Forecasting COVID-19 Cases with Neural Network and LSTM Models

Andy Chen, Boo Fullwood, Yunzhou Liu

May 05, 2020

Introduction

The COVID-19 (or coronavirus) pandemic has forced the widespread deployment of lockdowns, stay-at-home orders, and other social distancing measures to try and contain its spread, while policymakers and health researchers have been interested in forecasting the spread of the disease in order to help prepare and combat against it. This final project attempts to use machine learning techniques, specifically standard feed-forward and LSTM neural networks, to forecast future coronavirus cases in the United States for certain counties and across the state level in aggregate, based on data for past cases and deaths.

Related Work

A LSTM neural network addresses a particular shortcoming of traditional neural networks: an inability to learn long-term dependencies in the network. As a result, they are well-suited for modeling time series data, such as the number of cases per day in an epidemic. The coronavirus pandemic arrived only recently in the United States, but past work on applying feed-forward and LSTM neural network models to epidemic forecasting do exist in the literature. Wang et al. (2019) used multiple models, including LSTM neural networks, to model an HIV epidemic in Guangxi, China, finding evidence for its effectiveness over other time-series models. Mussumeci and Coelho (2020) used LSTM in comparison to LASSO and Random Forest regression to forecast weekly incidence of Dengue fever in Brazil, again finding evidence for LSTM's effectiveness. With regards to forecasting the coronavirus pandemic itself, Tomar and Gupta (2020) used an LSTM model to forecast coronavirus cases in India with somewhat decent results. Yang et al. (2020) found that an LSTM model trained on 2003 SARS epidemic data produced an incidence curve that fit surprisingly well to the real one. In contrast, Punn et al. (2020) predicted global cases over 10 days using only cases, deaths, and recovered data with both a standard deep neural network and an LSTM model, but found in comparison that polynomial regression was superior in the end. Nevertheless, these studies indicate potential for using an LSTM neural network for coronavirus cases forecasting.

Even non-LSTM models, however, can still demonstrate good prediction accuracy. A multilayer convolutional neural network using multiple inputs for cases and deaths forecasted the total number of confirmed cases in various Chinese cities with decent accuracy (Huang et al., 2020). As such, CNNs would also be good to try for predicting total cases.

Methodology

Dataset: We use COVID-19 U.S. cases and deaths data from the Center for Systems Science and Engineering at Johns Hopkins University, already organized by state and county upon download. After a few explorations for different models, we decided that the following data transformations should be applied for COVID-19 confirmed and death cases:

1. Since there are too many counties in the dataset, we cannot create one-hot encodings for each of the counties, and thus we tried to encode different counties by numbers, and view them as discrete quantitative variables.

2. Create past records for each county. For each past day, the record is added as a new column to the dataframe.

3. Normalization of the records. For each county, we calculate their normalized cumulative confirmed cases by $\frac{x-\mu_x}{\sigma x}$.

Originally, we also wanted to use Google Community Mobility data, obtained using Google's BigQuery API, as additional inputs into our neural networks. Since social distancing, if executed properly, can greatly affect the rate of spread of the coronavirus in an area, mobility data showing how much people tend to visit certain locations would in theory serve as a proxy for the level of social distancing in an area, and help predict the number of future cases. However, we ran out of time to properly implement these additional inputs.

Data Structure

Initial tests were done with five to ten days of lookback on case or death counts and the mobility data for the day prior to the desired prediction. In particular, we try to predict everydays' confirmed cases for each county by their past 5 (or 10) days cumulative confirmed cases, along with their encoded location. The model created has two densely connected hidden layers and one output layer, utilizing ReLU as the activation function. Since there are too many counties, we only chose counties in ten states, including North Carolina, New York, California, and Washington, as training sets, and counties in Illinois, Texas and Nevada as testing sets. And since we also noticed that though we already took a subset of the entire dataset, running the entire 1000 iterations for learning still is too time consuming, so we decided to take an alternative, and instruct the model to stop learning if it observes that the loss cannot be further improved for 15 consecutive iterations. For the error function, we used root mean-squared error (RMSE) of the predicted cases. Our predictions were for the next day.

Results

Model	MSE	MAE
CNN	0.042	0.002
LSTM (Single)	0.036	0.002
LSTM (Stacked)	0.031	0.001

Table 1: Model Loss and Error Values

Basic Neural Net

Convolutional Neural Net

The CNN model performed significantly better than expected for a simple model. It achieved comparable test loss results as the more complex LSTM models below and produced reasonable predictions out nearly 20 days from the last trained data point. The model was mostly linear, which matched the nature of the training data well, though it did diverge slightly from the test data as the result of moderate non-linear fluctuations.

