# Regime Switching Neural Network for Financial Time Series Forecasting

Robert Nardello, Hyeoncheol Baik PhD

School of Business

Stockton University

**Acknowledgements**

This research bears the mark of several people who have made a significant impact on my thinking. I am deeply indebted to the Board of Trustees at Stockton University, who made this research possible through their generous donation. I express my sincere appreciation for my advisor Dr. Hyeoncheol Baik for his patience, support, and guidance throughout this research. His insight and direction were paramount in the fulfillment of this study. I would also like to express my gratitude to the many teachers in the Business Program that have provided me an exceptional education here at Stockton University thus far.

1. Introduction...........................................................................................................................1

2. Neural Networks Models......................................................................................................3

2.1 Feed-Forward Neural Network...............................................................................3

2.2 Neural Network Autoregression.............................................................................5

3. Markov Models ....................................................................................................................5

3.1 Markov Chains.......................................................................................................6

3.2 Hidden Markov Model...........................................................................................6

4. Outline of Research .............................................................................................................8

4.1 Dataset....................................................................................................................8

4.2 Objective................................................................................................................8

4.3 Methodology..........................................................................................................9

4.4 HMM Parameters.................................................................................................10

4.5 Neural Network Parameters…….........................................................................11

I. Observations in Training Set......................................................................11

II. Model Architeture.....................................................................................11

4.6 Measuring Forecast Accuracy….........................................................................12

5. Results……………...........................................................................................................13

6.Conclusion……………......................................................................................................13

7. Extensions…….…............................................................................................................14

8. References…….…............................................................................................................15

***Abstract - In most real-life scenarios such as finance and business applications, time-series data exhibit dramatic shifts in behavior due to changes in regulation, economic policy, and secular conditions. These changes result in temporally varying statistics and nonlinear behavior in data, making financial systems notoriously difficult to model even with advanced forecasting methods like neural networks. In the past, Regime Switching Models have been useful in assessing the state of the economy and market, as they help model data with both trends and seasonal patterns. This paper proposes a Neural Network with a refined training set by using regime-switching. In the first stage, a Hidden Markov Model (HMM) is used to classify data into expansionary and recessionary periods. Following, a Neural Network trains on the previously classified data for time series prediction. Our proposed approach of a Feed-Forward Neural Network trained on data partitioned by an HMM model had significantly better prediction accuracy than a standard Feed-Forward Neural Network when forecasting a 20-year dataset of Real GDP.***

1. **Introduction**

Financial institutions, individual investors, and researchers frequently use financial time series data such as exchange rates, GDP, and other macroeconomic indicators in forecasts. In most business domains, including finance and economics, time-series data is extremely volatile and complex, due to behavioral changes over relatively long sample periods. These behavioral changes may occur due to some permanent change in the structure of the economy (Piger, 2007). Additionally, shifts can occur due to pathological economic episodes like the most recent financial crisis precipitated by the pandemic. Finally, changes can be both temporary, and recurrent as economic cycles loop between regimes, in the case of expansionary and recessionary periods (13).

Since the 1980s, analysts have been using Machine Learning (ML) techniques to identify complex market trends. While ML has had some success in forecasting financial markets, recent developments in deep learning have not dramatically improved model accuracy compared to the level of advancement in other fields modeling complex systems like natural language processing, and computer vision (Patel, 2018).

A fundamental reason for the poor accuracy of financial deep learning models’ is data distribution. To demonstrate, let us consider a neural network used for classifying handwritten digits. We can expect the distribution of pixel weights in number 1's class training set to be close to number 1's test set distribution. In simpler terms, images of the number 1 must contain the number 1 for a model to predict accurately. This property does not hold for financial datasets, due to the behavioral changes of financial data over time (Patel, 2018). In financial markets, what you may see in the future could be completely different from what has occurred in the past; demonstrated by our current recession. In financial applications of Machine Learning models, the learned model is only useful when forecasted data adhere to the train/validation set distribution (12).

