Regression and Neural Networks for COVID-19 Prediction

Bin Zhang (5660329599), Hanzhi Zhang (4395561906) , Yihang Chen (6338254416)
University of Southern California
Viterbi School of Engineering
bzhang97@usc.edu, hanzhizh@usc.com, yihangch@usc.edu

Abstract

A database recording the COVD-19 confirmed cases and fatalities from different countries and regions. This report chooses confirmed cases data from four countries, China, the US, Spain, and Italy to use three algorithms including polynomial regression, logistic regression, and Neural Networks for the existing data mathematical fitting and prediction. To reflect the time continuity of training data, the whole data is split into small pieces, which contains a continuous day group. In this case, the algorithm is applied to these data pieces to figure out the relationship between these pieces. According to the comparison among these three methods, Neural Networks which can have over 90% fitting and reasonable prediction for future tendency.

Introduction

COVID-19 is caused by a coronavirus called SARS-Cov-2 [1]. This virus was firstly confirmed in China. At the beginning of the spread, due to the lack of data and the unknown transmission method of this virus, it is hard to make an accurate prediction.

Unfortunately, on the 5th of May 2020, more than 3 million people are infected which already leads to almost 260,000 deaths [2]. The World and Health Organization announced this new Coronavirus become a worldwide outbreak pandemic. Furthermore, the majority transmission of the COVID-19 virus is confirmed by respiratory droplets and contact routes (WHO). Under this circumstance, it is possible and necessary to build a mathematical model for forecasting which can help handle the epidemic.

In this project, a dataset records the confirmed cases and fatalities across the world from January 23rd January to 9th May. To check the accuracy of prediction, we use three methods to exam it. Firstly, the dataset is divided into two parts, 90% train sets and 10% test sets. Secondly, the whole dataset is treated as train sets. The predicted results are compared with actual data to estimate the accuracy. The final step is used by the trained model from the second method to predict the daily confirmed cases for a further 29 days.

Data preparation

Data used by this project is a dataset from Kaggle *COVID19 Global Forecasting(week5) competition* (https://www.kaggle.com/c/covid19-global-forecasting-week-5/data). The evaluation data for the competition comes from John Hopkins CSSE. In the train.csv data, we got both "Confirmed" and "Fatalities" data based on county or province scale if applicable (Table 1). So, the first step of data pre-processing is to split data of "Confirmed" and "Fatalities" into two columns and grouped by country. Then we split the "train.csv" into files based on country. Here we just keep the rows whose county and provinces are both "Nan" cause these data are the sum of the whole country on the certain day. Through the mentioned steps of data pre-processing, we divided the data into sub-files based on country (Table 2).

Id	**County**	**Country_Region**	**Population**	**Weight**	**Date**	**Target**	**TargetValue**
1		Afghanistan	27657145	0.058359	23/01/2020	ConfirmedCases	0
2		Afghanistan	27657145	0.583587	23/01/2020	Fatalities	0
3		Afghanistan	27657145	0.058359	24/01/2020	ConfirmedCases	0
4		Afghanistan	27657145	0.583587	24/01/2020	Fatalities	0
5		Afghanistan	27657145	0.058359	25/01/2020	ConfirmedCases	0
6		Afghanistan	27657145	0.583587	25/01/2020	Fatalities	0

Table 1 Original CSV File

	County	**Country_Region**	**Date**	**Target_x**	**TargetValue_x**	**Target_y**	**TargetValue_y**
0		Afghanistan	23/01/2020	ConfirmedCases	0	Fatalities	0
1		Afghanistan	24/01/2020	ConfirmedCases	0	Fatalities	0
2		Afghanistan	25/01/2020	ConfirmedCases	0	Fatalities	0
3		Afghanistan	26/01/2020	ConfirmedCases	0	Fatalities	0
4		Afghanistan	27/01/2020	ConfirmedCases	0	Fatalities	0
5		Afghanistan	28/01/2020	ConfirmedCases	0	Fatalities	0
6		Afghanistan	29/01/2020	ConfirmedCases	0	Fatalities	0

Table 2 Afghanistan Fatalities

After the import of table2, the rows including Country, Province_State, Country_state can be Dropped. Then two models for predict confirmed cases and fatalities should be built. In this project, the focus would be applied to confirmed cases. Therefore, a table containing two columns including Date and TergetValue_x is built (Table 3).

	Date	TargetValue_x
0	20200123	95
1	20200124	277
2	20200125	486
3	20200126	669
4	20200127	802
5	20200128	2632
6	20200129	578
7	20200130	2054

Table 3 Used DataFrame for data processing

For the logistic growth model, the independent variable is the time(date) and the dependent variable is the confirmed case at that date, since the logistic function is used, the independent variable is automatically generated by the order index and the dependent variable keeps the same. The only problem needs to be solved is what is the time of an epidemic outbreak. Therefore, we choose the start day in Table 3, 2020/01/23. The confirmed cases in the start day would be compared with cases from the further 6 days. If all 7 days have at least one case and the tendency of cases is increasing, this time would be considered as the epidemic outbreaking point. If not, the next date would be chosen and repeat the same statement processing until the satisfied date is found. If there is no date is satisfied, this country would be considered as a non-infected country. The assumed number of confirmed cases is the average value from the last 15 days. For the infected country, the whole input for this model is
X: a list of index number [x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}…]. X_i is the date, day i, that is considered as the epidemic outbreaking point
Y: a list of the number confirmed case derived from the processed data. [Y_i, Y_{i+1}, Y_{i+2}, Y_{i+3}, Y_{i+3} …]. Y_i is the confirmed cases on day i.

