

# Long Short-Term Memory Recurrent Neural Networks for Forecasting U.S. Unemployment Rate

DANIEL SMITH

## I. INTRODUCTION

Unemployment rate is one of the key economic indicators for the health of the United States economy. The level of employment is of such importance that it is one of the two mandates of the Federal Reserve System.

Being able to accurately forecast the U.S unemployment rate is important because projections shape policy decisions that effect the whole country and the world economy.

Forecasting the unemployment rate is difficult due to the fact that the unemployment rate is a time series data set which is nonlinear and non-stationary. [1] Within economics, the dominant method for forecasting time series data with these characteristics is the autoregressive integrated moving average (ARIMA), which will be used as a baseline for this paper.

Long short-term memory recurrent neural networks (LSTM) have emerged as a powerful tool for forecasting time series data sets. The aim of this paper will be to compare LSTM to ARIMA on U.S. unemployment forecasting<sup>1</sup>.

## II. ARIMA

An ARIMA model is a useful model for forecasting non-stationary time series. For U.S. unemployment rate series  $u_t$  the model can be stated as

$$(1 - \phi_1 L - \cdots - \phi_p L^p)(1 - L)^d u_t = c + (1 - \theta_1 L - \cdots - \theta_q L^q) \varepsilon_t \quad (1)$$

<sup>1</sup>My proposal has my project being on four different data sets rather than just unemployment. After beginning my project I learned that it is probably better to do a more in depth review of one data set rather than just skim through four data sets.

where  $p$ ,  $d$ , and  $q$  are nonnegative integers.  $c$ ,  $\phi_i$  and  $\theta_j$  are parameters.  $L$  is a lag operator such that  $L u_t = u_{t-1}$ .  $\varepsilon$  is a normally distributed error term with mean 0 and variance  $\sigma^2$ .

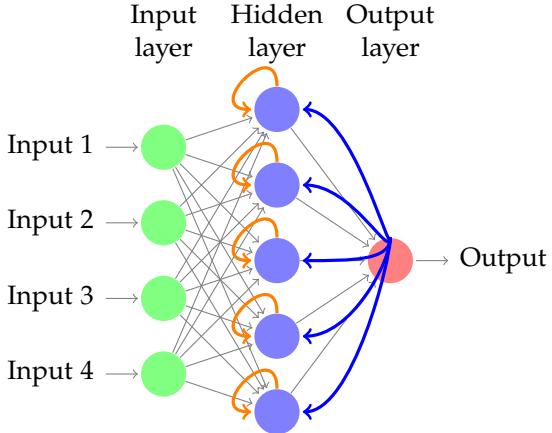
The model is typically stated as ARIMA( $p,d,q$ ). Where  $p$  is the number of time lags for the autoregressive model.  $q$  is the order of the moving average model.  $d$  is the degree of differencing. This differencing aims to reduce non-stationarity in the series. This model can be thought of as the combination of three good ideas: auto-regression, moving average and differencing.

The parameters for this model can be estimated using either nonlinear least squares or maximum likelihood estimation. The ARIMA model being used in this paper is from the python package statsmodels. For their implementation of ARIMA they fit the parameters using maximum likelihood estimation.

Finding the best  $p$ ,  $q$  and  $d$  for the model is typically done using akaike information criterion (AIC). Cross validation can also be done to determine the hyper-parameters. Due to the frequent use of ARIMA models for forecasting unemployment, economist have studied the best hyper-parameters for ARIMA. The two most frequent hyper-parameterizations for the model are ARIMA(1,1,0)[2] and ARIMA(2,1,2) [3].

