# 新冠肺炎疫情趋势预测-最终报告

小组成员：武星 3220190897
孙燕北 3120191045

## 1. 问题背景

2020 年突如其来的肺炎疫情，打乱了很多人的计划，也让很多人措手不及。新冠肺炎疫情传播在全社会范围内对企业、经济、人民生活造成了巨大影响。随着疫情的持续发展，各国政府鼓励运用大数据、人工智能等技术，在疫情监测分析方面发挥支撑作用。

本次课设选题为预测美国新冠肺炎疫情趋势，主要包括疫情导致的每天感染人数以及死亡人数的预测等，模型算法选用较为简单的 logistic 与 Exponential 回归模型拟合疫情数据，以及训练 LSTM 循环神经网络对疫情数据进行预测，主要方法为利用前一段时间内的疫情数据去预测之后一天疫情数据，包括感染人数以及死亡人数，从而得到疫情的发展趋势。

## 2. 数据获取及预处理

### 2.1 数据来源及说明

本次项目使用的数据来自 Kaggle 美国新冠肺炎公开数据集 Us counties COVID 19(https://www.kaggle.com/jieyingwu/covid19-us-countylevel-summaries)数据集包含了自 2020 年 1 月 21 日至近期(2020 年 6 月 13 日)美国各州郡的新冠肺炎患者以及死亡数据情况。

数据集共有 6 个字段：

- date：日期，例如：“2020-01-21”，(数据范围：2020.01.21 - 2020.06.13)
- county：郡县名，例如 “Snohomish”
- state：州名，例如 “Washington”
- fips：郡县区别代码
- cases：截止当前日期该郡县的累计确诊病例数量
- deaths：截止当前日期该郡县的累计死亡病例数量

例如：

| | date | county | state | fips | cases | deaths |
|---|---|---|---|---|---|---|
| 0 | 2020-01-21 | Snohomish | Washington | 53061.0 | 1 | 0 |
| 1 | 2020-01-22 | Snohomish | Washington | 53061.0 | 1 | 0 |
| 2 | 2020-01-23 | Snohomish | Washington | 53061.0 | 1 | 0 |
| 3 | 2020-01-24 | Cook | Illinois | 17031.0 | 1 | 0 |
| 4 | 2020-01-24 | Snohomish | Washington | 53061.0 | 1 | 0 |

Figure 1：**数据样例**

### 2.1 数据预处理

首先查看不同字段的数据缺失比例情况：

| | column_name | percentage |
|---|---|---|
| 0 | date | 0.000000 |
| 1 | county | 0.000000 |
| 2 | state | 0.000000 |
| 3 | fips | 1.057585 |
| 4 | cases | 0.000000 |
| 5 | deaths | 0.000000 |

Figure 2：字段缺失值比例

可以看到除 fip 字段以外，其他字段都没有缺失情况，而对于 fips 字段，也仅仅只有 1%左右的数据缺失，因为 fips 与之前的 county 字段是一一对应关系，因此在这里，我们根据之前的 county 字段数据进行相关填充，但是在之后的工作中，因为这两个字段的信息一致，因此并没有实际使用 fips 信息。