For the Neural Network and polynomial regression, both models cannot just use the input format of logistic regression. In this case, the timeline is divided into small pieces. Table 4 shows the data processing method. Continuous k days (yellow cell) is considered as a set of features to predict v days (red cell). In figure 1, the model can be called M (6,1). Figure 1 explains the principle of prediction. If Day 107 value is an unknown value, Day 107 can be predicted by the previous 6 days. Then the predicted value of Day 107 would be group by the previous 5 days' value to predict Day 108. Repeat this step until the required date is reached.

timeline	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9
	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7		
		Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	
			Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9

Table 4 Training Data format for Neural Network and Polynomial regression

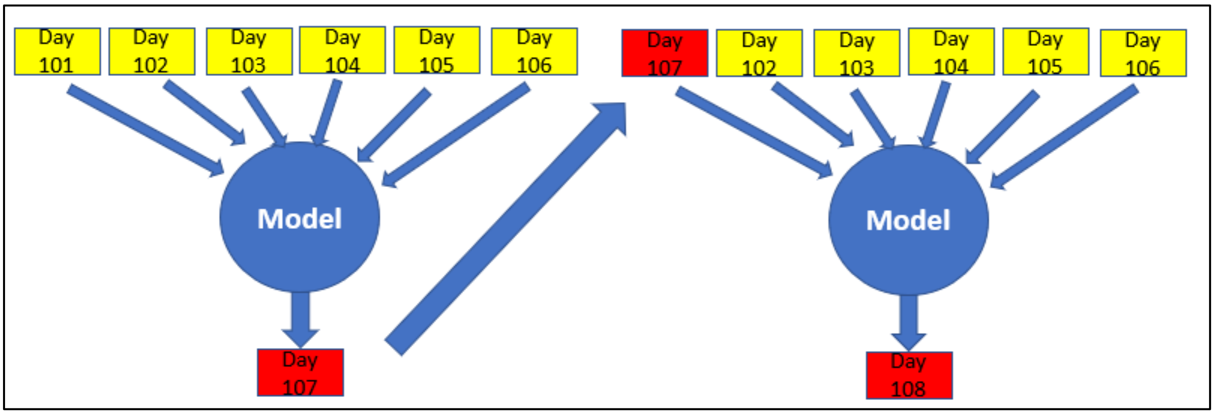


Figure 1 Prediction principle

For polynomial regression, it uses a model M (3,1) to predict. For Neural Network, it uses three types of model, M (7,1), M (15,3), M(30,7).

Regression

A. Polynomial regression

The basic pattern in machine learning is to use linear models trained on nonlinear functions. By adding data pre-processing layer in the pipeline, it can easily extend linear regressing to polynomial regression. The basic model of linear regression shows below:

$$y(w,x) = w_0 + w_1x_1 + w_2x_2 \quad (1)$$

Where,

w_0, w_1, w_2 are constant which choose randomly in the beginning.

x_1, x_2 are the value of training data in the x axis.

A polynomial transform that is showed below is used to the change linear regression to polynomial regression.

$$z = [x_1, x_2, x_1x_2, x_1^2, x_2^2] \quad (2)$$

The model equation would be the eq.3 which is the combination of eq.1 and eq.2

$$\mathrm{y}(w,x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2 \ (3)$$

That means the linear regression training model can still be used to train polynomial regression model.

In the experiment, a group of continuous 3 days is aggregated as the features to predict the next day data, which is the 4th day. The data is a 1x3 array for a single data point. And the y value is the 4th day's data. The degree of the polynomial transform is 2, so the new set of features would be

$$\mathrm{y}(w,x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1^2 + w_5x_2^2 + w_6x_3^2 + w_7x_1x_2 + w_8x_1x_3 + w_9x_2x_3 \ (4)$$

where,

x_1 is the date of day1.

x_2 is the date of day2.

x_3 is the date of day3.

The sum of the difference between the predicted value got from eq.4 and the actual value for each x_i is $\vec{w}$. The target of the regression is to find out the values of w_0, w_1, w_2 …. w_9, when $\vec{w}$ is the lowest, $E_{in}(\vec{w})$.

$$\begin{cases} E_{in}(\vec{w}) = \ y(w,x) - y_4 \\ \dfrac{\partial E_{in}(\vec{w})}{\partial w} = 0 \end{cases} \quad (5)$$

$$\begin{cases} \overrightarrow{w_{opt}} = (DD^T)D\vec{y} \\ D = (x_1 + x_2 + x_3) \ (6) \\ w = w{-}\propto \overrightarrow{w_{opt}} \end{cases}$$

Where,

y4 is the confirmed cases or fatalities on day4.

w is 1X9 matrix, (w_0, w_1, w_2 …. w_9);

Therefore, in the regression model, assign a random value to w_0, w_1, w_2 …. w_9. From the eq.5, we can get the eq.6. Use eq.6 to update the value of w for each iteration. Once the model reaches to the required times of iterations or the value of difference, $\vec{w}$, output the value of w.

In order to estimate the final accuracy of predicted results, a fitting score is used. The calculation of the fitting score is shown in eq.7

$$R^2 = 1 - \frac{\sum_i(\mathrm{y}_i(w,x) - \overline{y})^2}{\sum_i(y_i - \overline{y})^2} \ (7)$$

Where,

$\mathrm{y}_i(w,x)$ is the confirmed cases or fatalities obtaining the function from regression on day i;

y_i is the actual confirmed cases or fatalities on day I;

$\overline{y}$ is the actual average confirmed cases or fatalities;

In this project, the weights for each variable are learned by sklearn library which PolynomialFeatures is used for attribute building and LinearRegression is used for model fitting.

For polynomial regression, it is hard to predict the result of a long-time future. Thus, the other regression model, logistic regression would be introduced in the latter part.

