

# 基于 LSTM+BPNN 深度神经网络的新冠疫情预测模型

张栩菱

(营口理工学院 电气工程学院, 营口 邮编)

**摘要:** 新冠肺炎疫情传播在全社会范围内造成了巨大影响。本文对收集到的数据, 利用长短期记忆人工神经网络模型对疫情期间人员流动历史数据合理分析预测, 通过具有多层前馈结构的误差反向传播算法对预测过后的数据模型进行细化、调参、优化。从而提出一种基于 LSTM-BPNN 深度神经网络的新冠疫情预测模型, 以此为推动抗击疫情的发展贡献绵薄之力。

**关键词:** 长短期记忆模型; 误差反向传播算法; 深度学习

## 1 引言

新型冠状病毒肺炎(COVID-19)在全球范围内大范围传播流行,是全世界关注的公共卫生问题.本次疫情人员控制上,很多城市在人员的综合管理、疫情的实时管控、突发情况的处理等方面存在不足。但因我国强有力的疫情控制措施, 目前疫情趋于稳定。然而, 在疫情爆发早期, 政策实施相对宽松, 医疗资源相对缺乏, 疫情传播扩散速度较快。本次疫情爆发与农历新年春运时间较吻合, 且早期疫情传播受政策影响较小, 因此, 通过建立疫情早期流动人口数据与各地区疫情严重程度的模型对于疫情的检控预防具有积极的意义。

## 2 理论基础

### 2.1 传统 RNN 与 LSTM 神经网络结构模型分析对比

在传统的 RNN 模型中, 通过对一数据矩阵 $[x_1, x_2, x_3, \dots, x_n]$ 作为输入集传送入模型框架中, 经过初始化的权值矩阵 $[u_1, u_2, u_3, \dots, u_n]$ 时, 会将其输入矩阵  $x$  和矩阵  $w$  ( $w$  是上次数据处理后的隐藏层矩阵  $s$ ) 得到新生的隐藏层数据矩阵  $s=[s_1, s_2, s_3, \dots, s_n]$ , 矩阵  $s$  经过权值矩阵  $v=[v_1, v_2, v_3, \dots, v_n]$ 和  $w=[w_1, w_2, w_3, \dots, w_n]$ 通过计算方程式后得到输出层矩阵  $o=[o_1, o_2, o_3, \dots, o_n]$ 。如图 1

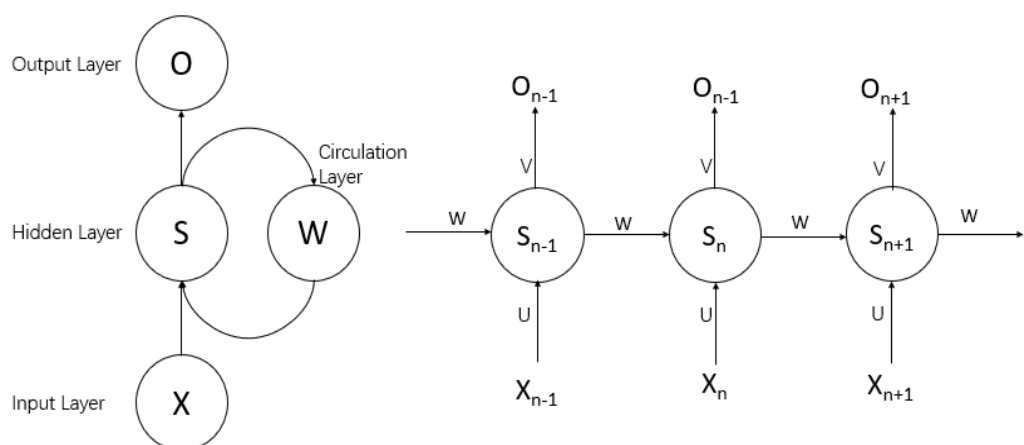


图 1 RNN 网络结构图及其展开

由此可见，传统 RNN 在理论上是利用过去的信息流信息与现如今的信息做一系列运算，如果是短期内的信息，确实可以得到较好的反馈，不过一旦时间久远，那么便会出现长依赖问题，网络难以去正确地分析处理该问题。或者说，极易学习到错误的知识。而 LSTM 可以更好地解决这一问题，它是一种特殊的循环神经网络结构，在原本的重复链接单元上添加到了四个网络层，如图 2

