Rafael Rockenbach da Silva Guimarães

Deep Learning Macroeconomics

Porto Alegre

2021

Rafael Rockenbach da Silva Guimarães

Deep Learning Macroeconomics

Thesis presented in partial fulfillment of the requirements for the degree of Doctor in Economics.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL – UFRGS
FACULDADE DE CIÊNCIAS ECONÔMICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ECONOMIA

Supervisor: Prof. Dr. Sérgio Marley Modesto Monteiro

Porto Alegre
2021

Rafael Rockenbach da Silva Guimarães

Deep Learning Macroeconomics

Thesis presented in partial fulfillment of the requirements for the degree of Doctor in Economics.

Aprovada em: Porto Alegre, xx de xxxxxxxxxxxxxx de 2021.
BANCA EXAMINADORA:

Prof. Dr. Sérgio Marley Modesto Monteiro-Orientador
Universidade Federal do Rio Grande do Sul
- UFRGS

Prof. Dr. xxxxx
Universidade xxx - UFxx

Prof. Dr. xxxxx
Universidade xxx - UFxx

Prof. Dr. xxxxx
Universidade xxx - UFxx

This thesis work is dedicated to Fernanda, Rafaela and Felipe Preto Guimarães.

Acknowledgements

Foremost,

Resumo

A econometria, o ramo da economia preocupado com a aplicação de métodos estatísticos e matemáticos aos problemas econômicos, é uma área de pesquisa relevante e em constante expansão. Uma das questões sempre abertas é onde usar um determinado método, principalmente quando apresentar desempenho notável em outras áreas do conhecimento, como a aprendizagem profunda atualmente. Várias técnicas de aprendizado de máquina são incorporadas à estrutura econométrica, mas o aprendizado profundo continua focado na previsão de séries temporais. As abordagens propostas aqui estão relacionadas ao reconhecimento de padrões, onde o aprendizado profundo alcançou desempenho de ponta. Quando aplicamos a econometria a problemas macroeconômicos, conjuntos de dados limitados e relações não lineares complexas estão entre os desafios que podem surgir. Esta pesquisa propõe a aprendizagem profunda como uma abordagem para transferir aprendizagem no primeiro caso e para mapear relações entre variáveis no último caso.

Primeiramente, a aprendizagem por transferência é proposta como uma estratégia adicional para a macroeconomia empírica. Ao desenvolver estratégias de modelagem econômica, a falta de dados pode ser um problema que a aprendizagem por transferência pode corrigir. Começamos por apresentar conceitos teóricos relacionados à transferência de aprendizagem e propomos uma conexão com uma tipologia relacionada a modelos macroeconômicos. Em seguida, exploramos a estratégia proposta empiricamente, mostrando que os dados de domínios diferentes, mas relacionados, um tipo de aprendizagem por transferência, ajudam a identificar as fases do ciclo de negócios quando não há comitê de datação do ciclo de negócios e a estimar rapidamente um hiato do produto de base econômica. Em ambos os casos, a estratégia também ajuda a melhorar o aprendizado quando os dados são limitados. A abordagem integra a ideia de armazenar conhecimento obtido de especialistas em economia de uma região e aplicá-lo a outras áreas geográficas. O primeiro é capturado com um modelo de rede neural profunda supervisionado e o segundo aplicando-o a outro conjunto de dados, um procedimento de adaptação de domínio. No geral, há uma melhora na classificação com a aprendizagem por transferência em comparação com os modelos de base. Até onde sabemos, a abordagem combinada de aprendizagem profunda e transferência é subutilizada para aplicação a problemas macroeconômicos, indicando que há muito espaço para o desenvolvimento de pesquisas.

Em segundo lugar, uma vez que os métodos de aprendizagem profunda são uma forma de aprender representações, aquelas que são formadas pela composição de várias transformações não lineares, para produzir representações mais abstratas, aplicamos a aprendizagem profunda para mapear variáveis de baixa frequência a partir de variáveis de alta frequência, uma tarefa também conhecido como interpolação, distribuição e extrapolação de séries temporais por séries relacionadas.

Os excelentes resultados obtidos mostram a adequação de modelos de aprendizagem profunda aplicados a problemas macroeconômicos. Primeiro, os modelos aprenderam a classificar os ciclos

de negócios dos Estados Unidos corretamente. Em seguida, aplicando o aprendizado de transferência, eles alcançaram um desempenho excepcional na identificação dos ciclos de negócios de dados brasileiros e europeus fora da amostra. Na mesma linha, os modelos aprenderam a estimar o hiato do produto com base nos dados americanos e obtiveram excelente desempenho frente aos dados brasileiros. Em ambos os casos, a estratégia proposta surge como uma ferramenta suplementar potencial para os governos e o setor privado conduzirem suas atividades à luz das condições econômicas nacionais e internacionais. Além disso, o aprendizado profundo se mostrou adequado para mapear variáveis de baixa frequência a partir de dados de alta frequência para interpolar, distribuir e extrapolar séries temporais por séries relacionadas. A aplicação dessa técnica em dados brasileiros mostrou-se compatível com benchmarks baseados em outras técnicas.

Abstract

Econometrics, the branch of economics concerned with applying statistical and mathematical methods to economic problems, is a relevant and constantly expanding area of research. One of the questions always open is where to use a particular method, especially when presenting remarkable performance in other areas of knowledge, as deep learning nowadays. Several machine learning techniques are incorporated into the econometric framework, but deep learning remains focused on time-series forecasting. The approaches proposed here are related to pattern recognition, where deep learning has achieved state-of-the-art performance. When we apply econometrics to macroeconomic problems, limited datasets and complex nonlinear relationships are among the challenges that may emerge. This research proposes deep learning as an approach to transfer learning in the former case and to map relationships between variables in the latter case.

Firstly, transfer learning is proposed as an additional strategy for empirical macroeconomics. When developing economics modeling strategies, the lack of data may be an issue that transfer learning can fix. We start presenting theoretical concepts related to transfer learning and proposed a connection with a typology related to macroeconomic models. Next, we explore the proposed strategy empirically, showing that data from different but related domains, a type of transfer learning, helps identify the business cycle phases when there is no business cycle dating committee and to quick estimate a economic-based output gap. In both cases, the strategy also helps to improve the learning when data is limited. The approach integrates the idea of storing knowledge gained from one region's economics experts and applying it to other geographic areas. The first is captured with a supervised deep neural network model, and the second by applying it to another dataset, a domain adaptation procedure. Overall, there is an improvement in the classification with transfer learning compared to baseline models. To the best of our knowledge, the combined deep and transfer learning approach is underused for application to macroeconomic problems, indicating that there is plenty of room for research development.

Secondly, since deep learning methods are a way of learning representations, those that are formed by the composition of multiple non-linear transformations, to yield more abstract representations, we apply deep learning for mapping low-frequency from high-frequency variables, a task also known as interpolation, distribution, and extrapolation of time series by related series.

The outstanding results obtained show the suitability of deep learning models applied to macroeconomic problems. First, models learned to classify United States business cycles correctly. Then, applying transfer learning, they achieved outstanding performance in identifying the business cycles of out-of-sample Brazilian and European data. Along the same lines, the models learned to estimate the output gap based on the U.S. data and obtained excellent performance when faced with Brazilian data. In both cases, the proposed strategy emerges as a potential supplementary tool for governments and the private sector to conduct their activities in the light of national and international economic conditions. Additionally, deep learning proved adequate for

mapping low-frequency variables from high-frequency data to interpolate, distribute, and extrapolate time series by related series. The application of this technique to Brazilian data proved to be compatible with benchmarks based on other techniques.

Keywords: business cycle, output gap, transfer learning, deep learning, frequency conversion.

List of Figures

List of Tables

List of abbreviations and acronyms

AI	Artificial intelligence
ANN	Artificial neural networks
AUC	Area under the ROC curve
BCB	Central Bank of Brazil
CBO	Congressional Budget Office
CEPR	Centre for Economic Policy Research
CODACE	Brazilian Business Cycle Dating Committee
COVID-19	Coronavirus Disease 2019
DFMS	Dynamic factor model with Markov-switching
DL	Deep learning
DSGE	Dynamic stochastic general equilibrium
EURO-19	The 19 selected countries in Europe.
FRED	Federal Reserve Economic Data
FNN	Forward neural networks
FUNCEX	Foundation for Foreign Trade Studies Center
GDP	Gross domestic product
GNP	Gross national product
HP	Hodrick-Prescott
IBC-Br	Brazilian Central Bank Economic Activity Index
IBGE	Brazilian Institute of Geography and Statistics
IPEA	Institute of Applied Economic Research
IS-LM	Investment-savings and liquidity preference-money supply
LSTM	Long short-term memory

MAE	Mean absolute error
ML	Machine learning
MLN	Markov logic networks
MS	Markov-switching
NBER	National Bureau of Economic Research
NLP	Natural language processing
RBC	Real business-cycle
ReLU	Rectified linear unit
RIDE	Representation learning for interpolation, distribution and extrapolation of time series by related series
RNN	Recurrent neural network
SGS	Series Management System
SVM	Support vector machines
Tanh	Hyperbolic tangent function
TFP	Total Factor Productivity
TL	Transfer learning
U.S.	United States of America
VAR	Vector autoregression

Contents

1 Introduction

Econometrics, the branch of economics concerned with applying statistical and mathematical methods to economic problems, is a relevant and constantly expanding area of research. One of the questions always open is where to use a particular method, especially when presenting remarkable performance in other areas of knowledge, as deep learning nowadays. Several machine learning techniques are incorporated into the econometric framework, but deep learning remains focused on time-series forecasting. The approaches proposed here are related to pattern recognition, where deep learning has achieved state-of-the-art performance. When we apply econometrics to macroeconomic problems, limited datasets and complex nonlinear relationships are among the challenges that may emerge. This research proposes deep learning as an approach to transfer learning in the former case and to map relationships between variables in the latter case.

A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold (Pan and Yang, 2010). Can we use information from *adult* humans to train an intelligent system for diagnosing *infant* heart disease? Such a problem is known as *transfer learning* (Figure 1). The population of interest is called the target domain, for which labels are usually not available and training a classifier might be not possible. However, if data from a similar population is available, it could be used as a source of additional information (Kouw and Loog, 2018, 2). For example, we may find that learning to recognize apples might help to recognize pears. Similarly, learning to play the eletronic organ may help facilitate learning the piano. The study of transfer learning (TL) is motivated by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions (Pan and Yang, 2010, 2). Thus, TL refers 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). The learner must perform two or more different tasks, but it is assumed that many of the factors that explain the variations in P1 are relevant to the variations that need to be captured for learning P2 (Goodfellow et al., 2016, 534).

Artificial intelligence (AI) has recently gained considerable prominence due to performances in autonomous vehicles, intelligent robots, image and speech recognition, automatic translations, and medical and law usage (Makridakis, 2017). In Economics, the application of machine learning (ML) methods, an AI technique, 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
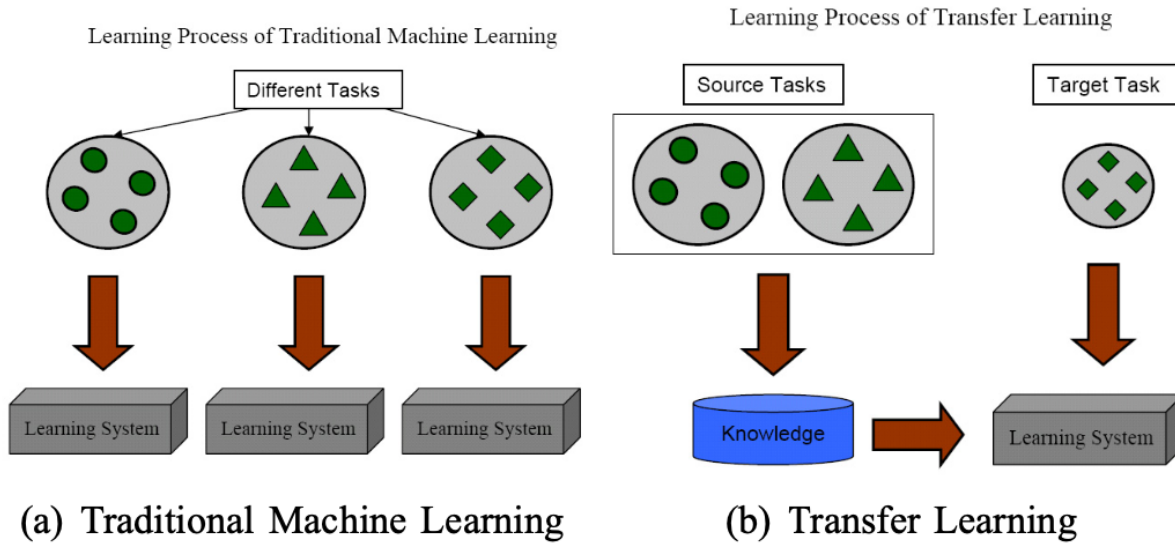
Figure 1 – Different Learning Process between Traditional Machine Learning and Transfer Learning - reproduced from Pan and Yang (2010)

has been observed in many areas, attracting attention and research funding.

Meanwhile in Macroeconomics, according to (Vines and Wills, 2018, 2), during the Great Moderation (mid-1980s to 2007), the New Keynesian Dynamic Stochastic General Equilibrium (DSGE) model had become the benchmark model: the one taught to students at the start of the first-year graduate macro course. But the benchmark model has let us down; it explained neither why the Great Financial Crisis (2008) happened, nor what to do about it. What new ideas are needed? What needs to be thrown away? What might a new benchmark model look like? Will there be a paradigm shift?

Focus on pattern recognition is one way to advance. If this is true that the transfer knowledge capacity, or more broadly the representation learning, observed in other scientific areas must hold in economics, we face an underutilized tool that can contribute to applied macroeconomics. First, in Chapter 2, we present theoretical concepts related to transfer learning and propose a connection with a typology related to macroeconomic models. Then, in Chapter 3 we discuss the empirical setup and, in Chapters 4 and 5, we advance empirically by exploring transfer learning capability when applied to business cycle identification and output gap estimation. Lastly, in Chapter 6 we explore deep learning for mapping the low-frequency gross domestic product from high-frequency variables.

2 Transfer Learning and Macroeconomics

2.1 Macroeconomic models

More than forty years ago, when Sims (1980) proposed vector autoregressions (VARs) as an alternative strategy for empirical macroeconomics[1], he asserted that existing econometric analysis strategies related to macroeconomics were subject to a number of serious objections, some recently formulated, some old. For instance, empirical macroeconomists sometimes express frustration at the limited amount of information in economic time series, and it does not infrequently turn out that models reflecting rather different behavioral hypotheses fit the data about equally well (Sims, 1980, 15).

Recently, Vines and Wills (2018), in the Rebuilding Macroeconomic Theory Project, claims that the need to change macroeconomic theory is similar to the situation in the 1930s, at the time of the Great Depression, and in the 1970s, when inflationary pressures were unsustainable. They asked a number of leading macroeconomists to describe how the benchmark New Keynesian model might be rebuilt in the wake of the 2008 crisis (Vines et al., 2018).

As one can see, it has been an area in constant evolution, but still with many challenges. The number of macroeconomic models is countless, and we see no restriction to use, or at least to try, the strategy proposed in Section 2.2 to any of them. Still, here we restrict our environment to the typology proposed by Blanchard (2018). According to him (Blanchard, 2018, 43-44), there should be five kinds of general equilibrium models: a common core, plus foundational theory, policy, toy, and forecasting models. The different classes of models have a lot to learn from each other, but the goal of full integration has proven counterproductive. We need different macroeconomic models for different purposes. One type is not better than the others. (Blanchard, 2018, 52-53) explains each type:

1. The purpose of the DSGE models is to explore the macro implications of distortions or sets of distortions. To allow for a productive discussion, they must be built around a largely agreed-upon *common core.* Each model then explores additional distortions, be it bounded rationality, asymmetric information, different forms of heterogeneity, etc. These models should aim to be close to reality, but not through *ad hoc* additions and repairs, such as arbitrary and undocumented higher-order costs introduced only to deliver more realistic lag structures. Fitting reality closely should be left to policy models.

---

[1] VARs are useful statistical devices for evaluating alternative macroeconomic models. His suggestion has stood the test of time well. In the early days, VARs played an important role in the evaluation of alternative models. They continue to play that role today (Christiano, 2012).

2. *Foundational* models' purpose is to make a deep theoretical point, likely of relevance to nearly any macro model, but not pretending to capture reality closely. Some examples of this type are the consumption-loan model of Paul Samuelson, the overlapping generation model of Peter Diamond, the models of money by Neil Wallace or Randy Wright, the equity premium model of Edward Prescott and Rajnish Mehra, the search models of Peter Diamond, Dale Mortensen, and Chris Pissarides.

3. The purpose of *policy* models is to help policy, study the dynamic effects of specific shocks, and explore alternative policies. For these models, capturing actual dynamics is clearly essential. So is having enough theoretical structure that the model can be used to trace the effects of shocks and policies. But the theoretical structure must by necessity be looser than for DSGE: aggregation and heterogeneity lead to much more complex aggregate dynamics than a tight theoretical model can hope to capture. Old-fashioned policy models started from theory as motivation and then let the data speak, equation by equation. Some new-fashioned models start from a DSGE structure and then let the data determine the richer dynamics. In any case, for this class of models, the rules of the game here must be different than for DSGEs. Does the model fit well, for example, in the sense of being consistent with the dynamics of a VAR characterization? Does it capture well the effects of past policies? Does it allow us to think about alternative policies?

4. Examples of *toy* models are the many variations of the IS-LM model, the Mundell–Fleming model, the RBC model, and the New Keynesian model. As the list indicates, some may be only loosely based on theory, others more explicitly so. But they have the same purpose. They allow for a quick first pass at some questions and present the essence of the answer from a more complicated model or a class of models. For the researcher, they may come before writing a more elaborate model or after, once the elaborate model has been worked out. How close they remain formally to theory is not a relevant criterion here. They are art as much science. But art is of much value.

5. The purpose of *forecasting* models is straightforward: give the best forecasts. And this is the only criterion by which to judge them. If a theory is useful in improving the forecasts, then the theory should be used. If it is not, it should be ignored. The issues are then statistical, from over-parameterization to how to deal with the instability of the underlying relations.

2.2 An additional strategy for empirical macroeconomics

When developing one or more modeling strategies described in Section 2.1, the lack of data may be an issue that transfer learning helps to fix. As we show empirically in chapters 4 and 5, data from different but related domains, a type of transfer learning, help improve the task at hand.