B. Logistic curve

The Logistic function or Logistic curve is a common sigmoid function, which was named by Pierre François Verhulst when he studied relationship with population growth in 1844 or 1845. This model describes such an evolving process: The initial stage is roughly exponential growth; then as the process evolves, the growth becomes saturated and the increase slows down; finally, the increase stops when it reaches maturity. The function formula and graph are as follows:

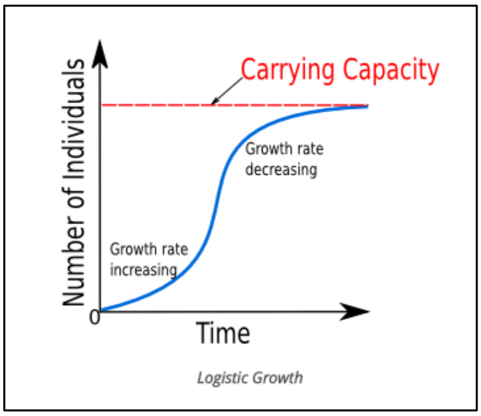


Figure 2 Logistic curve

$$P(t) = \frac{KP_0e^rt}{K + (P_0e^rt - 1)} \quad (8)$$

Where,

K is the maximum carrying capacity;

P_0 is the initial value.
R is the growth rate in this model, 0.3.
t is the date.
This regression model is like polynomial regression. Eq.5 and Eq.6 can be used to iterate to get the best value of K. K in eq.8 can be considered as w in eq.4. t in eq.8 can be considered as x in eq.4. The value of r in eq.8 is determined by r_0 in SEIR model. r_0 means the average number of people would be infected by patience. In the report, r_0 is 2~2.5 (say 2.1 in this project) [3]. The time to be admitted is 7 ± 4 days [4]. Therefore, r which means the growth rate of confirmed cases is 2.1/7 = 0.3.
The accuracy of this model is also evaluated by eq.7.
This regression model's main idea is to use the available data to fit an appropriate curve for the unknown independent variables.
Here the available data can be useful for seeking the best value of K and r. In this project, the parameters K and r are learned by the scipy library where curve_fit is used to get the appropriate parameters.
For the logistic curve, it only focuses on one variable time and ignores the dependency between each point, so it cannot predict future cases in an accurate way.

Neural Networks

In 1943, Warren MuCulloch (neuroscientist) and Walter Pitts (logician) proposed the first computational model of a neuron. Based on this model, the artificial neural network evolves today. MLP is a class of feedforward artificial neural network (ANN) which is helpful to solve the nonlinear problem. The MLP consists of three parts: an input layer, an output layer, and a number of hidden layers. Each perceptron between layers is fully connected. The structure of the MLP is as follows:

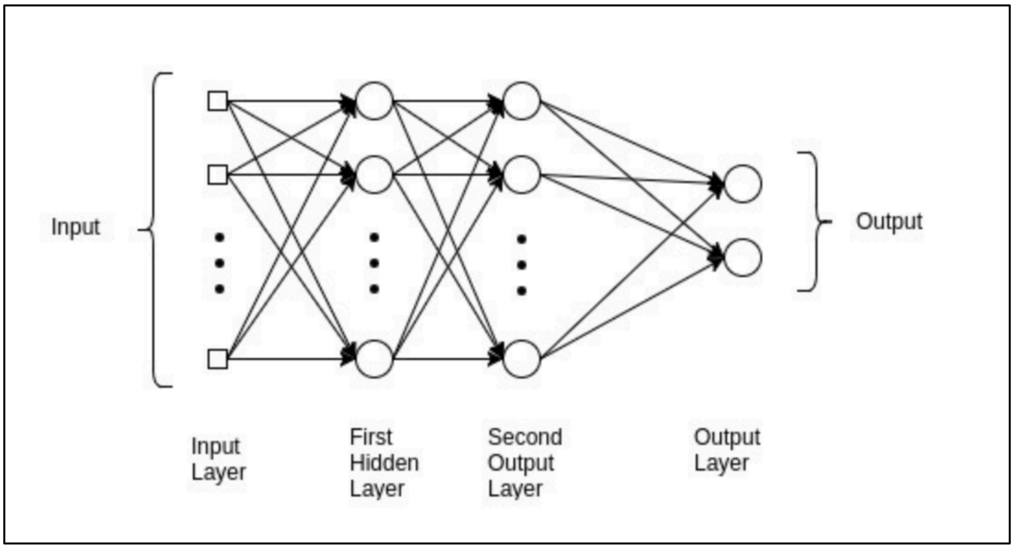


Figure 3 Overview of Neural Networks

For each single perceptron, it does two operations: linear function and activation:

