Irving Fisher Committee on
Central Bank Statistics

IFC-Bank of Italy Workshop on “Data Science in Central Banking: Applications and tools”

14-17 February 2022

Deep vector autoregression for macroeconomic data[1]

Marc Agustí and Ignacio Vidal-Quadras Costa, European Central Bank;
Patrick Altmeyer, Delft University of Technology

[1] This contribution was prepared for the workshop. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Italy, the BIS, the IFC or the other central banks and institutions represented at the event.

Deep Vector Autoregression for Macroeconomic Data

Patrick Altmeyer[1], Marc Agusti[2], Ignacio Vidal-Quadras Costa[3]

Vector Autoregression is a popular choice for forecasting time series data. Due to its simplicity and success at modelling monetary economic indicators VAR has become a standard tool for central bankers to construct economic forecasts. A crucial assumption underlying the conventional VAR is that interactions between variables through time can be modelled linearly. We propose Deep VAR: a novel approach towards VAR that leverages the power of deep learning in order to model non-linear relationships. By modelling each equation of the VAR system as a deep neural network, our proposed extension outperforms its conventional benchmark in terms of in-sample fit, out-of-sample fit and point forecasting accuracy. In particular, we find that the Deep VAR is able to better capture the structural economic changes during periods of uncertainty and recession. By staying methodologically as close as possible to the original benchmark, we hope that our approach is more likely to find acceptance in the economics domain.

1 Introduction

As stated by the European Central Bank, the monetary transmission mechanism is the process through which monetary policy decisions affect the economy in general and the price level in particular. Uncertainty with respect to this transmission is generally huge, given that it is characterized by long, variable and uncertain dependencies through time and variables. Hence, it is typically challenging to predict how changes in monetary policy actions affect real economic outcomes. It is therefore of foremost importance for policy-makers to use adequate tools to model the underlying mechanisms.

With this in mind, a lot of research on the forecasting of time series has been developed to assess the effect of current policy decisions on future economic variables. Thanks to this, over the last decades policy makers have had more information when taking decisions. This information usually comes in the form of point estimates and interval forecasts. To come up with these estimates, several methodologies have been developed and applied in the time series forecasting literature.

At the time of writing, one the most common methodologies to produce these estimates is the so-called Vector Autoregression (VAR). This framework, which belongs to the traditional toolkit of econometric forecasting techniques, has been shown to provide policy-makers with fairly good and consistent point and interval

[1] Delft University of Technology, p.altmeyer@tudelft.nl

[2] European Central Bank, marc.agusti _i _torres@ecb.europa.eu

[3] European Central Bank, ignacio.vidalquadrascosta@barcelonagse.eu

estimates. It has therefore been used extensively in the monetary policy divisions of central banks.

Simultaneously, with the recent advancements in computational power, and more importantly, the development of advanced machine learning algorithms and deep learning, interesting novel tools have become available that may be useful for forecasting time series. Whereas the good performance of techniques such as VAR is well established, it is still uncertain whether deep learning algorithms can be applied successfully to macroeconomic data.

To this end, this paper contributes a new and ground-breaking methodology that combines the VAR equation-by-equation structure with deep learning. We provide evidence that this improves the model's capacity to capture potentially highly non-linear relationships in the underlying data generating process. The primary objective of this paper is to develop a methodology that produces improved modelling outcomes while deviating as little as possible from the established VAR framework, thereby keeping things straight-forward and familiar to economists. We show that the existing VAR methodology can be easily extended to the broader class of Deep VAR models and provide solid empirical evidence that Deep VAR models consistently outperform the conventional approach.

To the best of our knowledge, this is the first paper to propose a Deep VAR framework of this structure, namely, to fit a deep neural network for each equation of the VAR process. Although previous work has explored the use of deep learning to forecast macroeconomic time series, previous proposed methodologies deviate more from the conventional VAR framework. For example, Verstyuk (2020) chooses to model the whole system through one unified deep neural network. We find that the equation-by-equation approach not only helps to maintain interpretability and simplicity, but also appears to produce better modelling outcomes. To enable researchers and practitioners to easily implement our proposed methodology, we have developed a unified framework for estimating Deep VAR models in R and plan to continue its development going forward.

We find that the Deep VAR methodology outperforms the traditional VAR framework in terms of in-sample and out-of-sample fit as well as with respect to forecasting accuracy. In particular, the Deep VAR appears to be better at capturing non-linear dynamics underlying the time series process. It therefore leads to consistently lower modelling errors than the VAR, especially during periods of economic downturn and uncertainty.

Arguably policy makers are not only interested in the forecasting accuracy of the model but are typically also concerned with inference. For example, central banks are often interested in knowing to what extent interest rates granger cause other variables within the monetary transmission mechanism. Another aspect policy makers and researchers care about is how the variables of the system evolve through time in response to innovations. This information is typically recovered using Impulse Response Functions (IRFs). The linear additive modelling assumption underlying the conventional VAR makes inference straight-forward. In the case of Deep VAR models inference is arguably more complicated, though promising avenues have recently been explored (Verstyuk 2020). We believe that the methodology proposed in this paper can be augmented to the inference realm in future work.

The remainder of the paper is structured as follows: in section 2 we present a literature review of prior research on the methodologies used to provide forecasts and on the monetary transmission mechanism in general. Section 3 provides a detailed description of the data we use for our empirical exercises. In section 4 we present the traditional VAR methodology and develop our proposed Deep VAR model. Sections 5 and 6 present our empirical findings and possible extensions and caveats, respectively. Finally, section 7 concludes.

2 Literature review

There is broad agreement among economists on the fact that monetary policy affects economic activity in the short and medium term. Friedman and Schwartz (2008) found that monetary policy actions are followed by movements in real output that may last for two years or more (Romer and Romer (1989); Bernanke (1990)). The underlying forces that trigger these outcomes is of great interest to most economists. Central bankers in particular aim to understand the monetary transmission mechanism. If monetary policy affects the real economy, then what exactly is the transmission mechanism through which these effects occur? This is one of the questions which is among the most important and controversial topics in modern-day macroeconomics.

In the aftermath of the oil price shock in the 1970's, interest emerged in understanding business cycles. To this end economists initially made use of large-scale macroeconomic models, which was criticized by Lucas (1976), stating that the assumption of invariant behavioural equations was inconsistent with dynamic maximizing behaviour. Hence, **New Classical** economists started to make use of so-called market clearing models of economic fluctuations. With the goal of really taking into account productivity shocks, **Real Business Cycle** models were developed (Kydland and Prescott (1982)).

Following the failure of large-scale macroeconomic models when trying to predict business cycles, the economic profession resorted to structural vector autoregressive (VAR) models to analyse business cycles, which proved to be useful for capturing the impact of policy actions. Sims et al. (1986) suggested that VAR models were an efficient tool to evaluate macroeconomic models. One of the advantages of VAR models is their simplicity, which makes it easy to estimate and interpret them.

Yet, this simplicity comes at a cost: conventional VAR models are typically not able to capture non-linear relationships in the data, which might be a significant limitation. In the very short run many time series can be expected to behave more or less according to their past and a linear model may be efficient to capture dynamics, but for longer term dependencies this is typically not the case. With respect to the economic time series that form part of the monetary transmission mechanism, specifically output, inflation, interest rates and labour market variables, non-linear dependencies are likely to form part of the data generating process as shown by Brock et al. (1991). This is true in particular during times of abrupt and significant economic fluctuations.

During past years, economists have therefore started to add non-linear techniques to their forecasting tool kit. Machine Learning has contributed a lot to this field of research. Some of the most popular machine learning techniques which do not assume a linear relationship between inputs and outputs include K-Nearest Neighbours (first introduced by Fix and Hodges (1951)), Support Vector Machines (mostly developed by Cortes and Vapnik (1995)), Random Forests (first introduced in

1995 by Ho (1995)) and Deep Artificial Neural Networks (first proposed in 1943 by McCulloch and Pitts (1990)). The latter have been explored previously in the realm of time series forecasting (Hamzaçebi (2008), G. P. Zhang (2003), Kihoro, Otieno, and Wafula (2004)). Neural networks are non-parametric models that have been shown to be particularly successful at capturing non-linearities (G. Zhang, Patuwo, and Hu (1998), G. P. Zhang (2003)).

A particular subclass of neural networks used primarily for sequential data are recurrent neural networks (RNN). RNNs propagate previous outputs recursively allowing the model to learn persistent dependencies and thereby making them very efficient for time series data (Dorffner 1996). A recent staff working paper published by the Bank of England provides some empirical support for the argument that deep learning can be successfully applied to macroeconomic data (Joseph et al. 2021). The authors run a horse race for forecasting inflation across different time horizons comparing the performance of linear and non-linear models. They find that neural networks in particular and other common machine learning algorithms are useful for forecasting particularly at a longer horizon.

3 Data

To evaluate our proposed methodology empirically we use a sample of monthly US data on leading economic indicators, which spans the period of January 1959 through March 2021. We use the relatively novel FRED-MD data base which is updated monthly and publicly available (McCracken and Ng 2016). The sample spans from March, 1959 to March, 2021 providing us with a relatively rich data set of macroeconomic time series with $T = 745$ observations.

In order to investigate the monetary transmission mechanism, the literature typically focuses on variables related to economic output, inflation, short and long term interest rates as well as labour market indicators. Some go beyond to also include stock price indices, money and credit aggregates, balance of payments figures, confidence indicators and some cases foreign domestic indicators. In this paper we limit our attention to the four main indicators mentioned above. In particular we use changes in the industrial production index (IP) to measure output growth, changes in the growth of the (all items) consumer price index (CPI) to measure inflation, the Federal Funds Rate (FFR) as our interest rate and the unemployment rate (UR) as our labour market indicator. Note that we use IP rather than the gross domestic product as a proxy for output, because the latter is only available at quarterly frequency.

Another strength of the FRED-MD is the fact that the data is already pre-processed. Specifically, the industrial production index comes in log differences, the CPI in second-order log differences and both the Fed Funds Rate and the unemployment rate in first-order differences. This has allowed us to let the data enter our estimations without any further adjustments, which should facilitate the reproducibility of our results.

4 Methodology

In conventional Vector Autoregression (VAR) dependencies of any system variable on past realizations of itself and its covariates are modelled through linear equations. This corresponds to a particular case of the broader class of Deep Vector Autoregressions investigated here and will serve as the baseline for our analysis.

4.1 Vector Autoregression

Let $\mathbf{y}_t$ denote the $(K \times 1)$ vector of variables at time t. Then the VAR(p) with p lags and a constant deterministic term is simply a linear system of stochastic equations of the following form:

$$\mathbf{y}_t \quad = \mathbf{c} + \mathbf{A}_1 \mathbf{y}_{t-1} + \mathbf{A}_2 \mathbf{y}_{t-2} + \ldots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{u}_t, \qquad \mathbf{u}_t \quad \sim \mathcal{N}(\mathbf{0}, \Sigma_u) \tag{4.1}$$

The matrices $\mathbf{A}_m \in \mathbb{R}^{K \times K}$, where $m \in \{1, \ldots, p\}$, contain the reduced form coefficients and $\mathbf{u}_t \in \mathbb{R}^{K \times 1}$ is a vector of errors for which $\mathbb{E}\mathbf{u}_t$, $\mathbb{E}\mathbf{u}_t\mathbf{u}_t^T = \Sigma$ and $\mathbb{E}\mathbf{u}_t\mathbf{u}_s^T = \mathbf{0}$ for all $t \neq s$. We refer to (4.1) as the **reduced form** representation of the VAR(p) because all right-hand side variables are predetermined (Kilian and Lütkepohl 2017).

We can restate (4.1) more compactly as

$$\mathbf{y}_t \quad = \mathbf{A}\mathbf{Z}_{t-1} + \mathbf{u}_t \tag{4.2}$$

where $\mathbf{A} = \left(\mathbf{c}, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_p\right) \in \mathbb{R}^{K \times (Kp+1)}$ and $\mathbf{Z}_{t-1} = \left(1, \mathbf{y}_{t-1}^T, \ldots, \mathbf{y}_{t-p}^T\right)^T \in \mathbb{R}^{(Kp+1) \times 1}$. The expression in (4.2) demonstrates that the VAR(p) can be considered as a **seemingly unrelated regression** (SUR) model composed of individual regressions with common regressors (Greene 2012). In fact, it is useful to note for our purposes that the VAR(p) can be estimated efficiently through equation-by-equation OLS regression. In particular, it follows from (4.2) that

$$y_{it} \quad = c_i + \sum_{m=1}^{p} \sum_{j=1}^{K} a_{jm}\, y_{jt-m} + u_{it} \qquad , \qquad \forall i = 1, \ldots, K \tag{4.3}$$

which corresponds to the key modelling assumption that at any point in time t any time series $i \in 1, \ldots, K$ is just a weighted sum of past realizations of itself and all other variables in the system. This assumption makes the estimation of VAR(p) processes remarkably simple. Perhaps more importantly, the assumption of linearity also greatly facilitates inference about VAR models.