Then, for empirical macroeconomics, it should be feasible to apply transfer learning on data as an additional strategy to either (Pan and Yang, 2010, 5)

- *instance-transfer*: re-weight some labeled data in the source domain for use in the target domain.

- *feature-representation-transfer*: find a good feature representation that reduces difference between the source and the target domains and the error of classification and regression models.

- *parameter-transfer*: discover shared parameters or priors between the source domain and target domain models, which can benefit for transfer learning.

- *relational-knowledge-transfer*: build mapping of relational knowledge between the source and the target domains.

Similar to the characterization of *toy models* by Blanchard (2018), applying transfer learning to macroeconomics is an art as much science, but art is of much value. The first step in developing such an approach is collecting, organizing, and evaluating the task-related available data. As a data-driven method, transfer learning performance relies on the quality of the data. Next, the researcher must choose one or more machine learning algorithms that will learn with the data. Pan and Yang (2010) and Weiss et al. (2016) review some options, and in Section 3.1.1, we describe deep learning, our choice. Finally, one evaluates the results, transfers learning, and compares it with the outcomes of other modeling strategies if it is the case.

In the conventional transfer learning approach, we first train a base network on a base dataset and task, and then we repurpose the learned features or transfer them to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task (Yosinski et al., 2014). Figure 2 illustrates these dynamics[2].

The foregoing discussion is based on a relevant, comprehensive survey about transfer learning by Pan and Yang (2010). Consider that a *domain* $\mathcal{D}$ consist of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$. For example, if our learning task is a document classification, and each term is taken as a binary feature, then $\mathcal{X}$ is the space of all term vectors, x_i is the ith term vector corresponding to some documents, and X is a particular learning sample. In general, if two domains are different, then they may have

---

[2] The base networks (top two rows) are trained using standard deep learning procedures on datasets A and B. The labeled rectangles (e.g., WA1) represent the weight vector learned for that layer, with the color indicating which dataset the layer was originally trained on. The vertical, ellipsoidal bars between weight vectors represent the activations of the network at each layer. The target networks (bottom two rows) represent transfer learning strategies. The first n weight layers of the network (in this example, n = 3) are copied from a network trained on one dataset (e.g., A), and then the entire network is trained on the other dataset (e.g., B). Usually, the first n layers are either locked during training or allowed to learn.
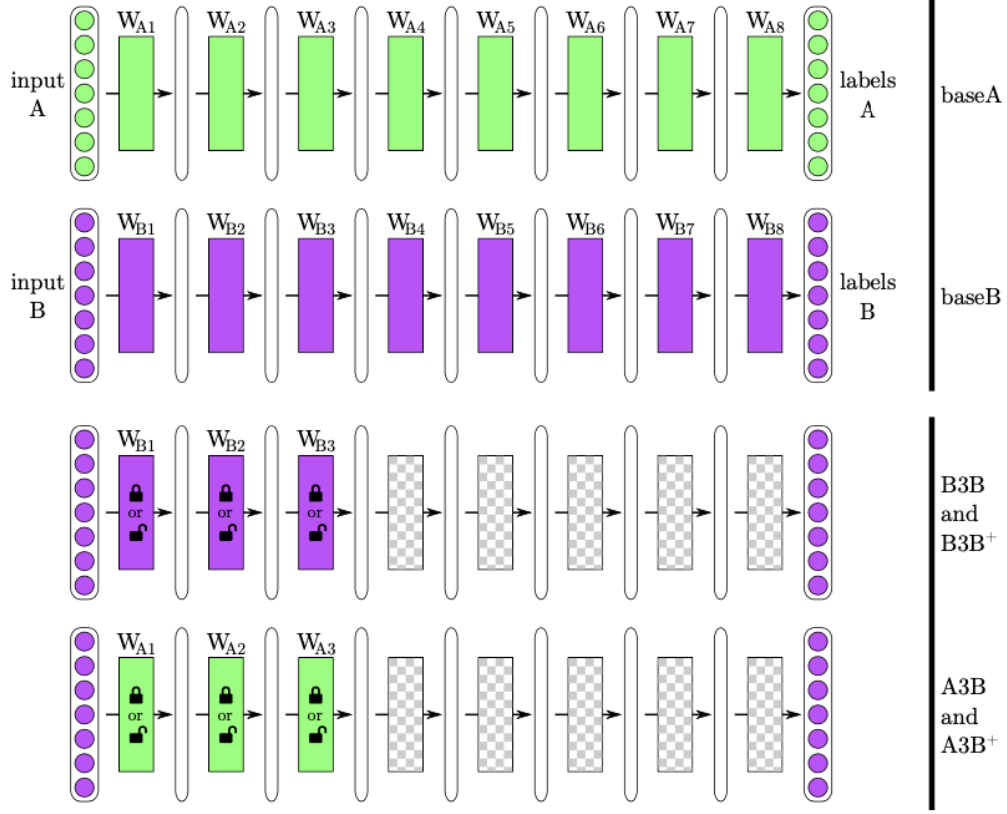
Figure 2 – Transfer learning overview - reproduced from Yosinski et al. (2014)

different feature spaces or different marginal probability distributions. Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a *task* consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$, denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, which is not observed but can be learned from the training data, which consist of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be used to predict the corresponding label, $f(x)$, of a new instance x. From a probabilistic viewpoint, $f(x)$ can be written as $P(y|x)$. In our document classification example, $\mathcal{Y}$ is the set of all labels, which is True, False for a binary classification task, and y_i is "True" or "False". For simplicity, lets consider the case where there is one source domain $\mathcal{D}_S$, and one target domain, $\mathcal{D}_T$, as this is by far the most popular of the research works in the literature. More specifically, lets denote the *source domain data* as $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_S}}, y_{S_{n_S}})\}$, where $x_{S_i} \in \mathcal{X}_S$ is the data instance and $y_{S_i} \in \mathcal{Y}_S$ is the corresponding class label. In the document classification example, $\mathcal{D}_S$ can be a set of term vectors together with their associated true or false class labels. Similarly, lets denote the *target domain data* as $\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), ..., (x_{T_{n_T}}, y_{T_{n_T}})\}$, where the input x_{T_i} is in $\mathcal{X}_T$ and $y_{T_i} \in \mathcal{Y}_T$ is the corresponding output. In most cases, $0 \leq n_T \ll n_S$ (Pan and Yang, 2010, 2-3).

Definition 1 *(Transfer Learning)* Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, *transfer learning* aims to help improve the learning of the target

predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

In the above definition, a domain is a pair $\mathcal{D} = \{\mathcal{X}, P(X)\}$. Thus the condition $\mathcal{D}_S \neq \mathcal{D}_T$ implies that either $\mathcal{X}_S \neq \mathcal{X}_T$ or $P_S(X) \neq P_T(X)$. Similarly, a task is defined as a pair $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$. Thus the condition $\mathcal{T}_S \neq \mathcal{T}_T$ implies that either $\mathcal{Y}_S \neq \mathcal{Y}_T$ or $P(Y_S|X_S) \neq P(Y_T|X_T)$. When the target and source domains are the same, i.e., $\mathcal{D}_S = \mathcal{D}_T$, and their learning tasks are the same, i.e., $\mathcal{T}_S = \mathcal{T}_T$, the learning problem becomes a traditional machine learning problem. When the domains are different, then either (1) the feature spaces between the domains are different, i.e., $\mathcal{X}_S \neq \mathcal{X}_T$, or (2) the feature spaces between the domains are the same but the marginal probability distributions between domain data are different; i.e., $P(X_S) \neq P(X_T)$, where $X_{S_i} \in \mathcal{X}_S$ and $X_{T_i} \in \mathcal{X}_T$. Pan and Yang (2010) summarized the relationship between traditional machine learning and various transfer learning settings, categorizing transfer learning under three sub-settings (Table 1), then we can read **Definition 1** as: Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$,

1. *inductive transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$.

2. *transductive transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$.

3. *unsupervised transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{Y}_S$ and $\mathcal{Y}_T$ are not observable.

Table 1 – Relationship between Traditional Machine Learning and Transfer Learning

Learning Settings	Source and Target Domains	Source and Target Tasks
Traditional Machine Learning	the same	the same
Inductive Transfer Learning	the same	different but related
Transductive Transfer Learning	different but related	the same
Unsupervised Transfer Learning	different but related	different but related

2.2.1 Inductive Transfer Learning

Definition 2 *(Inductive Transfer Learning)* Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a learning task $\mathcal{T}_T$, *inductive transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$.

Based on the above definition, a few labeled data in the target domain are required as the training data to *induce* the target predictive function. This setting has two cases: (1) labeled data in the source domain are available; (2) labeled data in the source domain are unavailable while unlabeled data in the source domain are available. Most transfer learning approaches in this setting focus on the former case (Pan and Yang, 2010, 4-5). Inductive transfer learning can be used to instance-transfer, feature-representation-transfer, parameter-transfer and relational-knowledge-transfer.

2.2.2   Transductive Transfer Learning

Definition 3 *(Transductive Transfer Learning)* Given a source domain $\mathcal{D}_S$ and a corresponding learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a corresponding learning task $\mathcal{T}_T$, *transductive transfer learning* aims to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$. In addition, some unlabeled target domain data must be available at training time.

This definition covers the work of Arnold et al. (2007), since they considered *domain adaptation*, where the difference lies between the marginal probability distribution of source and target data; i.e., the tasks are the same but the domains are different (Pan and Yang, 2010, 8). Transductive transfer learning can be used to instance-transfer and feature-representation-transfer, and it is the setting we explore in chapters 4 and 5 in two macroeconomic problems well suited for this approach, business cycle identification and output gap estimation, respectively. The intuitive behind the feature-representation-transfer is to learn a good feature representation for the target domain. In this case, the knowledge used to transfer across domains is encoded into the learned feature representation. With the new feature representation, the performance of the target task is expected to improve significantly (Pan and Yang, 2010, 4).

2.2.3   Unsupervised Transfer Learning

Definition 4 *(Unsupervised Transfer Learning)* Given a source domain $\mathcal{D}_S$ with a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a corresponding learning task $\mathcal{T}_T$, *unsupervised transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$ and $\mathcal{Y}_S$ and $\mathcal{Y}_T$ are not observable.

Based on this definition, no labeled data are observed in the source and target domains in training, since in unsupervised transfer learning, the predicted labels are latent variables, such as clusters or reduced dimensions (Pan and Yang, 2010, 10). Unsupervised transfer learning can be used to feature-representation-transfer.

2.2.4 Transfer Learning settings and Macroeconomic typologies

As mentioned in 2.2.2, we apply transductive transfer learning in chapters 4 and 5. However, depending on the problem type and the available data, the path may differ, as shown in Figure 3.
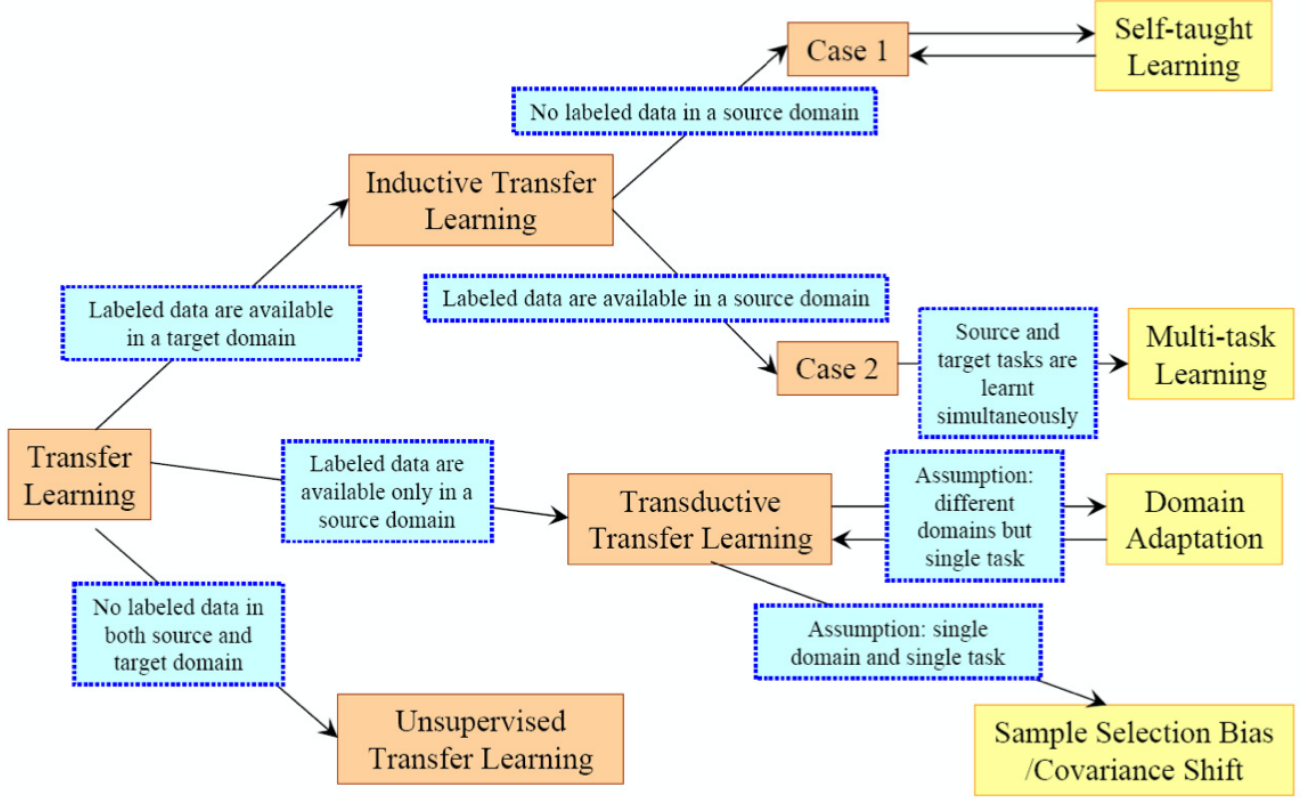
Figure 3 – Different settings of transfer - reproduced from Pan and Yang (2010)

Starting through the shortest path, when there is *no labeled data in both source and target domain*, generally a situation in which we have data, possibly even a lot of data, but not a clear definition or classification of the existing relationships. This scenario is one of the most challenging and one of the most promising, as it is minimal or even no human intervention in assigning the relationship between the data. Some possible macroeconomics applications are clustering for regional studies and dimensionality reduction for input variables.

Next, Figure 3 shows two related cases to inductive transfer learning, where *labeled data are available in a target domain*, and source and target domains are the same. In case 1, the self-taught learning, there is no labeled data in a source domain, while in case 2, the multi-task learning, labeled data are available in a source domain, where the source and target tasks are learned simultaneously. From a macroeconomics perspective, relevant application could be finding priors, or initialization parameters for models, or applying natural language processing (NLP) for sentiment analysis on monetary policy committee statements. Suppose a sentiment analysis

problem in which the researcher wants to classify monetary policy committee statements of a given country in hawkish or dovish. Still, there is insufficient textual data because it is a recently implemented regime in the country. The researcher can then use a model previously trained with data from other countries and then use an inductive transfer learning strategy.

Lastly, the two situations of transductive transfer learning, where *labeled data are available only in a source domain* with two alternative assumptions: different domains but single task, the domain adaptation case, and single domain and task, the sample selection bias/covariance shift. The latter is a recurring problem in macroeconomics and, therefore, with well-developed frameworks, which does not prevent considering transfer learning as an additional option. The former case, domain adaptation, is what we explore in chapters 4 and 5.

Not all combinations between the macroeconomic models (Blanchard, 2018) typologies and the transfer learning (Pan and Yang, 2010) settings will necessarily succeed, but certainly, each one can be explored. In Figure 4 we speculate about some possibilities.

DSGE model	Inductive transfer learning	Parameter-transfer
Task: prior view about the maximum length of the inflation deviation from the inflation target after a demand shock.	Limitation: there is just one demand shock in the data from country A (target). Approach: select "similar" countries B and C to compose the source dataset together with data from country A. Then, label data are available in the target and source domains, the multi-task learning case.	Result: transfer knowledge through the shared parameters of the source and target domain learner models to choose the prior's value.

Policy model	Transductive transfer learning	Instance-transfer
Task: study the dynamic effects of specific shocks and explore alternative policies with less theoretical structure than one required by a DSGE model.	Limitation: time and resources to develop a structural model. Approach: select another structural model for the country of interest containing labeled data to train a model focusing on new specific shocks. Then, label data are available only in the source domain.	Result: reweight some labeled data in the source domain for use in the target domain.

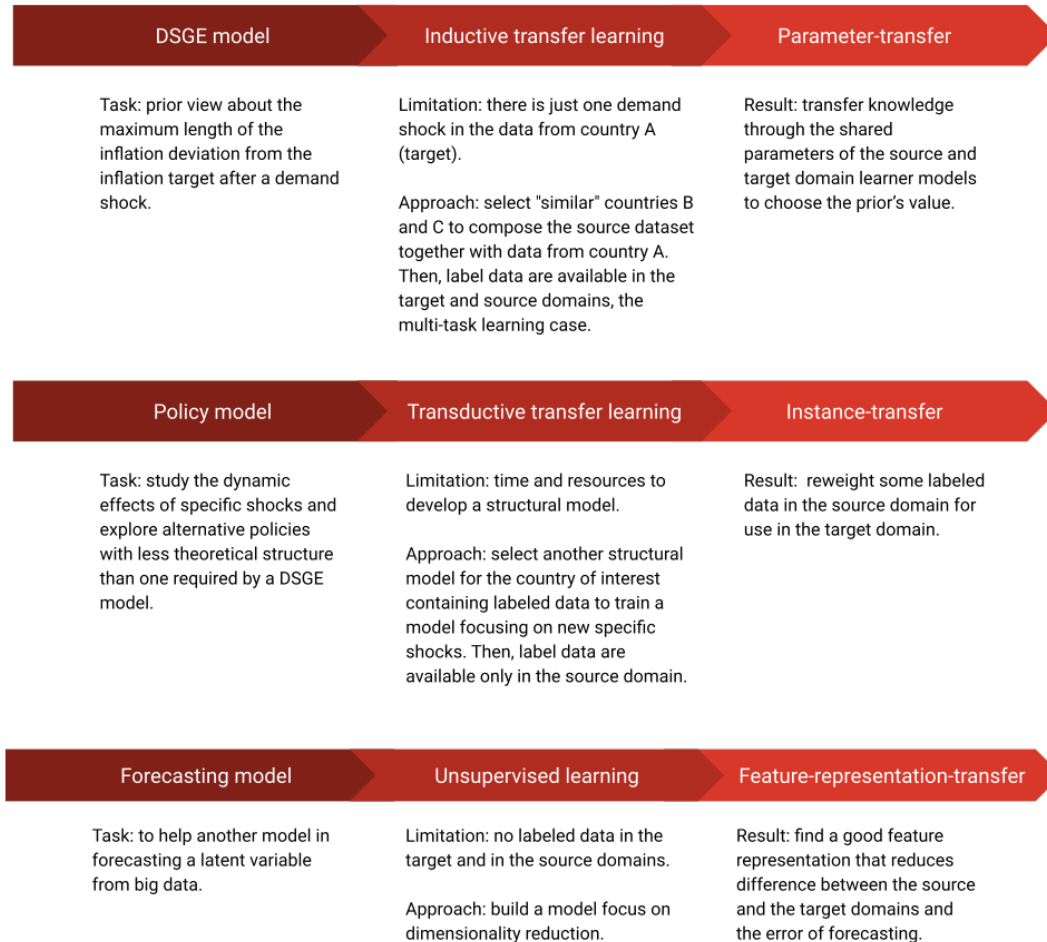| Forecasting model	Unsupervised learning	Feature-representation-transfer
Task: to help another model in forecasting a latent variable from big data.	Limitation: no labeled data in the target and in the source domains. Approach: build a model focus on dimensionality reduction.	Result: find a good feature representation that reduces difference between the source and the target domains and the error of forecasting.