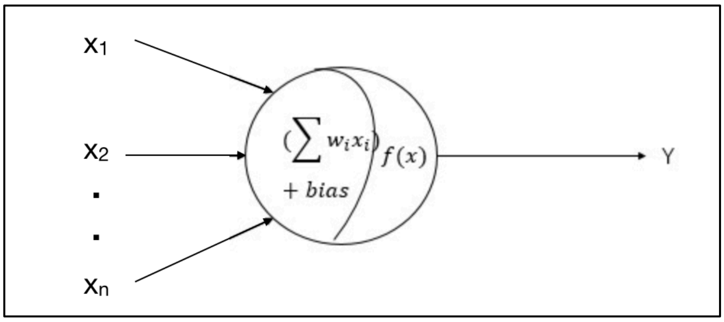


Figure 4 A Single Neural

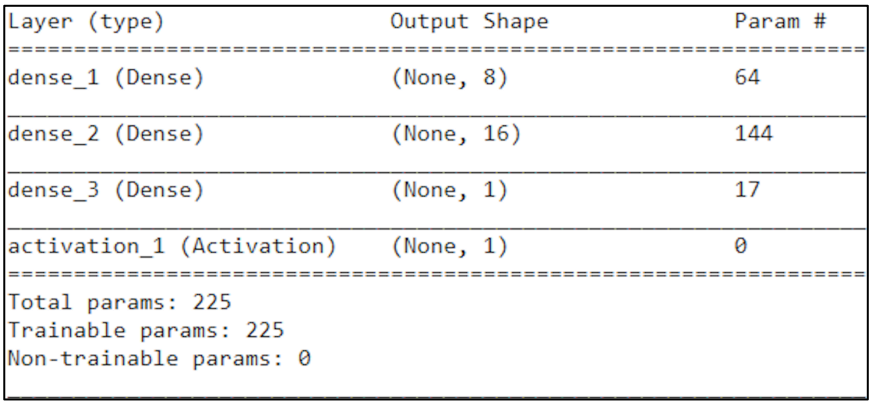
$$\delta = \sum_{i}^{n} x_i * w_i \ + bias \ (9)$$

$$a = f(\delta) \ (10)$$

In this project, considering that the good performance of the neural network in the regression problem, multilayer perceptron (MLP) is used to make predictions. Due to the amount of the data and in the case of overfitting, the model only has two hidden layers. For observing the time series prediction decencies, three MLP models are built. Each model has the same layer numbers (all of them have four layers) but different perceptron numbers in each layer.

Here are the details:

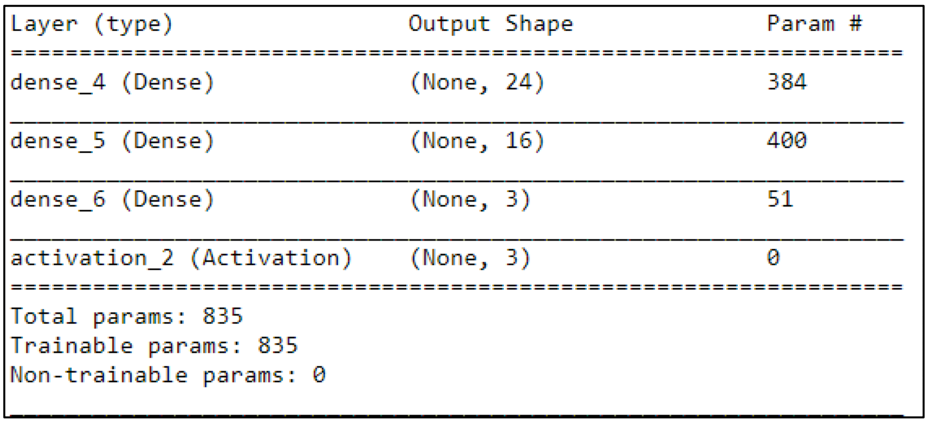
1. The first seven days predict the eighth day
   For each input, its shape is (7*1)
   For each input, its shape is (1,)



```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 8)                 64
_________________________________________________________________
dense_2 (Dense)              (None, 16)                144
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 17
_________________________________________________________________
activation_1 (Activation)    (None, 1)                 0
=================================================================
Total params: 225
Trainable params: 225
Non-trainable params: 0
_________________________________________________________________
```

Figure 5 model 1 structure

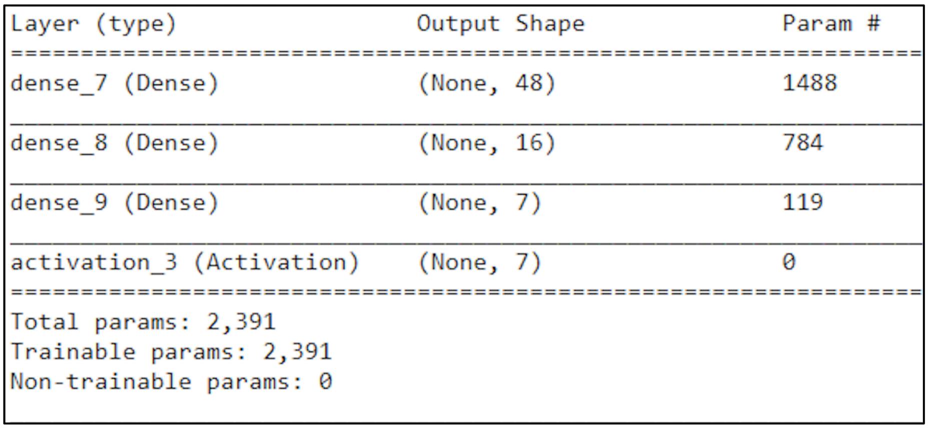
2. The first 15 days predict the later three days
   For each input, its shape is (15*1)
   For each input, its shape is (3,)



```
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 24)                384
_________________________________________________________________
dense_5 (Dense)              (None, 16)                400
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 51
_________________________________________________________________
activation_2 (Activation)    (None, 3)                 0
=================================================================
Total params: 835
Trainable params: 835
Non-trainable params: 0
_________________________________________________________________
```

Figure 6 model 2 structure

3. The first one month predict the later one week
   For each input, its shape is (30*1)
   For each input, its shape is (3,)



```
Layer (type)                 Output Shape              Param #
=================================================================
dense_7 (Dense)              (None, 48)                1488
_________________________________________________________________
dense_8 (Dense)              (None, 16)                784
_________________________________________________________________
dense_9 (Dense)              (None, 7)                 119
_________________________________________________________________
activation_3 (Activation)    (None, 7)                 0
=================================================================
Total params: 2,391
Trainable params: 2,391
Non-trainable params: 0
```

Figure7 model 3 structure

In this project, keras is used for model building and parameter learning, the activation for each layer is Rectified Linear Activation Function (ReLU).