## III. LSTM

Artificial neural networks have emerged as a powerful tool for statistical pattern recognition. Feed forward neural networks can't exploit previous observations in order to improve predictions. Recurrent neural networks(RNN) later emerged as a solution to this problem. Opposed to feed-forward neural networks, RNNs



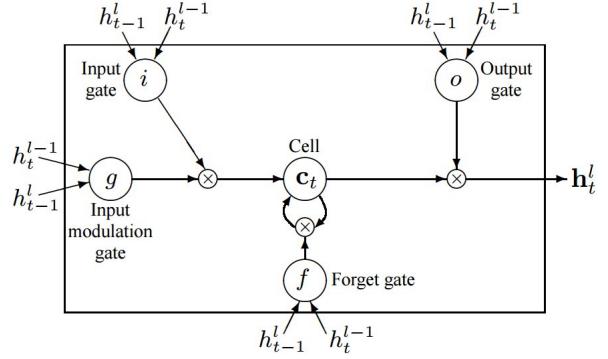
**Figure 1:** A Recurrent Neural Network

allow the data that was outputted in the hidden layers to be looped back as input. Figure 1 illustrates what this process looks like.

Creating this looping structure allows for the neural network to use the temporal structure of the data for its predictions. When training RNNs an issue that can occur is vanishing gradients. To combat this LSTMs can be used.

LSTM are a type of RNN which uses memory cells. These memory cells contain input, forget and output gates. The goal of these cells are to allow the network to be able to utilize long-term dependencies and "forget" non useful information. There are many different ways to construct these memory cells. For this paper the cells discussed in Zaremba, Vinyals and Sutskever's paper titled "Recurrent Neural Network Regularization" will be used.[4] The memory cells they describe were chosen for this paper because they describe a memory cell that has the ability train with drop out. This may help with any over fitting problems the model encounters.

Figure 2 shows an illustration used in Zaremba, Vinyals and Sutskever's paper. The memory cells are stated, Where  $D$  is the drop out operator, as:



**Figure 2:** LSTM Memory Cell. Picture taken from <https://arxiv.org/pdf/1409.2329v5.pdf>

$$\begin{aligned}
 LSTM : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l &\rightarrow h_t^l, c_t^l \\
 \begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} &= \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \left( \begin{matrix} D(h_t^{l-1}) \\ h_{t-1}^l \end{matrix} \right) \quad (2) \\
 c_t^l &= f \odot c_{t-1}^l + i \odot g \\
 h_t^l &= o \odot \tanh(c_t^l)
 \end{aligned}$$

For this paper this LSTM model was implemented in tensorflow. Because the unemployment rate  $\in \mathbb{R}$ , a regression output function is used. Different size architectures are tested out using cross validation. The objective function being used is mean square error and it is optimized by mini-batch gradient descent with a batch size of 100.

#### IV. DATA

The unemployment rate is released monthly by The Bureau of Labor Statistics. The data used for this project is the monthly seasonally adjusted unemployment rate from the period of 1/1/1948 - 9/1/2016.

Figure 3 in the appendix shows a graph of unemployment rate. As you can see unemployment rises during recessions. The time between recessions is often spread out which is why a LSTM model can be beneficial, it can

use events that happened many time steps in past to make predictions.

Next the data was tested for non-stationarity. Figure 4 in the appendix shows the rolling mean and standard deviation of the data set. To calculate these rolling statistics a window size of 10 was used. The rolling mean increases during times of recession. Also the data becomes more volatile during recessions.

To further determine non-stationarity an augmented dickey fuller test (ADF) was used. An ADF is a unit root test, the null hypothesis is that there is a unit root. The dickey-fuller test was performed with a constant and time trend in the regression. Choosing to include a constant or time-trend can greatly effect the results of the dickey-fuller test. This decision was based on domain knowledge.[3] The p-value of the test is .078 so we fail to reject the null hypothesis at the 95% confidence level.

As shown in Figure 4, the mean and standard deviations patterns depending on if the economy is in a recession or not; because of this I will be training and testing on two sections of the data, during a recession and an expansion. Table 1 summarizes the break up of the data. In the LSTM the validation data is inputed as a validation monitor into the model. This is a tool tensorflow offers which tests the validation data while training and stopping under certain conditions.