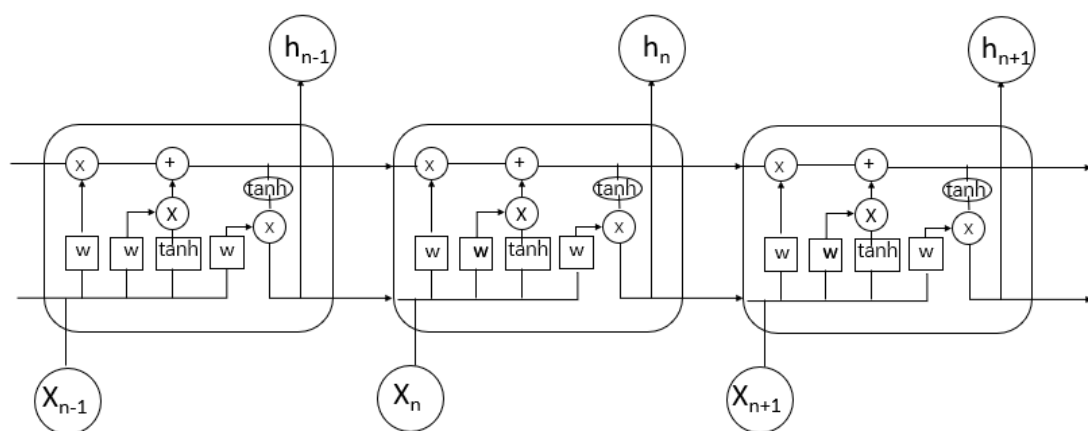


图 2 LSTM 网络结构图

LSTM 网络有着一种处理信息的特殊结构，称之为门，它由一个点乘运算以及一个限制数据输出大小范围为 0 到 1 的 Sigmoid 层组成。1 为通过，而 0 则阻挡，也就是说该结构有着对输入数据是否可以流通的决定权。

## 2.2 BP 神经网络结构模型分析

反向传播神经网络是一种凭借其卓越的内部复杂模式下非线性映射能力和优异的提取数据特征自我学习能力而被广泛应用的多层前馈神经网络，它是 McClelland 与 Rumelhart 等人于 1986 年独立开发用来处理多层神经网络隐含层连接权学习障碍的结构模型。它是