# 3.数据分析与可视化

## 3.1 全美整体疫情分析

首先我们将数据集中各州数据按照时间做了累计，以分析这段时间疫情的整体发展趋势：

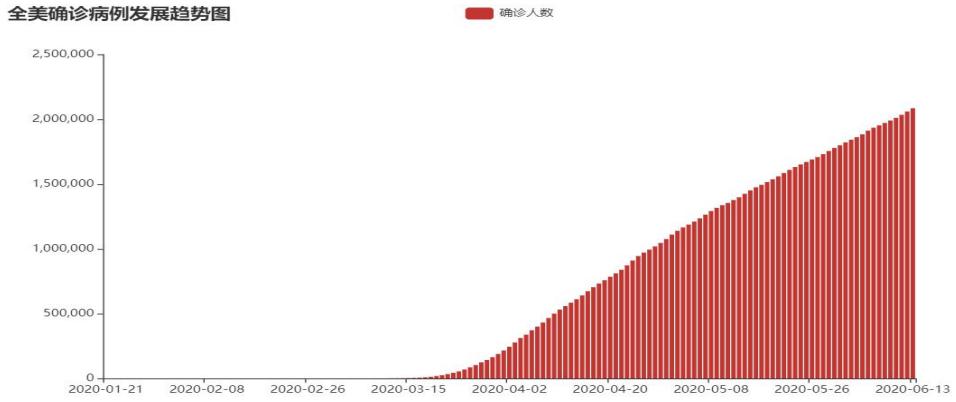


Figure 3：确诊病例总数发展趋势

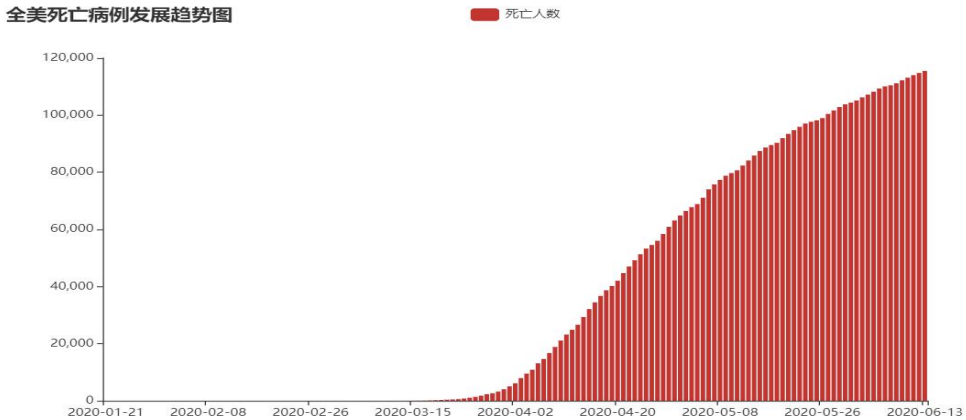


Figure 4：死亡病例总数发展趋势

上面两张图中展示了数据集中这段时间的全美疫情整体发展情况，可以看到自一月末开始的两个月里，几乎没有病例产生，但是自三月中旬开始，新冠肺炎进入了一个快速发展期，虽然六月数据比较少，但是我们从整体上依然可以看出无论是确诊病例数量还是死亡病例数量，都有所缓和。
接下来分析每日确诊以及死亡病例数量：


Figure 5：**日确诊病例增加量**

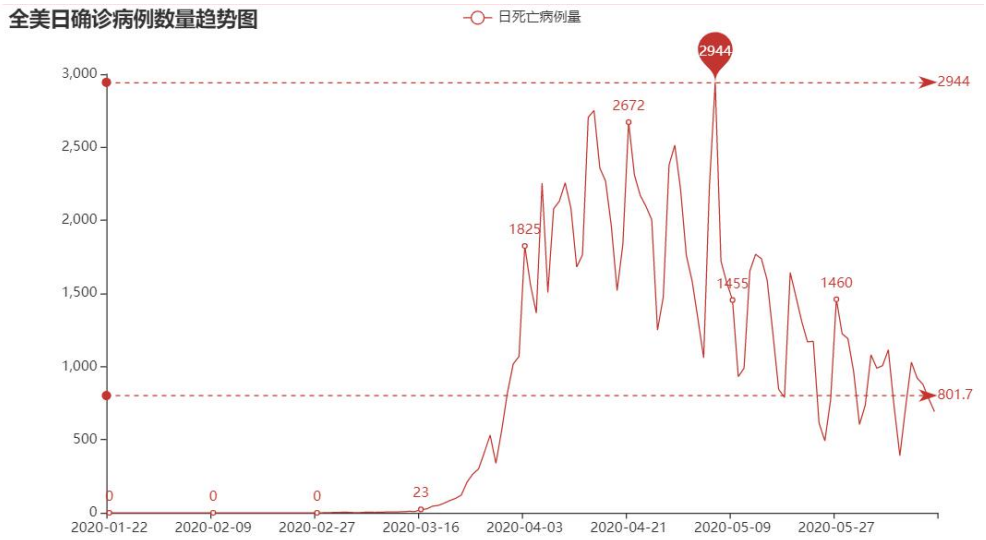


Figure 6：**日死亡病例增加量**

无论是确诊病例还是死亡病例的日增长数量也符合刚才的结论，在三月到四月是全美疫情最严重的时间段，但是自四月开始，虽然病例数量日增长都有波动，但是整体来看是可以看到呈现出一个下降的趋势的，这也说明了自四月开始，人们采取的一些策略开始起作用了，整体疫情得到了一定的控制。