For implementation purposes it is generally more useful to estimate the VAR(p) through one single OLS regression. To this end let $\widetilde{\mathbf{A}} = \mathbf{A}^T$ and note that (4.2) can be restated even more compactly as

$$\mathbf{y} \quad = \mathbf{Z}\widetilde{\mathbf{A}} + \mathbf{u}_t \tag{4.4}$$

with $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)^T \in \mathbb{R}^{T \times K}$ and $\mathbf{Z} \in \mathbb{R}^{T \times (Kp+1)}$. Then the closed form solution for OLS is simply $\widetilde{\mathbf{A}} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y}$ and hence

$$\mathbf{A} \quad = \mathbf{y}^T\mathbf{Z}(\mathbf{Z}\mathbf{Z}^T)^{-1} \tag{4.5}$$

4.2 Deep Vector Autoregression

We propose the term Deep Vector Autoregression to refer to the broad class of Vector Autoregressive models that use deep learning to model the dependences between system variables through time. In particular, as before, we let $\mathbf{y}_t$ denote the $(K \times 1)$ vector that describes the state of system at time t. Consistent with the conventional VAR structure we assume that each individual time series y_{it} can be modelled as a function of lagged realizations of all variables y_{jt-p}, $j = 1, \dots, K$, $m = 1, \dots, p$.

More specifically, we have

$$y_{it} \quad = f_i(\mathbf{y}_{t-1:t-p}; \theta) + v_{it} \qquad , \qquad \forall i = 1, \dots, K \qquad (4.6)$$

where $\mathbf{y}_{t-1:t-p} = \{y_{jt-m}\}_{j=1,\dots,K}^{m=1,\dots,p}$ is the vector of lagged realizations, f_i is a variable specific mapping from past lags to the present and θ is a vector of parameters. While in the conventional VAR above we assumed that the multivariate process can be modelled as a system of linear stochastic equations, our proposed Deep VAR(p) can similarly be understood as a system of potentially highly non-linear equations. As we argued earlier, Deep Learning has been shown to be remarkably successful at learning mappings of arbitrary functional forms (Goodfellow, Bengio, and Courville 2016).

Note that the input and output dimensions in (4.6) are exactly the same as in the conventional VAR(p) model (equation (4.3)): f_i maps from $\mathbf{y}_{t-1:t-p} \in \mathbb{R}^{Kp\times 1}$ to a scalar. Our proposed plain-vanilla approach to Deep VAR models diverges as little as possible from the conventional approach: it boils down to simply modelling each of the univariate outcomes in (4.6) as a deep neural network. We can restate this approach more compactly as

$$\mathbf{y}_t \quad = \mathbf{f}(\mathbf{y}_{t-1:t-p}; \theta) + \mathbf{v}_t \qquad (4.7)$$

where $\mathbf{f}(\cdot) = (f_1(\cdot), f_2(\cdot), \dots, f_K(\cdot))^T \in \mathbb{R}^{K\times 1}$ is just the stacked vector of mappings to univariate outcomes described in (4.6).

The notation in (4.7) gives rise to a more unified and general approach to Deep VAR models that would treat the whole process as one single dynamical system to be modelled through one deep neural network $\mathbf{g}$:

$$\mathbf{y}_t \quad = \mathbf{g}(\mathbf{y}_{t-1:t-p}; \theta) + \mathbf{v}_t \qquad (4.8)$$

This approach is in fact proposed and investigated by Verstyuk (2020) in his upcoming publication. We decided to go with the approach in (4.7) for two reasons: firstly, the link to conventional VAR models is made abundantly clear through this implementation and, secondly, we found that the equation-by-equation approach produces good modelling outcomes and is relatively easy to implement using state-of-the art software.

Finally, note that if f_i in (4.3) is assumed to be linear and additive for all $i = 1, \dots, K$ then we are back to the conventional VAR(p). This illustrates the point we made earlier that the linear VAR(p) is just a particular case of a Deep VAR(p). Since the model described in equations (4.6) and (4.7) is less restrictive but otherwise consistent with the conventional VAR framework, we expect that it outperforms the traditional approach towards modelling multivariate time series processes.

4.3 Deep Neural Networks - a whistle-stop tour

So far we have been speaking about deep learning in rather general terms. For example, above we have referred to our model of choice for learning the mapping $f_i: \mathbf{y}_{t-1:t-p} \mapsto y_{it}$ as a **deep neural network**. The class of deep neural networks can further be roughly divided into **feedforward neural networks** and **recurrent neural networks**. As the term suggests, the latter is generally used for sequential data and therefore our preferred model of choice. Nonetheless, below we will begin by briefly exploring feedforward neural networks first. This should serve as a good introduction to neural networks more generally and (even though we have not tested this empirically) there is good reason to believe that even Deep VAR models using feedforward neural networks perform well.

4.3.1 Deep Feedforward Neural Networks

The term **deep feedforward neural network** or **multilayer perceptron** (MLP) is used to describe a broad class of models that are composed of possibly many functions that together make up the directed acyclical graph. The functions $f_i(\cdot)$ - sometimes referred as layers $\mathbf{h}_i$ - are chained together hierarchically with the first layer feeding forward its outputs to the second layer and so on (Goodfellow, Bengio, and Courville 2016). Applied to our case, an MLP with H hidden layers can be loosely defined as follows:

$$f_i(\mathbf{y}_{t-1:t-p};\theta) \quad = f_i^{(H)}\left(f_i^{(H-1)}\left(\dots f_i^{(1)}(\mathbf{y}_{t-1:t-p})\right)\right) \qquad (4.9)$$

The depth of the MLP is defined by the number of hidden layers H, where, generally speaking, deeper networks are more complex.

The desired outputs of any $f_i^{(h)}$ that will serve as inputs for $f_i^{(h+1)}$ cannot be inferred from the training data $\mathbf{y}_{t-1:t-p}$ ex-ante, which is where the term **hidden** layer stems from. Each $f_i^{(h)}$ is typically valued on a vector of hidden units, each of them receiving a vector of inputs from $f_i^{(h-1)}$ and returning a scalar that is referred to as activation value. This approach is inspired by neuroscience, hence the term **neural** network (Goodfellow, Bengio, and Courville 2016).

4.3.2 Deep Recurrent Neural Networks

Recurrent neural networks (RNN) are based on the idea of persistent learning: a continuous process that evolves gradually and at each step uses information about its prior states instead of continuously reinventing itself and starting from scratch. To this end, RNNs develop the basic concepts underlying feedforward neural networks by incorporating feedback loops. Formally the loop is typically made explicit as follows

$$\mathbf{h}_t \quad = f(\mathbf{h}_{t-1}, \mathbf{x}_t; \theta) \qquad (4.10)$$

where $\mathbf{h}_t \in \mathbb{R}^{N\times 1}$ corresponds to the hidden state of the dynamical system at time t that the RNN learns (Goodfellow, Bengio, and Courville 2016), and N corresponds to the number of hidden units in each hidden layer, known as the width of the layer. In the given context we have that $\mathbf{x}_t = \mathbf{y}_{t-1:t-p}$ as specified in (4.7). Given some random initial hidden state vector $\mathbf{h}_0$ the RNN updates parameters sequentially at each time step t as follows

$$\begin{aligned} \mathbf{a}_t &= \mathbf{b} + \mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{h}_{-1} \\ \mathbf{h}_t &= \tanh(\mathbf{a}_t) \\ \hat{y}_{it} &= c + \mathbf{v}^T\mathbf{h}_t \qquad (4.11) \end{aligned}$$

where $\mathbf{b} \in \mathbb{R}^{N\times 1}$ is a vector of constants (biases), $c \in \mathbb{R}$ is a scalar that captures the deterministic term of the VAR, tanh is the hyperbolic tangent activation function, $\mathbf{W}, \mathbf{U} \in \mathbb{R}^{N\times N}$ are coefficient matrices and $\mathbf{v} \in \mathbb{R}^{N\times 1}$ is a vector of coefficients. Note that to simplify the notation we have omitted the layer index in (4.11): to be specific, $\mathbf{h}_t$ really represents $\mathbf{h}_t^{(H)}$ (the ultimate hidden layer), $\mathbf{h}_{-1}$ stands for $\mathbf{h}_t^{(H-1)}$ (the penultimate layer). Finally, at each step t the first layer $\mathbf{h}_t^{(0)}$ of the forward propagation corresponds to $\mathbf{y}_{t-1:t-p}$.

A shortfall of generic recurrent neural networks is that they fail to capture long-term dependencies. More specifically, if parameters are propagated over too many stages in a simple RNN, it typically suffers from the problem of **vanishing gradients** (Goodfellow, Bengio, and Courville 2016). Fortunately, there exist effective extensions of the RNN, most notably the long short-term memory (LSTM), which was introduced by Hochreiter and Schmidhuber (1997) and is our model of choice for Deep VAR models. The key idea underlying LSTMs is to regulate exactly how much information is propagated from one cell state vector $\mathbf{C}_{t-1}$ to the next $\mathbf{C}_t$ through the introduction of so called sigmoid gates:

> "The LSTM [has] the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through." — Olah (2015)

These regulating gate layers include a **forget gate** $\mathbf{f}_t$, an **input gate** $\mathbf{i}_t$ and a **output gate** $\mathbf{o}_t$. Each of them are vector-values sigmoid functions whose elements $\mathbf{f}_{it}, \mathbf{i}_{it}, \mathbf{o}_{it}$ are bound between 0 and 1. Their individual purposes are implied by their names: faced with $\mathbf{h}_{t-1}$ and $\mathbf{y}_{t-1:t-p}$, the forget gate regulates how much of each individual unit in $\mathbf{C}_{t-1}$ is retained. Then the input gate regulates which units of $\mathbf{C}_{t-1}$ should be updated and to what candidate values $\tilde{\mathbf{C}}_{t-1}$. Using the previous two steps the actual update is performed according to the following rule

$$\mathbf{C}_t \quad = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_{t-1} \qquad (4.12)$$

where $\odot$ indicates the element-wise product. Finally, the output gate acts like a filter on $\mathbf{C}_t$: the new hidden state is computed as $\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t)$ where as before we use the hyperbolic tangent as our activation function.[4] Formally, we can summarize the LSTM neural network underlying our Deep VAR framework as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma\big(\mathbf{b}_f + \mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{U}_f\mathbf{h}_{-1}\big) \\
\mathbf{i}_t &= \sigma(\mathbf{b}_i + \mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{U}_i\mathbf{h}_{-1}) \\
\mathbf{o}_t &= \sigma(\mathbf{b}_o + \mathbf{W}_o\mathbf{h}_{t-1} + \mathbf{U}_o\mathbf{h}_{-1}) \\
\mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{b}_C + \mathbf{W}_C\mathbf{h}_{t-1} + \mathbf{U}_C\mathbf{h}_{-1}) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \\
\hat{y}_{it} &= c + \mathbf{v}^T\mathbf{h}_t \qquad (4.13)
\end{aligned}$$

which is best understood when read from top to bottom. Once again we have simplified the notation by omitting the layer index in (4.11). The same notation as before applies.

4.4 Model selection

There are at least two important modelling choices to be made in the context of conventional VAR models. The first choice concerns properties of the time series data itself, in particular the order of integration and cointegration. The second choice is about the the lag order p. In order to arrive at appropriate decisions regarding these

[4] For a clear and detailed exposition see Olah (2015).

choices the VAR literature provides a set of guiding principles. We propose to apply these same principles to the Deep VAR, firstly because they are intuitive and simple and secondly because treating both models equally to begin with allows for a better comparison of the two models at the subsequent modelling stages.

