# Machine Learning Engineer  Nanodegree

## Capstone Project - Macroeconomic Indicator Forecasting with LSTM

**Meng Hsuan Tsai 2020/10**

## I. Introduction

The problem of  macroeconomic forecasting has been widely researched both before and after the popularization of machine learning. The macroeconomic indicator is related to financial risk and its prediction accuracy is very important in portfolio optimization. Time-series models are very popular given the time dimension inherent to trading. Key applications include the prediction of inflation and volatility, as well as the identification of co-movements of asset price series. Current methods are capable of generating relatively accurate forecasts of macroeconomic indicators. But these approaches bring a variety of undesirable properties, ranging from high sensitivity to model specification to high data requirements. There is considerable room to advance the state of macroeconomic forecasting by leveraging recent advancements in machine learning and neural networks.

The methods of estimation and forecast of macroeconomic indicator have been intensively studied (see, e.g., the references in Robert and Eric (2009) and in Anna(2018); Philippe(2019). Imputing economic fundamentals into volatility models pays off in terms of long horizon forecasting. Robert and Eric demonstrate macroeconomic fundamentals play a significant role even at short horizons. Anna examined machine learning techniques can be used to efficiently forecast macroeconomic time series. Artificial neural networks outperform a linear autoregressive (AR) and a random walk (RW) models in forecasting the monthly US CPI inflation. While current forecasting literature has focused on matching specific variables and horizons with a particularly successful algorithm, Philippe focuses on a wide range of horizons and variables and learn about the usefulness of the underlying features driving ML gains over standard macro-econometric methods

In this paper, I apply exploratory data analysis of the dataset used by Federal Reserve Economic Research. I also explore possible data augmentation and filtering, then begin with model implementation of LSTM. Additionally, our approach provides a number of advantages that lay the groundwork for developing new and increasingly accurate forecasts of economic indicators. The approach provides good single-series performance and can incorporate novel data if desired. Test results are discussed at the end along with possible future research.

## II. Problem Statement

Each datapoint includes a target variable, corresponding to Industrial Production: Manufacturing and Consumer Sentiment by the end of the sampling period. The problem of interest is whether macroeconomic indicator can be accurately predicted ahead of time. In other words, can the target variable generate time series prediction based on the feature set.

## III. Metrics

- Mean Absolute Error

For this task, I will be focusing on mean absolute error as evaluation metrics. From an investor's perspective, macroeconomic indicator prediction is most valuable for avoiding portfolio losses, which would be accomplished by implementing risk control before sudden volatility. Summarizing and assessing the quality of a machine learning model can be measure through mean absolute error. We do a subtraction of Predicted value from Actual Value as: Prediction Error → Actual Value - Predicted Value

This prediction error is taking for each record after which we convert all error to positive. This is achieved by taking Absolute value for each error as : Absolute Error → |Prediction Error|

Finally we calculate the mean for all recorded absolute errors (Average sum of all absolute errors).

MAE = Average of All absolute errors

$$mae = \frac{\sum_{i=1}^{n} abs\left(y_i - \lambda(x_i)\right)}{n}$$

The beauty of the MAE is that all of the errors will be weighted on the same linear scale. Thus, unlike the Mean Squared Error, we won't be putting too much weight on our outliers and our loss function provides a generic and even measure of how well our model is performing. For our time series datasets, we focus more on trends and do not focus on outliers which may be sudden impact from other economic circumstances.

- Recurrent Neural Network

The neural network architecture is mainly inspired by Kim & Kang (2010), who use a single hidden layer model with nodes roughly doubling the feature count. They also find that more complicated boosted models do not noticeably improve performance. The major innovation of RNN is that each output is a function of both previous output and new data. As a result, RNN gain the ability to incorporate information on previous observations into the computation it performs on a new feature vector, effectively creating a model with memory. This recurrent formulation enables parameter sharing across a much deeper computational graph that includes cycles. Prominent architectures include Long Short-Term Memory (LSTM) that aim to overcome the challenge of vanishing gradients associated with learning long-range dependencies, where errors need to be propagated over many connections.

- Long-Short Term Memory

RNNs with an LSTM architecture have more complex units that maintain an internal state and contain gates to keep track of dependencies between elements of the input sequence and regulate the cell's state accordingly. These gates recurrently connect to each other instead of the usual hidden units we encountered above. They aim to address the problem of vanishing and exploding gradients by letting gradients pass through unchanged.
A typical LSTM unit combines four parameterized layers that interact with each other and the cell state by transforming and passing along vectors. These layers usually involve an input gate, an output gate, and a forget gate, but there are variations that may have additional gates or lack some of these mechanisms.