## 3.2 各州疫情发展情况分析

首先我们统计了截止目前美国各州的疫情数据，并对这些数据进行了排序，例如，按照确诊病例数量进行排序：

| | index | state | cases | deaths |
|---|---|---|---|---|
| 0 | 33 | New York | 387402 | 30565 |
| 1 | 31 | New Jersey | 166605 | 12589 |
| 2 | 4 | California | 150418 | 5059 |
| 3 | 14 | Illinois | 133117 | 6491 |
| 4 | 22 | Massachusetts | 105395 | 7576 |

Figure 7：**各州具体疫情数据(按确诊数量)**

首先，使用条形图直观展示不同州的确诊以及死亡病例数量对比(由于州数量较多，因此这里展示的疫情比较严重的十个州)：

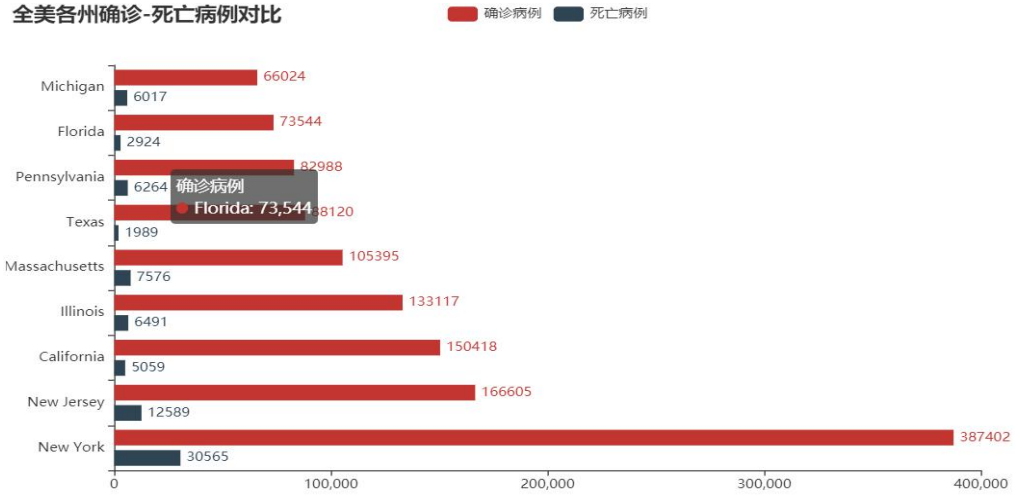


Figure 8：**各州确诊-死亡病例对比**

我们计算了这些州的确诊以及死亡病例所占比例情况：

| | index | state | cases | deaths | percent |
|---|---|---|---|---|---|
| 0 | 6 | Connecticut | 44994 | 4186 | 0.093035 |
| 1 | 23 | Michigan | 66024 | 6017 | 0.091134 |
| 2 | 49 | Virgin Islands | 72 | 6 | 0.083333 |
| 3 | 33 | New York | 387402 | 30565 | 0.078897 |
| 4 | 31 | New Jersey | 166605 | 12589 | 0.075562 |
| 5 | 40 | Pennsylvania | 82988 | 6264 | 0.075481 |
| 6 | 22 | Massachusetts | 105395 | 7576 | 0.071882 |
| 7 | 36 | Northern Mariana Islands | 30 | 2 | 0.066667 |
| 8 | 19 | Louisiana | 46396 | 3004 | 0.064747 |
| 9 | 37 | Ohio | 40848 | 2554 | 0.062524 |

Figure 9：**各州死亡率对比**

从各州确诊以及死亡病例比例来看，最高的州死亡率达到了 9%，有很大一部分州的死亡率为 5%左右，当然一些我们所熟知的州，尽管确诊病例众多，但是死亡率不高，这与美国各州经济发展不平衡也有很大关系，例如我们可以绘制比例图更直观得到这个结论。