**Table 1:** Data Breakdown

	Expansion	Recession
Train	1/1/1948 - 1/1/2003	1/1/1948 - 1/1/1996
Validation	2/1/2003 - 9/1/2009	2/1/1996 - 11/1/2007
Test	10/1/2009 - 9/1/2016	12/1/2007 - 9/1/2009

## V. RESULTS

### i. LSTM

There are a many number of hyper-parameters to be tunned in an LSTM model. The parameters that were experimented with for this paper were number of layers, number of nodes on layers, drop-out probability, and time-steps. Time-steps is defined as the number of previous

observations fed into the model. For example time-steps is initially set at 10 meaning for an observation  $X_t$  the input will be  $X_{t-1} \dots X_{t-10}$ .

First the number layers was varied with the number of nodes set at 10 and no drop out. The mean squared error was used to evaluate performance. The model predicts one observation ahead. Table 2 summarizes the findings.

**Table 2:** MSE for Different Layer Sizes on the Test Data

Layers	Expansion	Recession
1	0.052	0.198
2	0.049	0.169
4	0.031	0.161

With 4 layers and no drop out, the number of nodes per layer was varied. First there is a constant layer size then a pyramid architecture and a reverse pyramid architecture Table 4 contains the findings.

**Table 3:** MSE for Different Layer Sizes on the Test Data

Nodes	Expansion	Recession
(10,10,10,10)	0.031	0.161
(10,7,5,2)	0.027	0.11
(2,7,5,10)	0.035	0.079

Next returning back to a single layer with 10 nodes and no drop out, the number of time steps was varied.

**Table 4:** MSE for Different Time Steps on the Test Data

Time-Steps	Expansion	Recession
2	0.023	0.143
5	0.071	0.067
15	0.034	0.029

Finally, drop out was tested. The probabilities of drop out that were tried were 25 % and 10 %. Both of these drop out probabilities significantly worsened the results. They both resulted in MSE over 1 for both expansion and recession data sets. This shows that over fitting was not an issue. This makes sense we are only training on around 500 observations which is not very many for an LSTM.

In the end the architecture that was chosen to be the best is 4 layers with nodes of (2,5,7,10), 10 times steps and no drop out. The final predicted results are show in figures 5 and 6 in the appendix.

## ii. ARIMA

Next was fitting the ARIMA model. Cross validation was used to select the appropriate  $p$ ,  $d$ , and  $q$ . The first two elements of the table are the most common ARIMA parameterizations for forecasting unemployment. Table 5 summarizes the findings.

**Table 5:** MSE for Different  $p$ ,  $d$  and  $q$

$ARIMA(p, d, q)$	Expansion	Recession
(1,1,0)	0.027	0.122
(2,1,2)	0.026	0.059
(1,1,1)	0.026	0.056
(3,1,1)	0.027	0.057

## iii. Discussion

When it came to expansionary periods the ARIMA model performed about as well as the LSTM model. During expansionary periods, the data is less volatile which may explain why both models perform about the same.

During times of recessions the LSTM performed slightly better than the ARIMA model. The best MSE that was achinved on this data with an LSTM model was .029 while for the ARIMA model it was 0.056 . This indicates that the LSTM model may perform better than the ARIMA model during more volatile times.

Both of these models were run on a CPU. The ARIMA model took at most around one minute to run while the LSTM model took around three minutes to run. Computational time isn't a big deciding factor when choosing between the two models.

What can be a significant factor in choosing between models is construction and hyper-parameters tunning. While tensorflow offers many tools to construct RNNs, it still took a

good amount of time to build the model. Opposed to ARIMA models which only took a few lines of code to build. Hyper-parameter tuning also took significantly longer for LSTM. Although the best LSTM performed better than ARIMA in the recession period, there existed some parameterizations of the LSTM model that performed significantly worst than ARIMA. Another benefit to ARIMA is the wealth of literature that surrounds the model. Parameters for the ARIMA model have been studied extensively so its easy to try out a few of the common approaches. For LSTM it was pretty much a guess and check game for all the parameters.