4.4.1 Stationarity

When working with time series we are generally concerned about stationarity. Broadly speaking stationarity ensures that the future is like the past and hence any predictions we make based on past data adequately describe future outcomes. In order to state stationarity conditions in the VAR context it is convenient to restate the K-dimensional VAR(p) process in companion form as

$$\mathbf{Y}_t = \begin{pmatrix} \mathbf{c} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \mathbf{A}\mathbf{Y}_{t-1} + \begin{pmatrix} \mathbf{u}_t \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad (4.14)$$

where $\mathbf{Y}_t = \left(\mathbf{y}_t^T, \dots, \mathbf{y}_{t-p+1}^T\right)^T \in \mathbb{R}^{Kp\times 1}$ and $\mathbf{A} \in \mathbb{R}^{Kp\times Kp}$ is referred to as the companion matrix (Kilian and Lütkepohl 2017). Stationarity of the VAR(p) follows from stability: a VAR(p) is stable if the effects of shocks to the system eventually die out. Stability can be assessed through the system's autoregressive roots or equivalently by looking at the eigenvalues of the companion matrix $\mathbf{A}$ (Kilian and Lütkepohl 2017). In particular, for the VAR(p) in (4.14) to be stable we condition that the Kp eigenvalues λ that satisfy

$$\det\left(\mathbf{A} - \lambda \mathbf{I}_{Kp}\right) = 0$$

are all of absolute value less than one. Stability implies that the first and second moments of the VAR(p) process are time-invariant, hence ensuring weak stationarity (Kilian and Lütkepohl 2017).

A straight-forward way to deal with stationarity of VAR models is to simply ensure that the individual time series entering the system are stationary. This usually involves differencing the time series until they are stationary: for any time series y_i that is integrated of order $I(\delta)$, there exists a δ-order difference that is stationary. An immediate drawback of this approach is the loss of information contained in the levels of the time series. Modelling approaches that take into account conintegration of individual time series can ensure system stationarity and still let individually non-stationary time series enter the system in levels (Hamilton 2020).

4.4.2 Lag order

The VAR's lag order p can to some extent be thought of as the persistency of the process: past innovations that still affect outcomes in time t happened at most p periods ago. From a pure model selection perspective we can also think of additional lags in terms of additional regressors that add to the model's complexity. From that perspective, choosing a lower lag order corresponds to a form of regularization as it pertains to a more parsimonious model.

Various strategies have been proposed to estimate the true or optimal lag order p empirically (Kilian and Lütkepohl 2017). Among the most common ones are sequential testing procedures and selection based on information criteria. The former involves sequentially adding or removing lags - **bottom-up** and **top-down** testing, respectively - and then testing model outcomes in each iteration. A common point of criticism of sequential procedures is that the order tests matters (Lütkepohl (2005)).

Here we will focus on selection based on information criteria, which to some extent makes the trade-off between bias an variance explicit (Kilian and Lütkepohl 2017). In particular, it generally involves minimizing information criteria of the following form

$$C(m) \quad = \log\left(\det\left(\hat{\Sigma}(m)\right)\right) + \ell(m) \qquad (4.15)$$

where $\hat{\Sigma}$ is just the sample estimate of the covariance matrix or errors and ℓ is a loss function that penalizes high lag orders. In particular, we have that our best estimate of the optimal lag order p is simply

$$\hat{p} \quad = \underset{m \in \mathcal{P}}{\text{argmin}} C(m) \qquad (4.16)$$

where $\mathcal{P} = [m_{\text{min}}, m_{\text{max}}]$. We will consider all of the most common functional choices for (4.15).

4.4.3 Neural Network Architecture

By now it should be clear that deep neural networks come in many shapes and sizes. When thinking about the architecture of a neural network many different design choices can be made and networks can thus be tailored to specific use cases. Here, we intend to keep things simple and vary only the depth and width of the LSTMs underlying the Deep VAR. The number of hidden units per hidden layer is held constant across layers.

Figure 4.1 illustrates a simulated network architecture for the case of two lags ($p = 2$) and four variables ($K = 4$). We can see that the first layer corresponds to the inputs, that is, the input layer $\in \mathbb{R}^{Kp \times 1}$. This architecture consists of $H = 2$ hidden layers each counting twenty hidden units. Since we are modelling equation-by-equation, there is only one output unit, namely variable y_{it}.

With respect to network compilation, the popular Adam optimization algorithm is used (Kingma and Ba 2014). This algorithm can be used instead of the more traditional stochastic gradient descent to update network weights. There are several reasons to use this algorithm that are particularly appealing, among them its straightforward implementation and its computationally efficiency. Adam distinguishes itself from classic stochastic gradient descent in that it uses adaptive learning rates.

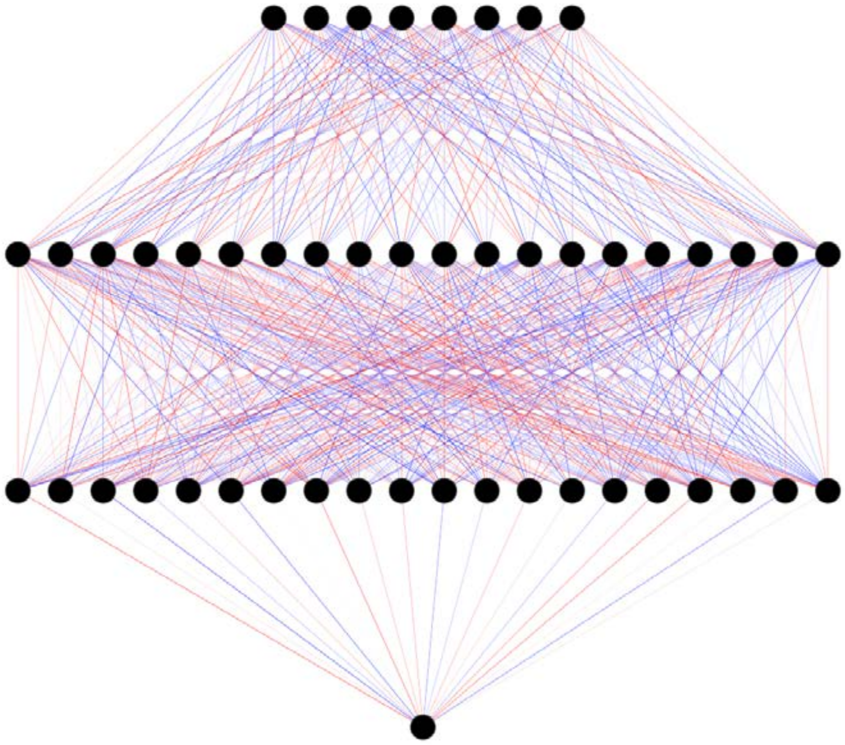


Figure 4.1: Neural Network Architecture.

As mentioned above, the estimation of deep neural networks involves a very large number of parameters and hence regularization is an important concern. One way to mitigate the risk of overfitting is to choose a neural network architecture that is neither excessively wide nor deep. Another way to regularize the neural network is to use the fact that optimization at the training phase is stochastic. One greedy way to reduce overfitting risk is therefore to simply retrain the network multiple times and then average over the obtained parameter estimates and predictions (Srivastava et al. 2014). While theoretically appealing, this approach is computationally prohibitive. Instead, another layer of stochasticity can be introduced at the training stage through **dropout**: at each training iteration and each stage of the forward propagation a share of the hidden units is simply dropped at random. This approach mimics the idea of repeated training. Dropout adds noise into the model and thereby avoids that hidden layers try to adapt to a mistake made by previous hidden layers.

5 Empirical results

We now proceed to benchmark the proposed Deep VAR model against the conventional VAR and other existing approaches using our macroeconomic time series data. To begin with, we compare the models in terms of their in-sample fit. For this part of the analysis the models will be strictly run under the same framing conditions. Due to the RNN's capacity to essentially model any possible function $f_i(\cdot)$ the Deep VAR dominates competing approaches in this realm. We investigate during what time periods the out-performance of the Deep VAR is particularly striking to gain a better understanding of when and why it pays off to relax the linearity constraint.

These findings with respect to in-sample performance provide some initial evidence in favour of the Deep VAR. But since a reduction in modelling bias is typically associated with an increase in variance, we are particularly interested in benchmarking the models with respect to their out-of-sample performance. To this end we split our sample into train and test subsamples. We then firstly benchmark the models in terms of their pseudo out-of-sample fit. Finally we also look at model performance with respect to n-step ahead pseudo out-of-sample forecasts.

The final part of this section relaxes the constraint on the framing conditions. In particular, we investigate how hyperparameter tuning with respect to the neural network architecture and lag length p can improve the performance of the Deep VAR.

5.1 In-sample fit

For this first empirical exercise all models are trained on the full sample. We have decided to include the post-Covid sample period despite the associated structural break, since it serves as interesting point of comparison. The optimal lag order as determined by the Akaike Information Criterium is $p = 6$, where we used a maximum possible lag of $p_{\text{max}} = 12$ corresponding to one year. A look at the eigenvalues of the companion matrix showed that the VAR(6) is stable. The LSTM models underlying the Deep VAR are composed of $H = 2$ that count $N = 32$ hidden units each. The dropout rate is set to $p = 0.25$.

To assess the fit of our models we use squared residuals. Figure 5.1 shows the cumulative loss of the Deep VAR model and its conventional benchmarks for each of the time series over the whole sample period. Aside from the linear VAR, we have added another popular approach towards VAR models that addresses non-linearity (Threshold VAR). We have also added a Random Forest Regressor (RF) for comparison, which was trained on the entire FRED-MD database, so far more variables than the four output variables. Previous studies have shown that RF tends to well at high-dimensional time series modelling (Masini, Medeiros, and Mendes 2021).

The first thing we can observe is that the RMSE of the Deep VAR is consistently flatter than the RMSE of its benchmarks. With respect to model fit, the Deep VAR dominates throughout the almost the entire sample period and for all of the considered variables. This empirical observation seems to confirm our expectation that the vector autoregressive process is characterized by important non-linear dependencies across time and variables that the conventional VAR and even the TVAR and RF fail to capture.

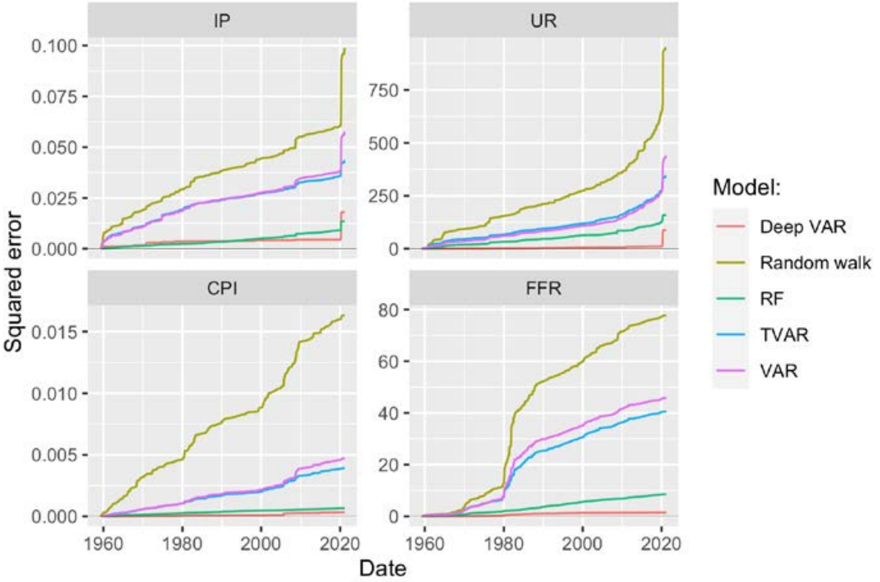


Figure 5.1: Comparison of cumulative loss over the entire sample period for Deep VAR and benchmarks.

Figure 5.1 is especially useful to asses in which specific periods the Deep VAR model fits the data better than alternative approaches. From the very beginning and across variables, we observe that the increase in cumulative loss for other models is greater than for the Deep VAR model.

The US economy during 1960s was influenced by John F. Kennedy's introduction of **New Economics**, which was informed by Keynesian ideas and characterized by increasing levels of inflation, a reduction in unemployment and output growth. The change in government certainly corresponded to a regime switch with respect to the economy (Perry and Tobin 2010) and in that sense it is interesting to observe that the Deep VAR appears to be doing a better job at capturing the underlying changes. The 1970s can be broadly thought of as a continuation of New Economics and loosely defined as a period of stagflation. The Deep VAR continues to outperform during that period.