Rectified Linear Activation Function (ReLU) is one kind of activation function, which is shown below:

$$f(x) = \max(0, x) \quad (11)$$

The ReLU is a one-side function, which is can simulate better of the action of biological activity. Using the ReLU as activation function, the whole network only has 50% activated hidden unites. Also, compared with sigmoid function, it has more efficiency in computation, which can accelerate the training progress.

For each model's learning process, using one data as a batch to update the parameters due to the lack of the amount of data, the optimization function is Root Mean Square Prop instead of gradient descent taught in the class. This optimization function in keras is used as optimizer='rmsprop'.

Since predicting case is a regression problem, the metric used here is mean absolute error (MAE) which is shown below:

$$MAE = \frac{\sum_{1}^{n} |h(y_i) - y_i|}{n} \ (12)$$

Results

A. Polynomial Regression:

In the experiment, the continuous 3 days cases are the features to predict the next day data, which is the 4th day. The data is a 1x3 array for a single data point. And the y value is the 4th day's data. The degree of the polynomial transform is 2, so the new set of features would be

$$y(w,x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1^2 + w_5x_2^2 + w_6x_3^2 + w_7x_1x_2 + w_8x_1x_3 + w_9x_2x_3 \ (13)$$

Here are the results for some typical countries:

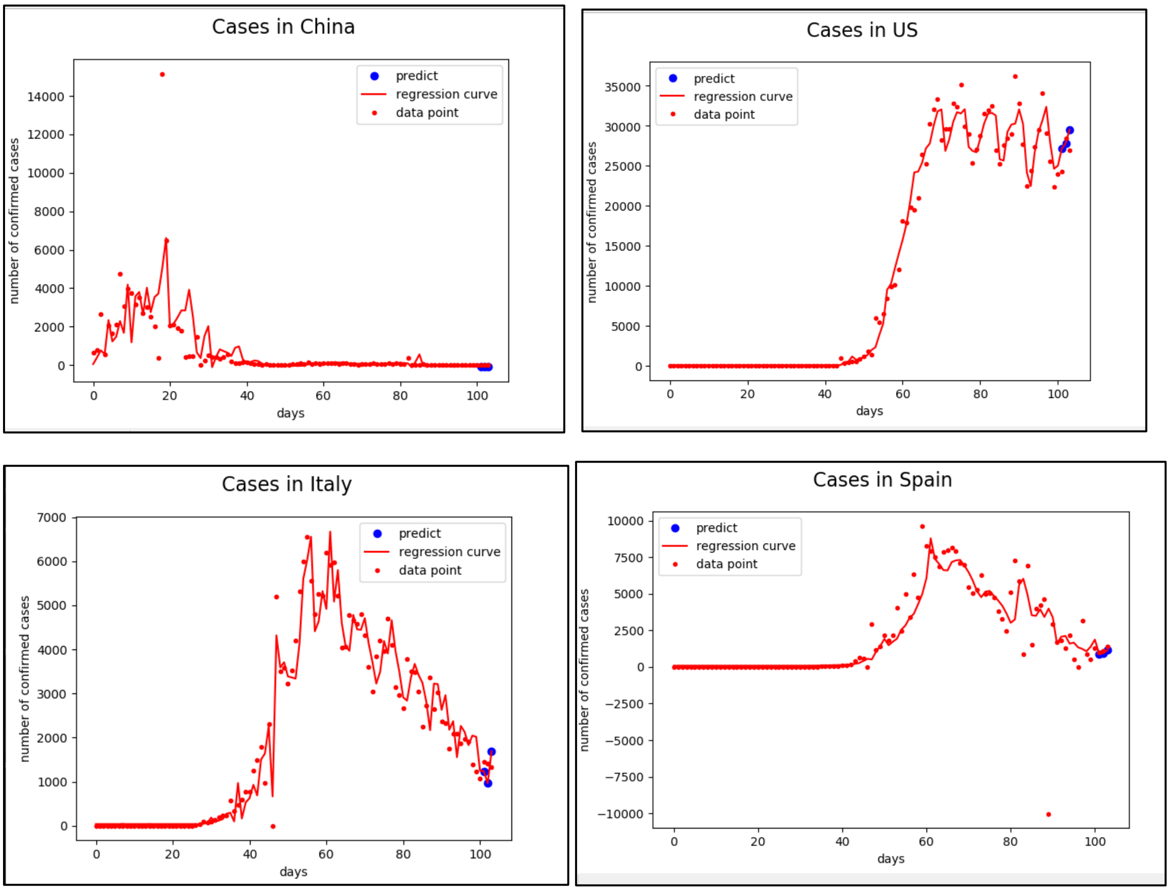


Figure 8 polynomial results in different countries

Now taking the U.S as an example:

The fitting score in the train set is 0.9866324548838642

The prediction upcoming 3 days is [27127.6355451 27801.08394533 29486.97998504]

The real upcoming 3 days is [24251. 28420. 26906.]