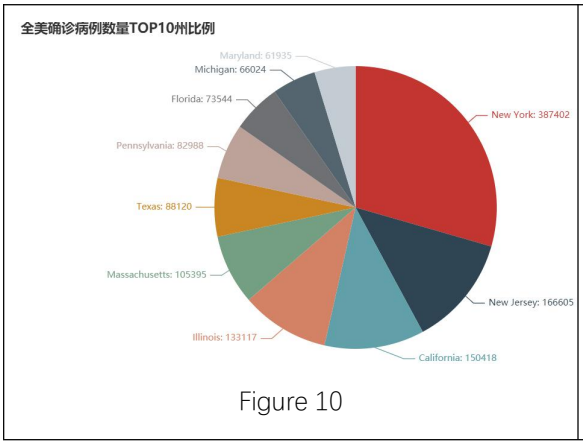


Figure 10

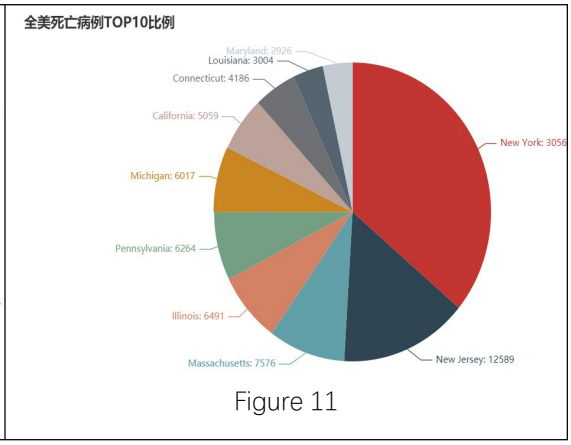


Figure 11

此外，为了分析地理因素对于各州疫情发展情况的影响，我们将确诊病例以及死亡病例数量呈现在美国地图上，可以得到：

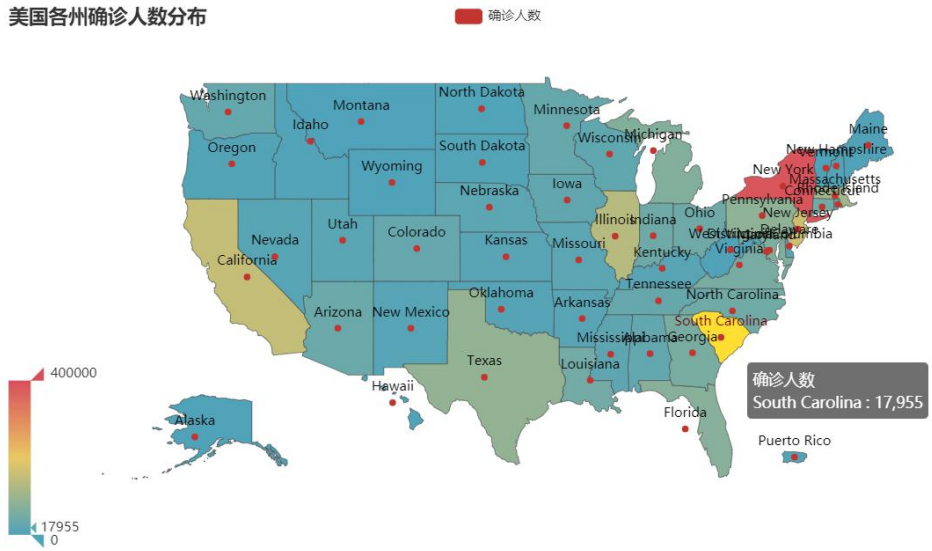


Figure 12：各州确诊病例对比

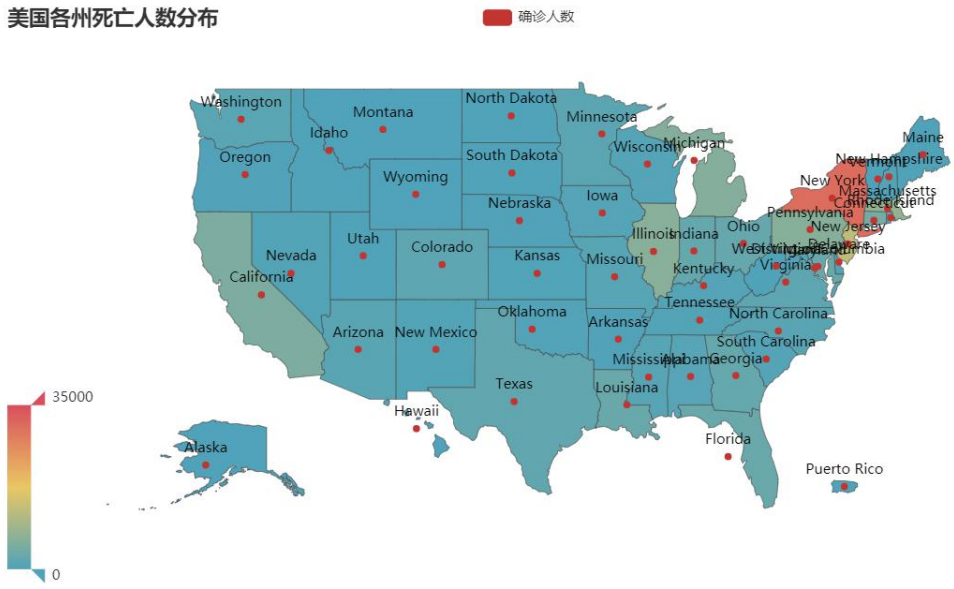


Figure 13：各州死亡病例对比

通过这两个地域分布图，我们可以看到，全美东北角的疫情较其他地方更严重，这部分区域也是整个美国经济最为发达的地区之一，人口密度高，人口流量大，特别是在早期对于疫情控制并未做好的情况下，造成了大量人员感染。此外，无论是确诊病例还是死亡病例数量，我们也可以看到另一个趋势：沿海岸线以及边境线的确诊人数更多，因此在疫情严重的现在，更应该把握对外交流的管控，防止人员流动带来的疫情问题。

# 4. 模型构建及实现

## 4.1 回归模型及实现

机器学习中有许多相关算法可以用来解决时序数据集的预测问题，其中比较常见模型有 logistic 回归模型与 Exponential 回归模型。

logistic 回归模型被广泛用于描述人口的增长，而传染病的传播过程会随着时间不断的更新，这与人口的增长有着类似的地方，所以可以使用 logistic 模型进行传染病趋势预测。

logistic 回归模型可以表示为将线性函数的结果映射到 sigmoid 函数中，其中 sigmoid 函数最一般的表达式如公式(1)所示：

$$\mathrm{h}(x,a,b,c)=\frac{c}{1+e^{-(x-\mathrm{b})/a}} \tag{1}$$

其中变量 x 表示时间，另外三个参数为待拟合的参数变量。模型实现方式采用定义函数与 curve_fit 拟合的方式，其中 logisti_model 函数如图 Figure14 所示，利用该函数进行 curve_fit 拟合即可实现该模型。

```
90 def logistic_model(x,a,b,c):
91     return c/(1+np.exp(-(x-b)/a))
```

Figure14：logistic 模型函数

