

# 基于RNN的社交网络谣言预测技术研究

崔佳程, 常 乐, 陈佳楠, 林其东, 王良康, 王李冬\*

杭州师范大学钱江学院, 浙江 杭州

Email: \*violet\_wld@163.com

收稿日期: 2021年8月2日; 录用日期: 2021年9月6日; 发布日期: 2021年9月13日

## 摘 要

随着科技的迅速发展与经济实力的卓越进步, 大数据时代在悄然无息中走进我们的生活, 随之而来的是互联网的进步与信息获取的便利, 但是, 谣言也逐渐地在社交平台上泛滥, 严重影响了网络生态环境的健康发展, 使得人们难以在种类繁杂的、质量参差不齐的信息中辨别出可靠的信息, 大大降低了人们获取信息的质量。本文借助RNN与CNN结合的方法实施谣言预测, 即通过CNN将谣言信息向量化, 并输入RNN模型进行谣言的预测。根据给定的数据进行高效的训练, 从而达到及时地、有效地甄别谣言的目的。

## 关键词

谣言预测, 机器学习, 卷积神经网络, 循环神经网络, 深度学习算法

# Research on Social Network Rumor Prediction Technology Based on RNN

Jiacheng Cui, Le Chang, Jianan Chen, Qidong Lin, Liangkang Wang, Lidong Wang\*

Hangzhou Normal University Qianjiang College, Hangzhou Zhejiang

Email: \*violet\_wld@163.com

Received: Aug. 2<sup>nd</sup>, 2021; accepted: Sep. 6<sup>th</sup>, 2021; published: Sep. 13<sup>th</sup>, 2021

## Abstract

With the rapid development of science and technology and the remarkable progress of economic strength, the era of big data has quietly entered our lives, followed by the progress of the Internet and the convenience of obtaining information. However, rumors have gradually spread on social platforms, seriously affecting the healthy development of the network ecological environment. It makes it difficult for people to distinguish the reliable information among the various kinds and

\*通讯作者。

文章引用: 崔佳程, 常乐, 陈佳楠, 林其东, 王良康, 王李冬. 基于 RNN 的社交网络谣言预测技术研究[J]. 数据挖掘, 2021, 11(4): 187-195. DOI: 10.12677/hjdm.2021.114017

uneven quality information, which greatly reduces the quality of people's access to information. The method combining RNN and CNN is used to implement rumor prediction and Python machine learning. We utilize CNN to generate feature vector for the rumor information by CNN model, then input the feature vectors to RNN model to obtain the prediction result. Finally, efficient training is carried out according to the given data, so as to achieve the purpose of timely and effective prediction of rumors.

## Keywords

Rumor Prediction, Machine Learning, Convolutional Neural Network, Recurrent Neural Networks, Deep Learning Algorithm

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着 Web 2.0 与移动互联网的发展和全球综合实力的卓越进步, 网络社交越来越成为人们学习、工作和生活中不可分割的一部分。人们在移动互联网平台中不仅仅成为了信息的接受者, 也成为了内容的创造者。例如百度贴吧、新浪微博、知乎、YouTube、Twitter、Flick 等。人们在这些平台上进行交流、对话和参与的时候, 虽然这大大加快了人与人之间的信息交流和沟通, 但是不容忽视的是, 这些移动互联网平台在给人们提供信息交流和便利互动的同时, 也大大降低了谣言传播的成本, 因此, 这些平台也日渐地成为了在众多谣言传播途径中最合适的场所, 而这些谣言往往相当的真实, 比如日本核电站事故引发的用盐的恐慌事件、各种冒充公安机关发布的文件等等, 或者是近期, 在新冠肺炎疫情防控的关键时期, 在全国人民都为疫情付出心血之时, 网上出现了各种有关新冠肺炎疫情防控的谣言, 造成了社会上错误的认知。这类事件不仅扰乱了市场的正常秩序, 而且还造成了群众的焦虑, 甚至还导致了各地的恐慌现象, 影响可谓是十分恶劣的。