Our proposed method of addressing behavioral changes, and subsequently changes in the data distribution of a financial time series utilizes a regime-switching model to partition training data for a neural network. Regime switching models are amongst the most useful models for forecasting and understanding a financial time series. One of the most popular regime-switching models in finance is the Markov Switching Model. (Chung-Ming, 2002). A Markov Switching Model involves multiple equations that characterize the behavior of a time series into different states (5). For example, Markov Switching models are used to identify business cycles, composed of expansionary and recessionary periods (Hamilton, 1989).

Multiple studies have developed Markov Switching forecasting models by dividing sequences into distinct states and performing switching between linear regressors (Zeevi, et al., 1996, Kozat & Singer, 2010). However, using linear regressors as experts can lead to poor model performance in challenging scenarios. Although linear methods such as autoregression and moving average models have remained popular, these methods fail to capture complex temporal patterns in data (Ratnadip, Agrawal, 2013). Neural Network based methods are becoming a favored method for time series prediction due to their ability to approximate highly nonlinear and complex functions. Artificial Neural Networks (ANNs) with multiple layers have also had success in the prediction of financial time series (Smalter, Cook, 2017). In this study we use the same switching mechanism as the traditional Markov Switching Model pioneered by Hamilton (1989), with a Feed-Forward Neural Network used as an expert opposed to a linear model. Multiple models are not employed, but training data “resets” when a regime shift occurs.

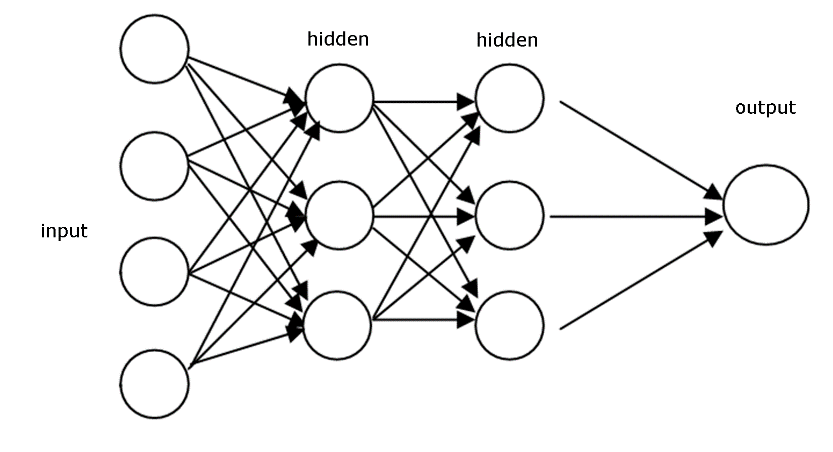
In this research a Hidden Markov Model (HMM) is applied to 20-year US Real GDP data to determine two-state underlying regimes. The training set for a Feed-Forward Neural Network model is then partitioned according to the probability of being in a particular regime state. Our Neural Network with a refined training set can detect different regimes and focus on each of them separately when training on data. This enables the Neural Network to adapt to dramatic shifts in a time series when forecasting. The results of this research indicate neural networks trained on data partitioned by the Hidden Markov Model possess greater prediction accuracy compared to a standard Feed-Forward Neural Network without HMM data partitioning. This affirms our approach to have potential application in financial time series forecasting and the possibility for use with more advanced deep learning models.

**2. Neural Network Models**

Neural networks are statistical learning models inspired by the way biological nervous systems process information. They are a structure of connected artificial neurons, or units which loosely model neurons found in our brain (Hyndman & Athanasopoulos, 2018). Through each connection, a signal to other neurons is transmitted. Neural networks “learn” or are trained through evaluating examples containing known "inputs" and "outputs". Neural Networks are organized in layers. The first layer consists of predictors (inputs), and forecasts form the last layer (outputs). Each layer is made up of several processing units, or nodes. Each node is connected to the nodes in the following layer via a weight. A learning algorithm that minimizes a cost function is used to select weights (7).

