Final Project: Nonlinear Regression Model and GPU Acceleration – COVID19
Hirokatsu (Hiro) Suzuki

Introduction

SARS-CoV-2, the virus that causes COVID-19 is not the deadliest disease in existence, but it is one of the viruses that brought a huge impact to our daily life. The goal of this project is to forecast the COVID-19 cases with one of the nonlinear regression methods, Random Forest regression, and use GPU server to accelerate the program runtime.

The importance of exploring the regression model and GPU acceleration is to deliver the predictions faster and more accurately. The prediction of COVID-19 cases has been one of the hottest research topics in the past years. However, the study is still extending as there are more factors, such as new variant and effectiveness of the public health interventions, revealing to the researchers. In order to avoid a massive pandemic, researchers need a powerful computing method, such as parallel computing, to perform regression analysis and constructing forecasting model.

Regression models are widely used in epidemiology modeling. Nonlinear regression is one of the regression forms that fits the data into a model and representing as a function. For instance, a nonlinear model can be used to explain the crop and soil processes in agricultural research. There are many types of mathematical expression discussed in Archontoulis's paper [1]. The general nonlinear regression model has a form of $y = f(x, \theta) + \varepsilon$, where y is the output, f is the function, x is the input variable, θ is the parameters, and ε is the error term. Similar approach can be applied to model the COVID-19 cases. This project used the Random Forest regression model to forecast the COVID-19 pandemic with Python.

Random Forest (RF) regression uses multiply decision trees to create prediction models and merges models to construct one model that is more accurate and stable [2]. A simple algorithm is shown below.

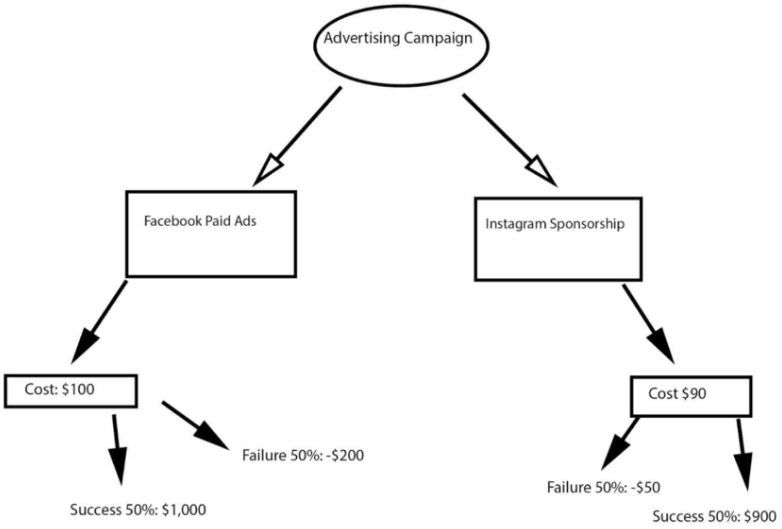


Figure.1 Algorithm of a Random Forest model

Furthermore, one of the Graphics Processing Unit (GPU) servers is used to accelerate the program to study the importance of computing acceleration. While Central Processing Unit (CPU) has multiple powerful cores and the ability of serial processing, GPU possess hundreds of thousands of weaker cores with parallel processing ability. Parallel processing is now a standard computing environment for a high-performance computing [6]. It uses multiple weaker cores to processes tasks simultaneously. The last part of the project compares the runtime of the Python program in CPU and GPU environments with difference number of cores.