另一种比较常见的模型是指数 Exponential 模型，与 logistic 模型不同的是与 logistic 模型描述的是在未来某一天传染病终会停止传染，而 Exponential 模型描述的是传染病将不可阻挡的持续感染，所以从模型本身考虑，logistic 模型与现实更加接近。

Exponential 模型的通用公式如公式(2)所示：

$$\mathrm{h}(x,a,b,c)=\mathrm{a}\cdot e^{b(x-c)} \tag{2}$$

其中变量 x 表示时间，另外三个参数为待拟合的参数变量。模型实现方式与 logistic 模型实现方式类似，采用定义函数与 curve_fit 拟合的方式，其中 exponential_model 函数如图 Figure15 所示，利用该函数进行 curve_fit 拟合即可实现该模型。

```
192 def exponential_model(x, a, b, c):
193     return a * np.exp(b * (x - c))
```

Figure15：exponential 模型函数

## 4.2 LSTM 模型

LSTM 算法全称为 Long short-term memory，是一种特定形式的 RNN（Recurrent neural network，循环神经网络），而 RNN 是一系列能够处理序列数据的神经网络的总称。

一般地，RNN 包含如下三个特性：

a）循环神经网络能够在每个时间节点产生一个输出，且隐单元间的连接是循环的。

b）循环神经网络能够在每个时间节点产生一个输出，且该时间节点上的输出仅与下一时间节点的隐单元有循环连接。

c）循环神经网络包含带有循环连接的隐单元，且能够处理序列数据并输出单一的预测。

RNN 还有许多变形，例如双向 RNN（Bidirectional RNN）等。然而，RNN 在处理长期依赖（时间序列上距离较远的节点）时会遇到巨大的困难，因为计算距离较远的节点之间的联系时会涉及雅可比矩阵的多次相乘，这会带来梯度消失（经常发生）或者梯度膨胀（较少发生）的问题，这样的现象被许多学者观察到并独立研究。为了解决该问题，研究人员提出了许多解决办法，例如 ESN（Echo State Network），增加有漏单元（Leaky Units）等等。其中最成功应用最广泛的就是门限 RNN（Gated RNN），而 LSTM 就是门限 RNN 中最著名的一种。有漏单元通过设计连接间的权重系数，从而允许 RNN 累积距离较远节点间的长期联系；而门限 RNN 则泛化了这样的思想，允许在不同时刻改变该系数，且允许网络忘记当前已经累积的信息。