**2.1 Feed-Forward Neural Network**

In a Feed-Forward Neural Network, the flow of information only moves forward, hence its name. Information moves from the input nodes, through the hidden nodes (if any) and to the output nodes. Each layer of nodes receives inputs from the previous layers, except the input. The output of the nodes in one layer are inputs to the next layer. Simple networks containing no hidden layers are equivalent to linear regressors. When an intermediate layer with hidden neurons is added, the network becomes nonlinear.



***Figure 2.1*** *Portrayal of a basic Feed-Forward Neural Network (15)*

Figured 2.1 is a stylized portrayal of a basic Feed-Forward Neural Network. Each circle represents one computational node, and each stack of nodes represents a computational layer. The inputs to each node are combined using a weighted linear combination. The result is then modified by a nonlinear function before being output. For example, the inputs into hidden neuron *j* are combined linearly by

To allow for a non-linear relationship between the weighted inputs and the output, a nonlinear function such as a sigmoid is applied.

The sigmoid function is applied on the linear model, yielding.

Application of the sigmoid function minimizes extreme input values, making the network more resilient for outliers (Hyndman & Athanasopoulos, 2018). The parameters and are “learned” from data. Initially, weights take a random value, and are update as new data is observed. Values of the weights are often restricted to prevent them from becoming too large. This is referred to as a decay parameter and is used to prevent overfitting. Networks are usually trained several times using different random starting points, and the results are averaged. However, if the model is trained with many backpropagation iterations, the weights will become large. This results in the network fitting to training data too well and failing to generalize to new observations (7).

**2.2 Neural Network Autoregression**

When modeling time series data with a neural network, lagged values are used as input for the network. This is referred to a neural network autoregression. For example, a neural network using the last 5 observations as input for forecasting the output can be denoted as ). When forecasting one step ahead, the previous historical data points as used as inputs. When forecasting multiple steps ahead, one-step ahead and all other forecasts are used as inputs for the forecasting result (Hyndman & Athanasopoulos, 2018).

1. **Markov Models**

Markov Models are often used to model dynamical systems, and chaotic phenomena that appear to vary randomly. In a Markov Process, the future state of a system is assumed to only depend on the systems current state. This assumption reduces computational complexity and is popular amongst those in the fields of predictive modeling and probabilistic forecasting (Chung-Ming, 2002). Regime-switching models have become an enormously popular modeling tool for financial time series modeling. Particularly regime-switching models measuring economic output, such as Real GDP. Examples of such models include Hamilton (1989), and Piger (2007). Markov-switching models popularized by Hamilton (1989), presume economic regime shifts emerge according to Markov Chains.

**3.1 Markov Chains**

Markov Chains are mathematical models made up of a finite number of states. , and assumed probabilities where is the probability from transition from state to . A Markov Chain is memoryless and guarantees the evolution of a process depends only on its present state, and not on its history. Therefore, a Markov Process satisfies this equation.

At any point in time, the state t +1 is dependent only on the previous state at time t. Modeling a financial time series like GDP as a Markov Process is logical as numerous studies suggest the behavior of financial time series exhibit drastically different patterns over time (Filardo, 1994). In other words, the state of GDP 20 years ago, may have nothing to do with the current state of GDP, and if so, should not be included as training data for our model.

**Recessionary**

**Expansionary**

.35

.6

.4

.7

***Figure 3.1*** *Visualization of Markov Chain for Financial Time Series*

**3.2 Hidden Markov Model**

A Hidden Markov Model is a Markov Chain with states that are not directly observable.

However, observations dependent on the state are. The “hidden” in Hidden Markov Model implies that we do not know with absolute certainty the state of the system but have a probabilistic idea. States can be thought of as probability distributions over a set of observations. Many time series of real-world processes have hidden variables (latent variables), which are unobservable in the data and available for learning (Russell, et al., 2009).

The structure of a hidden Markov Model can be defined as:

* Where the Variable S is a finite set of hidden states
* The Variable O are the observations of the model
* The Variable A is the models transition probability between states.
* The Variable B is the models emission probability. Emission Probabilities are the likelihood of being in a certain state at any given time step.
* The Variable is the models initial state, where is the probability of starting in state

There are three main problems associated with the Hidden Markov Model:

A. Evaluation

Given model Parameters and the sequence of observations how is the probability of the observation sequence computed? This problem is solved using he Expectation Maximization Algorithm.