Random Forest Regression

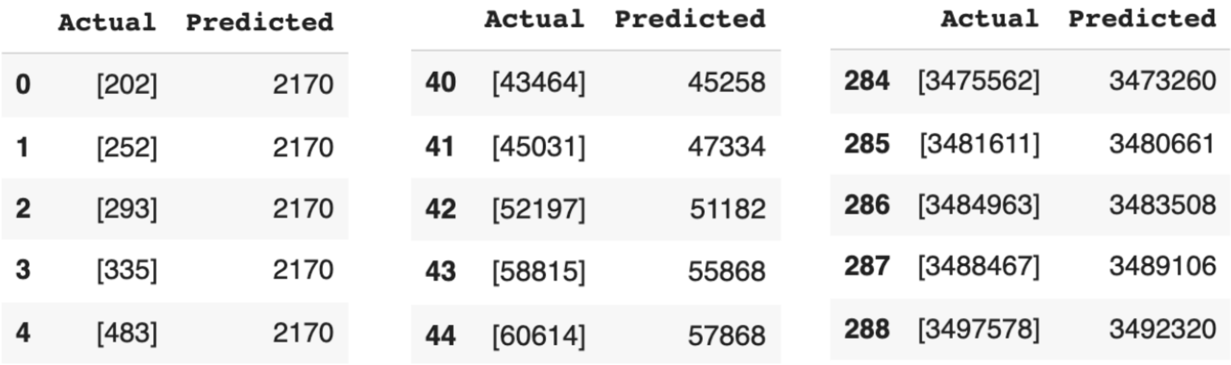
COVID data was collected from Centers for Disease Control and Prevention website. Only the data 03/12/2020 to 03/07/2021 in the state of California was used for the RF model. Since there are excessive data, only the 'date' and 'positive cases' are used to construct a RF model. The datetime was replaced with the number of days starting with 1.

Python is used to compute the regression model because external machine learning based packages are available on both CPU and GPU environments. The packages used to perform the regression are listed as: NumPy, Pandas, scikit-learn. NumPy and Pandas packages are used to read the file and formatting the dataset. Then the dataset is split into two parts with the split function, *train_test_split(parameters)*: 80% of data used to train the model, and 20% of data used to test the model. Lastly, the random forest regression function, *RandomForestRegressor(parameters)*, is used to build the model, and fit function, *regressor.fit(x, y)*, is used to train the model which x and y are the data values. Plotting of the model used matplotlib package.

GPU Acceleration

The GPU acceleration is performed by increasing the number of cores used for training the model. Within the regression function, one of the parameters, n_jobs, specify the number of cores used for the regressing process. Since there are only 4 cores on the CPU environment, same number of cores were used in the GPU environment. Once the code runs on a CPU environment, it is uploaded to one of the GPU servers.

Results: Modeling



	Actual	Predicted
0	[202]	2170
1	[252]	2170
2	[293]	2170
3	[335]	2170
4	[483]	2170

	Actual	Predicted
40	[43464]	45258
41	[45031]	47334
42	[52197]	51182
43	[58815]	55868
44	[60614]	57868

	Actual	Predicted
284	[3475562]	3473260
285	[3481611]	3480661
286	[3484963]	3483508
287	[3488467]	3489106
288	[3497578]	3492320

Figure.2 Comparing the predicted cases with real cases.

From the table above, the RF model showed some accuracy in forecasting the future cases. The model does not fit into the actual data as there are huge errors between the actual and predicted cases. However, looking at the fitting below, it is obvious that the model minimizes the error term as it predicts the more cases.

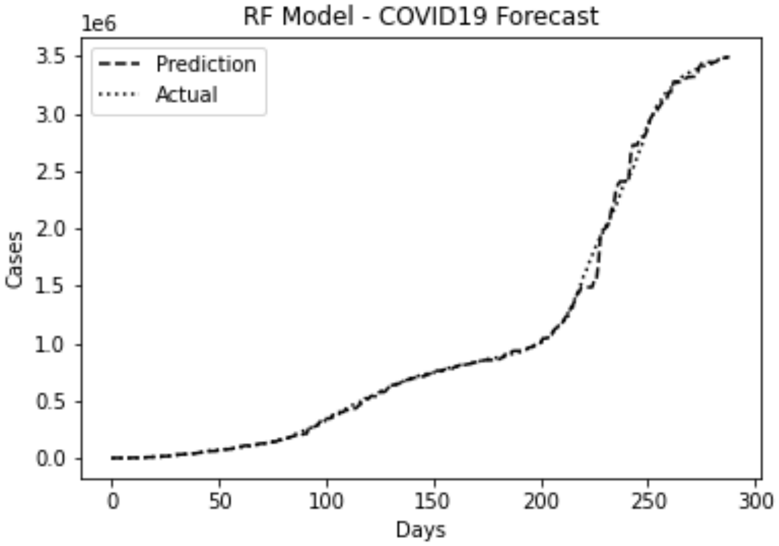


Figure.3 Comparing the RF model with actual data

Results: GPU Acceleration

Num. of core(s)	CPU (sec)	GPU (sec)	Comparison
1	13.729	9.377	1.5 faster
2	17.498	13.553	1.3 faster
3	17.901	14.099	1.3 faster
4	18.149	16.163	1.1 faster

Table.1 Comparing the runtime on CPU and GPU environments.