Figure 4 – Transfer Learning settings and Macroeconomic typologies

3 Empirical Setup

3.1 Algorithms for Transfer Learning

After discovering which knowledge can be transferred, learning algorithms need to be developed to transfer the knowledge (Pan and Yang, 2010, 3). To name a few, Dai et al. (2007a) proposed a boosting algorithm, *TrAdaBoost*, which is an extension of the *AdaBoost* algorithm, to address the inductive transfer learning problems, Jebara (2004) proposed to select features for multi-task learning with Support Vector Machines (SVM), Ruckert and Kramer (2008) designed a kernel-based approach to inductive transfer, which aims at finding a suitable kernel for the target data, Mihalkova et al. (2007) proposed an algorithm that transfers relational knowledge with Markov Logic Networks (MLN) across relational domains, Dai et al. (2007b) extended a traditional Naive Bayesian classifier for the transductive transfer learning problems. See Pan and Yang (2010) and Weiss et al. (2016) for many other examples.

Due to the availability of various machine learning (ML) methods and considering 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, a strategy known as horse-race, as in Tiffin (2016), Cook and Hall (2017), Garcia et al. (2017), Gu et al. (2018), and Piger (2020). Makridakis et al. (2018) go further to compare various non-ML and ML forecasting methods. Another usual approach, our choice, is selecting an appropriate strategy for the specific task in advance. Deep learning is the most suitable choice for having achieved state-of-the-art performance in pattern recognition and because, according to Bengio (2012), deep learning seems well suited to transfer learning because it focuses on learning representations and, in particular, on *abstract* representations which ideally disentangle the factors of variation present in the input. Learning representations of the data make it easier to extract useful information when building classifiers or other predictors, and, in the case of probabilistic models, a good representation is often one that captures the posterior distribution of the underlying explanatory factors for the observed input (Bengio et al., 2013).

3.1.1 Deep Learning

A deep neural network, also known as deep learning (DL), is an artificial neural network (ANN) with multiple layers hidden between the input and output layers (Bengio, 2009). Deep learning is a sub-field within machine learning that is based on algorithms for learning multiple levels of representation in order to model complex relationships among data (Deng and Yu, 2014, 200). In essence, almost all DL algorithms can be described as a combined specification of a data set, a

cost function, an optimization[1] procedure, and a model (Goodfellow et al., 2016). The analytical function corresponding to one of the simplest forms of an ANN, the feed-forward network, can be written as follows (Bishop, 1994, 118-9). In a feed-forward network having two layers there are d inputs, M hidden units and c output units, where x and y are the network input and output variables, respectively. The output of the jth hidden unit is obtained by first forming a weighted linear combination of the d input values, and adding a bias, to give

$$a_j = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \; . \tag{3.1}$$

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 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 (3.1) in the form

$$a_j = \sum_{i=0}^{d} w_{ji}^{(1)} x_i \; . \tag{3.2}$$

The activation of hidden unit j is then obtained by transforming the linear sum in (3.2) using an activation function $g(\cdot)$ to give

$$z_j = g(a_j) \; . \tag{3.3}$$

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

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \; . \tag{3.4}$$

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

$$a_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \; , \tag{3.5}$$

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

$$y_k = \tilde{g}(a_k) \; . \tag{3.6}$$

---

[1]   Deep learning algorithms involve optimization in many contexts. See (Goodfellow et al., 2016, 271-325) for a comprehensive discussion.

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 (3.2), (3.3), (3.5) and (3.6) we obtain an explicit expression for the complete function in the form

$$y_k = \tilde{g}\left(\sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right) . \tag{3.7}$$

These models are called feed-forward because information flows through the function being evaluated from inputs x, through the intermediate computations used to define the function, and finally to the output target y. There are no feedback connections in which the outputs of the models are fed back into itself. When processing sequential data is required, there are alternatives like recurrent neural networks (RNN), that process an input sequence one element at a time, maintaining in their hidden units a *state vector* that implicitly contains information about the history of all past elements of the sequence (LeCun et al., 2015). The long short-term memory (LSTM), a gated RNN, is one oh the most effective sequence models used in practical applications (Goodfellow et al., 2016, 404). The LSTM forward propagation equations for a shallow recurrent network architecture are given below (Goodfellow et al., 2016, 405-406). Instead of a unit that simply applies an element-wise nonlinearity to the affine transformation of inputs and recurrent units, LSTM recurrent networks have "LSTM cells" that have an internal recurrence (a self-loop), in addition to the outer recurrence of the RNN. Each cell has the same inputs and outputs as an ordinary recurrent network, but also has more parameters and a system of gating units that controls the flow of information. The most important component is the state unit $s_i^{(t)}$, which has a linear self-loop. The self-loop weight, or the associated time constant, is controlled by a **forget gate** unit $f_i^{(t)}$, for time step t and cell i, which sets this weight to a value between 0 and 1 via sigmoid unit:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) , \tag{3.8}$$

where $\mathbf{x}^{(t)}$ is the current input vector and $\mathbf{h}^{(t)}$ is the current hidden layer vector, containing the outputs of all the LSTM cells, and $\mathbf{b}^f$, $\mathbf{U}^f$, $\mathbf{W}^f$ are respectively biases, input weights, and recurrent weights for the forget gates. The LSTM cell internal state is thus updated as follows, but with a conditional self-loop weight $f_i^{(t)}$:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) , \tag{3.9}$$

where $\mathbf{b}$, $\mathbf{U}$ and $\mathbf{W}$ respectively denote the biases, input weights, and recurrent weights into the LSTM cell. The **external input gate** unit $g_i^{(t)}$ is computed similarly to the forget gate, with a sigmoid unit to obtain a gating value between 0 and 1, but with its own parameters:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) . \tag{3.10}$$

The output $h_i^{(t)}$ of the LSTM cell can also be shut off, via the **output gate** $q_i^{(t)}$, which also uses a sigmoid unit for gating:

$$h_i^{(t)} = tanh\left(s_i^{(t)}\right) q_i^{(t)} \,, \tag{3.11}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \,, \tag{3.12}$$

which has parameters $\mathbf{b}^o$, $\mathbf{U}^o$ and $\mathbf{W}^o$ for its biases, input weights and recurrent weights, respectively.

3.1.2   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 function used to train a neural network will often combine cost functions with a regularization term in order to prevent overfitting[2]. 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).

3.1.3   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

---

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

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. Algorithms 1 and 2, reproduced from (Goodfellow et al., 2016, 213), describe the general back-propagation procedure.

Algorithm 1 The outermost skeleton of the back-propagation algorithm. This portion does simple setup and cleanup work. Most of the important work happens in the *build_grad* subroutine of algorithm 2

Require: T, the target set of variables whose gradients must be computed
Require: G, the computational graph
Require: z, the variable to be differentiated
  1: Let G' be G pruned to contain only nodes that are ancestors of z and descendants of nodes in T.
  2: Initialize *grad_table*, a data structure associating tensors to gradients *grad_table*$[z] \leftarrow 1$
  3: **for V** in T **do**
  4:      *build_grad*($\mathbf{V}, G, G', grad\_table$)
  5: **end for**
  6: Return *grad_table* restricted to T

Algorithm 2 The inner loop subroutine *build_grad*($\mathbf{V}, G, G', grad\_table$) of the back-propagation algorithm, called by the back-propagation algorithm defined in algorithm 1

Require: $\mathbf{V}$, the variable whose gradient should be added to G and *grad_table*
Require: G, the graph to modify
Require: G', the restriction of G to nodes that participate in the gradient
Require: *grad_table*, a data structure mapping nodes to their gradients
  1: **if V** is in *grad_table* **then**
  2:      Return *grad_table*$[\mathbf{V}]$
  3: **end if**
  4: $i \leftarrow 1$
  5: **for C** in *get_consumers*($\mathbf{V}, G'$) **do**
  6:      op $\leftarrow$ *get_operation*($\mathbf{C}$)
  7:      $\mathbf{D} \leftarrow$ *build_grad*($\mathbf{C}, G, G', grad\_table$)
  8:      $\mathbf{G}^i \leftarrow$ op.*bprop*(*get_inputs*($\mathbf{C}, G'$)$, \mathbf{V}, \mathbf{D}$)
  9:      $i \leftarrow i + 1$
 10: **end for**
 11: $\mathbf{G} \leftarrow \sum_i \mathbf{G}^i$
 12: *grad_table*$[\mathbf{V}] = \mathbf{G}$
 13: Insert $\mathbf{G}$ and the operations creating it into G
 14: Return $\mathbf{G}$

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).

3.2   Pratical Methodology

Since machine learning practitioners cannot prove, at least until now, which model is the best for each situation, we search for successful deep artificial neural networks that can exhibit small differences between training and test performance (Zhang et al., 2016). According to Goodfellow et al. (2016), successful applying deep learning techniques requires more than just a good knowledge of what algorithms exist and the principles that explain how they work. A good machine learning practitioner also needs to know how to choose an algorithm for a particular application and how to monitor and respond to feedback obtained from experiments to improve a machine learning system. See (Goodfellow et al., 2016, 416-435) for more details about performance metrics, default baseline models, determining whether to gather more data, selecting hyperparameters, and debugging strategies.

The models in Chapters 4, 5 and 6 were built using TensorFlow[3], an interface for expressing machine learning algorithms, and an implementation for executing such algorithms. TensorFlow is flexible and can be used to express a wide variety of algorithms, including training and inference algorithms for deep neural network models. It has been used to conduct research and deploy machine learning systems into production across more than a dozen areas of computer science and other fields (Abadi et al., 2015).

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). The optimal set of hyperparameters was obtained using Hyperband (Li et al., 2018) from Keras Tuner[4].

3.3   Assumptions

As in the dynamic factor model with Markov-switching introduced by Chauvet (1998), the deep learning approach accounts for the idea of business cycles as the simultaneous, asymmetrical, and nonlinear movement of economic activity in various sectors. This data-driven framework is

---

[3]   https://www.tensorflow.org/.
[4]   https://www.tensorflow.org/tutorials/keras/keras_tuner.

flexible enough to be training with different features, the independent variables, and targets, the dependent variables. This characteristic has not yet been evaluated to monitor business cycle phases (Chapter 4) or to estimate the output gap (Chapter 5) to the best of our knowledge. Our goal follows Chauvet and Yu (2006) and Chauvet and Guimaraes (2021) in providing additional tools for governments and the private sector to conduct their activities in light of both national and international economic conditions. For that, we adopt a combined strategy of deep neural network and transfer learning to address two practical problems, (1) identifying the business cycle phases when there is no business cycle dating committee, and (2) a quick first economic-based estimating of the output gap. In both cases, the proposed strategy also helps to improve the learning when data is limited. Additionally, in Chapter 6 we apply deep learning associated with high-frequency indicators to disaggregate the GDP quarterly series, distributing this information for monthly periodicity, in line with Issler and Notini (2014) that proposed a state-space model to address this problem.

A supervised deep learning model learns from the features-targets without the need for strong assumptions about its relation, which prevents the expert from choosing an underlying economics school of thought to set up a model, although the variable selection might represent it. The algorithm maps the input-output relation variables according to the training and validation data. For instance, when we choose the National Bureau Economic Research (NBER)[5] turning points classification data as output label (Chapter 4), the output gap based on the potential output estimated by the U.S. Congressional Budget Office (CBO)[6] as the target (Chapter 5), the deep learning algorithm will learn from them how to classify the business cycle and to measure the output gap, implicitly following the same schools of thought. Additionally, a deep learning model is capable of handling a large number of explanatory variables (features) as indicator series (Chapter 6). However, there is a caveat to our strategy: the more features we use during learning, the more data preprocessing efforts will be required. For example, if we train for the United States output gap estimation using dozens of features, we will need the same quantity of features to transfer learning to other datasets or additional preprocessing strategies, as in Jackson and Rege (2019) that have fed an ANN with dynamic factors. We choose to leave it for future work to deal with high-dimensional data.

Transfer learning improves learning in a new task by transferring knowledge from a related task that has already been learned (Torrey and Shavlik, 2009). More specifically, domain adaptation can be considered a special set of transfer learning that aims at transferring shared knowledge across different but related tasks or domains. An example of the current state of the art is transfer learning with MobileNets (Howard et al., 2017). The MobileNet model is an open-source code that learned how to classify 1,000 categories from more than 1,500,000 images. It has some verification points to start from freezing the previous points' parameters and training a new model

---

[5]   The most followed classification for U.S. business cycle.
[6]   An economic-based estimation. See Shackleton (2018).

that recognizes images that are not necessarily among the 1,000 categories previously learned. This new model would start from the representation learning present in the parameters transferred from MobileNet. For example, in a simple exercise, presenting only 100 new images of gestures used for the game stone, paper, and scissors, a transfer learning model can identify each new category with high precision[7]. Without transfer learning from MobileNet, the number of images needed to train this new model would be much higher. A good feature representation should be able to reduce the difference in distributions between domains as much as possible, while at the same time preserving essential properties of the original data (Pan et al., 2011). When we transfer learning, though, there are additional issues. The challenge is to overcome the differences between the domains so that a classifier trained on the source domain generalizes well to the target domain, but generalizing across distributions can be difficult, and it is not clear which conditions have to be satisfied for a classifier to perform well (Kouw and Loog, 2018). From a macroeconometrics perspective, we could assume that the input variables' marginal contributions in explaining the business cycle phases are similar. However, they are probably not equal for the different economic areas, as well as the respective input variables' sample covariance matrix. Since we are applying transductive transfer learning (subsection 2.2.2), we can presume two circumstances (Pan and Yang, 2010, 4): (1) the feature spaces between the source and target domains are different; (2) the feature spaces between domains are the same, but the marginal probability distributions of the input data are different. The latter case is related to domain adaptation. Our approach handles this well when the models successfully build a common representation space for the different domains, which can be empirically verified when there are labels for the target domain, allowing evaluation by some error criteria. Otherwise, it remains an open question.

3.4   Robustness

Our baseline models are a logistic model (Chapter 4), a linear regression model (Chapter 5), and deep learning models without the transfer learning approach. They function as a reference to verify if negative transfer (Torrey and Shavlik, 2009) is not occurring, a situation in which the transfer reduces the learning. And, as benchmarks, we compare our results with those released by entities that apply widely accepted techniques to identify the business cycle and estimate the output gap, notably the Centre for Economic Policy Research (CEPR), the Brazilian Business Cycle Dating Committee (CODACE), and the Central Bank of Brazil. For Chapter 6, as benchmarks, we have the Chow and Lin (1971) method, the GDP Monitor from Instituto Brasileiro de Economia (2015), and the Central Bank Economic Activity Index (IBC-Br). The choice of these

---

[7]   https://github.com/lmoroney/dlaicourse/tree/master/TensorFlow, in TensorFlow Deployment, Course 1, Week 4.

benchmarks is relevant because of the risk of overfitting[8]. In other words, no matter how complex the relationships are between the data, there is always the risk of a neural network mapping the entire data set, resulting in overfitting and subsequently failing in terms of generalization by not correctly classifying data not previously observed. That is why the test set is so relevant. That is why our strategy of using data from a country other than the one used for network training as a test set is a critical factor in evaluating the models' performance.

The option for a feed-forward network as a deep learning model, which represents memory-less models, derives from the focus on contemporary movement between the selected input variables and the business cycle and the output gap to explore the informational content of their cross-section distributions. This implies disregarding the time dependence observed on the variables and shuffling the data to break it. The resulting model accounts just for coincident relations. To account for time dependency as a alternative strategy, we create additional models including a LSTM layer.

When transfer learning, the weight layers of the network for Euro and Brazillian data (Chapter 4), or just Brazillian data (Chapter 5), were copied from the network trained on the U.S. data, as in the last row of Figure 2, except that we do not retrain the parameters. It is as if these two datasets function as out-of-sample. Additionally, we unlock the last layers and retrain the parameters, a way of relaxing the macroeconometrics assumptions about input variables as mentioned before.

Finally, we retrained the models with others loss functions, the squared hinge (Chapter 4) and the adaptative gradient algorithm (AdaGrad) (Duchi et al., 2011) (Chapter 5), finding similar results.

---

[8]   According to Zhang et al. (2016), for every $k \geq 2$, there exists a neural network with ReLU activations of depth k, width $O(n/k)$, and $O(n + d)$ weights that can represent any function on a sample of size n in d dimensions.

4 Transfer Learning for Business Cycle

Monitoring business cycle phases is a traditional task in applied macroeconomics. Progressive market integration has induced a worldwide interest in analyzing cyclical fluctuations through economic indicators Chauvet (2001). Changes in exchange rates, outputs, consumption, inflation, and interest rates in different parts of the world can influence the effectiveness of government policies and the competitive position of businesses, even those not directly related to international operations (Chauvet and Yu, 2006, p.43). As a result, a wide range of techniques has been developed since the seminal work by Burns and Mitchell (1946). Recently, new approaches have emerged due to the progress in machine learning (ML) research, centering on building models that achieve better forecasting performance than the non-ML models or that identify turning points more timely (Piger, 2020).