The first truly interesting development we can observe in Figure 5.1 coincides with the onset of the Volcker disinflation period. Following years of sustained CPI growth, Paul Volcker set the Federal Reserve on course for a series of interest rate hikes as soon as he became chairperson of the central bank in August 1979. The shift in monetary policy triggered fundamental changes to the US economy and in particular the key economic indicators we are analysing here throughout the 1980s (Goodfriend and King 2005). Despite this structural break, the increase in the cumulative RMSE of the Deep VAR remains almost constant during this decade for most variables. The performance of the VAR on the other hand is unsurprisingly poor over the same period, in particularly so for the CPI and the Fed Funds Rate, which arguably were the two variables most directly affected by the change in policy. The Deep VAR also clearly dominates the VAR with respect to the output related variables (IP) and to a lesser extent unemployment.

These findings indicate that changes to the monetary transmission mechanism in response to sudden policy shifts are not well captured by a linear-additive vector autoregressive model. Instead they appear to unfold in a high-dimensional latent state space, which the Deep VAR by its very construction is designed to learn.

Following the Volcker disinflation period, Figure 5.1 does not reveal any clear outperformance of either of the models during the 1990s. Interestingly the dot-com bubble has little effect on either of the models, aside from a small pick-up in cumulative loss with respect to the CPI for both models. With all that noted, the Deep VAR still continuously outperforms the VAR since evidently its cumulative loss increases at a slower pace altogether.

As the Global Financial Crisis unfolds around 2007 the pattern we observed for the Volcker disinflation remerges, albeit to a lesser extent: there is a marked jump in the difference between the cumulative loss of the VAR and the Deep VAR, in particular so for the CPI, the Fed Funds rate and industrial production. The gap for all these variables continues to widen during the aftermath of the crisis. The Deep VAR once again does a better job at modelling the changes that the dynamical system undergoes: post-crisis US monetary policy was characterized by very low interest rates, low levels of inflation as well as the introduction of a range of non-conventional monetary policy tools including quantitative easing and forward guidance.

Finally, it is also interesting to observe how the different models perform in response to the unprecedented exogenous shock that Covid-19 constitutes. All models exhibit an abrupt and substantial increase in loss with respect to both IP and UR - the two series that were arguably most strongly affected by Covid. Evidently, the magnitude of that sudden increase is somewhat larger in absolute terms for VAR than for Deep VAR. Still, it is also worth pointing out that both the Threshold VAR and the Random Forest Regressor are less adversely affected by the Covid shock than Deep VAR.

As a sanity check we also visually inspected the distributional properties of the model residuals for the full-sample fit. The outcomes are broadly consistent across models: while for some variables residuals are clearly not Gaussian, we see no evidence of serial autocorrelation of residuals (see Figures 9.2 and 9.3 in the appendix).

5.2 Out-of-sample fit

In order to assess if the Deep VAR's outperformance is a consequence of overfitting, we now repeat the previous exercise, but this time we train the models on a subsample of our data. The training sample spans from March, 1959 to October, 2008, whereas the test data (including validation period) goes from November, 2008 to March, 2021. This corresponds to training the model on 80 percent of the data and retaining the remaining 20 percent for testing purposes. The optimal lag order for the training subsample is $p = 7$ where we use the same criterion and maximum lag order as before. Once again we find this VAR specification to be stable.

Tables 5.1 shows the Root Mean Squared Error (RMSE) for the in-sample and the out-of-sample predictions of both the VAR model and the Deep VAR model. We can see that the RMSE for the Deep VAR is lower than for the conventional VAR for both the training data and the test data and for all time series. The fifth column of the table shows us the ratio between the RMSEs of the Deep VAR and the VAR: the lower the ratio, the better the Deep VAR compared to the VAR. With respect to the training sample, the RMSE of the Deep VAR model is consistently less than 75% of that of the conventional VAR reflecting to some extent the results of the previous sections. Turning to the test data, there is no evidence that the Deep VAR is more prone to overfitting than the VAR. For both industrial production and unemployment, the Deep VAR yields an RMSE that is around half the size of that produced by the VAR. For inflation and interest rate predictions the out-performance on the test data is less striking, but still large.

Table 5.1: Root mean squared error (RMSE) for the two models across subsamples and variables.

Sample	Variable	DVAR	VAR	Ratio (DVAR / VAR)
test	IP	0.00485	0.01484	0.32703
test	UR	0.90300	1.65170	0.54671
test	CPI	0.00225	0.00342	0.65892
test	FFR	0.15743	0.23974	0.65665
train	IP	0.00267	0.00727	0.36737
train	UR	0.03701	0.43322	0.08543
train	CPI	0.00035	0.00232	0.14925
train	FFR	0.03658	0.25780	0.14191

5.3 Forecasts

Up until now we have been assessing the model fit, which has provided some initial evidence in favour of Deep VAR. Typically though in the time series context we are more interested in out-of-sample forecasts. which we shall turn to next.

We begin with a single forecasting exercise, where forecasts are produced recursively both for the VAR and the Deep VAR. Specifically, we use the models we trained on the training data to recursively predict one time period ahead, concatenate the predictions to the training data and repeat the process. Note that for the Deep VAR an alternative approach is to work with a different output dimension for the underlying neural networks.[5]

We produce one-year ahead forecasts beginning from the first date in the test sample (November, 2008). Table 5.2 shows the resulting root mean squared forecast errors (RMSFE) along with correlation between forecasts and realizations. As we can see in the table, the RMSFE of the Deep VAR is consistently lower than the one for the VAR. Regarding correlations the VAR produces forecasts that are negatively correlated with actual outcomes for all time series: in other words, when the time series evolves in one direction, the VAR forecast tends to evolve in the opposite direction. For industrial production, the Deep VAR forecast also has a highly negative correlation with the actual values. For the rest of time series the Deep VAR forecasts correlate positively with actual outcome, albeit weakly. Another general observation we made with respect to these forecasts is that the forecasts from the conventional VAR are fairly volatile, while the Deep VAR forecasts swiftly reverts to steady levels (see Figures 9.8 and 9.9 in the appendix).

Table 5.2: Comparison of n-step ahead pseudo out-of-sample forecasts.

Variable	VAR	Deep VAR	VAR	Deep-VAR
	RMSFE	RMSFE	correlations	correlations
IP	0.01870	0.01673	-0.30409	-0.09175
UR	0.85984	0.73402	-0.10093	0.31968

[5] In future work we plan to assess this approach further.

Variable	VAR RMSFE	Deep VAR RMSFE	VAR correlations	Deep-VAR correlations
CPI	0.00946	0.00710	-0.33567	0.03954
FFR	0.52321	0.39851	-0.55935	-0.01335

Finally, we repeat the forecasting exercise above using a rolling window approach: we train our models on a window of 240 months, compute and store 12-month ahead forecasts out of the training sample, roll the window one period forward and repeat the previous steps. This allows us to benchmark the different models in terms of their forecasting performance over the entire sample period. Once again forecasts are for now computed recursively: in other words, neural networks underlying the Deep VAR are not explicitly trained to forecast 12-steps ahead.

In Figure 5.2 we have plotted the cumulative loss incurred by each model: the different output variables are faceted across columns; each row corresponds to a different forecast horizon. For example, the panel in row 2 of column 3 shows the cumulative mean squared error incurred by each model for forecasts up to the 3-month horizon.

While the results are less striking than what we observed above for the in-sample fit, the Deep VAR nonetheless dominates its conventional benchmark overall. For both inflation (CPI) and interest rates (FFR), the Deep VAR forecasts incur substantially lower loss over the entire sample period and in particular at short horizons. We also see somewhat better forecasts overall for industrial production, while for the unemployment rate the Deep VAR is at par with its conventional benchmark. It is not altogether surprising that losses converge at longer horizons since we would expect that forecasts from both autoregressive models converge to their unconditional expectations.

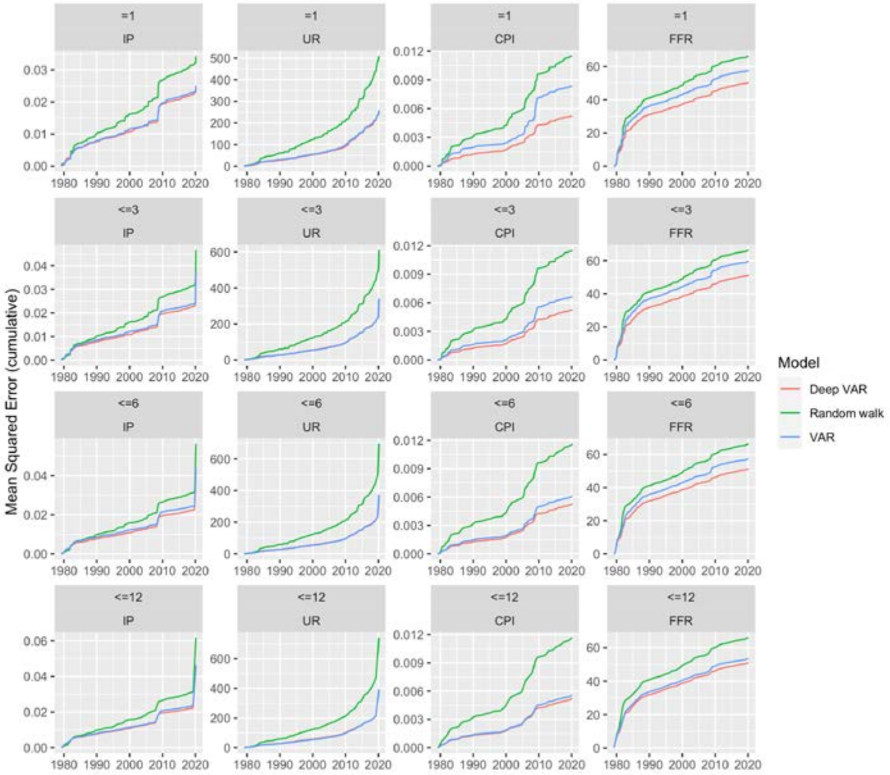


Figure 5.2: Comparison of cumulative rolling-window forecasting error over the entire sample period for Deep VAR and benchmarks. Forecasts are computed recursively.

5.4 Varying hyperparameters

While up until now with respect to model selection we have intentionally remained strictly within the conventional VAR framework, we will now relax that constraint and vary the lag length as well as hyperparameters of the Deep VAR. In particular, we perform a grid search where we vary the number of hidden layers $(1,2,5)$, number of hidden units per layer $(50,100,150)$, the dropout rate $(0.3,0.5,0.7)$ and the lag order $(10,\ 50,\ 100)$. For each combination of parameter choices we train the two models and compute the various performance measures introduced above.[6] Our expectation is that the conventional VAR is prone to overfitting and will produce poor out-of-sample outcomes for higher lag orders. For the Deep VAR we expect to interesting variation in the outcomes for different lag order and hyperparameter choices. It is not clear ex-ante that the Deep VAR should suffer from the same issue of overfitting for higher lag orders. The bulk of the corresponding visualizations can be found in the appendix.

5.4.1 Tuning the Deep VAR

To begin with, we shall forget about benchmarking for a moment and focus on the outcomes for the Deep VAR as we vary parameters. Recall that a higher number of hidden layers (depth), a higher number of hidden units (width) and a smaller choice for the dropout rate all correspond to an increase in neural network complexity. Consistent with this intuition we find that the in-sample loss for the Deep VAR improve as complexity increases (Figure 9.10): higher complexity leads to a reduction in bias and as we noted earlier the underlying recurrent neural networks should in principle be able to model arbitrary functions (Goodfellow, Bengio, and Courville 2016). Conversely, we observe exactly the opposite pattern for out-of-sample loss: as evident from Figure 9.11 a higher choice for the dropout rate and lower choices for the depth and width of the neural networks generally yields a smaller out-of-sample RMSE across variables.

Interestingly, both in- and out-of-sample loss tend to decrease significantly as the number of lags increases. In other words, the Deep VAR seems to be relatively insensitive to overfitting with respect to the lag order. With that in mind, we find that using standard lag order selection tools such as the AIC above may in fact not be appropriate for Deep VARs.