Then running the polynomial regression using continuous 7 days data and 15 days data for the U.S. data, the result is shown below

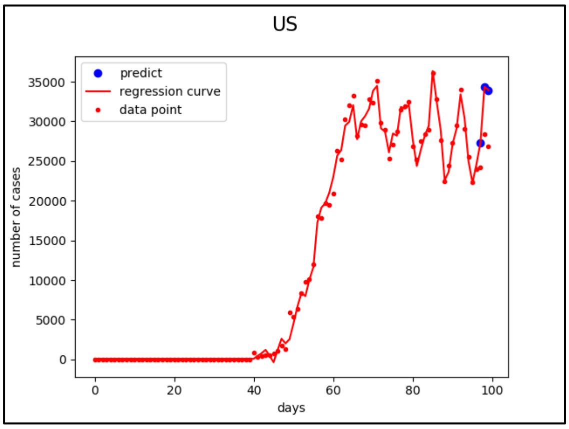


Figure 9 seven days' data for prediction

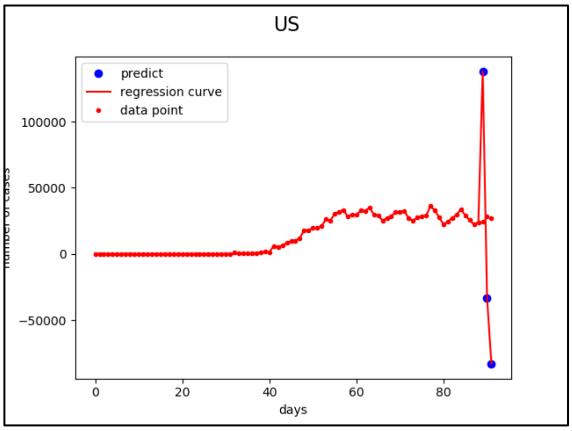


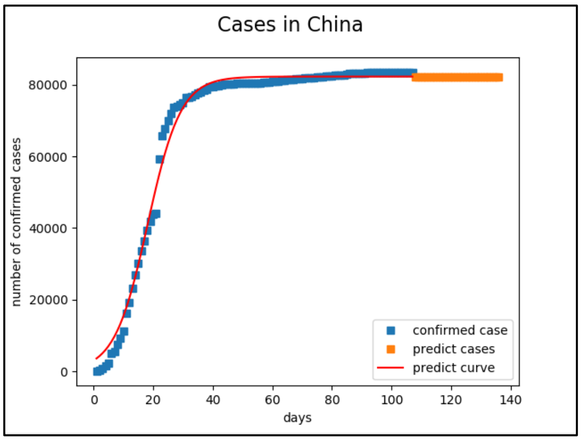
Figure10 fifteen days' data for prediction

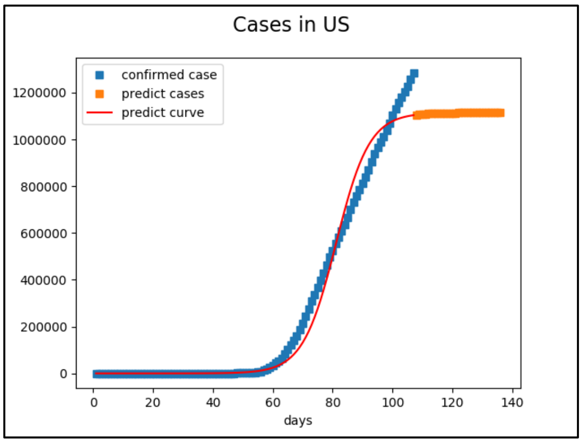
Could be seen that, when using longer sequence of data as feature, the result of the model easy to act as over fitted. Since the more features added to the model, the more the model will fit to the training set when regression. Therefore choosing 3 continuous days as features and do polynomial regression for the future case prediction, for the next unknown 10 days, the prediction results are:

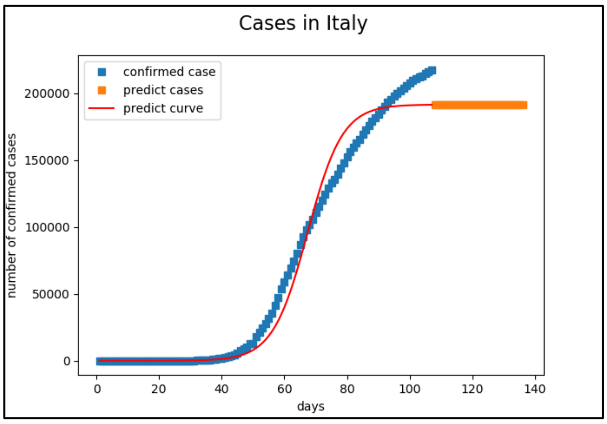
[29486.98,30611.38,31007.02,30984.92,30636.19,
30274.99,30067.75,30077.73,30216.65,30369.93]

B. Logistic growth curve

In the experiment, each day as a variable to calculate case happening at that day. So, this is a basic mapping function with one variable. Here are the results for some typical countries:







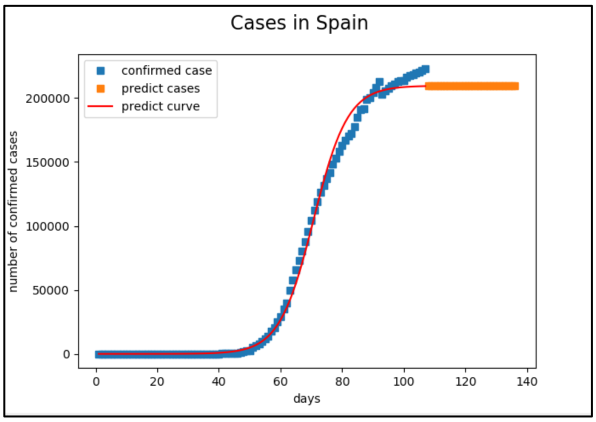


Figure 11 Logistic growth curve in different countries

Now taking the China as an example:

The K for this country is 82320 which means the total number of maximum cases is 82320 which is slightly different from the actually result and when using this model to make predictions, the next 10 days cases are 0 for China.