谣言的泛滥, 导致网络生态环境的健康发展受到了严重影响, 这使得人们难以在数量庞大的、样式五花八门的信息中进行分辨并获取到可靠的、有用的信息, 以至于人们鉴别、获取真实信息的可能性大大降低。在面对突发事件的时候, 例如一些涉及到社会安全的事件、卫生安全的事故, 重大自然的灾害等, 人们总是抱有质疑的态度去看待这类事件并从各自的、类似于社交平台这类的渠道去获取并传播貌似真实或虚假的信息。这样, 虽然网络社交媒体的便捷带给了人们无穷的好处, 但这也给谣言的传播提供了无穷的平台, 从而大大增加了谣言传播的深度和速度, 然后极易造成社会的慌乱和暴躁的情绪, 扰乱社会的秩序, 而往往这些负面影响是致命的。因此, 谣言检测不单单是为了还给人们一个良好的、健康的网络环境, 也要能够迅速帮助群众去分辨可靠的、有价值的信息。也正因为如此, 在谣言检测这个问题上已经有较多的研究人员提出了各种算法, 最常见的算法就是基于内容检测的谣言预测方法, 并且取得一定的效果。例如, Castillo 等[1]提出了基于各种特征来预测信息可信度的预测, 包括基于信息内容、基于用户特征、基于传播特征、基于话题等。Qazvinian 等[2]则分别基于内容文本、网络拓扑结构和微博特有的特征来识别谣言。然而, 对于谣言预测而言, 辨别某项信息是否是谣言固然重要, 但能够及时地从移动互联网社交平台中发现谣言从而能够禁止其流传, 然后避免可能带来的不良影响更为重要, 这便是谣言源检测的目的。因此, 谣言源的检测和监控相当重要, 能够有效地控制谣言的传播更为重要。

在百度贴吧、新浪微博、知乎、YouTube、Twitter、Flick 等众多平台中，新浪微博于 2010 年试运行其辟谣平台，并在之后正式推出其人工辟谣平台。根据新浪微博社区管理中心网站数据显示，截止 2020 年 1 月 1 日，共辟谣 38,186 条，换句话说，9 年来，平均每天辟谣 11 条，人工检测的效率可见一斑，因此，微博谣言自动检测逐渐成为研究热点[3]。

本次研究采用基于机器学习理论，提出卷积神经网络和循环神经网络相结合的谣言检测模型，将文本中的谣言事件向量化，通过循环神经网络的学习训练来挖掘表示文本深层的特征，避免了特征构建的问题，并能发现那些极其不容易被人发现的特征，从而达到更好的效果。

## 2. 算法描述

### 2.1. 数据预处理

本次研究采用的数据接口是来自新浪微博不实信息辟谣专区，并从此网页中爬取了中文谣言数据。在这些数据中，包含了一些与本次研究不相干的数据，所以接下来要进行聚焦爬虫操作。

#### ①提取网页内容

随着互联网的发展与 WEB 2.0 的到来，爬取的页面绝大多数都是动态页面，这就涉及到了 AJAX 技术。AJAX 技术可以看成是一种脚本，一种延时加载、异步更新的技术，并在特定的时候以少量数据的交换而进行部分页面的更新。而更新过后，爬取的数据更具有一定的时效性、真实性，这对聚焦爬虫有着巨大的好处。

#### ②爬取网页文本

在爬取的文本中，绝大多数都是带有 HTML 超文本标签，比如<head>、<title>等等，其中<div></div> 标签对里面的内容正是本次项目所需要的数据集。而为了后面方便进行数据的应用，本次研究采用将 HTML 语言转换成 JSON 语言(JSON 数据结构类似于 Python 里面的字典类型的数据结构)。