Finally, Figure 9.12 provides an overview of how pseudo out-of-sample forecasting errors behave as we vary the hyperparameters. As before we produce one-year ahead forecasts starting from the end of the 80% training sample. In this context, the pattern is less clear and varies across variables. As the lag order increases, for example, the forecast performance for the unemployment rate deteriorates. For inflation, forecasts are poor for the medium lag choice of $p = 50$ and much better for the low and high lag orders. The exact opposite relationship appears to hold for the Fed Funds Rate. With respect to the choices for the Deep VAR hyperparameters it is difficult to establish any clear pattern at all. The magnitude of differences in RMSFE is

[6] Of course, with respect to the conventional VAR only the lag order affects outcomes.

generally very small, so overall we conclude that to some extent the variation we do observe may be random.

In light of this evidence, we propose that for the purpose of hyperparameter tuning Deep VAR researchers should focus on the RMSE associated with the 1-step ahead fitted values. For the underlying data, a reasonable set of hyperparameter choices could be: 1 hidden layer, 50 hidden units and a dropout rate of 0.5.

5.4.2 Benchmark

Using the hyperparameter choices proposed above we now turn back to comparing the performance of the Deep VAR to the conventional VAR. Figure 5.3 shows the pseudo out-of-sample RMSE and RMSFE for both models across the different lag choices. For the sake of completeness we also include the performance measures we obtained when we initially ran both models in section 5.2 using the optimal lag order as determined by the AIC.

The first observation is that the Deep VAR outperforms the VAR across the board, reflecting our earlier findings. As expected, the VAR is subject to overfitting for when high lag order are chosen. This trend is observed both for the RMSE as well as the RMSFE. The fact that n-step ahead forecasts of the VAR are also subject to overfitting with respect to the lag order, while the Deep VAR appears unaffected, to some extent may reflect what we observed earlier: for the given data, Deep VAR forecasts swiftly converge to steady levels, while VAR forecasts are volatile, which may explain the relative outperformance of the Deep VAR. It appears that this effect is amplified for higher lag orders.

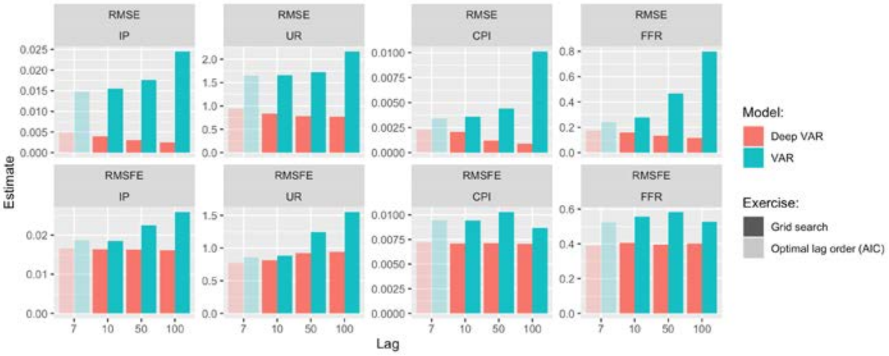


Figure 5.3: Pseudo out-of-sample RMSE and RMSFE for both models across the different lag choices. For the sake of completeness, we also include the performance measures we obtained when we initially ran both models using the optimal lag order as determined by the AIC.

To conclude this empirical section, we summarize our main findings:

1. We provide evidence that the conventional, linear VAR fails to capture important non-linear dependencies across time and variables that are typically used to model the monetary transmission mechanism.
2. Tapping into the broader class of Deep VAR leads to consistently better model performance.
3. Deep VAR appears to be relatively insensitive to very high lag orders at which conventional VAR models are prone to overfitting.

6 Caveats and extensions

In this work we have provided empirical evidence that the introduction of deep learning can lead to improved modelling and forecasting performance in the context of macroeconomic time series data. While we believe that our proposed methodology extends the conventional VAR framework quite naturally, it still comes with a lot of added complexity. Unfortunately, in the case of deep learning this added complexity also entails reduced interpretability: even though we have intentionally worked with a relatively small and simple neural network architecture, the number of parameters and interactions between neurons that they govern cannot possibly be interpreted by a human. This is why deep artificial neural networks are commonly referred to as black boxes.

Perhaps more importantly in the context of time series forecasting, it is also much harder to quantify predictive uncertainty of deep neural networks: while confidence intervals around point forecasts from a linear VAR can be computed using closed-form analytical expressions (Kilian and Lütkepohl 2017), no such expressions exist in the context of Deep VAR. Future work on this issue will most likely rely on probabilistic deep learning, which has gained popularity in recent years. Among the most widely used approaches to uncertainty quantification for deep learning are deep ensembles (Lakshminarayanan, Pritzel, and Blundell 2016) and Monte Carlo dropout (Gal and Ghahramani 2016). The former boils down to training not just one but multiple networks and effectively averaging over predictions: since weights are initialized randomly, predictions are stochastic. The latter similarly introduces stochasticity by activating dropout not only during training but also at the testing stage. A common drawback of these and other approaches that rely on Monte Carlo is the increased computational burden. As an alternative to Monte Carlo Daxberger et al. (2021) have recently shown that Laplace approximation can be used for effortless Bayesian deep learning.

Support for the estimation of impulse response functions is another missing cornerstone in the current version of our proposed framework. IRFs are used to understand how system variables change in response to unit shocks to any of the system variables. When estimating the model with the traditional VAR, IRFs can be readily derived from the reduced form model coefficients. Generalized (or structural) IRFs require the system to be fully identified, which is typically achieved through restrictions on contemporaneous (and likely correlated) reduced-form errors. In the context of Deep VAR further research is required concerning both computation of IRFs and the identification problem. Verstyuk (2020) computes impulse response functions for their proposed MLSTM numerically and relies on a Cholesky decomposition of the reduced-form covariance matrix, just like in the conventional setting. A more desirable approach may once again involve probabilistic deep learning: Ish-Horowicz et al. (2019) proposes a straight-forward approach towards producing global feature importance measures for input features of Bayesian neural networks. It might be possible to leverage these importance measures as proxies for the conventional VAR's linear coefficients and produce approximate impulse response functions for Deep VAR models in the same way as for conventional VAR models. Of course, these are merely rough ideas for future research.

7 Conclusions

Our initial motivation for this study was to see if by incorporating some of the latest developments from the machine learning and deep learning domains in the conventional VAR framework, we could attain improvements in the modelling and forecasting performance. In an effort not to deviate too much from the established framework, we only relax one single assumption to move from the conventional linear VAR to a broader class of models that we refer to as Deep VAR models.

To assess the modelling performance of Deep VAR models compared to linear VAR models we investigate a sample of monthly US economic data in the period 1959-2021. In particular, we look at variables typically analysed in the context of the monetary transmission mechanism including output, inflation, interest rates and unemployment. Our empirical findings show a consistent and significant improvement in modelling performance associated with Deep VAR models. In particular, our proposed Deep VAR produces much lower cumulative loss measures than the VAR over the entire period and for all of the analysed time series. The improvements in modelling performance are particularly striking during subsample periods of economic downturn and uncertainty. This appears to confirm or initial hypothesis that by modelling time series through Deep VAR models it is possible to capture complex, non-linear dependencies that seem to characterize periods of structural economic change.

When it comes to the out-of-sample performance, a priori it may seem that the Deep VAR is prone to overfitting, since it is much less parsimonious than the conventional VAR. On the contrary, we find that by using default hyperparameters the Deep VAR clearly dominates the conventional VAR in terms of out-of-sample prediction and forecast errors. An exercise in hyperparameter tuning shows that its out-of-sample performance can be further improved by appropriate regularization through adequate dropout rates and appropriate choices for the width and depth of the neural. Interestingly, we also find that the Deep VAR actually benefits from very high lag order choices at which the conventional VAR is prone to overfitting. In summary, we provide solid evidence that the introduction of deep learning into the VAR framework can be expected to lead to a significant boost in overall modelling performance. With respect to the main question posed at the beginning of this work, we therefore conclude that deep learning may be leveraged effectively in the context of macroeconomic time series modelling and vector autoregression.

We also point out several shortcomings of our proposed Deep VAR framework, which we believe can be alleviated through future research. In particular, policy-makers are typically concerned with uncertainty quantification, inference and overall model interpretability. Future research on Deep VAR models should therefore address the estimation of confidence intervals, impulse response functions as well as variance decompositions typically analysed in the context of VAR models. We point to a few possible avenues that involve probabilistic deep learning. We very much recognize the need for model interpretability especially in the context of policy-making and believe that the Deep VAR framework proposed here can be augmented to meet these demands.

References

Bernanke, Ben S. 1990. "The Federal Funds Rate and the Channels of Monetary Transnission." National Bureau of Economic Research Cambridge, Mass., USA.

Brock, William Allen, William A Brock, David Arthur Hsieh, Blake Dean LeBaron, and William E Brock. 1991. *Nonlinear Dynamics, Chaos, and Instability: Statistical Theory and Economic Evidence*. MIT press.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–97.

Daxberger, Erik, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. 2021. "Laplace Redux-Effortless Bayesian Deep Learning." *Advances in Neural Information Processing Systems* 34.

Dorffner, Georg. 1996. "Neural Networks for Time Series Processing." In *Neural Network World*. Citeseer.

Fix, E, and J Hodges. 1951. "An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation." *International Statistical Review* 3 (57): 233–38.

Friedman, Milton, and Anna Jacobson Schwartz. 2008. *A Monetary History of the United States, 1867-1960*. Vol. 14. Princeton University Press.

Gal, Yarin, and Zoubin Ghahramani. 2016. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." In *International Conference on Machine Learning*, 1050–59. PMLR.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Goodfriend, Marvin, and Robert G King. 2005. "The Incredible Volcker Disinflation." *Journal of Monetary Economics* 52 (5): 981–1015.

Greene, William H. 2012. "Econometric Analysis, 71e." *Stern School of Business, New York University*.

Hamilton, James Douglas. 2020. *Time Series Analysis*. Princeton university press.

Hamzaçebi, Coşkun. 2008. "Improving Artificial Neural Networks' Performance in Seasonal Time Series Forecasting." *Information Sciences* 178 (23): 4550–59.

Ho, Tin Kam. 1995. "Random Decision Forests." In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1:278–82. IEEE.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80.

Ish-Horowicz, Jonathan, Dana Udwin, Seth Flaxman, Sarah Filippi, and Lorin Crawford. 2019. "Interpreting Deep Neural Networks Through Variable Importance." *arXiv Preprint arXiv:1901.09839*.

Joseph, Andreas, Eleni Kalamara, George Kapetanios, and Galina Potjagailo. 2021. "Forecasting Uk Inflation Bottom Up."

Kihoro, J, RO Otieno, and C Wafula. 2004. "Seasonal Time Series Forecasting: A Comparative Study of ARIMA and ANN Models."

Kilian, Lutz, and Helmut Lütkepohl. 2017. *Structural Vector Autoregressive Analysis*. Cambridge University Press.

Kingma, Diederik P, and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv Preprint arXiv:1412.6980*.

Kydland, Finn E, and Edward C Prescott. 1982. "Time to Build and Aggregate Fluctuations." *Econometrica: Journal of the Econometric Society*, 1345–70.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2016. "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles." *arXiv Preprint arXiv:1612.01474*.

Lucas, JR. 1976. "Econometric Policy Evaluation: A Critique ', in k. Brunner and a Meltzer, the Phillips Curve and Labor Markets, North Holland."

Lütkepohl, Helmut. 2005. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media.

Masini, Ricardo P, Marcelo C Medeiros, and Eduardo F Mendes. 2021. "Machine Learning Advances for Time Series Forecasting." *Journal of Economic Surveys*.

McCracken, Michael W, and Serena Ng. 2016. "FRED-MD: A Monthly Database for Macroeconomic Research." *Journal of Business & Economic Statistics* 34 (4): 574–89.

McCulloch, Warren S, and Walter Pitts. 1990. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biology* 52 (1): 99–115.

Olah, Chris. 2015. "Understanding LSTM Networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Perry, George L, and James Tobin. 2010. *Economic Events, Ideas, and Policies: The 1960s and After*. Brookings Institution Press.

Romer, Christina D, and David H Romer. 1989. "Does Monetary Policy Matter? A New Test in the Spirit of Friedman and Schwartz." *NBER Macroeconomics Annual* 4: 121–70.