This Chapter[1] is organized as follows. Section 4.1 presents a literature review. In Section 4.2, we discuss the implementation details of the proposed new strategy (Section 2.2) when applied to Business Cycle identification, and, in Section 4.3, the empirical findings are presented.

4.1 Related Literature

Business cycles are recurrent sequences of alternating phases of expansion and contraction among many economic activities (Burns and Mitchell, 1946). According to Harding and Pagan (2005), there are three ways in the literature to describe what we mean by a cycle, depending on whether the main focus is on the fluctuation of the level of economic activity, the level of economic activity less a permanent component, or the growth rate of economic activity.

In the United States, the NBER Business Cycle Dating Committee provides a chronology of business cycle expansion and recession dates. According to Piger (2020), because the NBER methodology is not explicitly formalized, literature has worked to develop and evaluate formal statistical methods for establishing the historical dates of economic recessions and expansions in both the U.S. and international data. Estrella and Mishkin (1998), Estrella et al. (2000), Kauppi and Saikkonen (2008), Rudebusch and Williams (2009) and Fossati (2016) use an available historical indicator of the class, such as the NBER dates, to estimate the parameters of models such as logit or probit ones. This strategy is called a supervised classifier in the statistical classification literature, in contrast to unsupervised classifiers, which endogenously determine the classes. Unsupervised classifiers have also been used, with the primary example being the Markov-switching (MS) framework of Hamilton (1989), which become a relevant tool for applied work in economics. Chauvet

---

[1] A modified version of this Chapter was published in Central Bank of Brazil Working Paper Series (Chauvet and Guimaraes, 2021).

(1998) proposes a dynamic factor model with Markov-switching (DFMS) to identify expansion and recession phases from a group of coincident indicators and Chauvet and Hamilton (2005), Chauvet and Piger (2008) and Camacho et al. (2018) evaluate the performance of variants of this DFMS model to identify NBER turning points in real-time. See Piger (2020) for a comprehensive review.

Applied ML papers related to business cycles can be separated depending on whether the main focus is *predicting* or *identifying* turning points and phases. For example, Hoptroff et al. (1991), Qi (2001), Klinkenberg (2003), Nasr et al. (2007), Berge (2013), Ma (2015), Garbellano (2016), Nyman and Ormerod (2017), and James et al. (2019) have applied machine learning techniques such as artificial neural networks, support vector machines, boosting, k-nearest neighbor, and random forest to *forecasting* turning points, recessions, or business cycles phases mainly in the US, but also other countries[2]. These studies have generally reported some improvements over non-ML strategies. The other set of papers is concerned about *identifying* the turning points for real-time classification. Morik and Rüping (2002), Giusto and Piger (2017), Soybilgen (2018), Raffinot and Benoit (2019) and Jackson and Rege (2019) have applied inductive logic programming, learning vector quantization, random forest, boosting, k-nearest neighbor and artificial neural networks fed with dynamic factors. Piger (2020), in a comprehensive analysis, compares five ML techniques with DFMS. These studies have reported quickly and accurately turning points identification.

Lastly, some literature is dedicated to the study of business cycles worldwide, as in Chauvet and Yu (2006), Cuba-Borda et al. (2018), Abberger et al. (2020), and the reference turning points of the OECD Composite Leading Indicators[3].

4.2   Implementation

The feature selection comprises the coincident variables indicated by NBER[4] as the fundamental: gross domestic product (GDP), income, employment, industrial production, and wholesale-retail sales. Quarterly data is adopted because this is the frequency at which some relevant variables for the classification of the business cycle are available, and at this frequency, opposed to higher ones, the data usually carry less noise, which may facilitate the training and the transfer learning. We computed the first difference of the logarithm of the input features, capturing the growth rate (Harding and Pagan, 2005, 152-154). Alternatively, we run the model without this transformation, i.e., features in level. In both cases, the features are normalized. To have a common starting point for each dataset, we restrict the series' start to eliminate missing values. For example, we do not acquire data for the Euro area before 2005 because we do not have employment data

---

[2]   United Kingdom, Japan, West Germany and Lebanon.

[3]   Available at https://www.oecd.org/sdd/leading-indicators.

[4]   The NBER does not define a recession in terms of two consecutive quarters of decline in real GDP. Rather, a recession is a significant decline in economic activity spread across the economy, lasting more than a few months, normally visible in real GDP, real income, employment, industrial production, and wholesale-retail sales. Source: https://www.nber.org/cycles.html.

before this year for all the selected countries. We adopted a U.S. dataset for deep learning and two datasets, with data from Brazil and Europe, for transfer learning. The target values are the business cycle chronology provided by the NBER, the CODACE, and the CEPR Euro Area Business Cycle dating committees, respectively. Table 3 and Figure 18 in the Appendix summarizes the information about all series, mostly from the Federal Reserve Economic Data (FRED)[5] dataset, provided by the Federal Reserve Bank of St. Louis. Our data files are available at `https://github.com/rrsguim/PhD_Economics`.

Following Piger (2020), the objective function to maximize in the cross validation step is the area under the ROC curve (AUC). This metric is desirable here for being scale-invariant, measuring how well predictions are ranked, rather than their absolute values, and classification-threshold-invariant, measuring the quality of the model's predictions irrespective of what classification threshold is chosen.

Beginning with the deep learning step, we split the U.S. dataset into train, validation, and test sets. Then, we define a function that creates a neural network with hidden layers, ReLU as activation function, a dropout layer to reduce overfitting, and a sigmoid output layer that returns the probability of recession. Next, we retrain the model with the optimal hyperparameters, selected with *Hyperband*, to evaluate the results in both datasets, source and target, with binary cross-entropy as a loss function and Adam (Kingma and Ba, 2017) for optimization.

All codes used are available at `https://github.com/rrsguim/PhD_Economics`. In Appendix B we reproduce selected parts of the best performance transfer learning model for business cycle identification.

4.3   Results

Before presenting the results, it is relevant to assert that contributing to the turning points forecast improvement is not the objective of this chapter. The *predicting* of the next business cycle phase turning point, which becomes relevant at times like the current one when many countries are in the recession phase due to the COVID-19 pandemic, is a well-consolidated part of the business cycle research field (Section 4.1) and whose state of the art is similar to macroeconomics forecasting in general. However, when focusing on *identifying* the business cycle phase, we address challenges as the absence of a business cycle dating committee and the limited datasets. Another aspect to emphasize is that applying transfer learning to the Brazilian and the Euro area data is a choice related to the availability of committees that explicitly adopt the same, or very similar, classification approach than the committee (NBER) selected when training the models. This choice allows empirically evaluate the proposed method performance from labels in the source and target datasets. Good performance increases confidence to apply it to unlabeled datasets.

---

[5]   `https://fred.stlouisfed.org/`.

The method proposed in Subsection 2.2.1 involves two stages: the first one consists of a machine learning model that learns to perform a specific task based on a dataset; the second focuses on applying that learning to another dataset. The deep learning models, our choice as mentioned in 3.1.1, have learned how to classify business cycle phases. The transfer learning performance, stage 2, proved to be superior to our expectations. Figures 5, 6, 7, and 8 consolidate the results found. Results are slightly different each time the models are run due to different compositions in the selected data sets for training, validation, and test, especially in the cross-sectional approach.
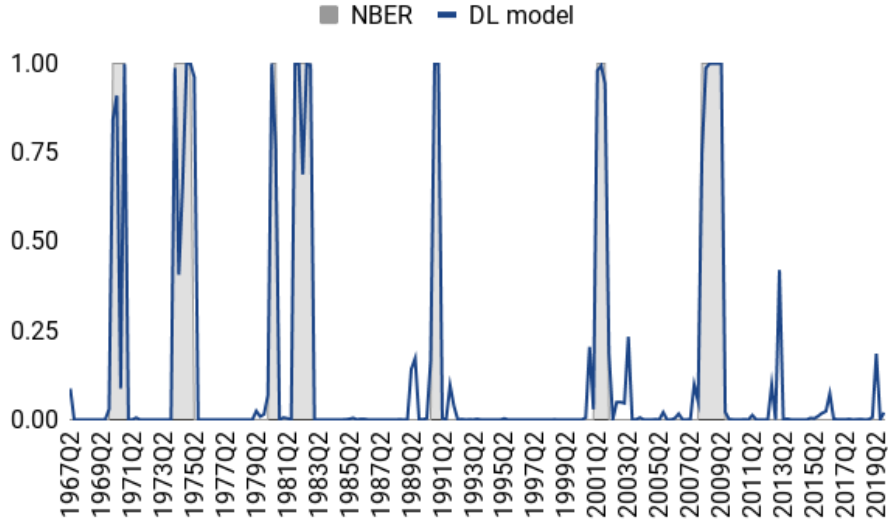

Figure 5 – Deep Learning (U.S)

The comparison in Figure 5 shows the excellent performance of the deep learning step: the U.S. business cycles, where shadow areas indicate recession phases according to NBER, and the feed-forward neural networks (FNN) model with the cross-sectional approach estimation, where the blue lines are showing the recession probability on each quarter. The alternative model that accounts for time dependency exhibited exceptional results as well, with almost all points correctly classified. Concerning the baseline models with U.S. data, we observed a significantly superior performance, measured by the AUC with out-of-sample data, of the deep learning models with data in first difference (1df). The outcomes of the baseline models for the nineteen selected countries in Europe (EURO-19), where there is no transfer learning step, do not motivate confidence because the FNN model has a perfect classification with AUC=1, while the alternative model that includes an LSTM layer is unable to identify crises (AUC=0.5). This point highlights one of the problems that the proposed methodology seeks to solve: identifying business cycles when data is limited. Note, in Figure 6, that there are only two recessions for the EURO-19 in the period under analysis so that models trained only on these data show inconsistent results. Regarding Brazil's outcomes, whose period contains more than one recession, the baseline models show more satisfactory performance

(Figure 7). We expected these results because machine learning methods are good at pattern recognition. The results align with the other machine learning strategies adopted in the literature (Section 4.1). There is no innovation here, but the confirmation that deep learning models correctly classify the business cycle phases.

However, the transfer learning step, which is not in the business cycle literature to the best of our knowledge, exceeded our expectations. Figure 6 compares EURO-19's CEPR classification, Brazil's CODACE classification, and the transfer learning models results. These results refer to the locked models, meaning the estimates for EURO-19 and Brazil operate as if they were out-of-sample because the parameters trained with U.S. data are locked in the transfer learning phase when applied to target EURO-19 and Brazil datasets. Figure 7 presents the details, and Figure 8 a summary comparison of each models' performance. Differences in the sizes of out-of-sample data sets reflect the need for adjustments according to time series, cross-sectional, locked, and unlocked strategies. Overall, there is an improvement in the classification with transfer learning compared to baseline models. Although both showed excellent results, the LSTM models performed better than the FNN models, and locked models overperform unlocked models. On the other hand, the models didn't learn well with data in level, requiring transformation. As mentioned in Section 3.2, successfully applying deep learning techniques requires monitoring and responding to feedback obtained from experiments, which is data-driven. Not necessarily the same configuration will emerge from different datasets, which demands finding the best hyperparameters to perform well the task at hand.
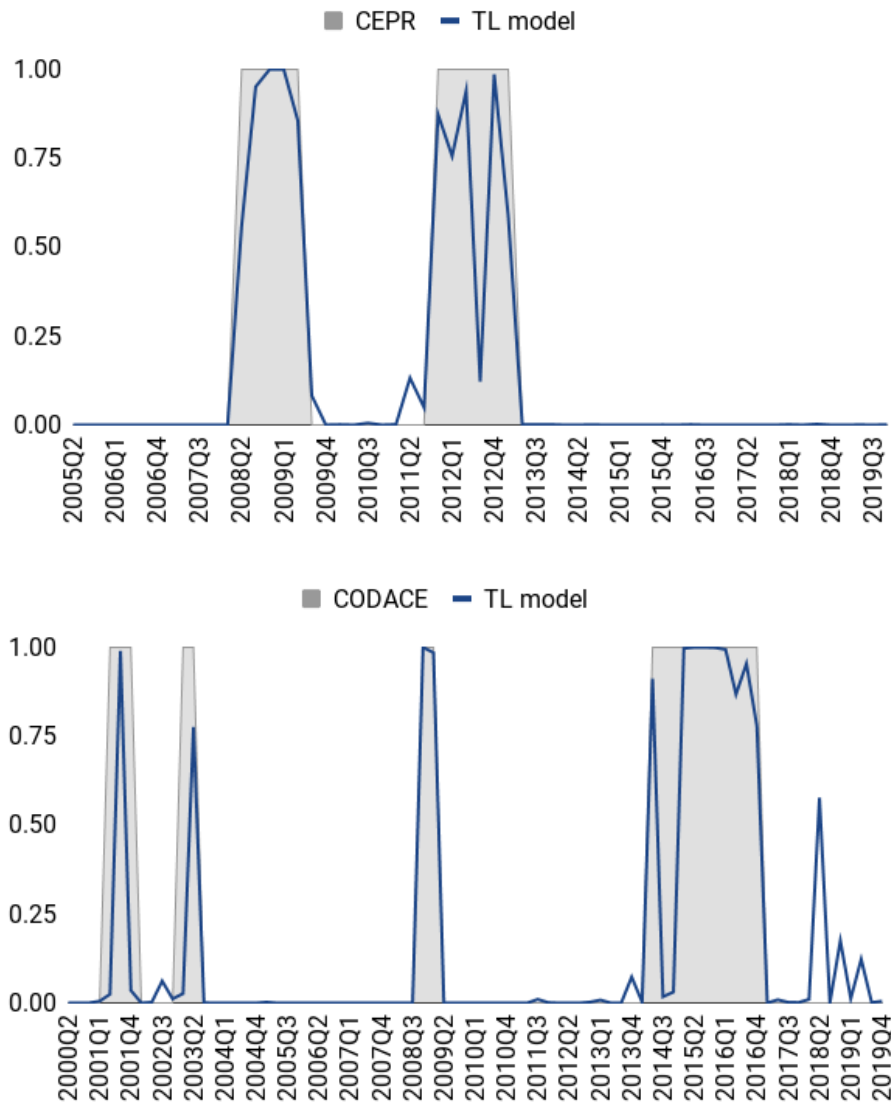
Figure 6 – Transfer Learning (Euro Area and Brazil)

	Dropout	Hidden layers		*Hyperband* best hyperparameters		Target	Out-of-sample						AUC
		LSTM	Dense	units	learning rate		Test size	Confusion matrix					
								TN	FP	FN	TP		
Baseline models													
Logit_1df_US	-	-	-	-	-	NBER	77	36	0	32	9	0.610
DL_FNN_1df_US	0.5	0	3	208	0.01	NBER	64	56	0	2	6	0.875
DL_LSTM_1df_US	0.5	1	4	176	0.01	NBER	85	71	5	0	9	**0.967**
DL_FNN_level_US	0.5	0	4	176	0.01	NBER	64	54	0	10	0	0.500
DL_LSTM_level_US	0.5	1	4	176	0.01	NBER	85	76	0	9	0	0.500
DL_FNN_1df_EUR	0.5	0	3	112	0.01	CEPR	18	15	0	0	3	1.000
DL_LSTM_1df_EUR	0.5	1	4	48	0.0001	CEPR	42	36	0	6	0	0.500
DL_FNN_1df_BR	0.5	0	3	64	0.01	CODACE	24	18	1	3	2	0.674
DL_LSTM_1df_BR	0.5	1	4	240	0.001	CODACE	32	17	4	1	10	0.859
Transfer learning models													
TL_FNN_1df_EUR_locked	0.5	0	3	208	0.01	CEPR	60	48	1	2	9	0.899
TL_FNN_1df_EUR_unlocked	0.5	0	4	208	0.01	CEPR	42	36	0	6	0	0.500
TL_LSTM_1df_EUR_locked	0.5	1	4	176	0.01	CEPR	60	48	1	1	10	**0.944**
TL_LSTM_1df_EUR_unlocked	0.5	1	4	64	0.01	CEPR	42	36	0	6	0	0.500
TL_FNN_1df_BR_locked	0.5	0	3	208	0.01	CODACE	80	60	2	6	12	0.817
TL_FNN_1df_BR_unlocked	0.5	0	4	208	0.01	CODACE	24	16	3	1	4	0.821
TL_LSTM_1df_BR_locked	0.5	1	4	176	0.01	CODACE	80	61	1	3	15	**0.909**
TL_LSTM_1df_BR_unlocked	0.5	1	4	64	0.01	CODACE	48	31	4	2	11	0.866

TN - true negative | FP - false positive | FN - false negative | TP - true positive | AUC - area under the ROC curve.
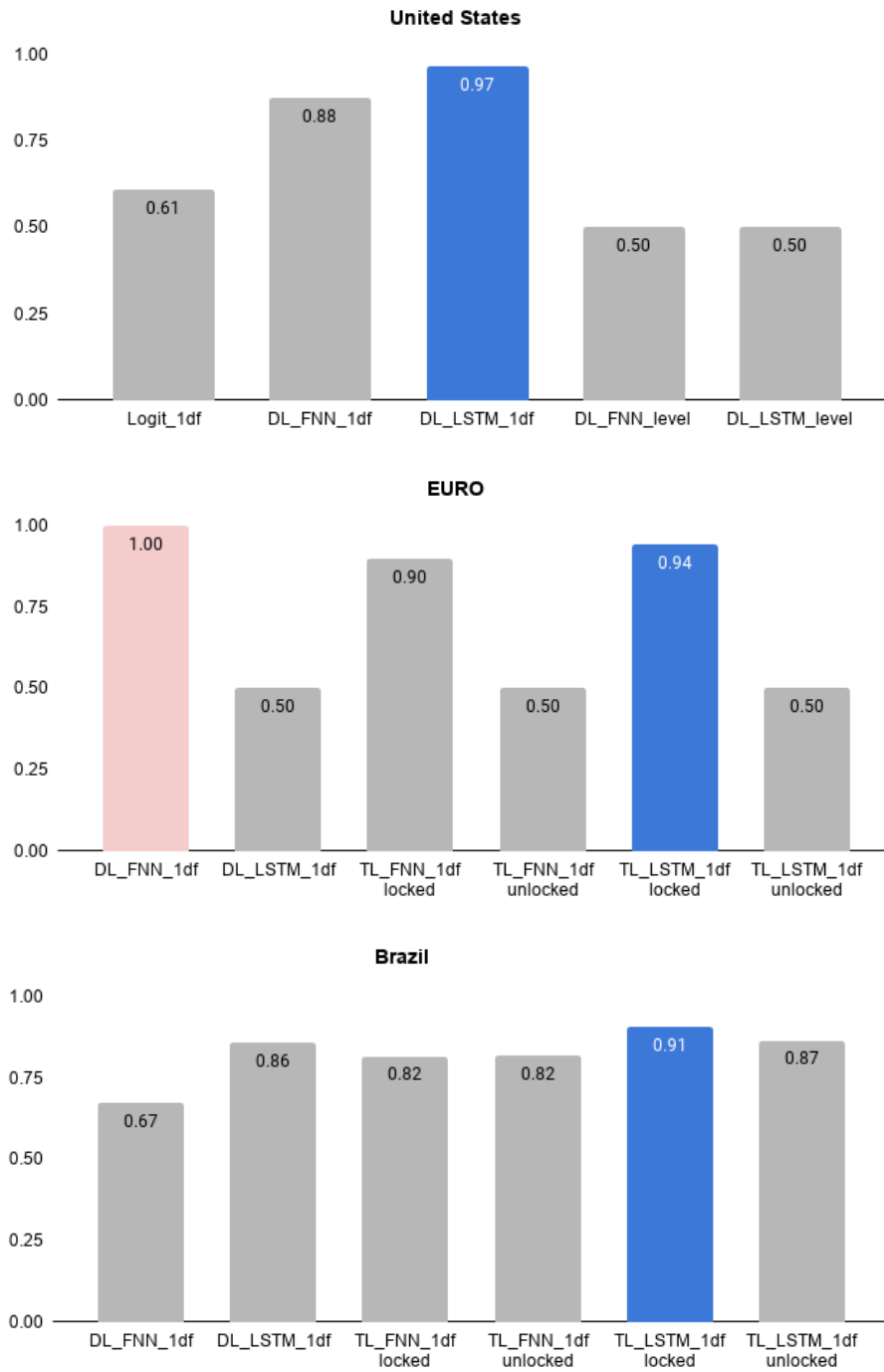
Figure 7 – Models specifications and results

Figure 8 – AUC out-of-sample

5 Transfer Learning for Output Gap

Measures of the gap between actual and potential activity are used frequently as indicators of the economic cycle and play a vital role in monetary and fiscal policy (Koske and Pain, 2008). As a result, the wide range of techniques that have been developed since the seminal work by Okun (1962) can be classified in different ways, such as between observed and unobserved components methods or between pure statistical versus economic base approaches each with its pros and cons. The emergence of new approaches applied to business cycles due to machine learning research progress (Piger, 2020) can also be explored for the output gap, given that they are related concepts.

