Yixiang Zhang & Zhaolong Wei

Background

We propose to develop and deploy a deep neural network (DNN) model to predict the daily number of Covid cases for a given region or country based on historical data and other relevant factors. The goal of this project is to provide accurate and timely predictions that can help public health officials and policymakers make informed decisions about resource allocation, risk management, and disease control strategies. To achieve this goal, we will collect and preprocess a large-scale dataset of Covid cases and related variables, such as wearing the masks or not, how long they spend in the restaurants and shops, whether they use the public transit or not, and so on. We will then design and train a DNN model that can learn and generalize from this data, using appropriate architectures and optimization techniques, such as LSTM, GRU, or Transformer, and hyper-parameter values. We will evaluate and validate the performance of the trained model on independent data, using appropriate metrics and statistical tests, such as root mean squared error or mean absolute percentage error, and interpret and visualize the results to gain insights into the underlying patterns and relationships.

Software packages:

1. Pytorch (torch): to build and train the DNN model.
2. Numpy : to process the data.
3. Matplotlib: to visualize the data and the train results.

Milestone Goals

1. **Get the data**: Collect and preprocess data from reliable sources, such as government or health organization websites, that provide daily Covid case counts for different regions or countries. Depending on the data sources, we may need to consider different methods for acquiring the data, such as web scraping, API calls, or downloading from online repositories. **Dataset is available at** https://www.kaggle.com/datasets/meirnizri/covid19-dataset.
2. **Clean and split the data**: Cleaning the data can be a time-consuming process, so it's important to plan accordingly and budget sufficient time for this task. We may need to consider using a script to ensure that our cleaning process can be easily repeated or modified if needed. Split the data into training, validation, and testing sets using appropriate strategies, such as time-based or random sampling, to ensure that the model is trained and evaluated on independent data.
3. **Get familiar with PyTorch and DNN lib**: PyTorch and DNN libraries offer many advanced features and techniques for deep learning. When working with PyTorch and DNN libraries, it's important to consider the computational requirements of our model and to ensure that we have sufficient resources (e.g., RAM, GPU) to train and evaluate the model.

4. **Plot some images to show some features of this dataset**: Depending on the nature of our dataset, we may want to consider different types of visualizations, such as scatter plots, histograms, or heatmaps.

Final project Goals

1. **Design and train a DNN model**: Design a DNN model architecture that takes into account the temporal nature of the data and can capture both short-term and long-term trends and patterns. You may want to consider using techniques like LSTM, GRU, or Transformer. Depending on the size and complexity of the dataset, training a DNN model can take a significant amount of time and computing resources. We may want to consider using distributed computing frameworks like Spark to speed up the training process. When training the model, we may want to consider techniques like transfer learning, fine-tuning, or regularization to improve performance and prevent overfitting.
2. **Setup hyper-parameters**: Hyper-parameters are the settings or values that control the behavior and performance of the DNN model. There are many different hyper-parameters to consider, such as learning rate, batch size, number of layers, and activation functions. It's important to carefully select and tune the hyper-parameters to optimize the performance of the model.
3. **Validation**: Validation is the process of evaluating the performance of the DNN model on a held-out dataset or validation set. Evaluate the performance of the trained model on the validation set using appropriate metrics and statistical tests, such as root mean squared error or mean absolute percentage error, to assess the accuracy and generalization of the model. If necessary, adjust the model architecture and hyper-parameters based on the validation results and retrain the model on the full training set.
4. **Testing**: Test the final model on the independent testing set to assess its performance and reliability in predicting Covid cases for unseen data.
5. **Evaluation**: Evaluation is the final step of the DNN model building process and involves a comprehensive assessment of the model's performance, strengths, weaknesses, and limitations.