LSTM 就是这样的门限 RNN，其巧妙之处在于通过增加输入门限，遗忘门限和输出门限，使得自循环的权重是变化的，这样一来在模型参数固定的情况下，不同时刻的积分尺度可以动态改变，从而避免了梯度消失或者梯度膨胀的问题。

## 4.3 LSTM 模型实现

本次实验采用基于 python 的 keras 开源人工神经网络库进行实现。

首先进行数据格式的预处理，对疫情数据进行归一化以提高训练的速度和训练效果，其次由于 lstm 网络的输入数据为时间序列，所以要将疫情数据切分为一段段的时间序列切片作为训练数据，下一段时间序列切片作为训练数据的标签，经过多次尝试选取 time_step 为 40 进行模型构建，即用 40 天的疫情数据去预测下一天的疫情结果，Figure16 为预处理后训练数据集标签的 shape：

```
In  [23]: dataX.shape,dataY.shape
Out[23]: ((104, 40), (104,))
```

Figure16：预处理后 shape

可以看到训练数据经过处理后总共有 104 条，对数据集进行划分，将前 80%即前 83 条数据划分为训练集，后 20%即剩下的 21 条数据划分为测试集，从而完成数据集的处理。

构建 lstm 网络模型，采用经典的 LSTM-->Drop_out-->Dense 划分结构，对于确诊人数数据集的模型整体结构如图 Figure17 所示，对于死亡人数数据集模型 lstm 层参数设置为 25。

```
In [24]: model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 100)               56400
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 56,501
Trainable params: 56,501
Non-trainable params: 0
_________________________________________________________________
```

Figure17：模型结构 summary

对于确诊人数数据集设置参数 batch_size 为 32，优化器为 adam，loss 采用 mse 评价指标，训练轮数 epochs=200 次；对于死亡人数数据集训练轮数设置为 epochs=100 次，训练部分过程如图 Figure18 所示。

```
74/74 [==============================] - 0s 330us/step - loss: 0.0160 - val_loss: 0.0090
Epoch 16/200
74/74 [==============================] - 0s 329us/step - loss: 0.0169 - val_loss: 0.0107
Epoch 17/200
74/74 [==============================] - 0s 316us/step - loss: 0.0166 - val_loss: 0.0122
Epoch 18/200
74/74 [==============================] - 0s 317us/step - loss: 0.0167 - val_loss: 0.0163
Epoch 19/200
74/74 [==============================] - 0s 329us/step - loss: 0.0147 - val_loss: 0.0207
Epoch 20/200
74/74 [==============================] - 0s 303us/step - loss: 0.0150 - val_loss: 0.0221
Epoch 21/200
74/74 [==============================] - 0s 329us/step - loss: 0.0157 - val_loss: 0.0197
Epoch 22/200
74/74 [==============================] - 0s 317us/step - loss: 0.0155 - val_loss: 0.0157
Epoch 23/200
74/74 [==============================] - 0s 303us/step - loss: 0.0145 - val_loss: 0.0134
Epoch 24/200
74/74 [==============================] - 0s 317us/step - loss: 0.0153 - val_loss: 0.0122
Epoch 25/200
74/74 [==============================] - 0s 330us/step - loss: 0.0141 - val_loss: 0.0103
Epoch 26/200
74/74 [==============================] - 0s 356us/step - loss: 0.0126 - val_loss: 0.0088
Epoch 27/200
```