## IV. Data Analysis

The dataset is a collection of financial data on Federal Reserve's data service.Using a single time series of monthly data on Industrial Production: Manufacturing (NAICS) and adding a monthly time series on University of Michigan: Consumer Sentiment(UMCSENT), both provided by the Federal Reserve's data service online. We will use the pandas-datareader library to retrieve data from 1980-01-01 through 2020-08-01 with total 536 entries.

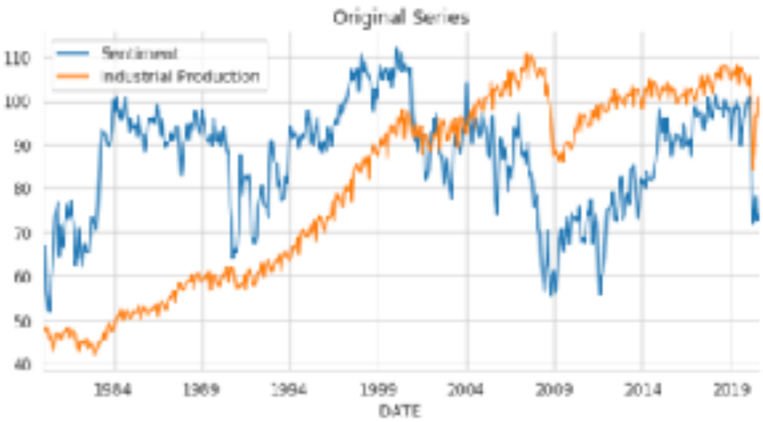

Data specification is as below:

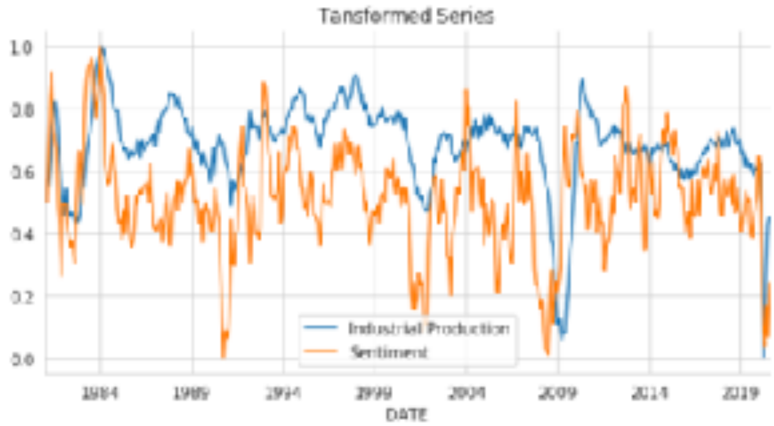
| DATE | sentiment | ip |
|---|---|---|
| 1980-01-01 | 67.0 | 46.8770 |
| 1980-02-01 | 66.9 | 47.9757 |
| 1980-03-01 | 56.5 | 48.4793 |
| 1980-04-01 | 52.7 | 47.0662 |
| 1980-05-01 | 51.7 | 45.6995 |

# V. Methodology

- Data Preprocessing

By checking with non-null data inputs with data type float, we apply data transformation as Log-transforming the industrial production series and seasonal differencing using lag 12 of both series yields stationary results to achieve stationarity. Also scaling the transformed data to the [0,1] interval to achieve the effects are proportionally take into considerations.




After reshaping to get non-overlapping series, we transform a dataset of several time series into the shape required by the Keras RNN layers. Using window size of 24 months and obtain the desired inputs for our RNN model, we split our data into a train and a test set, using the last 24 months to test the out-of-sample performance.

- LSTM Model Architecture

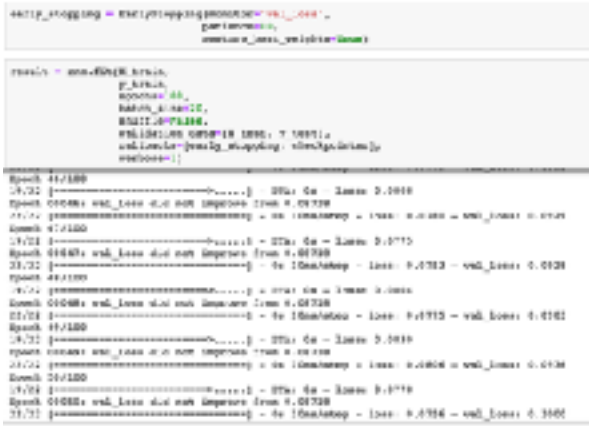
We use an architecture with two stacked LSTM layers with 12 and 6 units, respectively, followed by a fully-connected layer with 10 units. The output layer has two units, one for each time series. We compile them using mean absolute loss and the recommended RMSProp optimizer, as follows:

```
rnn = Sequential([
    LSTM(units=lstm_units,
         dropout=.1,
         recurrent_dropout=.1,
         input_shape=(window_size, n_features), name='LSTM',
         return_sequences=False),
    Dense(dense_units, name='FC'),
    Dense(output_size, name='Output')
])
```

```
rnn.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM (LSTM)                  (None, 12)                720
_________________________________________________________________
FC (Dense)                   (None, 6)                 78
_________________________________________________________________
Output (Dense)               (None, 2)                 14
=================================================================
Total params: 812
Trainable params: 812
Non-trainable params: 0
```