This Chapter is organized as follows. Section 5.1 presents a literature review. In Section 5.2, we discuss the implementation details of the proposed new strategy (Section 2.2) when applied to Output Gap estimation, and, in Section 5.3, the empirical findings are presented.

5.1 Related Literature

How much output can the economy produce under conditions of full employment? Okun's seminal paper (Okun, 1962) begins with this question and describes two related concepts, potential output and the output gap. Potential output differs from actual only because the potential concept depends on the assumption that aggregate demand is exactly at the level that yields a rate of unemployment equal to four percent of the civilian labor force[1]. If, in fact, aggregate demand is lower, part of potential GNP (Gross National Product, a measure of output.) is not produced; there is unrealized potential or a gap between actual and potential output (Okun, 1962). Since then, various definitions of potential output, as well as estimation strategies, have been proposed and used in the literature, depending on the investigator's objectives, although Okun's definition is still the main reference concept for economic policy-makers, including central banks (Proietti et al., 2007).

According to Proietti et al. (2007), a useful classification is between unobserved components and observed components methods. Harvey and Jaeger (1993) trend-cycle decomposition of output and the Hodrick and Prescott (1981) HP filter are examples of univariate approaches of unobserved components, while Kuttner (1994), Gerlach and Smets (1999) and Apel and Jansson (1999) focus on multivariate approaches. Observed components methods rely on the Beveridge and Nelson (1981) decomposition and on structural vector autoregressive (VAR) models, which have been used by Blanchard and Quah (1989) and St-Amant and van Norden (1997). St-Amant and van Norden (1997) states that univariate methods, such as the HP filter, are not reliable to measure the output gap. Hybrid methods that combine univariate and structural relationships have proved to be hard

---

[1] It was a reference to full employment at that time, as he discusses in the paper.

to estimate, they may not be robust to reasonable alternative calibrations, and it is not easy to calculate their appropriate confidence intervals. However, methods combining estimated dynamics with structural information offer an interesting alternative. Another conceivable classification, as in D'Auria et al. (2010), is pure statistical versus economic base approaches, the latter being preferred because of the possibility of examining the underlying economic factors which are driving any observed changes in the potential output indicator and consequently the opportunity of establishing a meaningful link between policy reform measures with actual outcomes. Hamilton (2018) goes further and claims that, although the HP filter method continues today to be adopted in academic research, policy studies, and analysis by private-sector economists, it is intended to produce a stationary component from an I(4) series. However, in practice, it can fail to do so and invariably imposes a high cost. It introduces spurious dynamic relations that are purely an artifact of the filter and have no basis in the true data-generating process. He proposes a regression of the variable at date $t + h$ on the four most recent values as an alternative approach. Okun (1962) foresaw the challenges mentioned above in his seminal work: *the quantification of potential output - and output gap - is, at best, an uncertain estimate and not a firm, precise measure.* Chauvet and Guimaraes (2021) proposed a transfer learning strategy to identify business cycle phases when data are limited or there is no business cycle data committee. The approach integrates the idea of storing knowledge gained from one region's economics experts and applying it to other geographic areas. Here, we adopt the same strategy to measure the output gap.

5.2   Implementation

The feature selection comprises variables usually incorporated in economic-based approaches: unemployment rate, capacity utilization, business cycle phase, and total factor productivity parity with U.S. when transfer learning. Quarterly data is adopted because this is the frequency at which some relevant variables for the estimation of the output gap are available, and at this frequency, opposed to higher ones, the data usually carry less noise what may facilitate the training and the transfer learning. We adopted a U.S. dataset for deep learning and data from Brazil for transfer learning. The target[2] values are calculated based on the CBO's potential output for the U.S. and the Brazilian Central Bank's output gap for Brazil. Table 4 and Figure 19 in the Appendix summarizes all series information from the FRED[3] dataset provided by the Federal Reserve Bank of St. Louis and from the Series Management System (SGS)[4] dataset provided by the Brazilian Central Bank. Our data files are available at https://github.com/rrsguim/PhD_Economics.

Beginning with the deep learning step, we split the U.S. dataset into train, validation, and test

---

[2]   This Chapter aims to observe the results of an empirical application to the method proposed in 2.2, and not evaluate the output gaps disclosed by different sources.

[3]   https://fred.stlouisfed.org/.

[4]   https://www3.bcb.gov.br/sgspub.

sets. Then, we define a function that creates a neural network with hidden layers, ReLU, tanh, or sigmoid as activation functions, and L1 (Lasso) regularization to reduce overfitting. Next, we retrain the model with the optimal hyperparameters, selected with Hyperband, to evaluate the results in both datasets, source, and target, with mean squared error as a loss function and the adaptive moment estimation (Adam) (Kingma and Ba, 2017) for optimization. All codes used and other results are available at https://github.com/rrsguim/PhD_Economics. See Appendix B for selected codes.

5.3 Results

Before presenting the results, it is relevant to assert that evaluate the output gaps disclosed by different sources is not the objective of this chapter. As it is an unobservable variable, there will always be some dispute about its most appropriate value. It is common for analysts to observe more than one source for decision-making. However, by focusing on learning how to estimate the output gap based on data disclosure from a selected source, we address challenges like quickly assess an economic-based output gap and the limited datasets. Another aspect to emphasize is that applying transfer learning to the Brazilian area data allows empirically evaluate the proposed method performance from labels in the source and target datasets. The theoretical model, however, enables the application to any region. Good performance increases confidence to apply it to unlabeled datasets also.

The method proposed in Subsection 2.2 involves two stages: the first one consists of a machine learning model that learns to perform a specific task based on a dataset; the second focuses on applying that learning to another dataset. The deep learning models, our choice as mentioned in 3.1.1, have learned how to estimate the output gap. The transfer learning performance, stage 2, proved to be superior to our expectations. Figures 9, 10, 11, and 12 consolidate the results found. Results are slightly different each time the models are run due to different compositions in the selected data sets for training, validation, and test.

Figures 9 shows the performance of the deep learning step: the U.S. output gap based on the CBO's potential output, compared with the FNN and the LSTM models. Concerning the baseline models with U.S. and Brazilian data, Figure 10, we observed a superior performance, measured by the mean absolute error (MAE) with out-of-sample data, of the feed-forward neural network (FNN) models. We expected these results because machine learning methods are good at fitting curves. There is no innovation here, but the confirmation that deep learning models correctly identify the output gap.
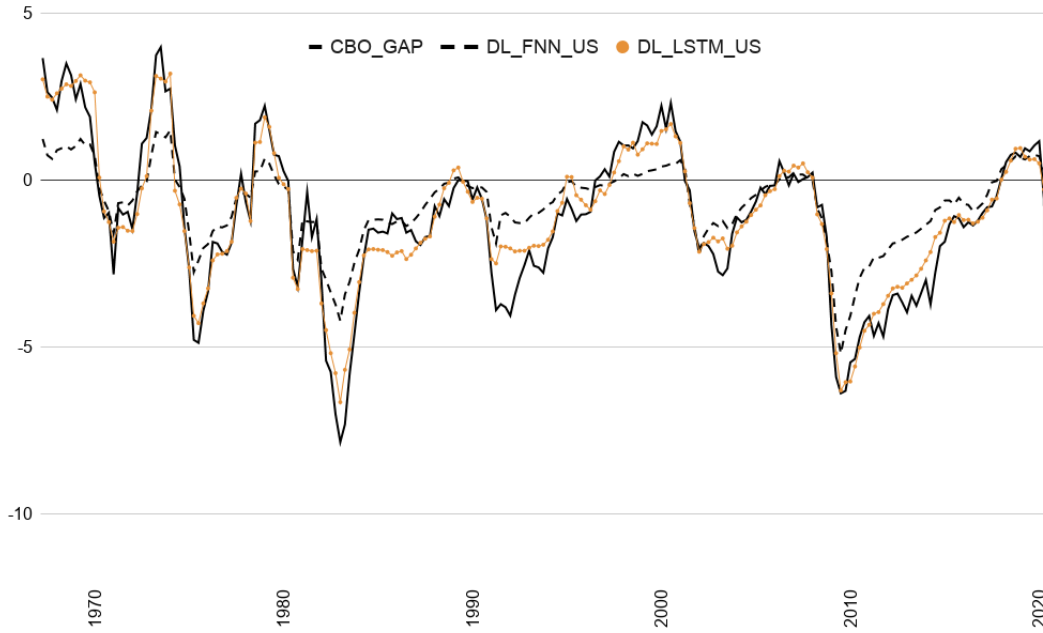
Figure 9 – US Output GAP - Congress Budget Office (CBO) and Deep Learning models

	Hidden layers		*Hyperband* best hyperparameters					Optmizer	In-sample		Out-of-sample	
	LSTM	Dense	LSTM		Dense				Size	MAE	Size	MAE
			Units	Dropout	Units	Activation	L1 lambda	Learning rate					
Baseline models												
Linear_US_1967_2020	-	-	-	-	-	-	-	-	Adam	172	0.615	43	0.668
DL_FNN_US_1967_2020	-	4	-	-	112	ReLu	0.001	0.01	AdaGrad	172	0.431	43	0.558
DL_LSTM_US_1967_2020	1	4	128	0.3	240	ReLu	0.01	0.001	Adam	172	0.985	43	0.654
DL_FNN_BR_2012-2020	-	4	-	-	112	ReLu	0.001	0.01	AdaGrad	27	0.361	7	0.366
DL_LSTM_BR_2012-2020	1	4	16	0.4	240	Tanh	0.01	0.01	Adam	27	0.493	7	1.390
Transfer learning models for Brazil dataset 2012-2020												
TL_FNN_BR_locked	-	4	-	-	112	ReLu	0.001	0.01	AdaGrad	-	-	34	0.646
TL_FNN_BR_unlocked	-	4	-	-	112	ReLu	0.001	0.01	AdaGrad	17	0.524	17	0.452
TL_LSTM_BR_locked	1	4	128	0.3	240	ReLu	0.01	0.001	Adam	-	-	34	0.614
TL_LSTM_BR_unlocked	1	4	112	0.4	224	Tanh	0.01	0.01	Adam	17	1.359	17	3.845

CBO - United States Congressional Budget Office | BCB - Central Bank of Brazil.

Figure 10 – Models specifications and results

However, the transfer learning step, which is not in the output gap literature to the best of our knowledge, exceeded our expectations. Figure 11 shows a comparison between the output gap estimated by the locked transfer learning model and the one by the Central Bank of Brazil which the methodology is based on a Bayesian model estimation (Banco Central do Brasil, 2020a), a different approach. It must be emphasized that, since it refers to the locked model, Brazil's estimate operates as if it was out-of-sample because the parameters trained with U.S data are locked in the transfer learning phase, with no retraining with Brazilian data. The similarity is

impressive because the *machine learned*, based on the U.S. data, how to estimate the output GAP and then transfer this knowledge to Brazilian data obtaining similar results of the Brazilian Central Bank specialists[5], with just one extra piece of information about Brazil, the total factor productivity (TFP) level at current purchasing power parities with the U.S. Figure 12 presents a summary comparison of the performance of each model.
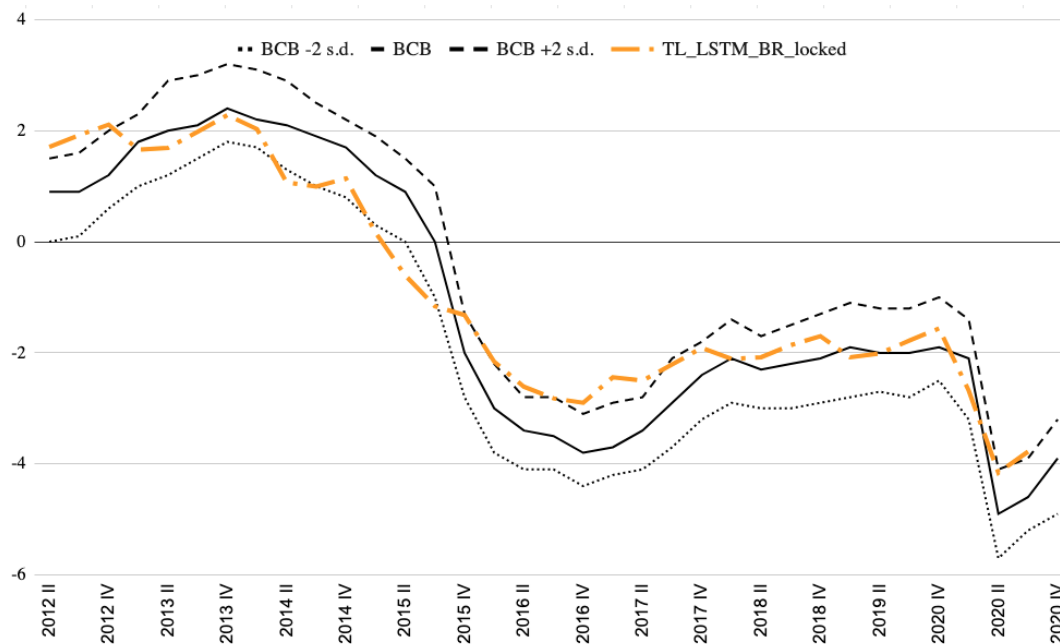

Figure 11 – Output GAP - Central Bank of Brazil (BCB) and Transfer Learning Model

---

[5]   We rebuilt the dataset based on the graph and text data at (Banco Central do Brasil, 2020b, 56) because it does not publish the output gap series.
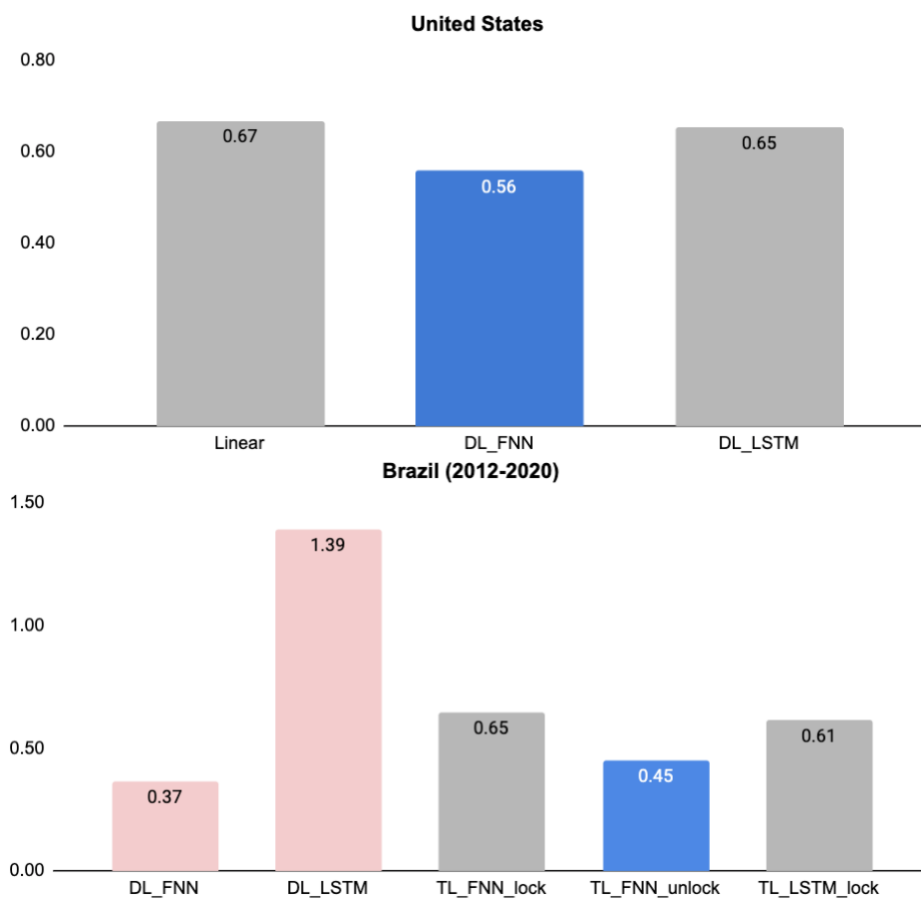
Figure 12 – MAE out-of-sample

6 Representation Learning for Interpolation, Distribution, and Extrapolation of Time Series by Related Series (RIDE)

6.1 Introduction

Economic and financial analysts commonly use time series modeling to predict future values, analyze their salient properties and characteristics, and monitor the current state of the economy. However, as it is not uncommon that some relevant variables are not available with the desired time and frequency, there is a research program aimed at transforming low-frequency economic variables into high-frequency variables (Pavía-Miralles, 2010). The procedure by Chow and Lin (1971), a generalization of Friedman (1962), is a seminal reference to traditional statistical methods that use related series, also called indicators. The main attraction of this approach, based on a linear regression model set of indicators with first-order stationary autoregressive errors, is its simplicity (Mazzi and Proietti, 2015). However, these methods are restricted to a limited number of indicator series due, until the recent past, to data and computation restrictions. Currently, large data sets are readily available, and models with hundreds of parameters are easily estimated (Angelini et al., 2016), allowing new approaches to open questions[1], such as using data with daily frequency, which is increasingly available[2].

