Julian Wiley

CSCI E-86 Deep Learning

12/13/22

Final Project – Case Study in Economics – Predicting Economic Recessions using Macroeconomic Indicators and Gold Prices

# Abstract

Predicting economic recessions is one of the most important applications of machine learning for economics. Recessions are major events that have significant implications for all manner of economic policies and business decisions. Currently, making predictions about future economic conditions is done by economists using traditional statistical models. This approach is slow and subjective, as experts can disagree until well into the start of a recession. This project aims to use machine learning models to predict future economic conditions quickly so that policymakers and corporate actors can begin making plans to mitigate the consequences of these shifts as early as possible.

The project lifecycle is relatively straight forward, with the workflow closely adhering to the basic machine learning project skeleton. We begin by downloading each indicator separately using the Fred API and yfinance libraries. We then explore each variable before joining them into a single data frame. After the variables are joined, we create the univariate and multivariate feature and label sets, reshaping the dataset into the format required by RNNs. Both datasets are shaped such that the models will be making predictions based on the previous 60 days of data. After, we build and train the univariate and multivariate LSTM, GRU, autoencoder, and CNN models. Our results show that the univariate LSTM has the best performance, with its predictions most closely resembling the test set.

Given the structure of our dataset, we chose deep learning models that perform well in time series forecasting. We implement each model on both univariate and multivariate datasets, first using just the probability that we are in a recession and then using the full dataset. The first type of model that we implement is an RNN, building several different LSTM variations. Secondly, we built a variational autoencoder (VAE) on both datasets that performed poorly. Lastly (too chunky), we built out convolutional neural networks (CNNs), which also performed quite poorly.

* Two minute (short): https://youtu.be/50ws4F889gU
* 15 minutes (long): https://youtu.be/-9HwrHZVedg