Sims, Christopher A et al. 1986. "Are Forecasting Models Usable for Policy Analysis?" *Quarterly Review* 10 (Win): 2–16.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *The Journal of Machine Learning Research* 15 (1): 1929–58.

Verstyuk, Sergiy. 2020. "Modeling Multivariate Time Series in Economics: From Auto-Regressions to Recurrent Neural Networks." *Available at SSRN 3589337*.

Zhang, G Peter. 2003. "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model." *Neurocomputing* 50: 159–75.

Zhang, Guoqiang, B Eddy Patuwo, and Michael Y Hu. 1998. "Forecasting with Artificial Neural Networks:: The State of the Art." *International Journal of Forecasting* 14 (1): 35–62.

Appendix

8 Tables

9 Figures

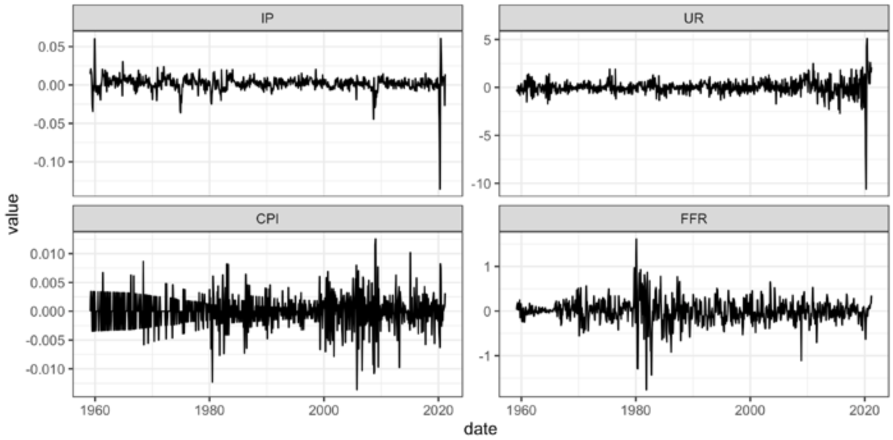


Figure 9.1: Time series

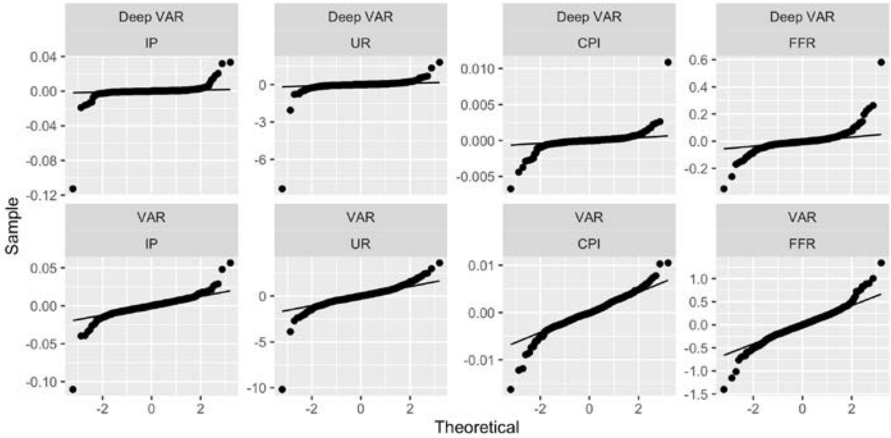


Figure 9.2: Quantile-quantile plots of full-sample residuals.

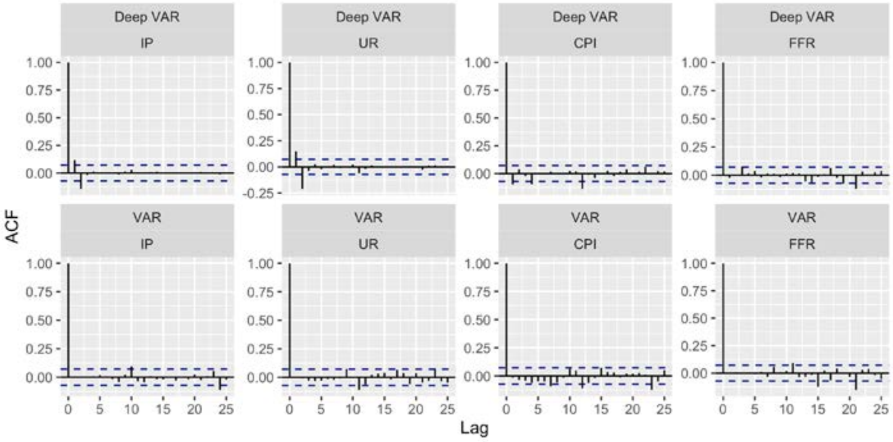


Figure 9.3: ACF plots of full-sample residuals.

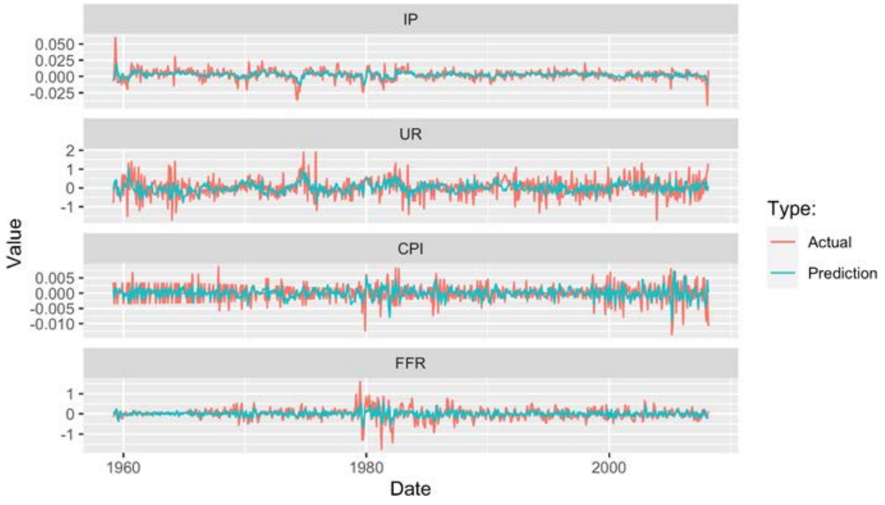


Figure 9.4: VAR fitted values plotted against observed values for the training sample.

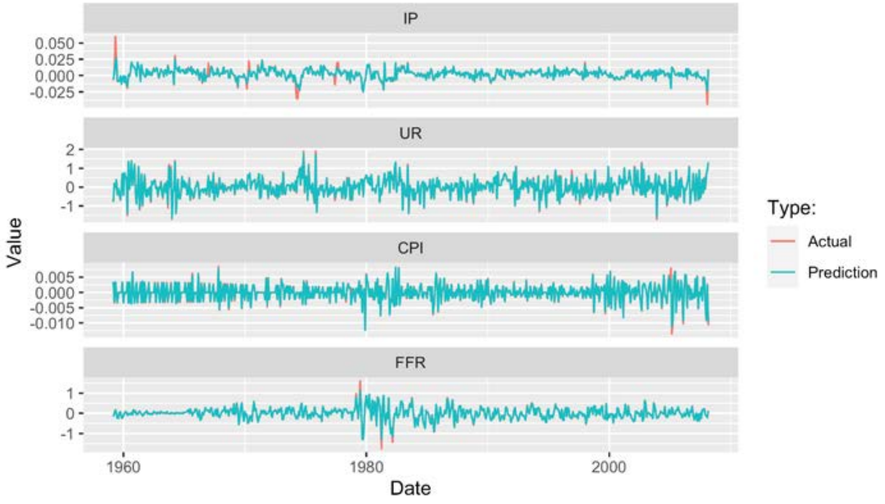


Figure 9.5: Deep VAR fitted values plotted against observed values for the training sample.

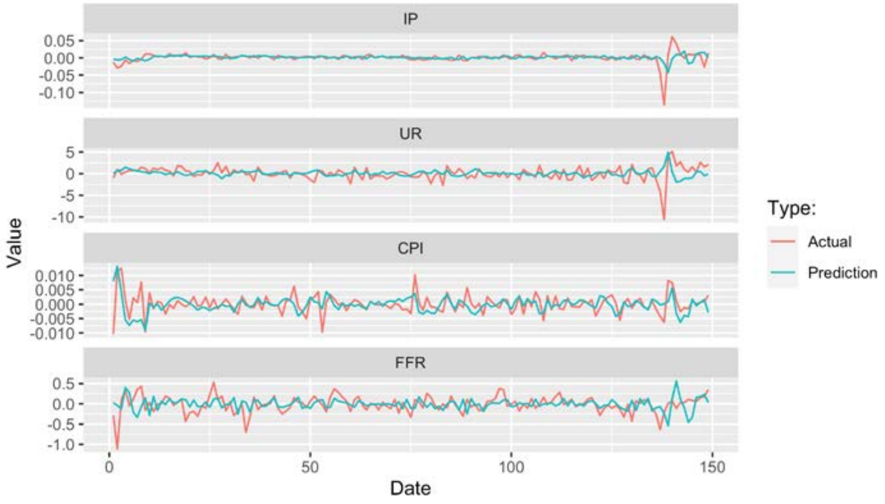


Figure 9.6: VAR fitted values plotted against observed values for the test sample.

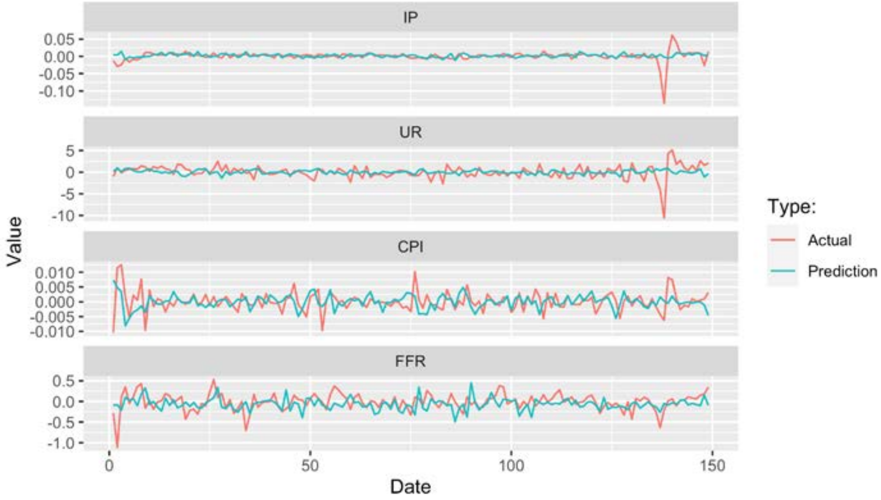


Figure 9.7: Deep VAR fitted values plotted against observed values for the test sample.

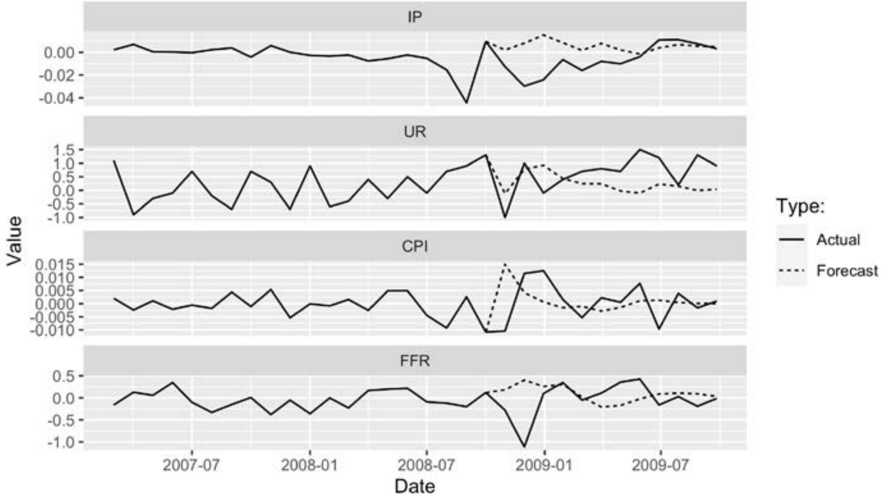


Figure 9.8: VAR n-step ahead forecasts plotted against observed values. Forecasts are for the first year of the test sample.