B. Decoding

What sequence of states best explains the sequence of observations? Or in this case, what periods of recession and expansion explains the time series of Real GDP best? Given model Parameters and the sequence of observations the Viterbi algorithm finds a maximum over all possible state sequences.

C. Learning

Given a set of observations } how do we learn the probability of our model to maximize the sequence of observations *P*?This is done through use of the forward-backward algorithm originated by Baum & Pierre (1966).

1. **Outline of Research**

The main objective of the proposed approach is to improve the accuracy of our neural network model via refined data partitioning. An HMM model was trained to classify the time series of 20-year Real GDP into expansionary and recessionary regimes. Our neural network model was then trained solely on data from its corresponding regime, compared to its entire training set. The intuition behind this methodology is to resolve the issue of changing data distributions in financial time series modeling, providing the neural network with pertinent training data to learn from. In theory, the more relevant training data is to forecasted values the greater predictive capability a ML model will have.

**4.2 Dataset**

Data of Real US Gross Domestic Product (GDP) was retrieved from the Federal Reserve Bank of St. Louis website https://fred.stlouisfed.org/series/GDPC1. Real gross domestic product is the inflation adjusted value of the goods and services produced by labor and property located in the United States and is one of the most important aggregate economic indicators for understanding our economy (Federal Reserve Bank of St. Louis, 2020). A description of the dataset is outlined in table 4.1 and visualized in Figure 4.1.

**Table 1: Description of dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Date** | **Observations** | **Units** | **Frequency** |
| U.S. Real GDP | Jan-2000 to Mar-2020 | 81 | Billions | Quarterly |