C. Neural Network

In the experiment, each model proposes a prediction for the further days regarding the available data, here are the results for some typical countries:

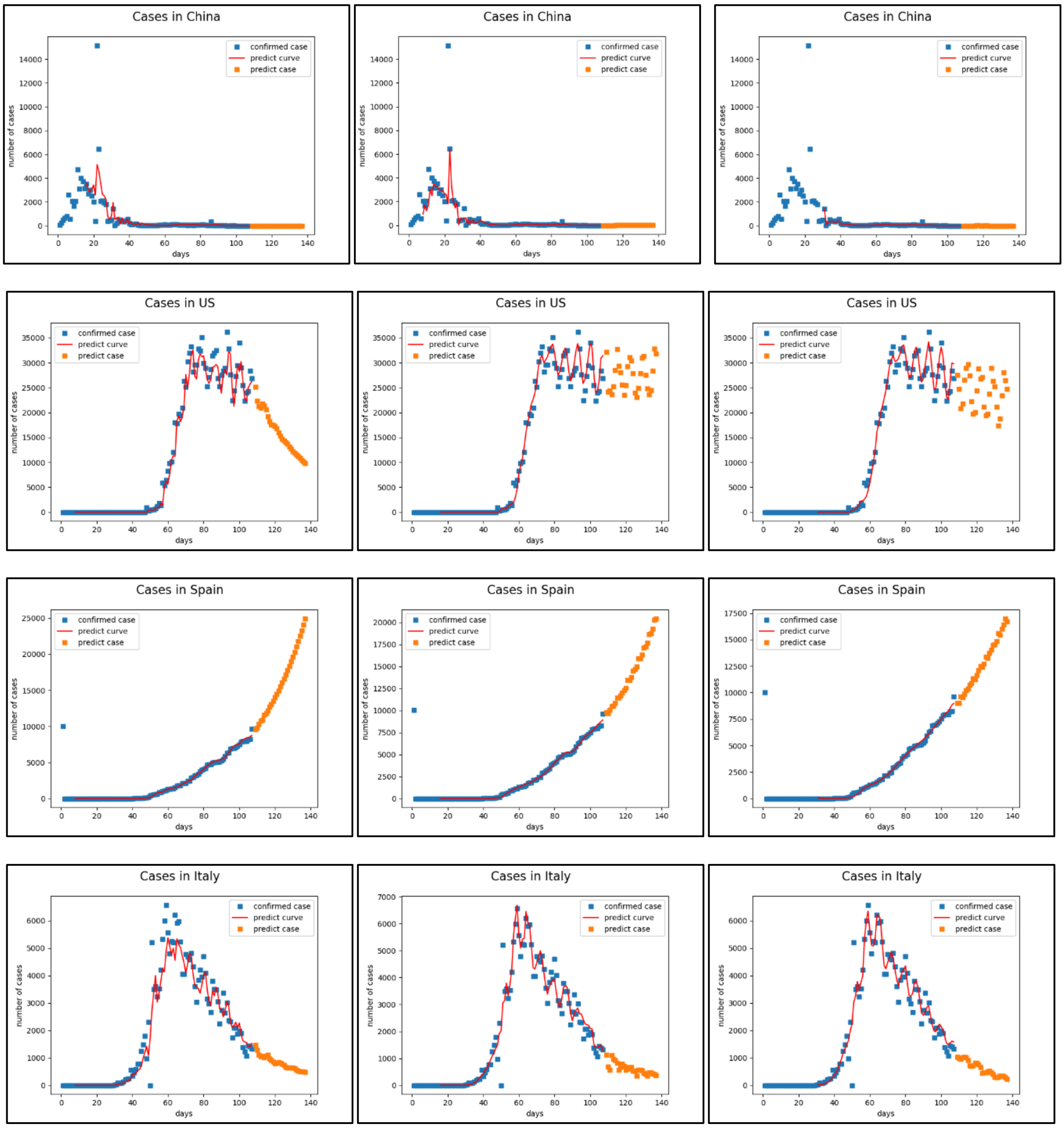


Figure 12 one week predict 8th day fifteen days predict 3 days one month predict 1 week

For each row of three graphs, the left one shows the prediction of 8th day case by the former 7 days, the middle one shows the prediction of the later 3 days' case by the former 15 days, the right one shows the prediction of the later one week by the former one month data. For each country, the prediction by each model has a slight difference. Using the first the model, the prediction will depend on the former 7 days. It could be a little bit problematic for periodic prediction. When using the second model, this problem can be solved to some extent. Using the third model, this problem can be solved in a larger scale. However, model 2 and model 3 can be influenced by the outliers.

Here taking the Spain as an example:

For the next unknown 10 days, the prediction results from three models are:

Model1(using the 7 days before to predict the 8th day):
[9771.69, 10144.42, 10770.62, 11476.56,12078.81,
12915.90, 13581.31, 14393.74,15296.95,16235.23]
Model2(using the 15 days before to predict the later 3 days):
[9405.86, 9923.88, 10405.75, 10281.97, 10834.46,
11337.77, 11339.02, 12160.86, 12602.01, 12215.78]
Model3(using the 1 month before to predict next week):
[9193.89, 8976.92, 8994.02,9827.00, 9348.65,
10076.22, 9600.03, 10665.23, 9850.82,10043.66]

The performance of each model is representing as the MAE and the R square. Therefore, weights for each model are added based on their performances. The final prediction for a single day is the combination of three predictions from three models multiplied by their weights. The formula is below:

$$y_{pre} = weight1 * y_{pre_{model1}} + weight2 * y_{pre_{model2}} + weight3 * y_{pre_{model3}} \quad (14)$$

Discussion

The model trained by polynomial regression can do a good job to predict the next 3 days confirmed case number, however the results turns bad when predict more. Since the polynomial transform is not a periodic form, so if the pattern is periodic, it can only predict the trend based on the last few points. Basically, it can be used to do data interpolation.
Another problem with polynomial regression is the choose of degree. With the increase of degree, the model can do a better job on the train set, however it will come with noise sensitive, overfitting problem, and the coefficient will have the explosion problem. Limited by the number of samples, we set a small degree (in the experiment, degree is set to 2) to limit the coefficient explosion and overfitting.