#### ③数据向量化

在数据集全部转化成 JSON 语言后，将键值对中我们需要的值进行提取特征，而这些特征必须能够最大化数据的特征，能够完全表现该数据的特性。

### 2.2. 卷积神经网络 CNN

卷积神经网络(Convolutional Neural Network, CNN)，人们通常情况下会把它看成是一种人工神经元，其神经元之间使用类似动物视觉皮层组织的链接方式，大多数情况下用于处理计算机视觉相关的任务，例如分类、分割、检测等。卷积神经网络最先出现于 20 世纪 80 年代到 90 年代，LeCun 提出了 LeNet 用于解决手写数字识别的问题，随着深度学习理论的不完善，计算机硬件水平的提高，卷积神经网络也随之快速发展。

卷积神经网络是近几年才发展起来的一种前馈神经网络，并成为了一种高效的识别方法。与其它的网络算法相相似，在训练时会使用梯度下降法等对参数进行修改、更新，因此所有的输入都需要进行在通道或时间维度归一化或标准化的预处理过程，其中，归一化是指通过计算极值将所有样本的特征值映射到层与层之间。而标准化则是通过计算均值、方差将数据分布转化为标准正态分布。

一般地，卷积神经网络通常由一个输入层(Input Layer)和一个输出层(Output Layer)以及多个隐藏层组成。隐藏包括卷积层(Convolutional Layer)、激活层(Activation Layer)、池化层(Pooling Layer)以及全连接层(Fully-connected Layer)等。

在本项目中，卷积神经网络用于前期神经网络框架的搭建。具体如下：因为文本的顺序是从上到下、从左到右的顺序，所以实验用到的向量为行向量。一般地，此行向量取五个字为标准，即以某个字为中

心, 分别向前和向后取两个字作为单个字向量的前缀和后缀, 以此组成基本行向量并作为卷积层每次迭代的输入。

### 2.3. 循环神经网络 RNN

在 20 世纪 80~90 年代, 科学家们开始研究循环神经网络, 它是一类以序列数据为输入, 在序列的演进方向进行递归且所有节点(循环单元)按链式连接的递归神经网络。到了 21 世纪初, RNN 发展为深度学习算法之一[4], 其中双向循环神经网络(Bi-RNN)和长短期记忆网络(LSTM)是常见的循环神经网络[5]。

循环神经网络具有记忆性、参数共享并且图灵完备(Turing completeness), 因此在对序列的非线性特征进行学习时具有一定优势。循环神经网络在自然语言处理(Natural Language Processing, NLP), 例如机器翻译、语音识别、语言建模等领域有应用, 也被用于各类时间序列预报[6]。引入了卷积神经网络构筑的循环神经网络可以处理包含序列输入的计算机视觉问题。

RNN 与卷积神经网络相结合的常见例子是循环卷积神经网络(Recurrent CNN, RCNN)。RCNN 将卷积神经网络的卷积层替换为内部具有递归结构的循环卷积层(Recurrent Convolutional Layer, RCL), 并按前馈连接建立深度结构[7]。

除 RCNN 外, RNN 和卷积神经网络还可以通过其它方式相结合, 例如使用卷积神经网络在每个时间步上对序列化的格点输入进行特征学习(time-distributed), 并将结果输入 RNN [8]。

循环神经网络 RNN 与卷积神经网络 CNN 相比, 其总参数量大大降低了, 这是因为循环神经网络 RNN 的具有权重共享的性质, 这种性质表示循环神经网络 RNN 的权重系数具有共享性, 即在每一次的迭代中, 循环节点使用相同的权重系数处理所有的时间步。相比于前馈神经网络, 权重共享降低了 RNN 的总参数量。权重共享也意味着循环神经网络 RNN 可以提取序列中随时间变化的特征, 因此其在学习和测试序列具有不同长度时可表现出泛化能力[9]。

作为时间序列模型, 循环神经网络 RNN 也具有这样的性质, 即在时间序列建模的观点下, RNN 是一个无限冲激响应滤波器(Infinite Impulse Response filter, IIR) [9]。这将 RNN 与其它应用于序列数据的权重共享模型, 例如一维的卷积神经网络相区分, 后者以时间延迟网络为代表, 是有限冲激响应滤波器(Finite Impulse Response filter, FIR) [9]。