由输入层、输出层、隐含层所组成的结构。

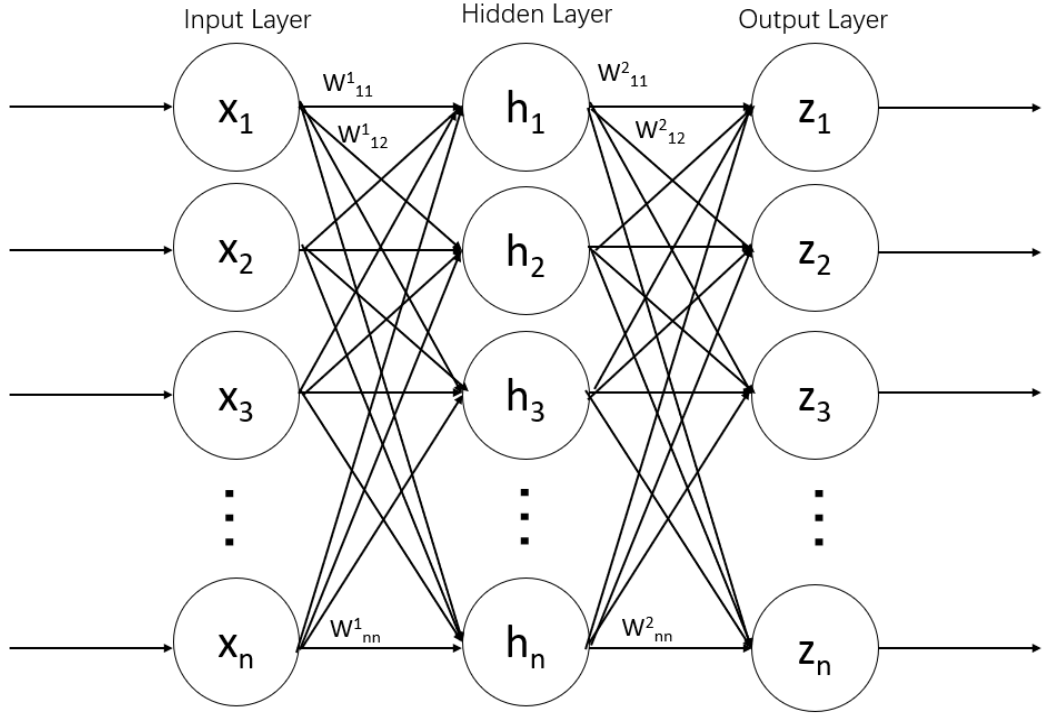


图 3 BPNN 网络结构图

如图 3 所示,  $[x_1, x_2, x_3, \dots, x_n]$  是本文输入层的  $n$  维输入数据, 在经过两层之间的  $[w_{11}^1, w_{12}^1, w_{13}^1, \dots, w_{nn}^1]$  和  $[w_{11}^2, w_{12}^2, w_{13}^2, \dots, w_{nn}^2]$  权值点乘相加运算后, 通过激活函数将数据压缩到隐含层并作为该层的输入向输出层传播, 最终得到  $[z_1, z_2, z_3, \dots, z_n]$ 。

$$\sum_{i=1}^n \sum_{j=1}^n h_i = x_j * w_{ji}^1$$

$$\sum_{i=1}^n \sum_{j=1}^n z_i = h_j * w_{ji}^2$$

### 3 实验准备

#### 3.1 实验环境平台

实验所使用计算机的配置如下: 处理器为 Inter(R) Core(TM) i5-8250U, CPU 频率为 1.60GHz 和 1.80GHz; 内存为 8.00GB; 操作系统为 Windows 10(64 位); 程序设计语言为 Python 3.8(64 位); 集成开发环境为 PyCharm Community Edition 2019.1.1。神经网络搭建中, LSTM、BPNN 模型由 Python 的 torch-1.6.0+cpu.dist-info 和 torchvision-0.7.0+cpu.dist-info 程序库实现。

#### 3.2 数据集

实验数据集提供 2020 年中国各地 1 月 10 号到 1 月 18 号的迁入迁出人数和患新冠肺炎人数记录。该记录如表 1、表 2、表 3 所示。为讨论三者之间的联系以及预测未来可能患病人数，本文选择 200 个地点的迁入和迁出以及患病人数作为训练集，余下 40 个作为测试集。

表 1 各地各时间迁迁出入人数

地点 时间	武汉			孝感			.....	吐鲁番		
	迁入	迁出	患病	迁入	迁出	患病		迁入	迁出	患病
1.10	4.6	6.62	572	1.41	0.97	26	.....	0.26	0.2	0
1.11	4.64	7.56	2639	1.69	1.11	541	.....	0.27	0.22	1
1.12	4.37	6.22	3215	1.58	1.09	628	.....	0.25	0.19	1
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
1.18	4.23	7.71	11618	2.03	1.00	2141	.....	0.24	0.21	1

### 3.3 数据预处理

本文的数据集中提取出各地各时间的迁入迁出人数和患病人数。先用 LSTM 对迁入数据集进行时间序列预测，以某地  $n$  天前的迁入或迁出人数 $[x_{n-4}, x_{n-3}, x_{n-2}, x_{n-1}, x_n]$ 为一组作为输入训练集，训练得到  $[x_{n-3}, x_{n-2}, x_{n-1}, x_n, x_{n+1}]$ ，这个  $x_{n+1}$  就是本文预测的该地第  $n+1$  天的迁入或迁出人数。