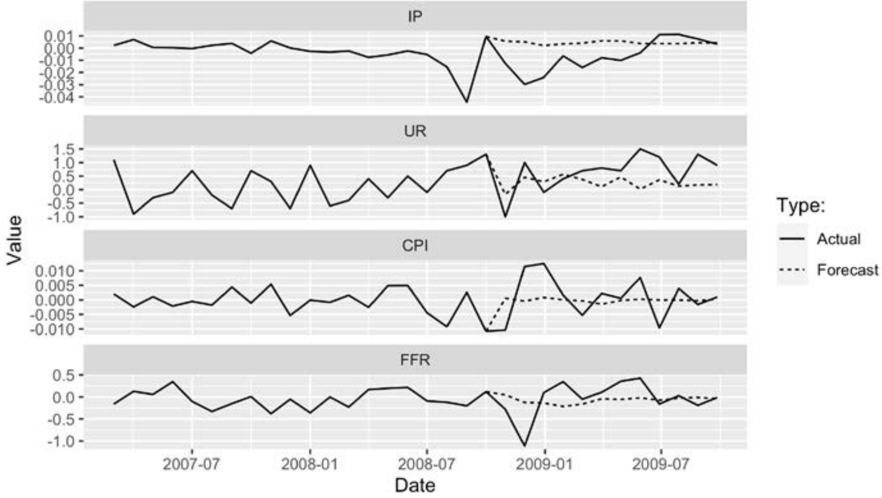


Figure 9.9: Deep VAR n-step ahead forecasts plotted against observed values. Forecasts are for the first year of the test sample.

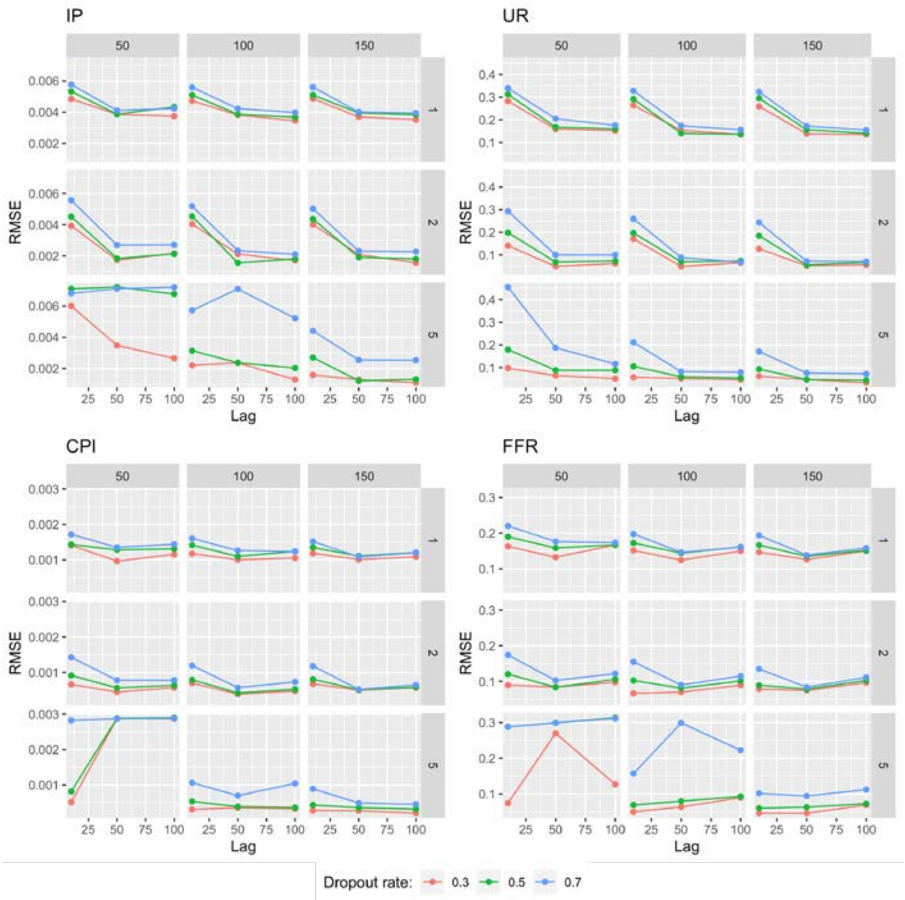


Figure 9.10: Train sample RMSE for Deep VAR for different variables.

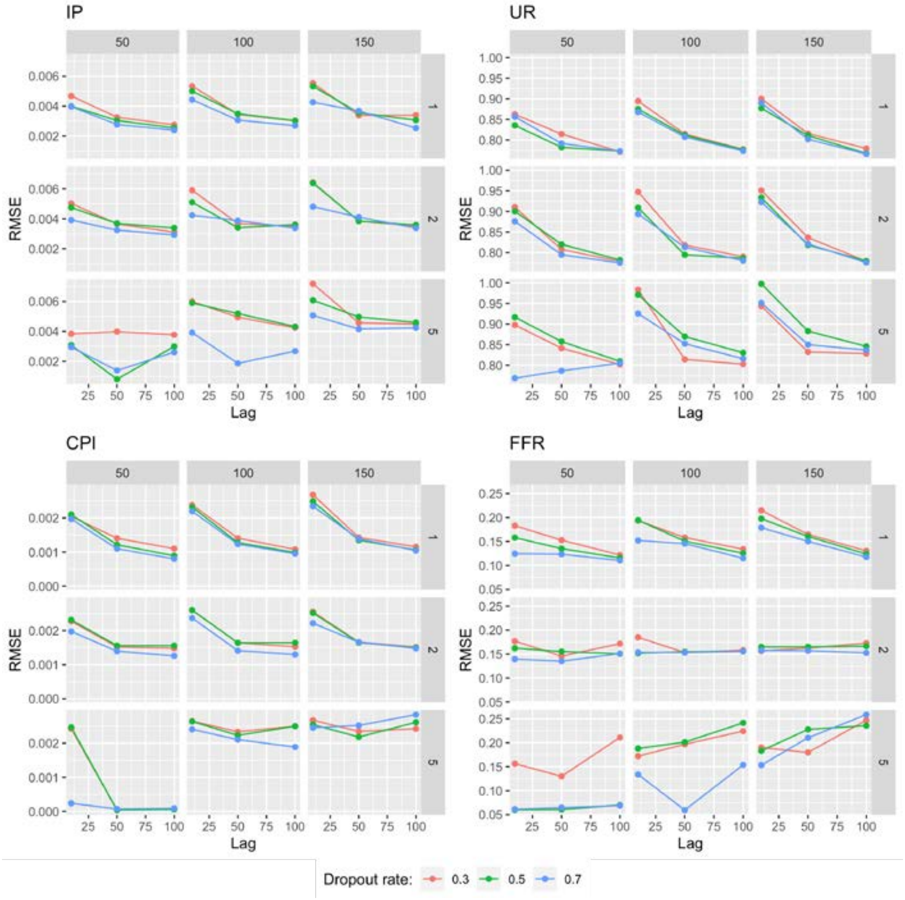


Figure 9.11: Test sample RMSE for Deep VAR for different variables.

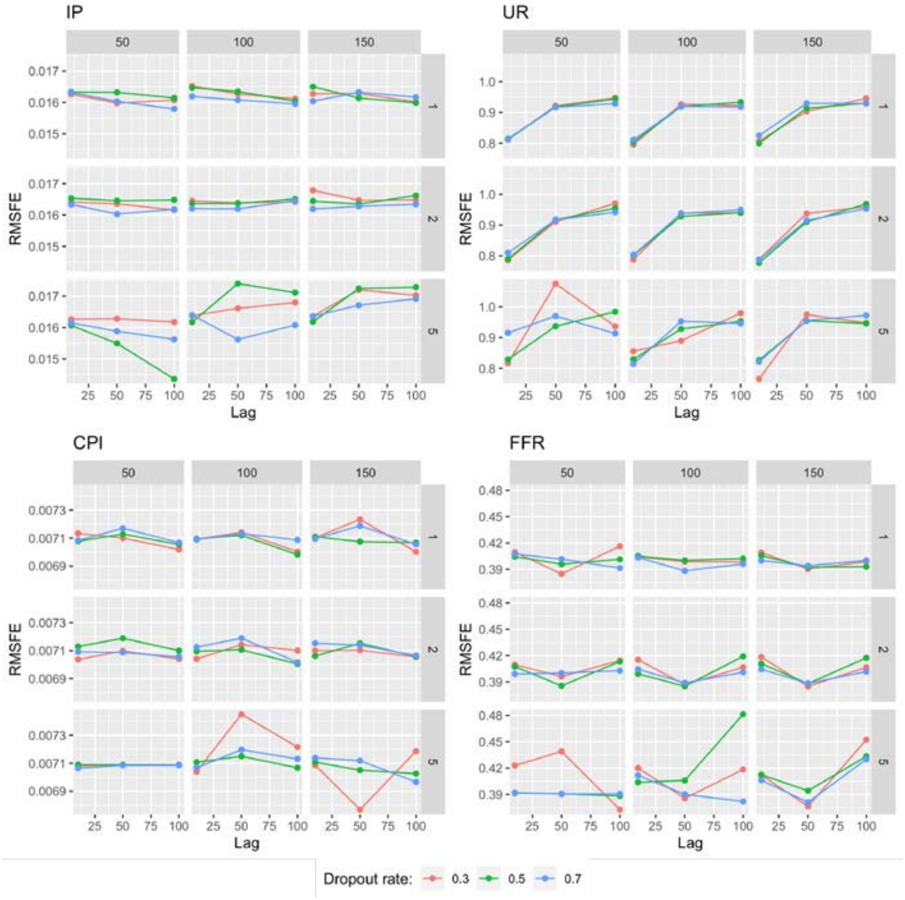


Figure 9.12: Pseudo out-of-sample RMSFE for Deep VAR for different variables.

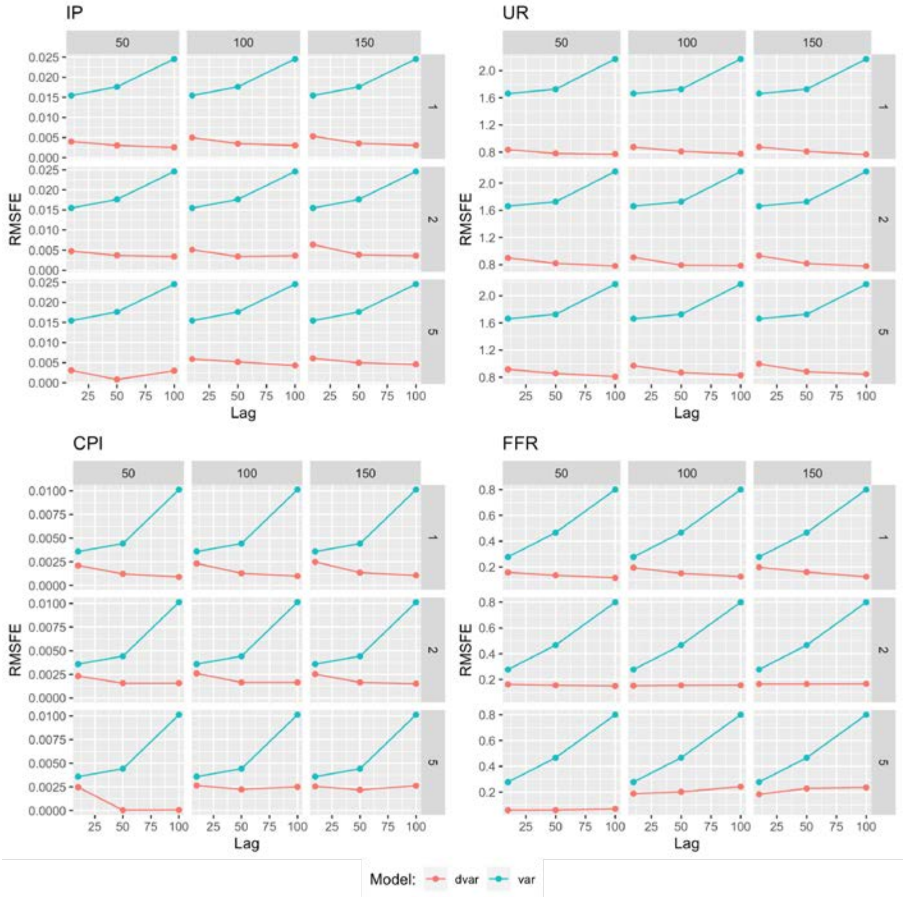


Figure 9.13: Comparison of out-of-sample RMSE for conventional VAR and Deep VAR for different variables.