Figure18：部分训练结果

训练过程中每个 batch 的 loss 变化曲线如图 Figure19 所示。

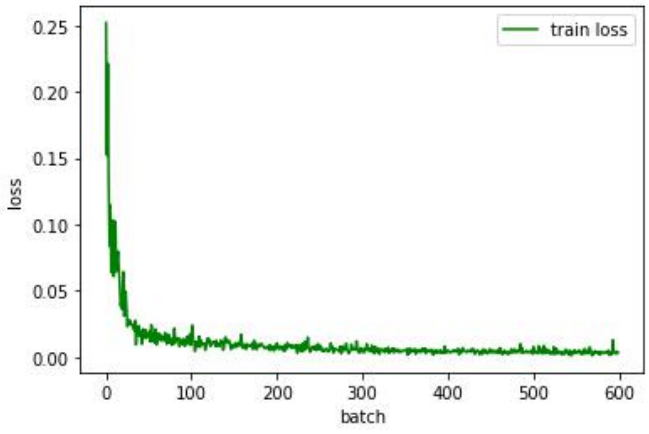


Figure19：loss 变化曲线

# 5. 实验结果及分析

## 5.1 回归模型结果分析

采用 logistic 模型与 Exponential 模型对从 1 月 21 日起至 6 月 13 为止美国新冠肺炎确诊人数以及死亡人数数据进行拟合，拟合结果如图 Figure20 所示，左侧为确诊人数数据拟合结果，右侧为死亡人数数据拟合结果，图表下方是两种模型拟合结果的评估，评价指标采用的是 RMSE。

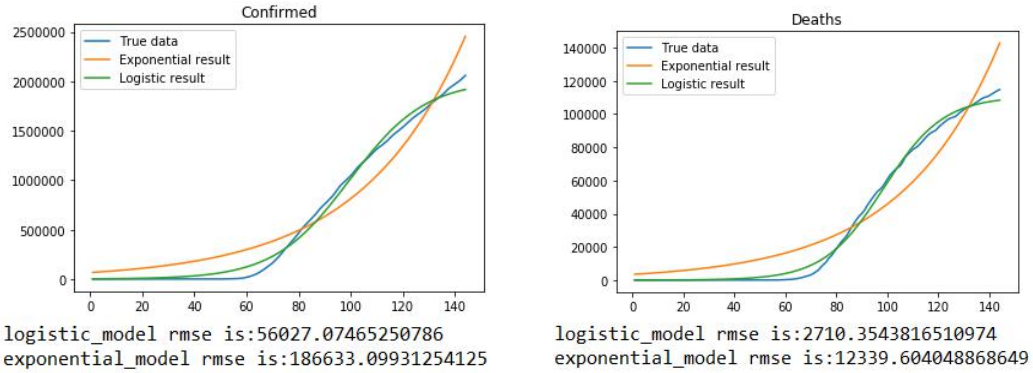


Figure20：回归模型实验结果

由 rmse 结果可以看出只有关于美国新冠肺炎的死亡人数统计基本符合 logistic 模型并且 rmse 相对较小，其他的实验结果显示 rmse 指标过大且并不符合 logistic 模型与 Exponential 模型的描述。

经过分析 logistic 是呈现出 s 曲线的函数，描述的是理想情况下且无外界干扰的发展趋势，而由实验结果可以看出美国疫情的发展趋势较为复杂，与传统 logistic 模型并不十分吻合。而 Exponential 模型本身并没有呈现出 s 曲线的趋势，并不适合流行病发展趋势的描述。

## 5.2 LSTM 模型结果分析

对训练好的模型，按照测试集的长度即 21 进行接下来 21 天的疫情发展趋势预测，每次只计算出下一天的疫情数据，并将该数据与之前 39 天数据进行拼接重新送入到模型中进行计算以得到下一天预测结果，上述过程循环 21 次从而得到了接下来 21 天的疫情发展趋势数据。