LSTM Nerual Net

The single layer LSTM model achieved moderate predictive performance. With data scaled to the range $[0, 1]$, the models predictive power declined more rapidly from the last trained data point than the other models. This resulted in an overall lower performance on the test data. The stacked LSTM model gave markedly better performance, maintaining low error even at high distances from the training point. However, both LSTM models demonstrated an interesting pattern in that both produced a model that decreases in growth rate as the days progress. This may be the influence of the memory capability of the LSTM layers, as the CNN did not produce this effect. It is this curve that primarily contributes to the single layer LSTM model's difficulty maintaining accuracy.

Extensions

Although predicting the number of cases on the next day can still be useful, a more relevant metric to train our model on would be the accuracy of the predictions for multiple days ahead.

Additionally, major limitations exist on the in-person testing that generated the coronavirus cases data used, as different states and counties may have had different levels of access to testing kits or laboratories to process test results, or may have had differences in how or when cases were reported. The Google Mobility dataset also was collected from users that opted into a certain program, restricting how representative it may be. The hope of this project was to nevertheless produce a neural network model that could predict future coronavirus cases with some degree of accuracy in spite of the noise in the data.

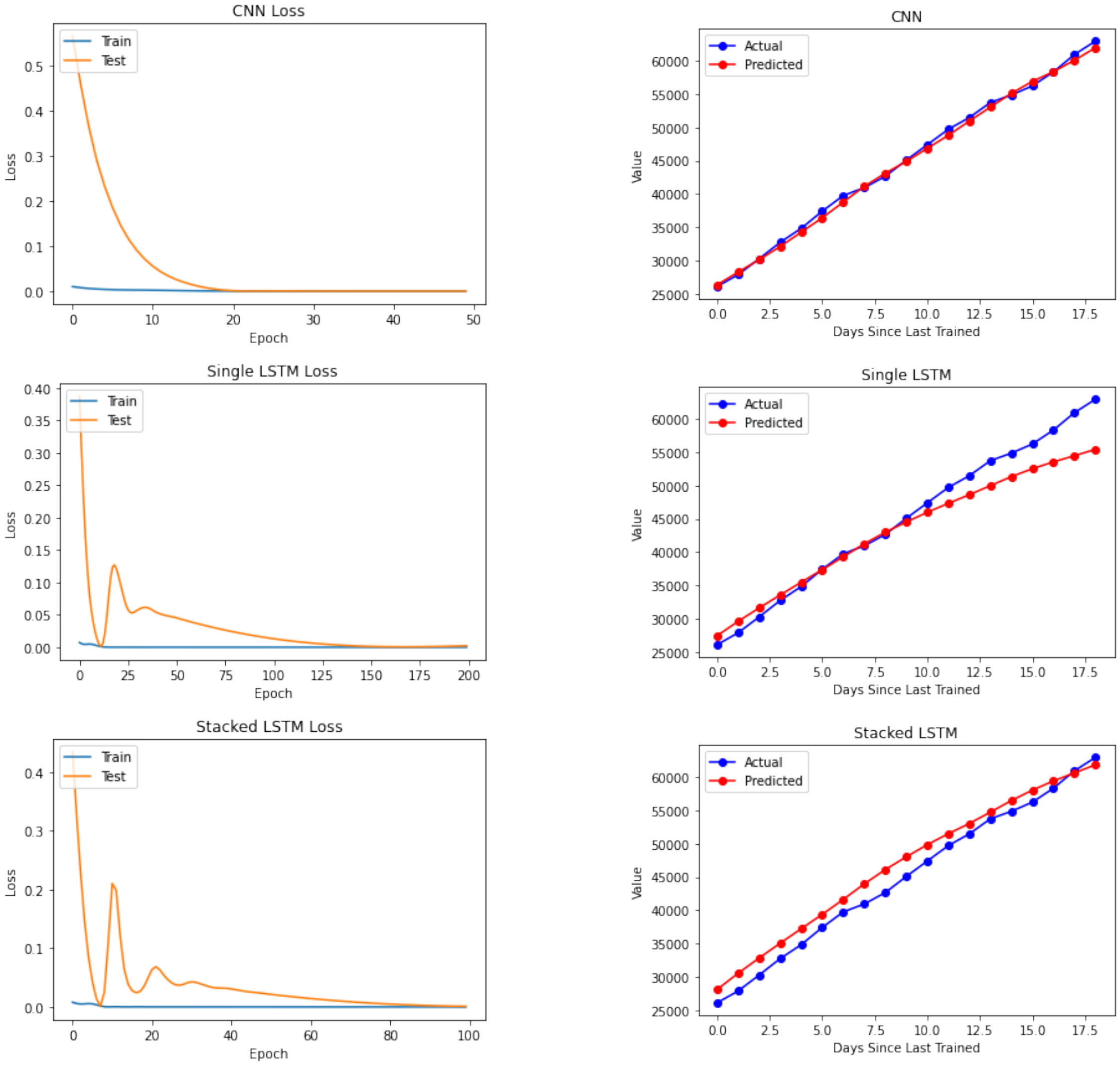

Figure 1: Model Training and Predictive Performance