本文的主要目的是对未来某地疫情爆发情况进行分析，为了预测 A 城市第  $n+1$  天的疫情爆发情况，使用 A 城市 15 天前的迁入人数 $[x_{n-14}, x_{n-13}, \dots, x_n]$ 和这 15 天的迁出人数 $[y_{n-14}, y_{n-13}, \dots, y_n]$ 作为 BPNN 的输入集  $I_1$ ，如下

$$I_1 = [x_1 \ y_1 \ x_2 \ y_2 \ \cdots \ x_{15} \ y_{15}]$$

为了让本文的网络有更好的适应泛化能力，本文对大量地区的数据同时进行训练，为此将 A、B、C...Z 各地的  $I_1$  输入集组合为矩阵  $I$ ，如下：

$$I = \begin{bmatrix} x_{a1} & y_{a1} & x_{a2} & y_{a2} & \cdots & x_{a15} & y_{a15} \\ x_{b1} & y_{b1} & x_{b2} & y_{b2} & \cdots & x_{b15} & y_{b15} \\ x_{c1} & y_{c1} & x_{c2} & y_{c2} & \cdots & x_{c15} & y_{c15} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{z1} & y_{z1} & x_{z2} & y_{z2} & \cdots & x_{z15} & y_{z15} \end{bmatrix}$$

在这里本文采用 MinMaxScaler 函数对输入矩阵  $I$  进行数据归一化,最终得到的矩阵最终被压缩在 $[0,1]$ 间,得到最终进入神经网络的输入矩阵  $Input$ ，提高网络的工作效率。

$$Input = \frac{(I - I.min)}{(I.max - I.min)}$$

最后在输出层使用 MinMaxScaler 的归一规则对输出矩阵  $Output$  反归一化得到预测的数据矩阵。

$$\text{Output} = \begin{bmatrix} Z_a \\ Z_b \\ Z_c \\ \vdots \\ Z_z \end{bmatrix}$$

### 3.4 激活函数与损失函数

为度量网络的性能，提高预测效果，采用均方差平方根 **MSE** 函数作为损失函数对预测结果  $Z_{\text{predict}}$  和实际数据  $Z_{\text{train}}$  求误差，并在训练中不断更新误差调整整体网络精度。

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Z_{\text{train},i} - Z_{\text{predict},i})^2$$

在各网络层之间选用 ELU 作为激活函数。

$$\text{ELU}(x) = \begin{cases} x, & x > 0 \\ a(e^x - 1), & x \leq 0 \end{cases}$$