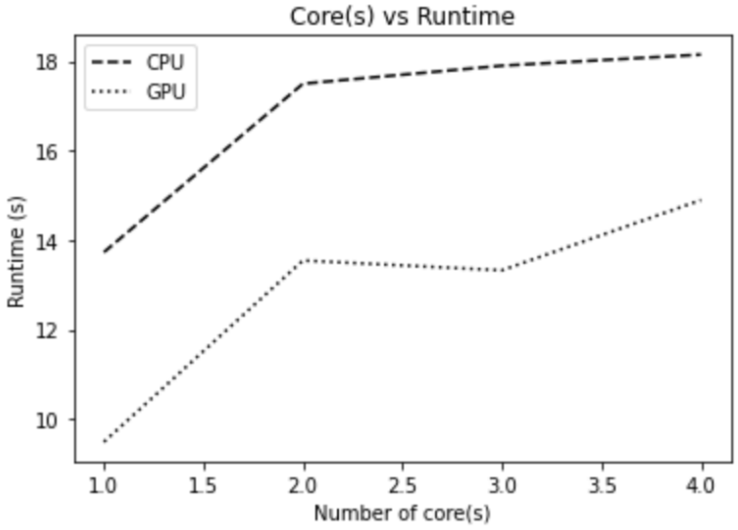


Figure.4 Comparing the runtime

From the table above, the code ends at difference runtimes with different number of cores on different environments. As shown in table 1, the runtime increases with the number of cores used. Nevertheless, the data shows that the GPU processes the code about 1.3 times faster than CPU. This is very unusual as GPU should process the code considerably faster than the CPU. This will be discussed in the next section.

Conclusion/Comparison/Evaluation

Overall, the objective of the project was reached supporting with the RF regression model and the faster runtime on a GPU environment. The model is only constructed to simulate the COVID-19 cases in about one year range. The analysis shows that there is a huge error in the model as the mean absolute error (MAE) is about 20918 which means the model is unlikely to be used in forecasting the COVID-19 cases in the future. One solution to improve the accuracy of the model is to increase the number of trees, or the number of estimators, in the RF model. The runtimes of the code on GPU environment somehow demonstrated the importance of parallel computing. The runtimes on GPU are not tremendously faster than on CPU for several reasons. One of them is the size of dataset. Since the GPU is meant to compute large number of tasks simultaneously, it may be more effective to use larger dataset constructing the RF model with a greater number of estimators. Another reason will be the GPU version of package used in the code. cuML is a GPU based implementations of scikit-learn package. The RF regression model may be optimized if such method is used.

References

[1] Sotirios V. A. and Fernando E. M. (2015). “Nonlinear Regression Models and Applications in Agricultural Research”. Agronomy Journal. https://doi.org/10.2134/agronj2012.0506

[2] Leo B. (2001). “Random Forests”. Machine Learning. https://doi.org/10.1023/A:1010933404324

[3] Segal, M. R. (2004). “Machine Learning Benchmarks and Random Forest Regression”. UCSF: Center for Bioinformatics and Molecular Biostatistics. https://escholarship.org/uc/item/35x3v9t4

[4] Harvey J. M. and Lennart A. R. (1987). “Fitting curves to data using nonlinear regression: a practical and nonmathematical review”. The FASEB Journal. https://doi.org/10.1096/fasebj.1.5.3315805

[5] Hengjian C. and Tao H. (2020). “Nonlinear regression in COVID-19 forecasting”. Scientia Sinica Mathematica. doi: 10.1360/SSM-2020-0055

[6] Guillem P. and Lei X. (2011). “GPU Computing in Medical Physics: A Review”. The International Journal of Medical Physics Research and Practice. https://doi.org/10.1118/1.3578605