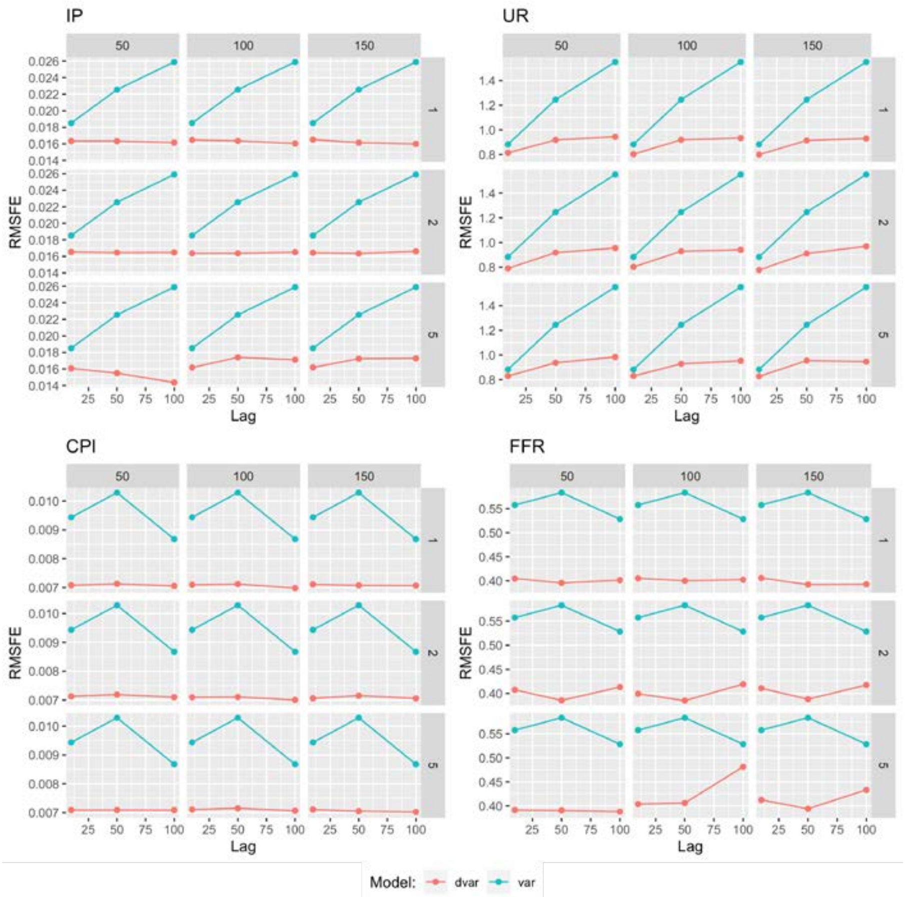


Figure 9.14: Comparison of pseudo out-of-sample RMSFE for conventional VAR and Deep VAR for different variables.

10 Code and Package

All code used for the empirical analysis presented in this article can be found on the corresponding GitHub repository. Researchers interested in using Deep VARs more generally for their own empirical work may find the R deepvars package useful which is being maintained by one of the authors. The package is still under development and as of now only available on GitHub. To install the package in R simply run:

```
devtools::install_github("pat-alt/deepvars", build_vignettes=TRUE)
```

Package vignettes will take you through the basic package functionality. Once the package has been installed simply run utils::browseVignettes() to access the documentation.

Deep Vector Autoregression for Macroeconomic Data

Patrick Altmeyer[1] Marc Agusti[2] Ignacio Vidal-Quadras Costa[3]

31 January, 2022

[1] Delft University of Technology, p.altmeyer@tudelft.nl

[2] European Central Bank, marc.agusti _i _torres@ecb.europa.eu

[3] European Central Bank, ignacio.vidalquadrascosta@barcelonagse.eu

Motivation

Can we leverage the power of deep learning in VAR models?

- We propose **Deep VAR**: a novel approach towards VAR that leverages the power of deep learning in order to model non-linear relationships.
- Worked under the following premise: **maximize performance** of an existing and trusted framework under **minimal intervention**.
- We maintain the additive structure of the VAR, but relax the assumption of linearity by modelling each equation of the VAR system as a recurrent neural network.
- By staying methodologically as close as possible to the original benchmark, we hope that our approach is more likely to find acceptance in the economics domain.

Key contributions

- Simple methodology close in spirit to conventional benchmark.
- Significant improvement in model fit and forecasting accuracy.
- Open source R package `deepvars` to facilitate reproducibility.

Work-in-progress:

- Master's thesis was selected for publication by Universitat Pompeu Fabra.
- Feedback rounds with Eddie Gerba (Bank of England, LSE) and Chiara Osbat (ECB).
- Presented an updated version of the paper at NeurIPS 2021 MLECON workshop in December.

Previous literature

- Non-linear dependencies are likely to form part of the data generating process of variables commonly used to model the monetary transmission mechanism (Brock et al. 1991).
- A range of machine learning models has previously been used in the context of time series forecasting Kihoro, Otieno, and Wafula (2004). Deep learning has been shown to be particularly successful at capturing non-linearities G. P. Zhang (2003).
- Joseph et al. (2021) review both machine learning and deep learning methods for forecasting inflation and find that neural networks in particular are useful for forecasting especially at a longer horizon.

Methodology

- Relax the assumption of linearity and instead model the process as system of potentially highly non-linear equations:

$$y_{it} = f_i\left(\mathbf{y}_{t-1:t-p};\theta\right) + v_{it} \quad , \quad \forall i = 1, ..., K \tag{1}$$

- Each single variable is model is modelled as a recurrent neural network:

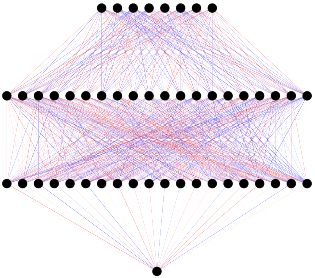


Figure 1: Neural Network Architecture.

Data

- To evaluate our proposed methodology empirically we use the FRED-MD data base to collect a sample of monthly US data on:
  - output (IP)
  - unemployment (UR)
  - inflation (CPI)
  - interest rates (FFR)
- Our sample spans the period from January 1959 through March 2021.

Model fit

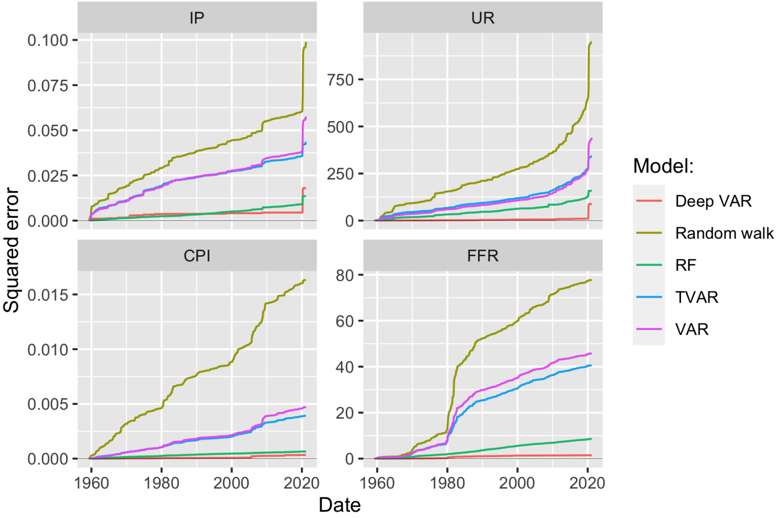


Figure 2: Comparison of cumulative loss over the entire sample period for Deep VAR and benchmarks.

Forecasting

Question: recursive forecasts like in conventional VAR or training on n outputs?

- Initially we opted for the former approach and provided anecdotal evidence that Deep VAR outperforms
- Have since tested this more rigorously using rolling window:
  - Deep VAR still outperforms VAR especially at short horizon
  - Currently investigating if training on *n* outputs provides additional edge.

Concluding remarks

- Simple framework that relies on the premise of minimal intervention in the conventional and trusted framework.
- Deep learning appears to do a good job at capturing non-linear dependencies.

But...

- Added complexity is (often) coupled with lack of interpretability:
  - No analytical expressions for impulse response functions and variance decompositions
  - Verstyuk (2020) manages to recover IRFs numerically; should be readily applicable to our Deep VAR framework.
- Uncertainty estimation can be done through Bayesian methods: deep ensemble, MC dropout, Variational Inference:
  - All of the above entail an added layer (layers really!) of computational complexity.
  - Laplace Redux for effortless Bayesian Deep Learning (Daxberger et al. 2021) holds promise, but not yet implemented.

Your questions and comments

References I

Brock, William Allen, William A Brock, David Arthur Hsieh, Blake Dean LeBaron, and William E Brock. 1991. *Nonlinear Dynamics, Chaos, and Instability: Statistical Theory and Economic Evidence*. MIT press.

Daxberger, Erik, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. 2021. "Laplace Redux-Effortless Bayesian Deep Learning." *Advances in Neural Information Processing Systems* 34.

Hamzaçebi, Coşkun. 2008. "Improving Artificial Neural Networks' Performance in Seasonal Time Series Forecasting." *Information Sciences* 178 (23): 4550–59.

Joseph, Andreas, Eleni Kalamara, George Kapetanios, and Galina Potjagailo. 2021. "Forecasting Uk Inflation Bottom Up."

Kihoro, J, RO Otieno, and C Wafula. 2004. "Seasonal Time Series Forecasting: A Comparative Study of ARIMA and ANN Models."

References II

Olah, Chris. 2015. "Understanding LSTM Networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Verstyuk, Sergiy. 2020. "Modeling Multivariate Time Series in Economics: From Auto-Regressions to Recurrent Neural Networks." *Available at SSRN 3589337*.

Zhang, G Peter. 2003. "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model." *Neurocomputing* 50: 159–75.

Zhang, Guoqiang, B Eddy Patuwo, and Michael Y Hu. 1998. "Forecasting with Artificial Neural Networks:: The State of the Art." *International Journal of Forecasting* 14 (1): 35–62.

Hiddens

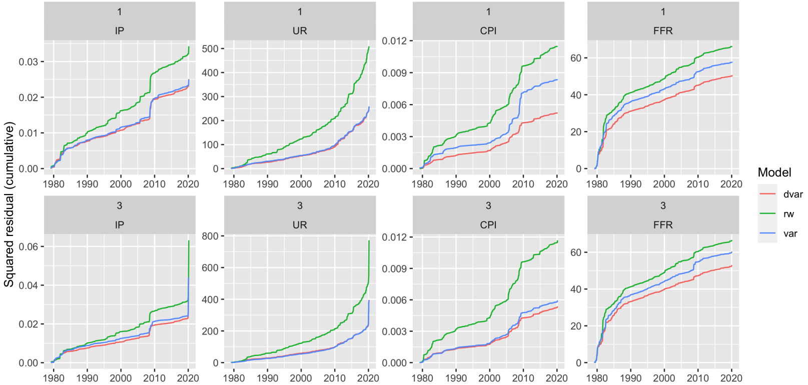
Long Short-Term Memory

- The most common choice of neural networks architectures for modelling persistent time series is the LSTM:

  "The LSTM [has] the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through." — Olah (2015)

$$
\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{b}_f + \mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{h}_{-1}\right) \\
\mathbf{i}_t &= \sigma\left(\mathbf{b}_i + \mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{h}_{-\mathbf{1}}\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{b}_o + \mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{h}_{-1}\right) \\
\mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tanh\left(\mathbf{b}_C + \mathbf{W}_C \mathbf{h}_{t-1} + \mathbf{U}_C \mathbf{h}_{-1}\right) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \\
\hat{y}_{it} &= c + \mathbf{v}^T \mathbf{h}_t
\end{aligned}
\tag{2}
$$

Rolling window forecasts


IP
UR
CPI
FFR
IP
UR
CPI
FFR
Squared residual (cumulative)
Model
dvar
rw
var