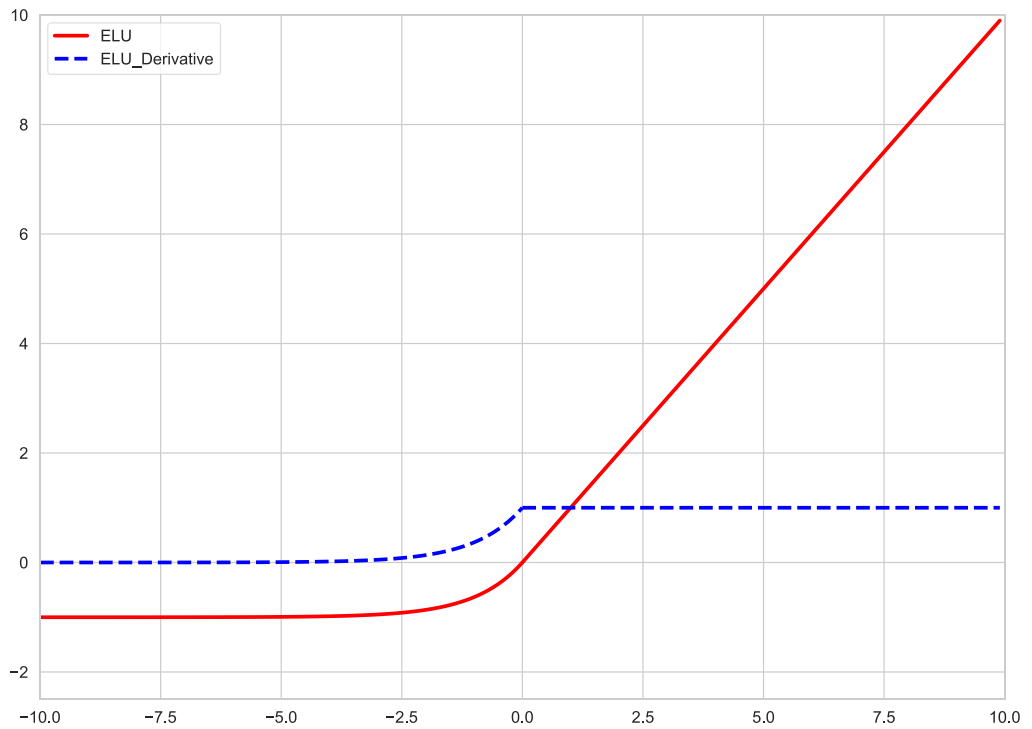


图 4 EUL 函数及其导数图

ELU 函数融合了 Sigmoid 和 Relu 函数的特点，左侧具有软饱和性，右侧线性部分可以缓解梯度消失，而左侧软饱能够让 ELU 对输入变化或噪声更鲁棒。ELU 的输出均值接近于零，所以收敛速度更快,如图 4。为了验证本文选用 ELU 函数的正确性，本文选用 100 次迭代训练过程中的损失值作为判断的标准，如图 5，可见 ELU 函数是最佳选择。

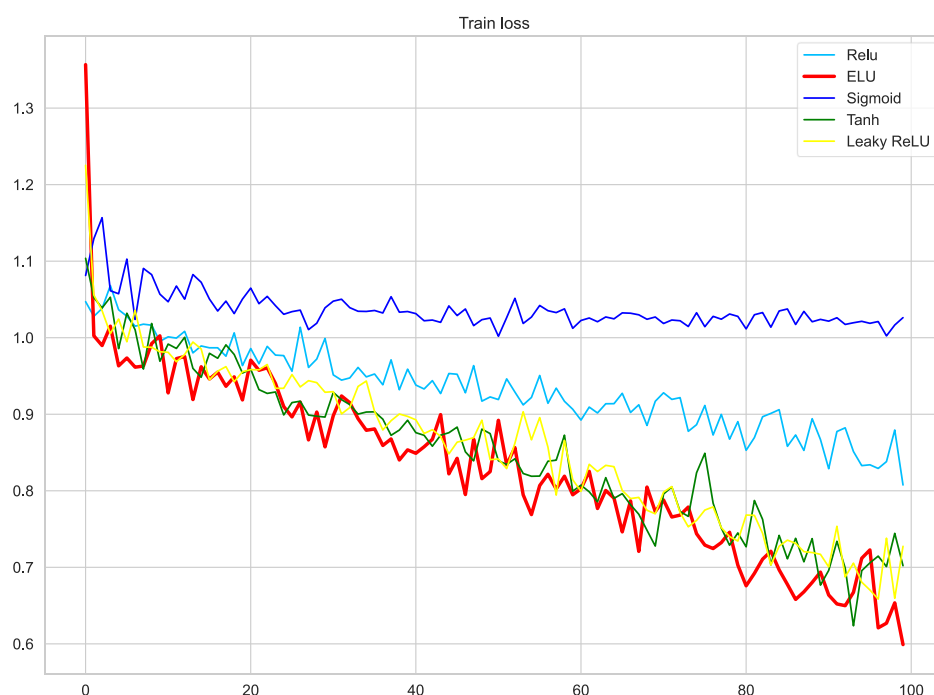


图 5 各函数损失对比图

### 3.5 异步随机梯度下降优化器 Asgd

传统 SGD 算法公式如下：

$$w_{t+1} = w_t - n * g(w_t; x_t, y_t)$$

其中  $w_t$  表示当前模型  $x_t, y_t$  是随机抽样的数据， $g(w_t; x_t, y_t)$  表示  $x_t, y_t$  对应的模型梯度， $n$  表示学习率。

同步随机梯度下降法在优化的每轮迭代中，会等待所有的计算节点完成梯度计算，然后将每个工作节点上计算的随机梯度进行汇总、平均并上面的公式更新模型。之后，工作节点接收更新之后的模型，并进入下一轮迭代。由于 SGD 要等待所有的计算节点完成梯度计算，因此好比木桶效应，SGD 的计算速度会被运算效率最低的工作节点所拖累。