***Table 4.1***

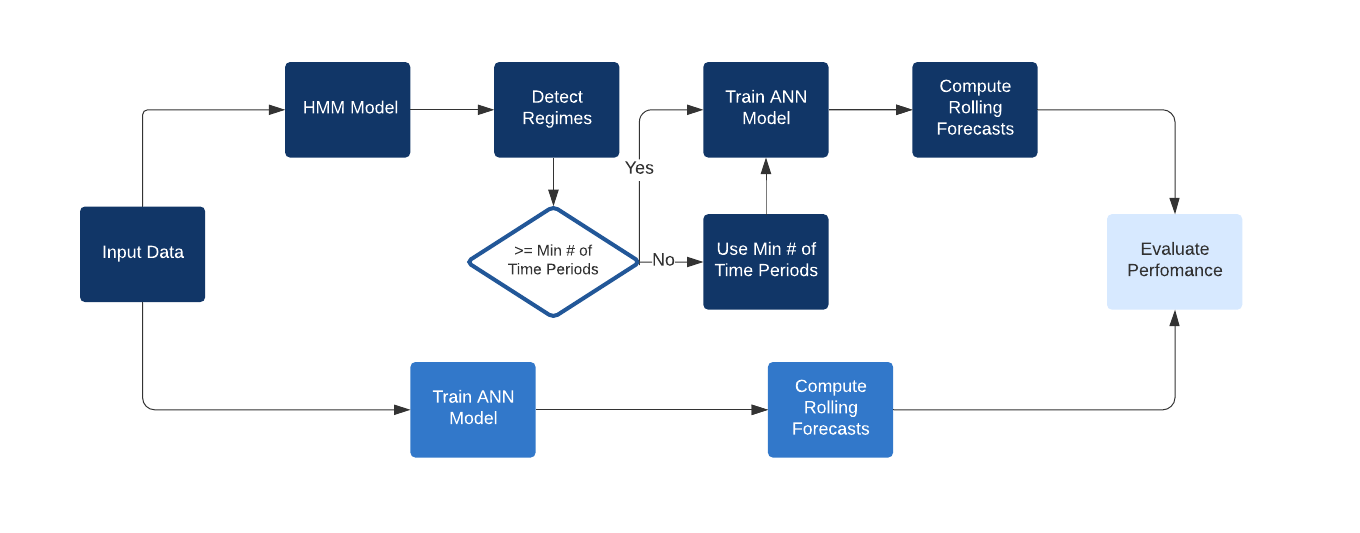
Chart, line chart

Description automatically generated

***Figure 4.1*** *Time Series of Real GDP, Jan. 2000-Mar. 2020*

**4.3 Methodology**

Data was partitioned by the HMM into two distinct regimes. Neural Networks tend to perform poorly when training on limited observations, a problem outlined in section 4.5, so a minimum number of training set observations is established for our data partitioning algorithm. If observations partitioned by the HMM model are larger than the minimum number of observations, the full batch is used as training data for our network. If the partition is less than the minimum number of training observations, then we simply use the minimum number of observations for a given training set. To validate our model, we compute 1-step ahead rolling forecasts. Model performance is evaluated by averaging the returned Mean Squared Prediction Error (MSPE) of the rolling forecasts and comparing results to MSPE of the benchmark Feed-Forward Neural Network without the use of HMM data partitioning. The general framework of our research is outlined in figure 4.2.

***Figure 4.2*** *Flow chart describing general approach of research.*

**4.4 HMM Model Parameters**

The number of latent states in a HMM must be set in advance, before training**.** Usually, GDP growth rates fluctuate at a higher level and are more persistent during periods of growth, but remain at a relatively lower level and less persistent during recessionary phases (Filardo, 1994). Because of this phenomenon, a 2-regime system has been widely accepted in economic literature, and the number of latent states chosen for data partitioning (Piger, 2007).

Before fitting the HMM model, it is possible to plot the posterior probabilities of being in a particular regime state. Regime states are designated as expansionary and recessionary. Posterior probabilities are compared with the underlying true states. Partitioned regime states, and their underlying posterior probabilities are as shown in Figure 4.3.

Chart

Description automatically generated

***Figure 4.3*** *Regimes Partitioned by the HMM model, and their underlying probabilities.*

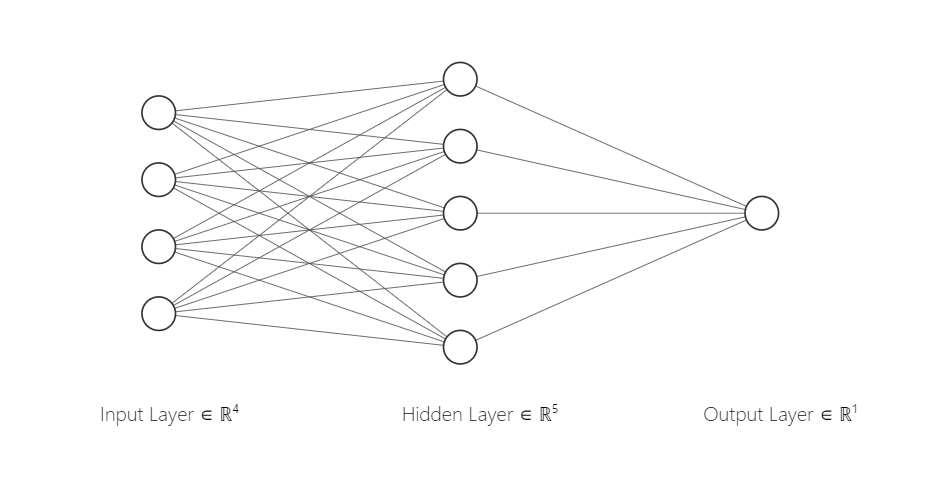
* 1. **Neural Network Model Parameters**

1. **Observations in Training Set**

A downside for this proposed approach of data partitioning is that it often leads to fewer observations for a Neural Network model to “learn” from. Deep learning models favor common observations and lose their ability to generalize when training on smaller datasets (Dilip, 1996). To account for this problem, our model trains on a minimum of 5 observations in its training set per forecast. If the number of time periods within the same regime state is less than the minimum number of time periods, then we use the minimum number of time periods as our training set. Although in some cases there may be less data for our model to train on, the intuition is that our Neural Network with a refined training set will outperform a typical Feed-Forward Neural Network in handling “edge” cases, where drastic changes in the economic system have occurred. This intuition was confirmed by the results in section 5, as our model utilizing the refined training set better adapted to major inflection points in 2008, 2014, and the most recent recession brought on by the pandemic.