RNN 的大致流程如下:

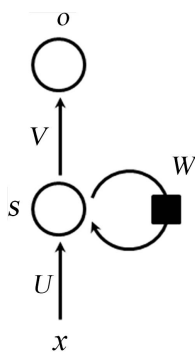


Figure 1. Schematic diagram of RNN basic principle  
图 1. RNN 基本原理示意图

在图 1 中,  $x$ 、 $s$ 、 $o$  均为向量, 分别代表着 RNN 的输入层、隐藏层和输出层的结果值,  $U$  表示从输入层到隐藏层的权重系数矩阵,  $V$  表示从隐藏层到输出层的权重系数矩阵,  $W$  表示距离现在最近的上一时刻输入的权重系数矩阵。其中前一刻的输出  $x$ 、 $s$ 、 $o$  均作为下一时刻的输入, 即

$$x_t = x_{t-1} \& x_{t+1} = x_t \quad (1)$$

$$s_t = s_{t-1} \& s_{t+1} = s_t \quad (2)$$

$$o_t = o_{t-1} \& o_{t+1} = o_t \quad (3)$$

这就表明，前一次迭代的结果必将在后一次迭代中体现出来。因此有隐藏层计算公式：

$$s_t = f(Ux_t + Ws_{t-1} + b) \quad (4)$$

其中，激活函数  $f$  通常情况下为  $f(x) = \tanh(x)$  ( $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ )， $U$  为输入层  $x_{t-1}$  到隐藏层  $s_t$  的权重系数矩阵， $W$  为距离现在最近的上一个时刻的隐藏层  $s_{t-1}$  到当前时刻隐藏层  $s_t$  的权重系数矩阵， $b$  为偏差向量。这体现了神经网络之循环，也体现出了与卷积神经网络最大的不同。亦有输出层计算公式：

$$o_t = f(Vs_t + c) \quad (5)$$

其中， $V$  是隐藏层  $s_t$  到输出层  $o_t$  的权重系数矩阵， $c$  是偏差向量，激活函数  $f$  通常情况下为归一化函数 (Softmax 函数)，其输入是从  $K$  个不同的线性函数得到的结果，而样本向量  $x$  属于第  $i$  个分类的概率的计算公式：

$$f(x) = P(x = i) = \frac{e^{x^T W_i}}{\sum_{k=1}^K e^{x^T W_k}} \quad (6)$$

这样，只需将  $Vs_t + c$  代入到上式中的  $x$  即可得到结果。

在本项目中，为了更好地体现实验的高效性，将数据预处理后的数据加入到 RNN 训练的序列中，即作为输入向量  $x$ ，然后对每个输入向量  $x$  添加权重得到输入权重向量  $s$ 。此时，向量  $s$  需要添加与距离现在最近的上一个时刻输入的权重系数(此为 RNN 特点)。同样的， $V$  为向量  $s$  需要添加的权重， $o$  为最后的输出。另外，每次都要将预测值与真实值进行比较来得到损失与精确度。这里多次提到权重，权重应该如何选取便成了一个大问题。但通过大量的实验发现，多个损失函数，数值会在某一个邻域内上下波动(基本上大致接近或是几乎无区别)。就像一些国际单位及其换算，在其数值上最多只是差了几个数量级，此时，加权变得毫无意义。为了解决这个问题，解决思路应该从损失函数角度考虑：

#### ①损失函数度量单位

每个损失函数的度量单位不尽相同。在其进行归一化处理时，应注意将损失函数的输入尽量保证量纲一致。比如如果是用正、余弦夹角计算的，本身就是 0~1 的，这时，不用做任何处理。如果是欧式距离，则对欧式距离进行归一化处理，将 Softmax 函数(不带 loss)作为损失函数时，在其进行归一化期间，在 caffe 的欧式距离层里面可以直接进行归一化处理。