For the general logistic growth model, it has high reliability in the early stage and low reliability in the middle and late stages. The reasons are as follows:
1.There is basically no control of the disease in the early stage of transmission, but the community, medical units have strictly controlled the disease in the middle and late stages. Also, the government's policy may have a positive effect on the spread control. Therefore, the transmission may be reduced.

2.The infection base is large, and some cases die or recover, reducing the number of diagnoses. The logistic model does not consider these factors. It only considers the daily confirmed cases.
Therefore, the logistic growth model only predicts the disease trend but cannot accurately make prediction.

For the neural network, using a single model to make prediction is slightly unreliable, so three models are combined to help strengthen the dependency among time series. Based on the performance on the available case data, give different weight for the model and combine each model's output as the final prediction.

Here is the prediction made by three models above to display the next ten days cases for China (Table 5) and the U.S (Table 6)

	Polynomial Regression	Logistic Growth	Neural Network
Day1	0	0	4.03
Day2	0	0	3.26
Day3	0	0	4.54
Day4	0	0	3.87
Day5	0	0	4.66
Day6	0	0	4.08
Day7	0	0	2.9
Day8	0	0	4.38
Day9	0	0	7.17
Day10	0	0	4.53

Table 5 prediction for the cases in the next 10 days for China

	Polynomial Regression	Logistic Growth	Neural Network
Day1	29486.98	-17850	27795.3
Day2	30611.38	1446	25237.2
Day3	30984.92	1210.7	21557.5
Day4	30636.19	1013.3	19810.3
Day5	30274.99	847.8	22890
Day6	30067.75	709.14	26762.4
Day7	30077.73	593.01	27636.4
Day8	30067.75	495.81	27415
Day9	30216.65	414.48	24377.2
Day10	30369.93	346.43	19745.6

Table 6 prediction for the cases in the next 10 days for U.S

Here is the explanation for the numbers in the tables above:
For the polynomial regression, the next one day's data depends on the three days before, so the trend form the three days before directly determine the following day's cases. For the logistic growth, the prediction depends on the whole curve and the maximum capacity(K), so for the U.S prediction, since the fitting curve has exceeded the maximum capacity, the next day's prediction will be a negative number which is wrong. For the neural network, the next day's prediction depends on former week, half week and month, which is more reliable and convincing.

Conclusion

Polynomial regression, logistic regression, and Neural Network (NN) are used to predict the confirmed cases in different countries. The results show that the measure that split a time continuity data into small pieces can still save the part of continuity property of original data. Both Polynomial and NN can plot high fit curve. Both logistic and NN can provide reasonable predictions in future 29 days. In summary, the best model for COVID-19 confirmed cases should be NN. Three model, Model1(using the 7 days before to predict the 8th day), Model2(using the 15 days before to predict the 3th day), Model1(using the 30 days before to predict the 7th day) of NN is used to make the prediction. A combination plot by using all three models is supposed to have a stable performance.

Acknowledge

Appreciate the excellent teaching from Dr. Satish. All of us enjoyed his class in this semester.
Appreciate the Kaggle for providing the updated dataset about the COVID-19 cases.
Appreciate each group member's hard working, this project cannot be done without anyone's corporation.

Reference

[1]V. Chamola, V. Hassija, V. Gupta and M. Guizani, "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain and 5G in Managing its Impact," in IEEE Access, doi: 10.1109/ACCESS.2020.2992341.

[2] "Coronavirus disease (COVID-19) Pandemic." [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019 [3] T. Singhal, "A Review of Coronavirus Disease-2019 (COVID-19)

[3] Report of the WHO-China Joint Mission on Coronavirus Disease 2019 (COVID-19). WHO, Feb 2020. [Online]. Available: https://www.who.int/docs/default-source/coronaviruse/whochina-joint-mission-on-covid-19-final-report.pdf

[4]Lauer SA, Grantz KH, Bi Q, et al. The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application. Ann Intern Med. 2020;172:577–582. [Epub ahead of print 10 March 2020]. doi: https://doi.org/10.7326/M20-0504

[5] datamonday, Time series prediction 04: TF2.1 develops multi-layer perceptrons (MLPs) time series prediction model in detail.2020 [Online]. Available: https://blog.csdn.net/weixin_39653948/article/details/105341180

[6] XRX,Xiangrui, Python implements logistic growth model fitting 2019-nCov confirmed number of people updated on February 1st. 2020 [Online]. Available: https://blog.csdn.net/weixin_36474809/article/details/104119494

[7] COVID19 Global Forecasting (Week 5) Forecast daily COVID-19 spread in regions around world2020 [Online]. Available: https://www.kaggle.com/c/covid19-global-forecasting-week-5/discussion]

Group member and contribution

Bin Zhang (5660329599): Logistic growth function building, neural network building and report for logistic curve, neural network
Yihang Chen (6338254416): Data preparation, polynomial regression model building, report for the data preparation, polynomial regression part

Hanzhi Zhang (4395561906): model optimization, report for the abstract, introduction, conclusion part

All of us spare no effort to take part in this project, so other the tasks are done by all team members.