1. **Model Architecture**

Both the Feed-Forward Neural Network utilizing the refined training set, and benchmark network have 4 input nodes, 5 hidden nodes, and 1 output node as depicted in Figure 4.4. This is a univariate model, subsequently the sole inputs for the network model are lagged variables. This is a simple neural architecture, and more complex deep learning models with additional features and hidden layers may have greater prediction capability when modeling a complex series like Real GDP. However, the purpose of this study was to assess the effectiveness of our regime switching data partitioning algorithm in improving model performance, so a simple architecture was used for the sake of result interpretability.



***Figure 4.4*** *Neural Network Architecture for proposed research*

* 1. **Measuring Forecast Accuracy**

To measure forecast accuracy, we compute 1-step ahead rolling forecasts, conducting regressions on subsamples of the 20-year Real GDP dataset. The intuition for this approach is that bias in the back-testing process can lead to inaccurate results. Factors that have driven the market in the past might not be important to current activity, so conducting forecasting across the entirety of the time series yields more reliable results. Mean Squared Prediction Error (MSPE) is averaged from the result of 67 rolling forecasts used to calculate model accuracy. Model accuracy is then benchmarked against the averaged MSPE from the same Feed-Forward Neural Network in the same forecasted periods trained on full batches of training data. Visualization of this approach outlined in figure 4.5.

**Chart

Description automatically generated**

***Figure 4.5*** *Approach for measuring forecast accuracy (8).*

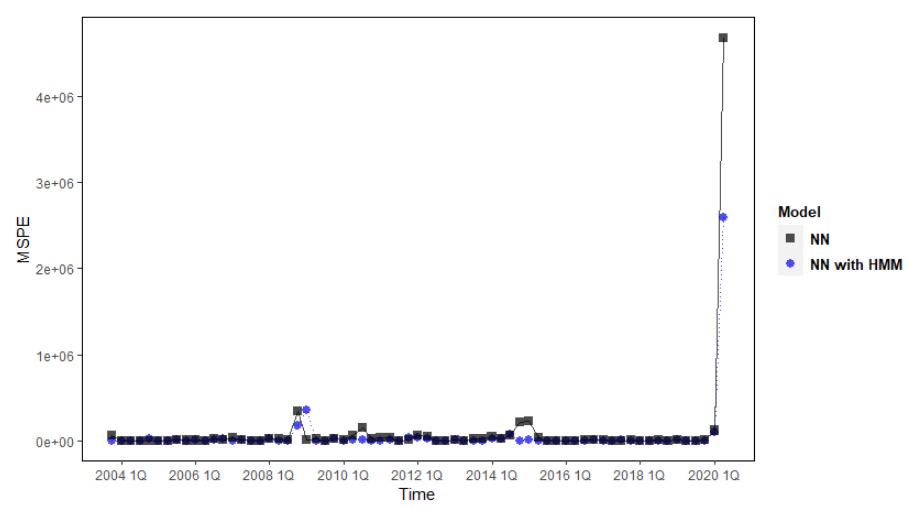
1. **Results**

The Neural Network model utilizing HMM data partitioning realized an averaged Mean Squared Prediction Error (MPSE) of 57173.4 compared to an MSPE of 101749.3 from the benchmark Feed-Forward Network without the use of our approach. Seen in figure 5.1, MPSE is significantly lower for our approach during the recessionary period of 2008, and the first 2 quarters of 2015, and the most recent recession beginning in March of 2020.