异步随机梯度下降法在每轮迭代中，每个工作节点在计算出随机梯度后直接更新到模型上，不再等待所有的计算节点完成梯度计算，如图 6。因此，异步随机梯度下降法的迭代速度较快，也被广泛应用到深度神经网络的训练中。

ASGD 算法公式如下

$$w_{t+r+1} = w_{t+r} - n * g(w_t)$$

$w_t$  表示当前模型的梯度，模型被其他工作节点更新了  $r$  轮，变成了  $w_{t+r}$ ， $n$  为设定的模型学习率。

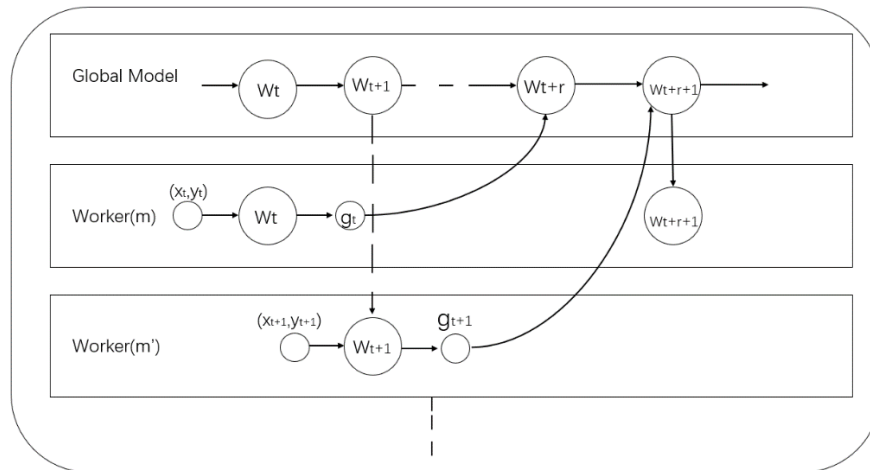


图 6 ASGD 优化器原理

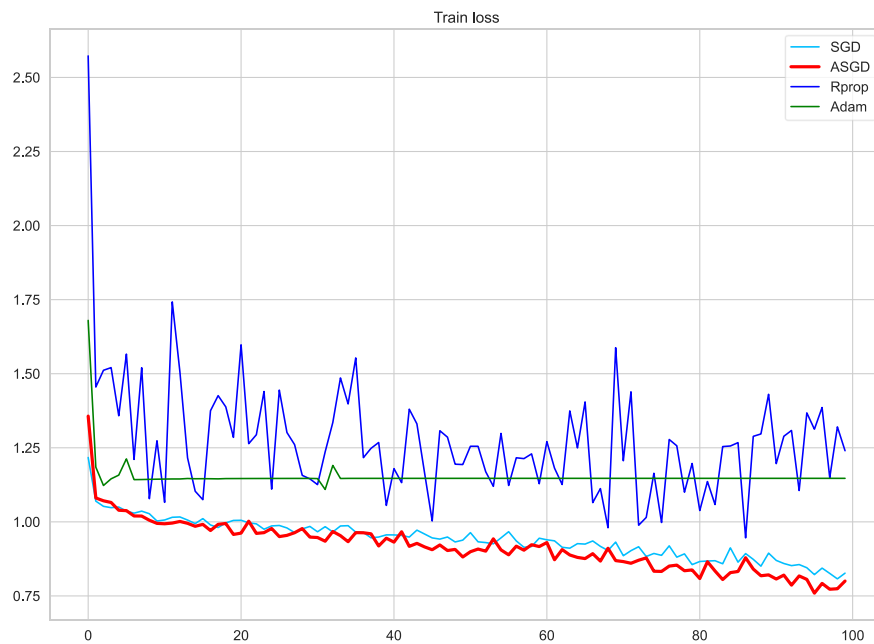


图 7 各优化器效果对比图

本文选用迭代过程中的误差值来描述在不同优化器下的预测效果，并取常见的 SGD、Adam、Rprop 优化器做对比。由图 7 可见，SGD、Adam、ASGD 优化器相对于 Rprop 有着较高的稳定性，SGD、ASGD 远超其他优化器的效果，但 SGD 与被优化过后的 ASGD 相比，其效果还是略逊一筹。故 ASGD 在本文所述的网络结构中可以达到最好的优化效果，极大地减小预测误差。

