–10.

Hristev, R. (1998). The ann book.

Kim, T. Y., Oh, K. J., Kim, C., & Do, J. D. (2004). Artificial neural networks for non-stationary time series. *Neurocomputing*, *61*, 439–447.

Kuzin, V., Marcellino, M., & Schumacher, C. (2011). Midas vs. mixed-frequency var: Nowcasting gdp in the euro area. *International Journal of Forecasting*, *27*(2), 529–542.

Mengyin, F., Jiapeng, K., Tong, L., Xingyv, L., & Kai, W. (2019). The kp index nowcast method based on neural network. In *2019 Chinese Control Conference (CCC)*, (pp. 3411–3415). IEEE.

Mushtaq, R. (2011). Augmented dickey fuller test.

Pasero, E. & Moniaci, W. (2004). Artificial neural networks for meteorological nowcast. In *2004 IEEE International Conference onComputational Intelligence for Measurement Systems and Applications, 2004. CIMSA.*, (pp. 36–39). IEEE.