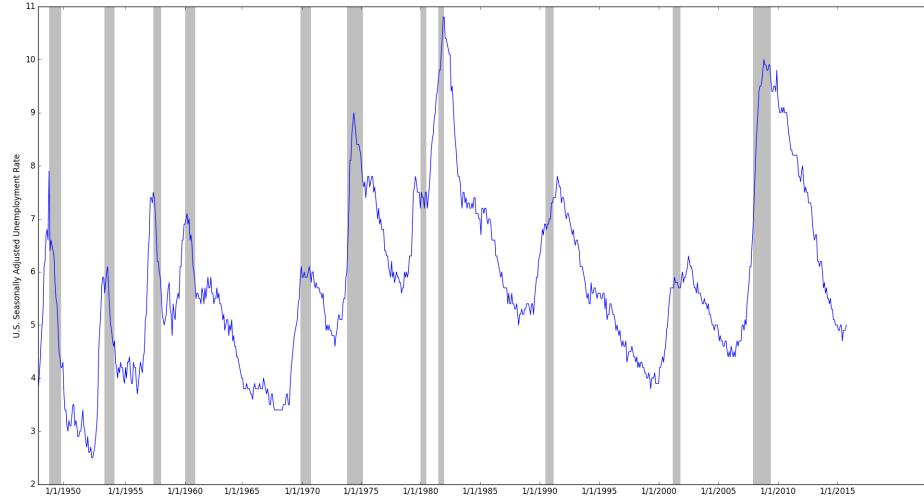
## VI. CONCLUSION

Neural networks have been used to forecast unemployment in previous research.[6] The author was unable to find any research into using LSTMs for unemployment. This paper showed that LSTM has the potential to outperform ARIMA and therefore should be explored further. Future research into this topic would include forecasting more than one time step ahead or experimenting with more parameterization of the LSTM.

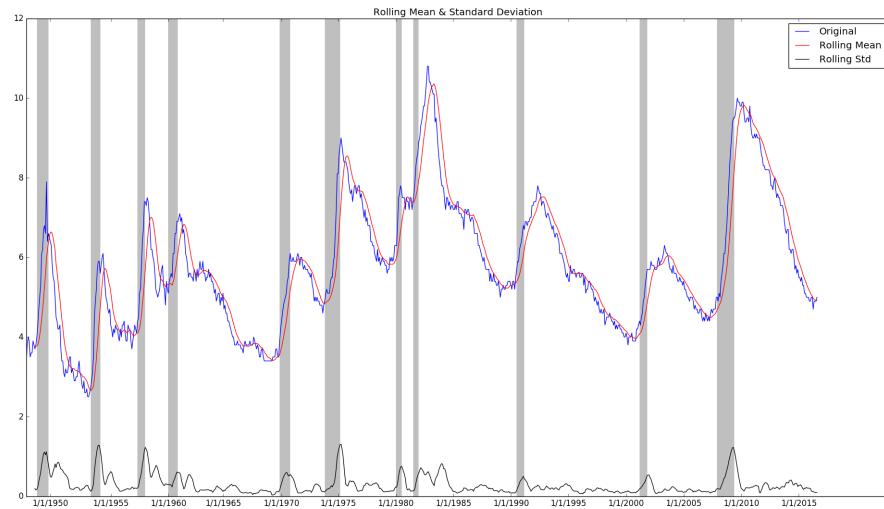
## REFERENCES

- [1] Veli Yilanci. *Are Unemployment Rates Nonstationary or Nonlinear? Evidence from 19 OECD Countries* . Economics Bulletin
- [2] Sims, C. A., and Todd, R. M. *Evaluating Bayesian Vector Autoregressive Forecasting Procedures for Macroeconomic Data*., paper presented at the NSF-NBER Seminar on Bayesian Inference in Econometrics and Statistics, St. Paul, April 1991.
- [3] Ion Dobre and Adriana AnaMaria Alexandru. *MODELLING UNEMPLOYMENT RATE USING BOX-JENKINS PROCEDURE* . Journal of Applied Quantitative methods.
- [4] Wojciech Zaremba and Oriol Vinyals and Ilya Sutskever *RECURRENT NEURAL NETWORK REGULARIZATION*. conference paper at ICLR 2015
- [5] Denis Kwiatkowski and Peter C.B. Phillips and Peter Schmidt and Yongcheol Shin *Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?* . Journal of Econometrics 1992
- [6] Geraint Johnes *Forecasting unemployment*. Applied Economics Letters