The purpose of this Chapter is to apply deep learning for mapping low-frequency from high-frequency variables[3]. Since deep learning methods are a way of learning representations, those that are formed by the composition of multiple non-linear transformations, with the goal of yielding more abstract – and ultimately more useful – representations (Bengio et al., 2013), and also inspired by Chow and Lin (1971), we denominate our strategy Representation Learning for Interpolation, Distribution, and Extrapolation of Time Series by Related Series (RIDE).

---

[1] In a comprehensive analysis, Pavía-Miralles (2010) describes the main techniques used, their advantages and disadvantages, as well as the open questions related to this field of research.

[2] (Chow and Lin, 1971, 372) mention that despite dealing only with estimating a monthly series from their quarterly data and considering related monthly series, the theory, with minimal modifications, applies to the estimation of others frequencies.

[3] 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).

6.2 Literature

6.2.1 Interpolation, Distribution and Extrapolation of Time Series by Related Series

Milton Friedman was an American statistician and economist awarded the Nobel Prize in Economics in 1976 and known as the father of monetarism. He was an intellectual leader with vast production. In 1962, Friedman (1962) published *The Interpolation of Time Series by Related Series*, a detailed analysis of the procedures adopted to construct time series related to economic phenomena. He highlighted the problem of estimating intermediate values that emerge, for example, when using data from a biennial census for annual estimation of national income or even aggregating accounting information between groups of banks that report them in different periods and frequencies. One of the most common operations performed in this process is interpolation: estimating some component for dates for which it is not directly available from known values of that component for other dates (Friedman, 1962, 729). A closely related operation, also widely used, is the distribution of a known total of a time unit among the shorter time units, of which the longest unit is composed (Friedman, 1962, 730). Another related operation is the extrapolation of some components to more recent dates to obtain current numbers before available reference data that will be used later in the interpolation (Friedman, 1962, 730). Directing his analysis to interpolation, he first states that this process, when done based only on the series being interpolated, takes the form of an old and well-known mathematical problem with extensive literature but rarely used in the construction of economics time series. The procedure most used incorporates series known or assumed to be related to the series to be interpolated (Friedman, 1962, 730). He argues, however, that this procedure has not been explored more rigorously, noting that each researcher uses their *ad hoc* procedure. Friedman establishes that the interpolation of a series from related series involves two steps: 1) selection of the related series to be used; 2) interpolation of the target series from related series. He clarifies that his focus is on techniques for interpolation and that he will not deal with selection (step 1). More specifically, he will deal with the restricted case where only a related series is used (Friedman, 1962, 731). After systematically presenting the methods used and associated errors and discussing various technical details involved in the choices to be made, he presents his conclusions (Friedman, 1962, 751-2), highlighting: i) mathematical interpolation and interpolation from related series are not substitute methods, they are complementary; ii) related series can improve interpolation; iii) correlation between series and deviations from trends are relevant points; iv) interpolation must be performed only on the part of the series that is unknown for the dates for which the interpolation will be performed, never generally covering the entire period.

Chow and Lin (1971) propose a unified approach to the problems of interpolation, extrapolation, and distribution that Friedman (1962), despite acknowledging to be related, dealt with them in isolation. Like Friedman (1962), they point out that the variable selection process is beyond their

scope (Chow and Lin, 1971, 372). As a solution, they present the derivation of an unbiased linear best estimator, assuming that the monthly observations of the y series to be estimated satisfy a multiple regression relationship with p related series $x1, ..., x_p$. In the sample period of $3n$ months, where n is the number of quarters, the relation is

$$y = X\beta + u , \tag{6.1}$$

where y is $3n$x1, X is $3n$xp and u is a random vector with zero mean and covariance matrix V, and

$$\beta = \left(X'_. V_.^{-1} X_. \right)^{-1} X'_. V_.^{-1} y_. \tag{6.2}$$

is the least squares estimator resulting from the regression coefficients using n quarterly observations of the sample, with the vectors subscripted with a dot indicating that they refer to quarterly data, and

$$\hat{u}_. = y_. - X_. \hat{\beta} \tag{6.3}$$

is the nx1 residual vector of the regression using quarterly data.

Chow and Lin (1971) describe details of their approach, such as the treatment distinction between flow and stock series or the estimation of V in three different ways. Thus, according to the authors, with a straightforward application of the unbiased linear best estimate theory to a regression model, the problems of interpolation, distribution, and extrapolation of a time series by related series can be solved in a unified way. The resulting estimator applies to all these cases. The usefulness of this method in practice, concerning the estimation of monthly economic time series, certainly depends on the validity of the assumed regression model or the possibility of finding related series that make the regression model a good approximation to reality. According to (Pavía-Miralles, 2010, 455), Chow and Lin (1971) is probably the most influential and cited article in the field of research related to interpolation, distribution, and extrapolation of time series that use related series, or indicators. This procedure has two main advantages over those that do not use indicators: i) better foundations in constructing hypotheses, and ii) they are more efficient because they adopt relevant economic and statistical information. Moreover, as the main disadvantage, it indicates that the results obtained are influenced by choice of indicators.

6.2.2 Artificial Neural Networks and Economics

Directing our attention to applied papers in economics, 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 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, Varian and Choi (2009) found that simple seasonal autoregressive 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. Lastly, 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.

After reviewing this literature, we realize that artificial neural networks, to the best of our knowledge, were not applied to interpolate, distribute, and extrapolate time series by related series until now. Table 2 summarizes a comparison between the Chow and Lin (1971) and our approach.

Table 2 – A comparison between two approaches

	Chow and Lin	RIDE
Performance metrics	Minimize error	Minimize error
Influenced by the indicator's choice	Yes	Yes
Unified approach	Yes	Yes
Assumption about variables relationship	Linear	Non-linear
Analytic demonstration	Yes	No
Big data input	Hard to optimize	Fits well

6.3   Mapping GDP

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 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 the National Account systems. As a result, many coincident indicators with a wide range of techniques have been developed in economics. The relevance of this variable is indisputable, both for the monetary authority and for diverse economic agents. *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?* (Hinds

et al., 2017, 35). For that purpose, economists have developed models to estimate economic activity in response to the regular flow of data.

Here, we apply deep learning (3.1.1) as a strategy for mapping the Brazilian GDP. With the end-to-end approach allowed by this technique, we propose to map a set of high-frequency variables to fit the GDP. When evaluating an end-to-end strategy, we want to observe whether deep learning models, known as universal approximators, perform well to this task. Since the higher frequency variable is generally not observable, as in the typical case where we want a monthly from a quarterly available GDP, we do not know the actual value to verify the performance, so we adopted two approaches, i) as highlighted in 3.1.3, the factors that determine how well a machine learning algorithm will perform are its ability to make the training error small, and make the gap between training and test error small (Goodfellow et al., 2016, 109), and ii) comparing the results with other approaches, but understanding that none finds the true values.

6.3.1   Implementation

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 twenty three monthly variables as indicators, as presented in the Appendix A, Table 5, from January 1996 until March 2021. As mentioned in 6.2, the selection of variables is a relevant point. In the first place, our choice reflects the knowledge about which types of variables are related to GDP. It is an *ad hoc* decision and follows the logic of two of the most relevant activity 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 indexes, mainly followed by agents interested in evaluating the current state of the Brazilian economy, adopt the strategy known as accounting indices, that is, an index composed of many proxy variables that are known to be related to the variables used to prepare the official GDP. Naturally, this choice is not perfect, and it encounters additional obstacles. Specifically for our application, we opted for series available for more extended periods, which reduces their quantity. Proxies variables appear, change, and are discarded over time. If we restrict our sample to a shorter period, we would have more but smaller series. Choosing a longer versus a wider database is a common trade-off in this type of problem.

We computed the input features and the target in level, normalized[4]. We split the dataset into train, validation, and test sets. Then, we define a function that creates a neural network with hidden layers, ReLU, tanh, or sigmoid as activation functions, and L1 (Lasso) or L2 (Ridge) regularization to reduce overfitting. Next, we retrain the model with the optimal hyperparameters, selected with Hyperband, to evaluate the results, with mean squared error as a loss function and the adaptive moment estimation (Adam) (Kingma and Ba, 2017) for optimization. A browser-based version of the proposed model, without the grid search function, is available at http://www.deeplearningeconomics.com/RIDE/.

6.3.2   Results

The results demonstrate the deep learning's suitability for interpolation, distribution, and extrapolation of time series by related series. Figures 13, 14, 15, 16, and 17 consolidate the results found, that are slightly different each time the models are run due to different compositions in the selected data sets for training, validation, and test.

In terms of learning, the curves of the training and validation set behave as expected (Figure 13), that is, over time (epochs), the highest error reduces and stabilizes at a low level, being the validation error a little higher than the training error.

6.3.2.1   Interpolation and Distribution

The interpolation and distribution concepts (Friedman, 1962) described in 6.2 have evolved in ways that are sometimes taken to be similar, as noted by Issler and Notini (2014)[5]. A subtle difference in the application to GDP data can be seen when some quarters referring to the target (GDP) within the sample are missing, but there are monthly indicators for these periods, in which case it would be necessary to interpolate GDP values, in addition to distributing them in the quarter.

The performance of the deep learning model to generate monthly data is shown in Figure 14, where we see the good fitting related to the official quarterly data, and in figures 15 and 16 the comparison of RIDE model with benchmarks for monthly GDP data, namely the Chow and Lin (1971) method, the GDP Monitor from Instituto Brasileiro de Economia (2015), and the Central Bank Economic Activity Index (IBC-Br)[6] from Banco Central do Brasil (2016). The RIDE correlation (year-over-year growth) with IBC-Br is 0.9266 and with GDP Monitor is 0.9213.

---

[4]   The resulting monthly GDP is not seasonally adjusted as we do not seasonally adjust input features. Instead, we apply the minimum necessary data transformations for good learning performance. Seasonal adjustment, commonly required for monthly GDP analyses, can be made a posteriori outside the proposed model.

[5]   *First, a word of caution about using the term interpolation here. GDP is a flow variable for which we possess quarterly observations we want to distribute within the months in the quarter. [...] the problem of allocating a quarterly flow as such is referred to as distribution, whereas interpolation estimates monthly values of stock variables from quarterly values. Despite this technical distinction for flow and stock variables, several authors still refer to interpolated GDP, a term that is now ingrained in the literature, being the reason why we employ it here* (Issler and Notini, 2014, 7).

[6]   The IBC-Br presents a level shift in 2015 due to methodological issues.

Despite using different strategies, the different approaches result in a monthly variable that follows GDP. As we have commented before, none of them can be taken as the authentic higher frequency movement, but they are still essential to guide the decision-making process of governments and the private sector.

6.3.2.2 Extrapolation

Lastly, we can use the higher frequency indicator series to infer the GDP already realized but not yet disclosed. There are many questions related to this area of research, one of the most relevant being the difference between near real-time versus real-time proxies[7]. However, these details are not the object of this work. What we can show is the extrapolation capacity of the proposed model. The Figure 17 compares the Brazilian GDP realized between 2020Q3 and 2021Q1 with an out-of-sample RIDE estimate. Like Chow and Lin (1971) and others, RIDE is a global model where the parameters are calculated based on the entire sample; therefore, in cases where the indicator series extends beyond the period cover by the endpoints of the benchmark series, the target will be extrapolated with the global target/indicator ratio. Furthermore, like in the others approaches, when using RIDE the analyst can diversify the use of variables, sample periods and evaluate performance one or several steps ahead, in real or near real-time.

---

[7]   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.
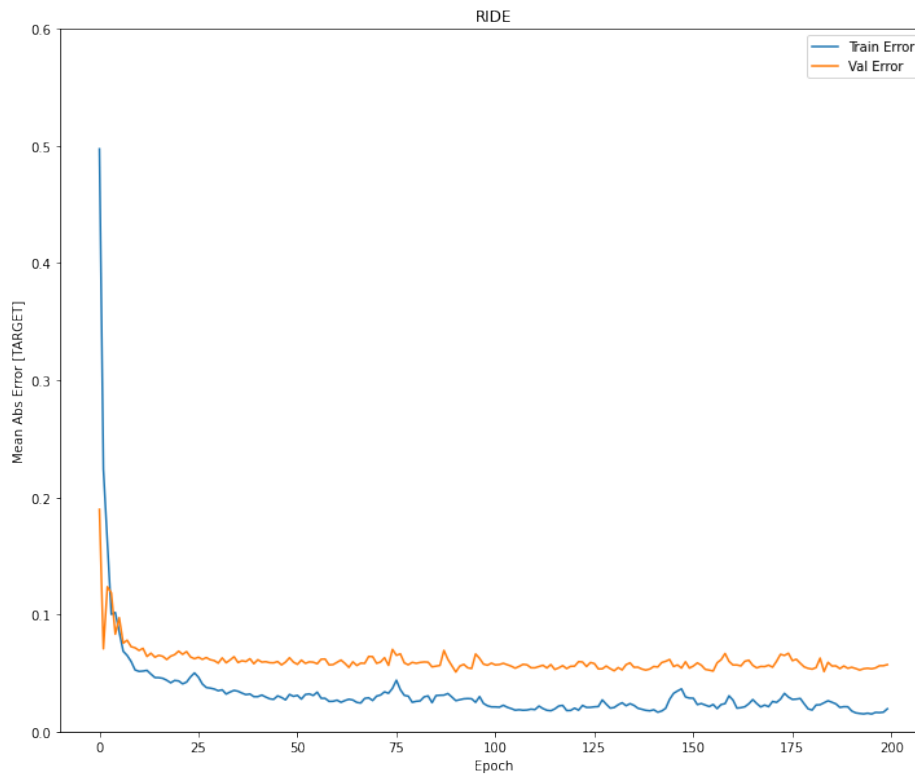
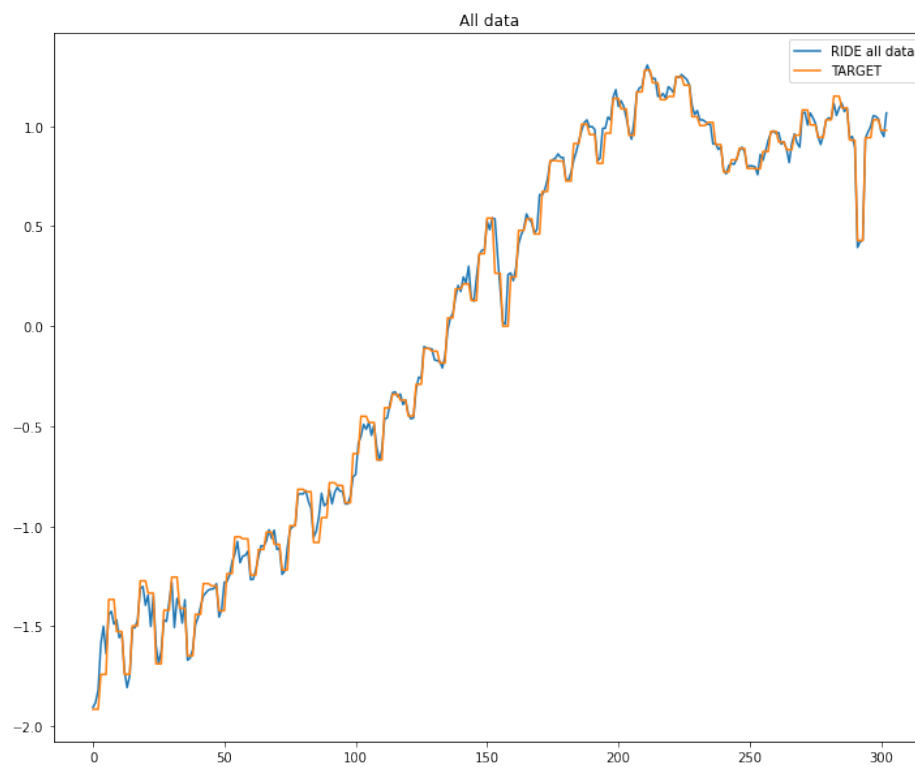Figure 13 – Training and validation curves
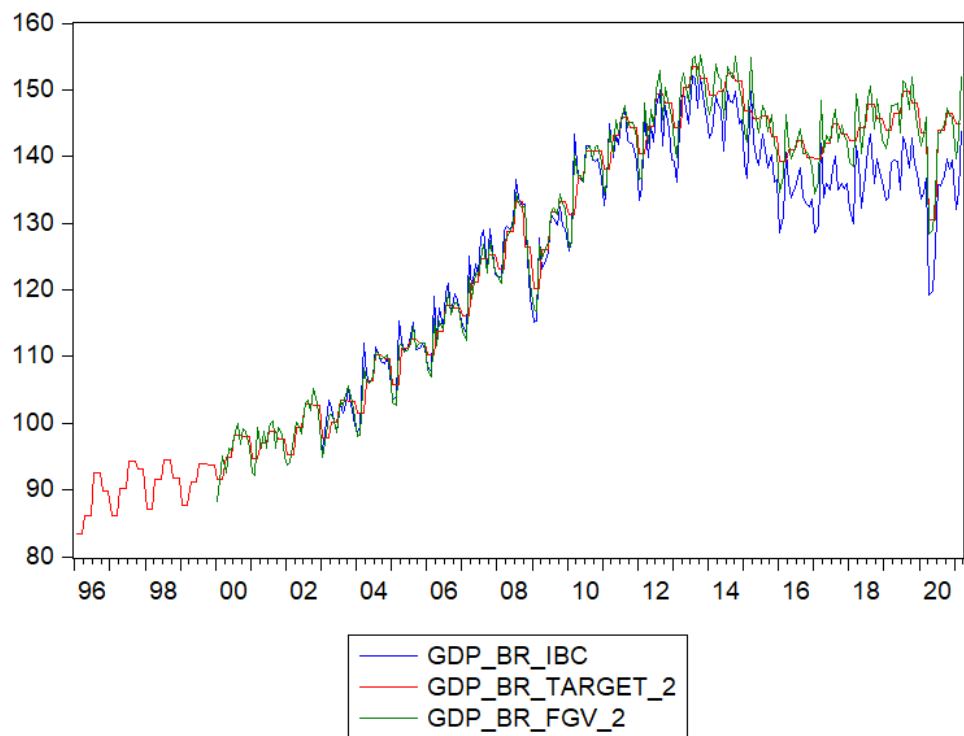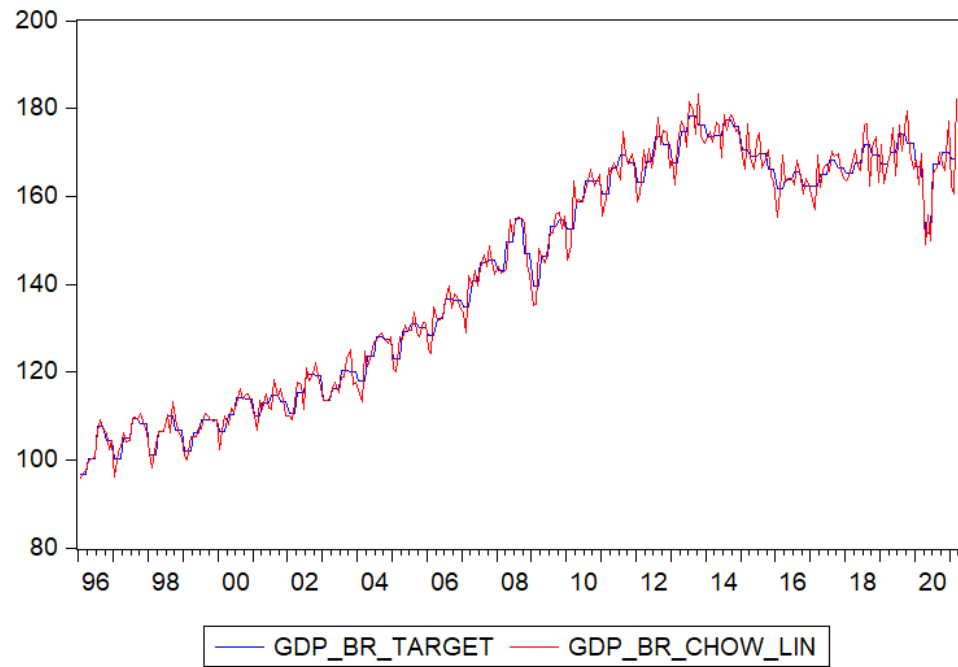