在整体数据集上的实验结果如图 Figure21 所示。

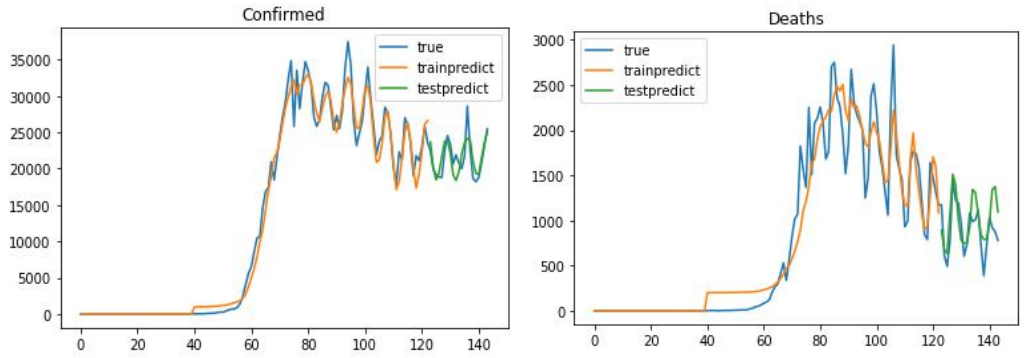


Figure21：整体数据集上实验结果对比

其中左侧为确诊数据结果，右侧为死亡数据结果，图中蓝色数据表示每一天的真实值，橙色数据表示将训练数据集通过网络计算得到的结果，绿色数据为预测得到的 21 天数据。

将经过模型计算得出的后 21 天数据与测试集中的 21 天数据进行 rmse 指标的计算，计算结果如图 Figure22 所示，综合 21 天确诊数据的 rmse 指标为 1763，死亡数据的 emse 指标为 245，可以看出相较于回归模型的拟合结果有了明显的提升。

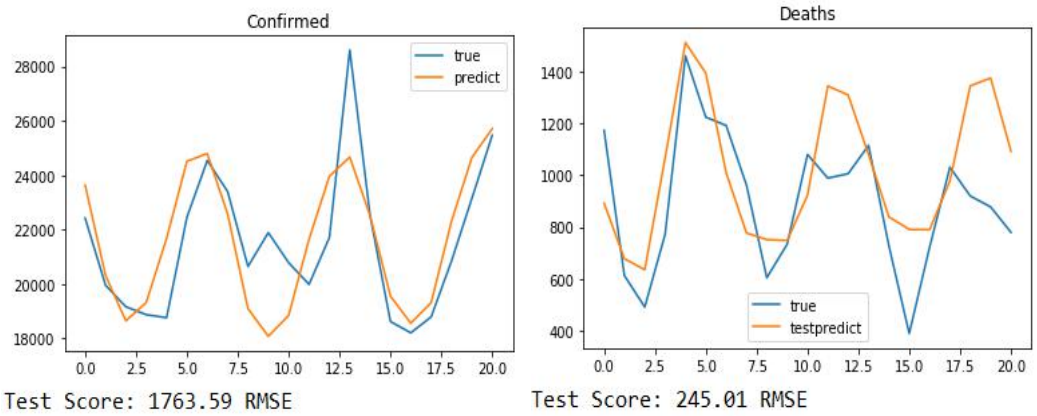


Figure22：**测试集上实验结果对比**

由实验结果可以看出经过训练出的 lstm 模型能够较好的模拟出疫情发展的大趋势，但对于趋势的峰值以及一些细微的变化并没有很好的描述出来，该结果虽然对于预测人数的精度较低，但对于趋势的预测仍然具有一定的价值。另外由于现阶段美国疫情的确诊数据趋势大都处于不断上升或是平稳的趋势，训练的 lstm 模型学习到的主要是上升及平稳阶段的数据拟合，所以确诊人数的预测结果最终不会收敛至 0，而是收敛到一个平衡的模式，所以未能给出疫情结束的日期。

# 6. 总结

在这次关于美国新冠肺炎疫情分析与预测项目中，我们首先对数据进行了前期处理，例如根据字段相关性进行缺失值填充，随后从整体和各州两种视角对当前疫情的发展做了分析，从整体来看，全美的疫情发展已经过了爆发期，政府采取的一些疫情控制措施开始起作用，但是整体确诊人数依然很多，因此依然需要更有力的疫情把控，从局部各州的情况来看，沿海以及边境线各州由于人口流量较大，疫情处理有一定难度，目前疫情也较为严重，仍需要对人员流动做好检测等控制措施。在对疫情趋势预测的过程中，我们首先尝试使用较为常用的逻辑回归以及指数回归模型进行拟合，但实验结果较差，并不能较好的拟合出疫情的发展变化趋势，之后我们利用 lstm 神经网络对疫情数据进行训练，由于训练数据条数较少且特征只利用了每日的人数变化，所以预测结果的精度比较低，但整体的疫情趋势比较清晰，具有一定的参考意义，之后可以尝试引用一些其他特征(例如美国的天气气候因素以及美国的医疗部署情况等)参与训练，可能会得到更高精度的结果。

## 项目分工情况

孙燕北：构建模型与结果分析，文档编写

武星：数据分析与预处理，文档编写

## 项目代码仓库地址：

数据分析代码：https://github.com/MirrorN/Data_Mining_Report/tree/master/Final
模型构建代码：https://github.com/syb-5213/Data_Mining