#### 4 模型对比

为了表现出不同的神经网络算法的预测效果，本文抽样了 10 个地区的数据，对其均方误差统计取平均值，本文采用将四条曲线放在同一张折线图的数据可视化处理方式，如图 8 所示。横坐标表示 10 个区域，纵坐标表示患病人数，绿色实线表示真实数值，绿线表示 BP 神经网络预测的各地数值，深蓝线表示 LSTM 预测的各地数值，红线表示 BPNN-LSTM 混合模型预测的各地数值。实验结果表明单纯 BP 神经网络的均方误差最大，LSTM-BPNN 混合模型误差最小，且不存在小量级数据规模的依赖，故而在可以极佳地避免过拟合问题的同时，也保留着有极强的泛化学习能力，如表 8 所示。

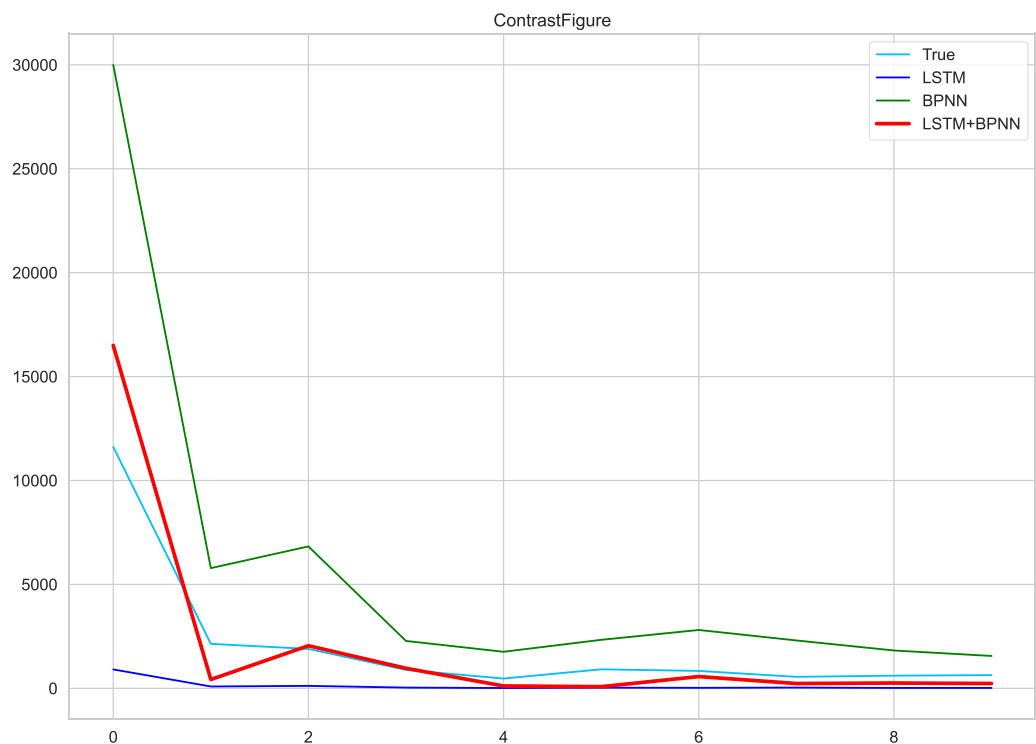


图 8 基于不同神经网络模型的预测折线图

表 2 基于不同神经网络模型的均方差比较（部分）

区域	实际值	LSTM		BPNN		LSTM-BPNN	
		预测值	均方差	预测值	均方差	预测值	均方差
1	11618	909.51	5.73e+07	30000	1.69e+08	16500	1.19e+07
2	2141	88.89	2.11e+06	5784	6.66e+06	425	1.47e+06
3	1897	118.47	1.58e+06	6828	1.22e+07	2050	1.17e+04
4	885	35.96	3.60e+05	2280	9.73e+05	964	3.12e+03
5	471	14.9	1.04e+05	1759	8.29e+05	114.06	6.37e+04
6	915	32.00	3.90e+05	2334	1.01e+06	78.94	3.49e+05
7	838	24.86	3.31e+05	2808	1.94e+06	567	3.67e+04
8	553	36.78	1.33e+05	2304	1.53e+06	231.5	5.17e+04