Figure 14 – RIDE mapping GDP

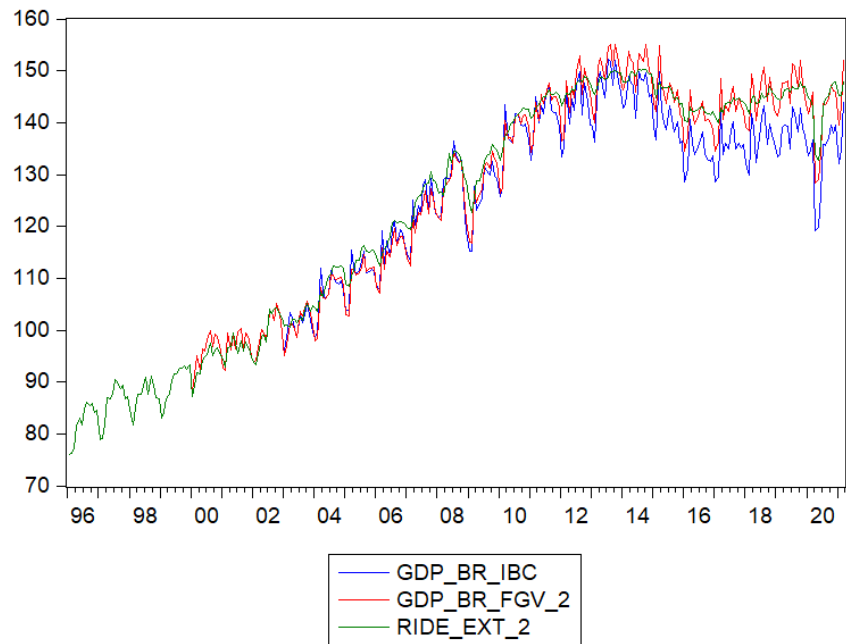Figure 15 – Chow Lin, IBC-Br and FGV Monitor fitting GDP

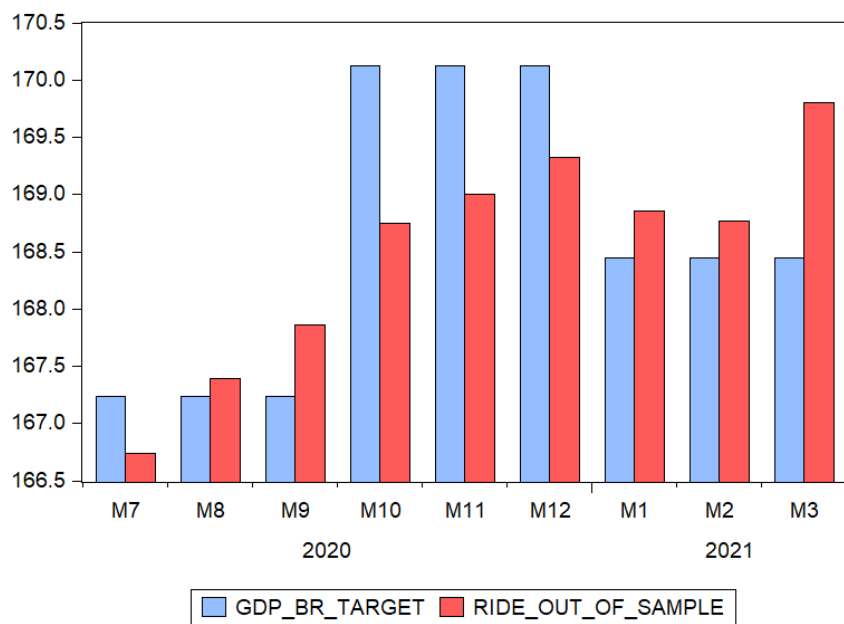Figure 16 – RIDE, IBC-Br and FGV Monitor

Figure 17 – RIDE extrapolation

7 Conclusion

In this work, deep learning was applied to macroeconomics in two approaches: to transfer learning and to interpolate, distribute, and extrapolate time series by related series.

Transfer learning was proposed as an additional strategy for empirical macroeconomics. First, we presented theoretical concepts related to transfer learning and proposed a connection with a typology related to macroeconomic models. At this point, there is already an enormous potential to be explored in the connection between transfer learning and applied macroeconomics. Then, we refer to several algorithms used for transfer learning and detail our choice, artificial neural networks with multiple layers hidden between the input and output layers, also known as deep learning. Secondly, we explore the proposed strategy empirically. As we showed, data from different but related domains, a type of transfer learning, helped to identify the business cycle phases when there was no business cycle dating committee or to estimate a quick first economic-based output gap. In both cases, the proposed strategy also helped to improve the learning when data was limited. The approach demonstrated excellent empirical performance with data from the US, Europe and Brazil, emerging as a potential supplementary tool for governments and the private sector to conduct their activities in the light of national and international economic conditions. The proposed strategy allows a quick application of the economic-based models to other geographic areas. Once trained, the machine inferred the business cycle state and the output gap based on the theory chosen for training. To the best of our knowledge, the combined deep and transfer learning approach is underused for application to macroeconomic problems, indicating that there is plenty of room for research development.

Additionally, since deep learning methods are a way of learning representations, we applied deep learning for mapping low-frequency from high-frequency variables, a task also known as interpolation, distribution, and extrapolation of time series by related series. The results obtained from Brazilian data demonstrate deep learning's suitability for this task.

Next, this research aims to improve the models to (i) enable other data types such as texts and images as inputs and (ii) build general representation learning models such as MobileNets for tasks applied to economics.
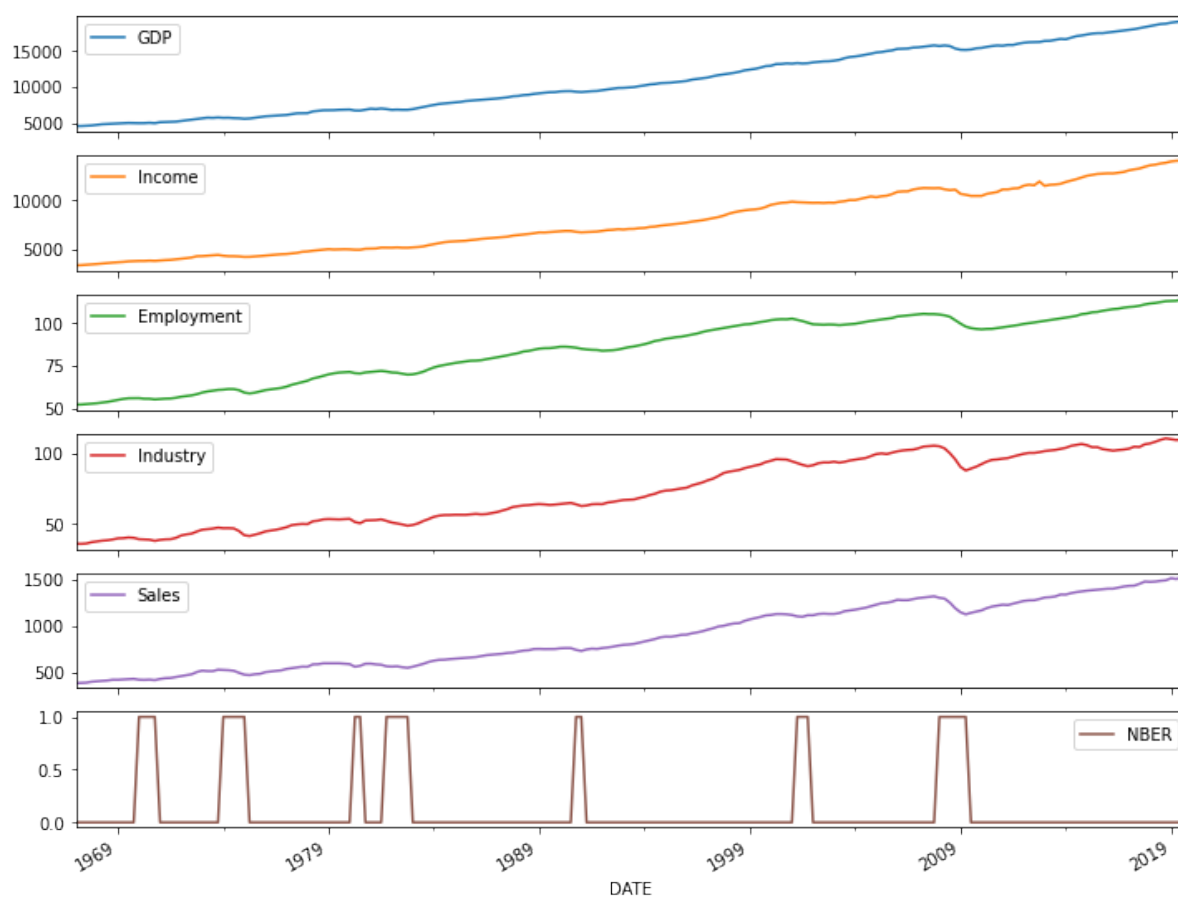
A Data description

Figure 18 – U.S. raw data for learning Business Cycle.

Table 3 – Dataset for Business cycle identification

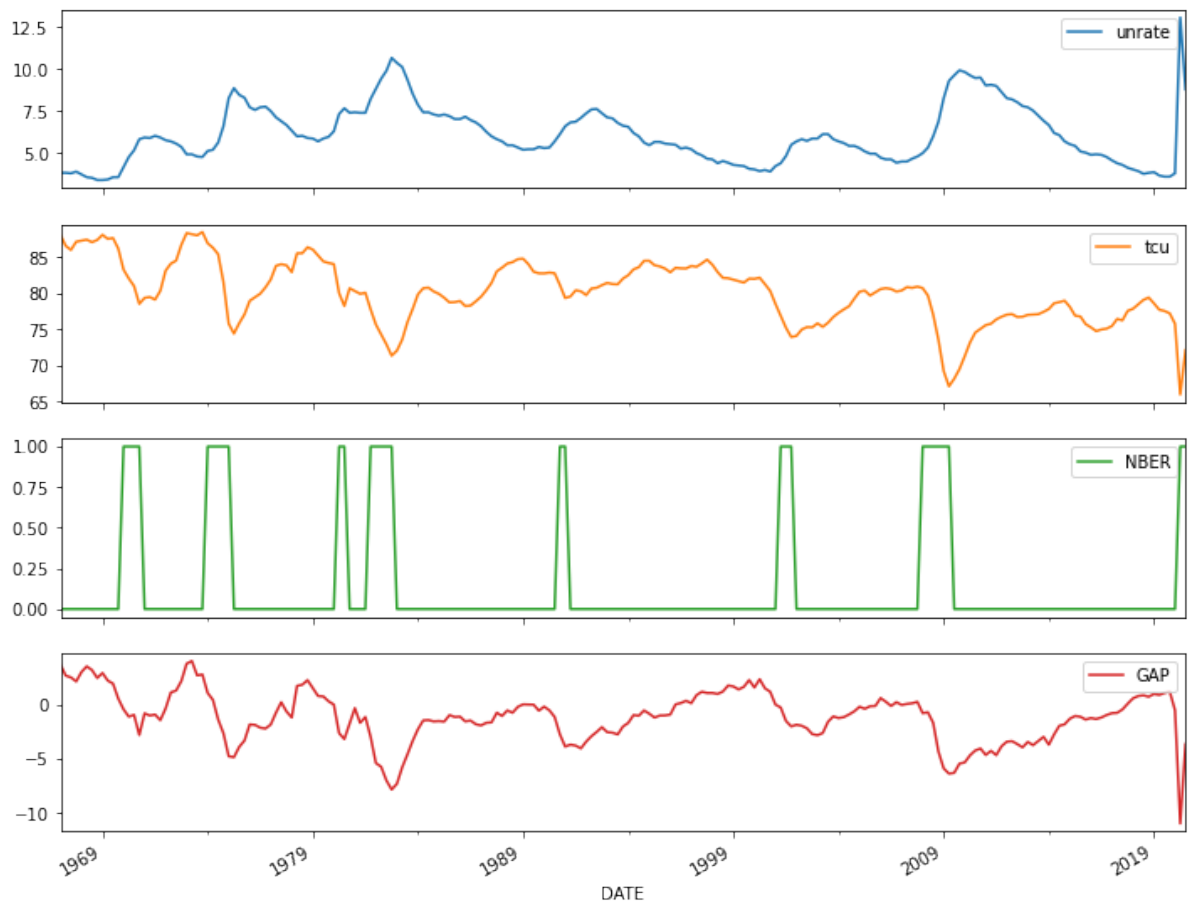| FRED Code	Series	Start	Characteristics of the raw data	Source
	UNITED STATES			
USRECQ	NBER based Recession Indicator	1967:Q1	Recession: 1 = true ; 0 = false	NBER
GDPC1	Real Gross Domestic Product	1967:Q1	Billions of Chained 2012 Dollars, s.a.	FRED
PIECTR	Real personal income ex current transfers	1967:Q1	Billions of Chained 2012 Dollars, s.a.	FRED
PRS85006013	Nonfarm Business Sector employment index	1967:Q1	Index 2012 = 100, s.a.	FRED
IPB50001SQ	Industrial production index	1967:Q1	Index 2012 = 100, s.a.	FRED
CQRMTSPL	Real manufacturing and trade ind. sales	1967:Q1	Millions of Chained 2012 Dollars, s.a.,	FRED
	EURO AREA			
N/A	CEPR based Recession Indicator	2005:Q1	Recession: 1 = true — 0 = false	CEPR
CLVMNACSCAB1GQEA19	Real Gross Domestic Product (19 countries)	2005:Q1	Millions of Chained 2010 Euros, s.a.	FRED
NAEXKP02EZQ189S	Private Final Consumption Expenditure	2005:Q1	Billions of Chained 2012 Dollars, s.a.	FRED
LFESEETTEZQ647S	Employees	2005:Q1	Persons, s.a.	FRED
PRMNTO01EZQ657S	Total Manufacturing Production	2005:Q1	Growth Rate Previous Period, s.a.	FRED
SLRTTO01EZQ657S	Volume of Total Retail Trade sales	2005:Q1	Growth Rate Previous Period	FRED
	BRAZIL			
N/A	CODACE based Recession Indicator	2000:Q1	Recession: 1 = true — 0 = false	CODACE
NAEXKP01BRQ652S	Total Gross Domestic Product	2000:Q1	Chained 2000 Real, s.a.	FRED
NAEXKP02BRQ189S	Private Final Consumption Expenditure	2000:Q1	Chained 2000 Real, s.a.	FRED
N/A	Registered Employees Index	2000:Q1	Index Dez-2001 = 100, s.a.	BCB
BRAPROINDQISMEI	Production of Total Industry	2000:Q1	Index 2015 = 100, s.a.	FRED
BRASARTQISMEI	Total Retail Trade	2000:Q1	Index 2015 =100, s.a.	FRED

Figure 19 – U.S. raw data for learning Output Gap

Table 4 – Dataset for Output Gap Estimation

FRED Code	Series	Start	Characteristics of the raw data	Source
	UNITED STATES			
USRECQ	NBER based Recession Indicator	1967:Q1	Recession: 1 = true ; 0 = false	NBER
GDPC1	Real Gross Domestic Product	1967:Q1	Billions of Chained 2012 Dollars, s.a.	FRED
GDPPOT	Real potential Gross DOmestic Product	1967:Q1	Billions of Chained 2012 Dollars	FRED
UNRATE	Unemployment rate	1967:Q1	percent, s.a., quarterly average	FRED
TCU	Capacity utilization index	1967:Q1	percent, s.a., quarterly average	FRED
	BRAZIL			
N/A	CODACE based Recession Indicator	2012:Q2	Recession: 1 = true — 0 = false	CODACE
NAEXKP01BR	Total Gross Domestic Product	2012:Q2	Chained 2000 Real, s.a.	FRED
TFP	Total Factor Productivity Level (PPP Brazil and U.S.)	2012:Q2	Index U.S. = 1	FRED
N/A	Registered Employees Index	2000:Q1	Index Dez-2001 = 100, s.a.	BCB