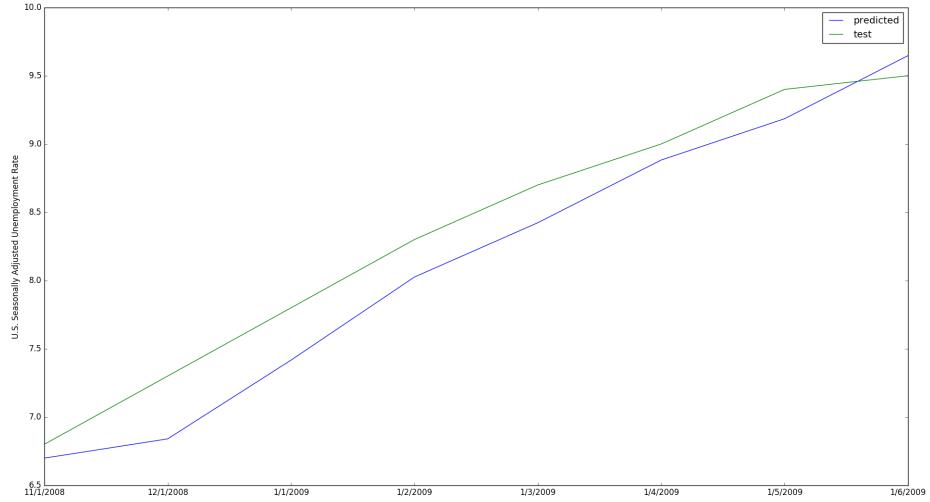
## VII. APPENDIX



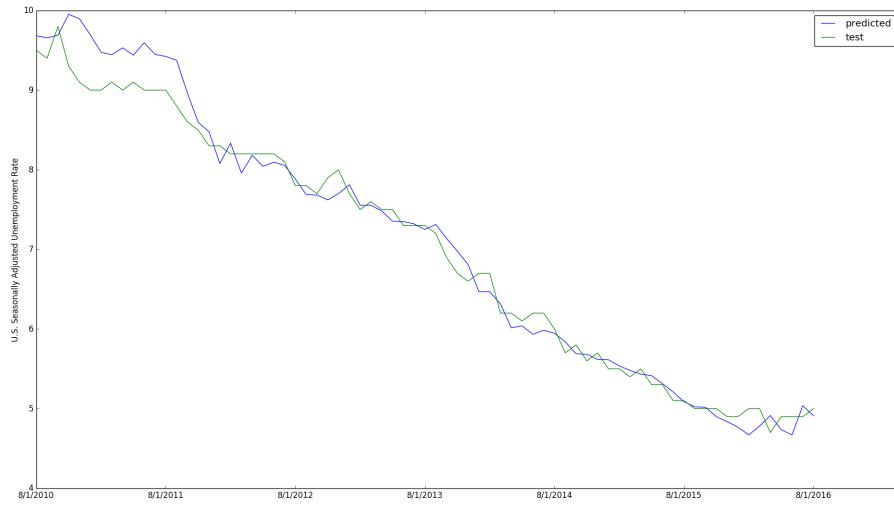
**Figure 3:** The U.S unemployment rate from 1/1/1948 - 9/1/2016. The grey bars are recessions.



**Figure 4:** The U.S unemployment rate from 1/1/1948 - 9/1/2016, with its rolling mean and standard deviation. The window for the rolling statistics is 10. The grey bars are recessions.



**Figure 5:** The U.S unemployment rate from 11/1/2008 - 1/6/2009 vs its predicted value by the LSTM model. This was during the great recession of 2007 - 2009



**Figure 6:** The U.S unemployment rate from 8/1/2010 - 8/1/2016 vs its predicted value by the LSTM model. This was during a period of expansion in the U.S. economy.