**Table 2: Results**

|  |  |
| --- | --- |
| **NN with HMM (Our Approach)** | **NN (Benchmark)** |
| **MSPE:** 57173.49 | **MSPE:** 101749.3 |

***Table 5.1***



***Figure 5.1*** *Comparative Plot of MSPE for NN and NN with HMM partitioning*

*on 20-year Real GDP dataset.*

1. **Conclusion**

Through our research, we demonstrate the performance gains of our Feed-Forward Neural Network model with HMM based data partitioning compared to a standard Feed-Forward Neural Network, frequently utilized in economic and financial time series forecasting. We show that our introduced method performs significantly better in terms of MSPE compared to the benchmark model, thanks to the joint optimization and efficient combination of HMM data partitioning, and Neural Networks. We show that the HMM algorithm can properly determine regimes, and that our model can switch between them to make more accurate predictions. As our analysis indicates, we can successfully capture distributional shifts in financial data, and adapt Machine Learning Models to make more accurate predictions on a complex financial time series.

1. **Extensions**

Variants of Recurrent Neural Networks have produced favorable results in time series prediction, as the architecture allows for connections between nodes capturing feedback in dynamic behavior (Hochreiter, Schmidhuber, 1997). Using HMM data partitioning algorithm in combination with these more complex model architectures could yield increased prediction accuracy of a complex financial time series. Additional experimentation with modifying parameters of both HMM and Feed Forward Neural Network could also lead to improved model performance.

**8. References**

1. Adhikari, Ratnadip, and R. K Agrawal, *An Introductory Study on Time Series Modeling and Forecasting.* 2013, arxiv.org/ftp/arxiv/papers/1302/1302.6613.pdf.

*2. Artificial Intelligence: A Modern Approach*, by Stuart J. Russell and Peter Norvig, Pearson, 2009, p. 816.

3. Baum LE, Petrie T (1966). “Statistical Inference for Probabilistic Functions of Finite State Markov Chains.” Annals of Mathematical Statistics, 67, 1554–40.

4. Ciaburro, Giuseppe, and Balaji Venkateswaran. *Feed Forward Neural Network*. Sept. 2017, subscription.packtpub.com/book/big\_data\_and\_business\_intelligence/9781788397872/1/ch01lvl1sec21/feed-forward-and-feedback-networks. Accessed Oct. 2020.

5. Federal Reserve Bank of St. Louis (2020). *Real Gross Domestic Product (GDPC1)* Available from Federal Reserve Bank of St. Louis Web site: <https://fred.stlouisfed.org/series/GDPC1>

6. Hall, Aaron Smalter, and Thomas R. Cook. “Macroeconomic Indicator Forecasting with Deep Neural Networks.” *SSRN Electronic Journal*, 2017, doi:10.2139/ssrn.3046657.

7. Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on Oct.2020

8. Hyndman, Rob J. “Time Series Cross-Validation.” *Hyndsight*, 2016, robjhyndman.com/hyndsight/tscv/.

9. J. D. Hamilton, “A new approach to the economic analysis of nonstationary time series and the business cycle,” Econometrica, vol. 57, no. 2, pp. 357–384, 1989

10 J. Filardo, “Business-cycle phases and their transitional dynamics,” Journal of Business & Economic Statistics, vol. 12, no. 3, pp. 299–308, 1994.

10. J. Zeevi, R. Meir, and R. J. Adler, “Time series prediction using mixtures of experts,” in NIPS, 1996.

11. Kuan, Chung-Ming. “Lecture on the Markov Switching Model.” *NTU*, NTU.edu, 2002, homepage.ntu.edu.tw/~ckuan/pdf/Lec-Markov\_note.pdf.

12. Patel, Hardik. “Why Is Machine Learning in Finance so Hard?” *Hardik Patel*, 11 Feb. 2018, www.hardikp.com/2018/02/11/why-is-machine-learning-in-finance-so-hard/.

13. Piger, Jeremy. (2007). Econometrics: Models of Regime Changes. Complex Systems in Finance and Econometrics. 10.1007/978-0-387-30440-3\_165.

14. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, pp. 1735–80, 12 1997.

15. S. Kozat and A. C. Singer, “Universal randomized switching,” Trans. Sig. Proc., vol. 58, no. 3, p. 19221927, Mar. 2010.

16. Sarkar, Dilip. “Empirical Estimation of Generalization Ability of Neural Networks.” *Applications and Science of Artificial Neural Networks II*, 1996, doi:10.1117/12.235979.