Table 5 – Dataset for RIDE

Code	Series	Range	Source
gdp	Gross domestic product - **Target**	1996:M1 to 2021:M3	IBGE
epeRes	Electricity consumption: household	1996:M1 to 2021:M3	Empresa de Pesquisa Energética
epeCom	Electricity consumption: commerce	1996:M1 to 2021:M3	Empresa de Pesquisa Energética
epeInd	Electricity consumption: industry	1996:M1 to 2021:M3	Empresa de Pesquisa Energética
expQ	Export goods: quantum	1996:M1 to 2021:M3	Funcex
impQ	Import good: quantum	1996:M1 to 2021:M3	Funcex
expBasic	Export goods: raw material	1996:M1 to 2021:M3	Funcex
expSemi	Export goods: semi	1996:M1 to 2021:M3	Funcex
expManuf	Export goods: industrial	1996:M1 to 2021:M3	Funcex
expAgroDolar	Export goods: agro	1996:M1 to 2021:M3	Funcex
anfaveaProd	Vehicles production	1996:M1 to 2021:M3	IPEA data
anfaveaLic	Vehicles sales	1996:M1 to 2021:M3	IPEA data
cagedTransf	Formal jobs: industry	1996:M1 to 2021:M3	Ministry of Labor
cagedConstr	Formal jobs: construction	1996:M1 to 2021:M3	Ministry of Labor
cagedCom	Formal jobs: commerce	1996:M1 to 2021:M3	Ministry of Labor
cagedServ	Formal jobs: services	1996:M1 to 2021:M3	Ministry of Labor
ipa	Industrial index prices	1996:M1 to 2021:M3	IPEA data
cambio	Exchange rate	1996:M1 to 2021:M3	IPEA data
poup	Savings	1996:M1 to 2021:M3	BCB
m1	M1 money supply	1996:M1 to 2021:M3	BCB
m2	M2 money supply	1996:M1 to 2021:M3	BCB
ibov	Stock market	1996:M1 to 2021:M3	IPEA data
embi	Sovereign risk	1996:M1 to 2021:M3	IPEA data
RecRFBIPCA	Taxes	1996:M1 to 2021:M3	Receita Federal do Brasil

B Selected codes

```python
1 neg, pos = np.bincount(raw_data['NBER'])
2 total = neg + pos
3 # set initial bias
4 initial_bias = np.log([pos/neg])
```

Listing B.1 – Python code for examine the class label imbalance.

```python
1 # Use a utility from sklearn to split out dataset
2 train_df, test_df = train_test_split(log_1df, test_size=0.4, random_state=0,
      shuffle=False)
3 train_df, val_df = train_test_split(train_1df, test_size=0.3, random_state=0,
      shuffle=False)
4 # Form np arrays of labels and features
5 train_labels = np.array(train_df.pop('NBER'))
6 val_labels = np.array(val_df.pop('NBER'))
7 test_labels = np.array(test_df.pop('NBER'))
8
9 train_features = np.array(train_df)
10 val_features = np.array(val_df)
11 test_features = np.array(test_df)
```

Listing B.2 – Python code for split the dataset.

```python
1 scaler = StandardScaler()
2 train_features = scaler.fit_transform(train_features)
3 val_features = scaler.transform(val_features)
4 test_features = scaler.transform(test_features)
```

Listing B.3 – Python code for normalize the input features

```python
1 METRICS = [
2     keras.metrics.TruePositives(name='tp'),
3     keras.metrics.FalsePositives(name='fp'),
4     keras.metrics.TrueNegatives(name='tn'),
5     keras.metrics.FalseNegatives(name='fn'),
6     keras.metrics.BinaryAccuracy(name='accuracy'),
7     keras.metrics.Precision(name='precision'),
8     keras.metrics.Recall(name='recall'),
9     keras.metrics.AUC(name='auc'),
10 ]
11
12 def make_model(hp, metrics = METRICS, output_bias = initial_bias):
13     output_bias = tf.keras.initializers.Constant(output_bias)
```

```
14
15    model = keras.Sequential()
16
17    # Tune the number of units in the layers
18    # Choose an optimal value between 16−256
19    hp_units = hp.Int('units', min_value = 16, max_value = 256, step = 16)
20
21    model.add(keras.layers.LSTM(units = hp_units, input_shape = (1,train_features.
          shape[−1],), dropout = 0.3)) #LSTM layer
22
23    model.add(keras.layers.Dense(units = hp_units, activation = 'relu')) #Dense1
24    model.add(keras.layers.Dense(units = hp_units, activation = 'relu')) #Dense2
25    model.add(keras.layers.Dense(units = hp_units, activation = 'relu')) #Dense3
26    model.add(keras.layers.Dense(units = hp_units, activation = 'relu')) #Dense4
27
28    model.add(keras.layers.Dropout(0.5)) # To prevent overfiting
29
30    model.add(keras.layers.Dense(1, activation='sigmoid', bias_initializer=
          output_bias)) # Output layer
31
32    # Tune the learning rate for the optimizer
33    # Choose an optimal value from 0.01, 0.001, or 0.0001
34    hp_learning_rate = hp.Choice('learning_rate', values = [1e−2, 1e−3, 1e−4])
35
36    model.compile(optimizer = keras.optimizers.Adam(learning_rate = hp_learning_rate)
          ,
37      loss = keras.losses.BinaryCrossentropy(),
38      metrics = metrics)
39
40    return model
```

Listing B.4 – Python code for define the model and metrics

```
1  tuner = kt.Hyperband(make_model,
2             kt.Objective('val_auc', direction='max'), #maximizes area under the ROC
                curve
3             max_epochs = 10,
4             factor = 3,)
5
6  tuner.search(train_features, train_labels,
7        epochs=50,
8        validation_data=(val_features, val_labels), callbacks = [ClearTrainingOutput
            ()])
9
10 #Get the optimal hyperparameters
11 best_hps = tuner.get_best_hyperparameters(num_trials = 1)[0]
```

```
12
13 #Retrain the model with the optimal hyperparameters
14 model = tuner.hypermodel.build(best_hps)
15 baseline_history = model.fit(train_features, train_labels,
16                epochs=50,
17                validation_data=(val_features, val_labels))
```

Listing B.5 – Python code for hyperparameters search and retrain

```
1 # Fine−tune from this layer onwards
2 fine_tune_at = 3
3 #Freeze all the layers before the fine_tune_at layer
4 for layer in model.layers[:fine_tune_at]:
5   layer.trainable = False
```

Listing B.6 – Python code for unlock some layers

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: https://www.tensorflow.org/. software available from tensorflow.org.

Abberger, K., Graff, M., Campelo, A.J., Gouveia, A.C.L., Müller, O., Sturm, J.E., 2020. The Global Economic Barometers: Composite indicators for the world economy. Research Working Paper KOF WP 471-20. KOF Swiss Economic Institute.

Angelini, E., Henry, J., Marcellino, M., 2016. Interpolation and backdating with a large information set. Journal of Economic Dynamics and Control 30, 2693–2724.

Apel, M., Jansson, P., 1999. A thoery-consistent system approach for estimating potential output and the nairu. Economics Letters , 271–275.

Arnold, A., Nallapati, R., Cohen, W., 2007. A comparative study of methods for transductive transfer learning, in: Proceedings of the 7th IEEE International Conference on Data Mining Workshops, pp. 77–82.

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

Banco Central do Brasil, 2020a. Inflation report, n.3, 22, 57–62.

Banco Central do Brasil, 2020b. Inflation report, n.4, 22, 56.

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

Bengio, Y., 2012. Deep learning of representations for unsupervised and transfer learning, in: Guyon, I., Dror, G., Lemaire, V., Taylor, G., Silver, D. (Eds.), Proceedings of ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, Bellevue, Washington, USA. pp. 17–36.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1798–1828.

Berge, T.J., 2013. Predicting recessions with leading indicators: model averaging and selection over the business cycle. Research Working Paper RWP 13-05. Federal Reserve Bank of Kansas City.

Beveridge, S., Nelson, C.R., 1981. A new approach to the decomposition of economic time series into permanent and transitory components with particular attention to the measurement of the business cycle. Journal of Monetary Economics , 151–174.

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.

Blanchard, O., 2018. On the future of macroeconomic models, in: Vines, D., Wills, S. (Eds.), Rebuilding macroeconomic theory. Oxford Review of Economic Policy, pp. 43–54.

Blanchard, O., Quah, D., 1989. The dynamic effects of aggregate demand and aggregate supply disturbances. American Economic Review , 653–658.

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

Camacho, M., Perez-Quiros, G., Poncela, P., 2018. Markov-switching dynamic factor models in real time. International Journal of Forecasting 34, 598–611.

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

Chauvet, M., 1998. An econometric characterization of business cycle dynamics with factor structure and regime switching. International Economic Review 39, 969–96.

Chauvet, M., 2001. A monthly indicator of brazilian gdp. Brazilian Review of Econometrics 21, 1–47.

Chauvet, M., Guimaraes, R., 2021. Transfer Learning for Business Cycle Identification. Working Paper Series 545. Banco Central do Brasil.

Chauvet, M., Hamilton, J.D., 2005. Dating Business Cycle Turning Points. NBER Working Papers. National Bureau of Economic Research, Inc.

Chauvet, M., Piger, J., 2008. A comparison of the real-time performance of business cycle dating methods. Journal of Business Economic Statistics 26, 42–49.

Chauvet, M., Yu, C., 2006. International business cycles: G7 and OECD countries. Economic Review 91, 43–54.

Chow, G., Lin, A., 1971. Best linear unbiased interpolation, distribution ans extrapolation of time series by related series. The Review of Economics and Statistics 53, 372–375.

Christiano, L.J., 2012. Christopher a. sims and vector autoregressions. The Scandinavian Journal of Economics 4, 1–23.

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

Cuba-Borda, P., Mechanick, A., Raffo, A., 2018. Monitoring the World Economy : A Global Conditions Index. IFDP Notes. Board of Governors of the Federal Reserve System (U.S.).

Dai, W., Yang, Q., Xue, G., Yu, Y., 2007a. Boosting for transfer learning, in: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200.

Dai, W., Yang, Q., Xue, G., Yu, Y., 2007b. Transferring naive bayes classifiers for text classification, in: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 540–545.

D'Auria, F., Denis, C., Havik, K., Morrow, K.M., Planas, C., Raciborski, R., Roger, W., Rossi, A., 2010. The production function methodology for calculating potential growth rates and output gaps. European Economy - Economic Papers 2008 - 2015 420. Directorate General Economic and Financial Affairs (DG ECFIN), European Commission.

Deng, L., Yu, D., 2014. Deep Learning: Methods and Applications. Technical Report. Microsoft.

Duchi, J.C., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159.

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.

Estrella, A., Mishkin, F., 1998. Predicting u.s. recessions: Financial variables as leading indicators. The Review of Economics and Statistics 80, 45–61.

Estrella, A., Rodrigues, A.P., Schich, S., 2000. How stable is the predictive power of the yield curve? evidence from Germany and the United States. Staff Reports 113. Federal Reserve Bank of New York.

Fossati, S., 2016. Dating US business cycles with macro factors. Studies in Nonlinear Dynamics & Econometrics 20, 529–547.

Friedman, M., 1962. The interpolation of time series by related series. Journal of the American Statistical Association 57, 729–757.

Garbellano, J., 2016. Nowcasting recessions with machine learning: New tools for predicting the business cycle, in: Thesis.

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.

Gerlach, S., Smets, F., 1999. Output gaps and monetary policy in the emu area. European Economic Review , 801–812.

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

Giusto, A., Piger, J., 2017. Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting 33, 174–184.

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.

Hamilton, J.D., 1989. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. Econometrica 57, 357–384.

Hamilton, J.D., 2018. Why you should never use the hodrick-prescott filter. The Review of Economics and Statistics , 831–843.

Harding, D., Pagan, A., 2005. A suggested framework for classifying the modes of cycle research. Journal of Applied Econometrics 20, 151–159.

Harvey, A.C., Jaeger, A., 1993. Detrending, stylized facts and the business cycle. Journal of Applied Econometrics , 231–247.

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.

Hodrick, R.J., Prescott, E., 1981. Post-War U.S. Business Cycles: An Empirical Investigation. Discussion Papers 451. Northwestern University, Center for Mathematical Studies in Economics and Management Science.

Hoptroff, R.G., Bramson, M.J., Hall, T.J., 1991. Forecasting economic turning points with neural nets, in: IJCNN-91-Seattle International Joint Conference on Neural Networks, pp. 347–352.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.

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

Issler, J.V., Notini, H.H., 2014. Estimating brazilian monthly gdp: a state-space approach. Ensaios Econômicos 757, 1–24.

Jackson, B., Rege, M., 2019. Machine learning for classification of economic recessions, in: 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), pp. 31–38.

James, A., Abu-Mostafa, Y.S., Qiao, X., 2019. Nowcasting recessions using the svm machine learning algorithm. arXiv:1903.03202.

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, 790–794.

Jebara, T., 2004. Multi-task feature and kernel selection for svms, in: Proceedings of the 21th International Conference on Machine Learning.

Kauppi, H., Saikkonen, P., 2008. Predicting u.s. recessions with dynamic binary response models. The Review of Economics and Statistics 90, 777–791.

Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. arXiv:1412.6980.

Kliesen, K.L., McCracken, M.W., 2016. Tracking the U.S. economy with nowcasts. The Regional Economist. Federal Reserve Bank of St. Louis.

Klinkenberg, R., 2003. Predicting phases in business cycles under concept drift.

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, 463–474.

Koske, I., Pain, N., 2008. The Usefulness of Output Gaps for Policy Analysis. OECD Economics Department Working Papers 621. OECD Publishing.

Kouw, W., Loog, M., 2018. An introduction to domain adaptation and transfer learning. CoRR abs/1812.11806. URL: http://arxiv.org/abs/1812.11806.

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

Kuttner, K.N., 1994. Estimating potential output as a latent variable. Journal of Business and Economic Statistics , 361–368.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. arXiv:1603.06560.

Ma, J.B., 2015. Applications of machine learning in forecasting recessions: boosting united states and japan.

Maasoumi, E., Khotanzed, A., Abaye, A., 1994. Artificial neural networks for some macroeconomic series: A first report. Econometric Reviews 13, 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.

Mazzi, G.L., Proietti, T., 2015. Multivariate temporal disaggregation. Eurostat Handbook on Rapid Estimates 13, 729–757.

Mihalkova, L., Huynh, T., Mooney, R.J., 2007. Mapping and revising markov logic networks for transfer learning, in: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 608–614.

Morik, K., Rüping, S., 2002. A multistrategy approach to the classification of phases in business cycles, in: Elomaa, T., Mannila, H., Toivonen, H. (Eds.), Machine Learning: ECML 2002, Springer Berlin Heidelberg. pp. 307–318.

Nasr, G.E., Dibeh, G., Achkar, A., 2007. Predicting business cycle turning points with neural networks in an information-poor economy, in: SCSC.

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

Nyman, R., Ormerod, P., 2017. Predicting economic recessions using machine learning algorithms. arXiv:1701.01428.

Okun, A., 1962. Potential gnp: Its measurement and significance, in: Proceedings of the Business and Economic Statistics Section of the American Statistical Association.

Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q., 2011. Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22, 199–210.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359.

Pavía-Miralles, J., 2010. A survey of methods to interpolate, distribute and extrapolate time series. Journal of Service Science and Management , 449–463.

Piger, J., 2020. Turning points and classification, in: Fuleky, P. (Ed.), Macroeconomic Forecasting in the Era of Big Data. Springer International Publishing, pp. 585–624.

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

Proietti, T., Musso, A., Westermann, T., 2007. Estimating potential output and the output gap for the euro area: a model-based production function approach. Empirical Economics , 85–113.

Qi, M.H., 2001. Predicting us recessions with leading indicators via neural network models, in: Princeton University Senior Theses.

Raffinot, T., Benoit, S., 2019. Investing Through Economic Cycles with Ensemble Machine Learning Algorithms. Working Papers. HAL.

Ruckert, U., Kramer, S., 2008. Kernel-based inductive transfer, in: Machine Learning and Knowledge Discovery in Databases, ser. Lecture Notes in Computer Science. European Conference, ECML/PKDD 2008, pp. 220–233.

Rudebusch, G., Williams, J., 2009. Forecasting recessions: The puzzle of the enduring power of the yield curve. Journal of Business Economic Statistics 27, 492–503.

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

Shackleton, R., 2018. Estimating and Projecting Potential Output Using CBO's Forecasting Growth Model. Working Paper 2018-03. Congressional Budget Office.

Sims, C.A., 1980. Macroeconomics and reality. Econometrica 48, 1–48.

Soybilgen, B., 2018. Identifying US business cycle regimes using dynamic factors and neural network models. MPRA Paper. University Library of Munich, Germany.

St-Amant, P., van Norden, S., 1997. Measurement of the Output Gap: A Discussion of Recent Research at the Bank of Canada. Technical Report 79. Bank of Canada.

Stergiou, C., Siganos, D., 2011. Neural networks. Technical Report. 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, 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 .

Tkacz, G., Hu, S., 1999. Forecasting GDP Growth Using Artificial Neural Networks. Staff Working Papers 99-3. Bank of Canada.

Torrey, L., Shavlik, J., 2009. Transfer learning, in: Olivas, E.S., Guerrero, J.D.M., Sober, M.M., Benedito, J.R.M., López, A. (Eds.), Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques (2 Volumes). chapter 11.

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

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

Vines, D., Wills, S., 2018. The rebuilding macroeconomic theory project: an analytical assessment, in: Vines, D., Wills, S. (Eds.), Rebuilding macroeconomic theory. Oxford Review of Economic Policy, pp. 1–42.

Vines, D., Wills, S., (eds.), 2018. Rebuilding macroeconomic theory. volume 34. Oxford Review of Economic Policy.

Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. Journal of Big Data 3, 1–40.

White, H., 1988. Economic prediction using neural networks: the case of ibm daily stock returns, in: IEEE 1988 International Conference on Neural Networks, 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.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks?, in: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 27. Curran Associates, Inc., pp. 3320–3328. URL: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2016. Understanding deep learning requires rethinking generalization. CoRR abs/1611.03530. URL: http://arxiv.org/abs/1611.03530.