9	610	19.06	1.75e+05	1821	7.33e+05	254.8	6.31e+04
10	635	18.84	1.90e+05	1558	4.26e+05	231.8	8.13e+04
平均均方误差			6.27e+06		1.95e+07		1.40e+06

为有效展现 BPNN+LSTM 网络结构的泛化性，选取个 6 省份所包含所有地区，分别预测他们的未来爆发情况，用预测值与实际值做对比，用比值的平均值来代表该省份的平均预测精准度。如图，6 个圆环分别代表 6 个省份的预测准确度，由内而外分别表示 BPNN+LSTM、LSTM 和 BPNN。实验分析得出所有地区中 BPNN 对于时序的预测效果最差，LSTM 偶尔可达到 70%以上的精准度，但极不稳定，在不同的省份大量的地区中难以做到准确预测。而 LSTM+BPNN 预测精准度可维持在 70%以上，可见其融合了 BPNN 的稳定性和 LSTM 的精准度的特性。



图 9 基于不同神经网络模型精准对比环形图

## 5 结论

本文提出了基于 LSTM+BPNN 深度神经网络的新冠疫情预测模型，包括对网络模型的参数选择、网络结构、最终预测和模型对比等相关内容。通过实验证明：

1) 与传统的 RNN 模型和 LSTM 模型相比，LSTM+BPNN 深度神经网络结构模型汲取了 LSTM 优秀的拟合时间序列能力，并且因为在 BPNN 中合理选择权值、阈值和更加适当的网络层次结构解决了庞大数据难以精准处理的障碍。

2) 与其他类型的神经网络相比，由于 BP 神经网络需要对大量特征数值分析纠正再学习，LSTM 需要对较长时间序列进行拟合，故而该混合模型在训练过程中也需要更多的时间去调整精度。

3) LSTM+BPNN 深度神经网络模型的均方误差更小更集中，表明与 BPNN、LSTM 模型相比，LSTM+BPNN 深度神经网络预测得出的结果更加准确稳定，更能合理地预测新冠疫情。

LSTM+BPNN 深度神经网络完善了单独 LSTM 和 BPNN 的部分缺陷，经过 LSTM 时间序列的预测得出新的迁入和迁出数据，再用 BPNN 对不同时间、不同地点的两者数据提取精炼得出最终结果。目前尚处于初步研究阶段，该网络结构的优化和其他预测模型相比能否还拥有卓越的性能等问题将成为后续开展工作的主要目标，由此使得该网络能更进一步。

#### 参考文献 (References)

- [1] 张烈平,周德俭,牛秦洲. 基于 BP 神经网络的预测建模系统的研究与实现 [J]. 计算机仿真, 2004, 21(9):48—50.
- [2] 李晓峰,徐玖平,王荫清,等,BP 神经网络自适应学习算法的建立及其应用 [J]. 系统工程理论与实践, 2004, 24
- [3] 王鑫,吴际,刘超,等. 奇异谱分析在故障时间序列分析中的应用 [J]. 北京航空航天大学学报, 2016, 42(11):2321-2331. WANG X, WU J, LIU C, et al. Application of singular spectrum analysis for failure time series [J]. Journal of Beijing University of Aeronautics and Astronautics, 2016, 42(11):2321-2331 (in Chinese).
- [4] TASCİKARAĞLU A, UZUNOĞLU M. A review of combined approaches for prediction of short-term wind speed and power[J]. Renewable
- [5] Kessler M M. Bibliographic Coupling Between Scientific Papers [J]. Journal of the American Society for Information Science & Technology, 1963, 14 (1):10—25.
- [6] 李杰,改进粒子群算法优化支持向量机的工程造价预测 [J]. 计算机系统应用, 2016, 25(6):202—206. [21] 张宇航,邱才明,贺兴,等. 一种基于 LSTM 神经网络的短期用电负荷预测方法 [J]. 电力信息与通信技术, 2017, (9):19—25.