#### ②损失函数本身

比如多个超参数之间的联系，多个 batch 的总和或平均。这种情况必须经过大量的实验才能比较出具体超参数取多少值才会达到加权最优的情况。

### 3. 实验材料

本项目数据集采用经数据预处理阶段过后的数据，此数据集是公开的微博谣言预测数据集。为了保证实验的可靠性、有效性，实验集中爬取了大量需要的信息，经预处理，数据集中包含 1538 条谣言数据、1849 条非谣言数据，这些数据具有真实性、可靠性的特点，足以供实验需要。通过这些数据集，生成数据字典，并将训练所用到的数据的每个字转换成数据字典中的数字，即数据中的每个字与数据字典中的每个编号一一对应。然后定义一个数据读取器。再应用循环神经网络 RNN 与卷积神经网络 CNN 结合的

训练模型进行模拟训练。最后给定特定的信息进行真实的预测。

1、在本项目训练阶段之前，构建单池化层，包括以下几个参数：

①num\_channels: 通道数(The number of channels in the input text.)

②num\_filters: 卷积核数量(The number of filter. It is as same as the output feature map.)

③filter\_size: 卷积核大小(The filter size. If filter\_size is a tuple, it must contain two integers, (filter\_size\_H, filter\_size\_W). Otherwise, the filter will be a square.)

④batch\_size: 批次大小(一般为 16)

⑤act: 激活函数(Activation type, if it is set to None, activation is not appended. Default: None.)

⑥pool\_size: 池化层大小(The pool kernel size. If pool kernel size is a tuple or list, it must contain two integers, (pool\_size\_Height, pool\_size\_Width). Otherwise, the pool kernel size will be a square of an int.)

⑦pool\_type: 池化层类型(The pooling type, can be “max” for max-pooling and “avg” for average-pooling. Default: max.)

⑧pool\_stride: 步幅(The pool stride size. If pool stride size is a tuple or list, it must contain two integers, (pool\_stride\_Height, pool\_stride\_Width). Otherwise, the pool stride size will be a square of an int. Default: 1.)

2、构建完池化层以后，搭建 CNN 神经网络，包括一下几个参数：

①dict\_dim: 数据字典大小(4409)

②emb\_dim: 纬度

③hid\_dim: 卷积核数量(一般为 32)

④fc\_hid\_dim: fc 参数纬度

⑤class\_dim: 分类数

⑥channels: 输入通道数

⑦win\_size: 卷积核尺寸

⑧batch\_size: 批次大小

⑨seq\_len: padding 纬度

⑩embedding: 嵌入(It implements the function of the Embedding Layer. This layer is used to lookup embeddings vector of ids provided by input . It automatically constructs a 2D embedding matrix based on the input size (vocab\_size, emb\_size) and dtype.)