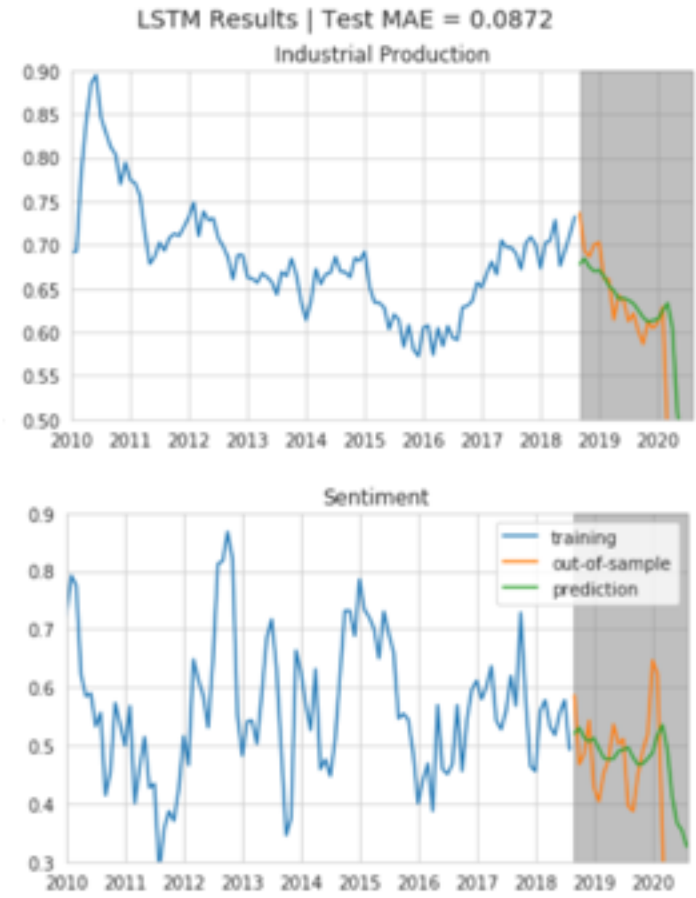
We train for 50 epochs with a batch size value of 20 using early stopping as below.



## VI. Results & Conclusion

- Model Evaluation and Validation

Training stops early after 22 epochs as below with test MAE 0.0872

- Conclusion

The model presented here use a novel technique for forecasting Industrial Production: Manufacturing and Consumer Sentiment. The model architecture provided competitive long-term forecasting performance. The implementation of neural network models, however, are not without practical challenges. First, a number of degrees of freedom exist in form of algorithm parameters the calibration of which may be highly costly in terms of computational capacity. Furthermore, forecast performances are sensitive to training and test split specifications and generally machine learning algorithms are strongly driven by the underlying data. Third, and relatedly, it is easier to train networks on data that is scaled to (-1,1) or (0,1). It is more difficult with non-stationary metrics such as nominal gross domestic product.

There are a few ways that we expect to improve model performance as we continue to develop this project. First, we anticipate that the inclusion of additional information as model inputs will improve model considerably. In this paper we have deliberately restricted ourselves to the use of a single series as the basis for model inputs. Other macro-indicators as model inputs should provide more information for model forecasts. Second, deep learning models are undergoing rapid development, with new techniques published every week. As new techniques emerge, we expect that we will be able to integrate them into the networks presented above to further improve performance.

## VII. Works Cited:

Robert F. Engle, Eric Ghysels, Bumjean Sohn(2009). Stock Market Volatility and Macroeconomic Fundamentals

Anna Almosova, Niek Andresen(2018). Forecasting Inflation with Recurrent Neural Networks

Philippe Goulet Coulombe1, Maxime Leroux, Dalibor Stevanovic, Stéphane Surprenant(2019). How is Machine Learning Useful for Macroeconomic Forecasting

Christopher Olah(2015) Understanding LSTM Networks

Rafal Jozefowicz, Ilya Sutskever(2015) An Empirical Exploration of Recurrent Network Architectures

Board of Governors of the Federal Reserve System (US), Industrial Production: Manufacturing (NAICS) [IPGMFN], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/IPGMFN, October 24, 2020.

University of Michigan, University of Michigan: Consumer Sentiment [UMCSENT], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/UMCSENT, October 24, 2020.

Working with RNNs(https://keras.io/guides/working_with_rnns/)

tf.keras.layers.LSTM(https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM)

Choosing the right estimator(https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html)

A Comprehensive Guide to Data Exploration(https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/)