3、在本项目的训练阶段中，包含了诸多超参数：

①epoch: 训练轮次

②batch\_size: 批次大小

③adam: 学习率

④padding\_size: padding 纬度

⑤vocab\_size: 字典大小

⑥skip\_steps: 每 N 个批次输出一结果

⑦save\_steps: 每 M 个批次保存一次

⑧checkpoints: 保存路径

最后进行模型训练与评估即可。

## 4. 实验测试

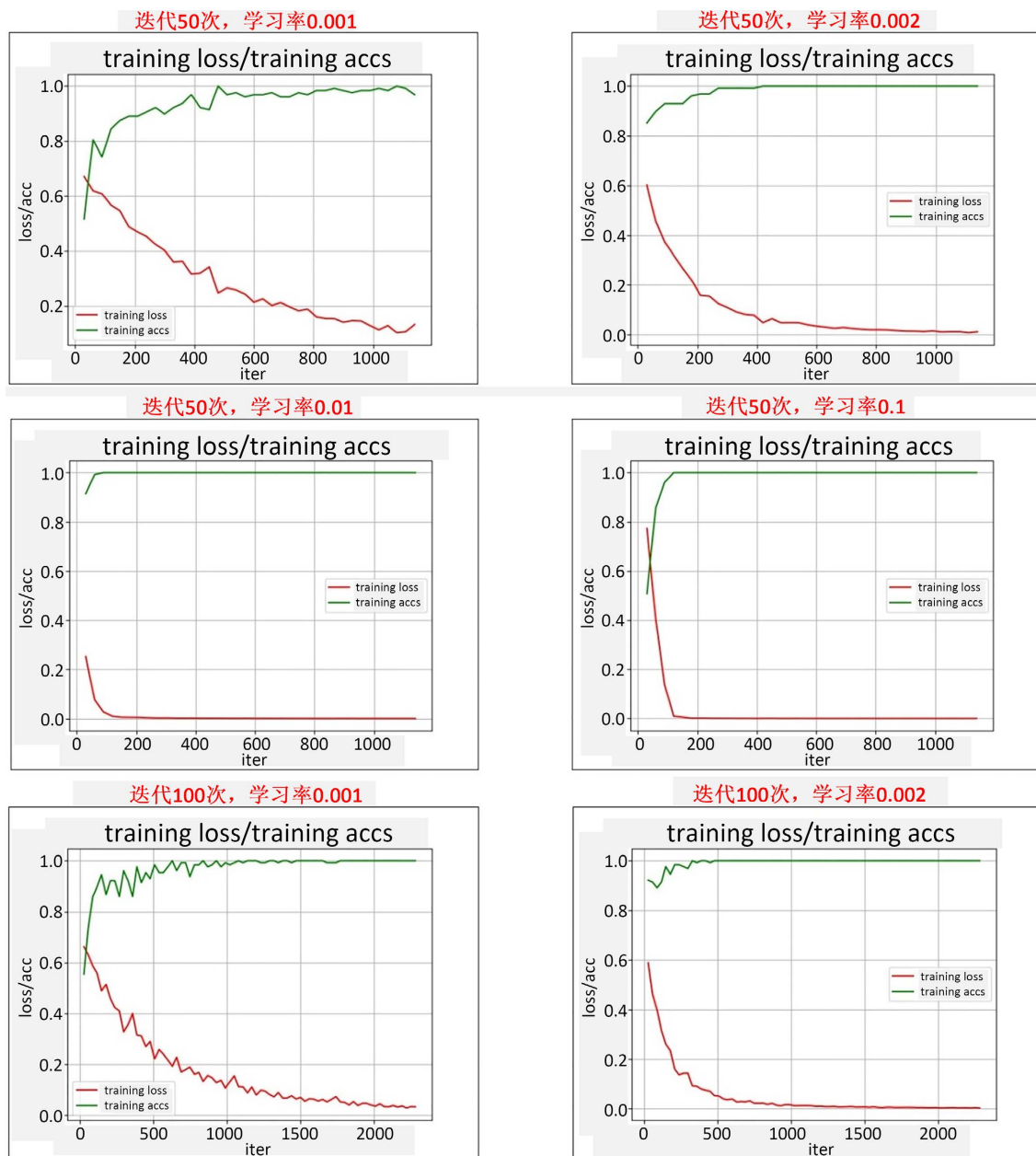
正式模型训练之前一定要先得到数据字典和数据列表。具体结果如图 2 所示：

```

谣言数据总量为: 1538
非谣言数据总量为: 1849
all_data.txt已生成
数据字典生成完成!      字典长度为: 4409
数据列表生成完成!
    
```

**Figure 2.** Generate data dictionaries and data lists  
**图 2.** 生成数据字典和数据列表

在实验过程中多次调整超参数发现：参数不同会影响结果的收敛速度不同。从图中可以看出，迭代的次数、学习率等的不同导致了精度和损失的截然不同。学习率越低，收敛的速度越慢，收敛的时间也越长。在相同学习率的情况下，迭代次数越多，则准确率越高。具体结果如图 3 所示：



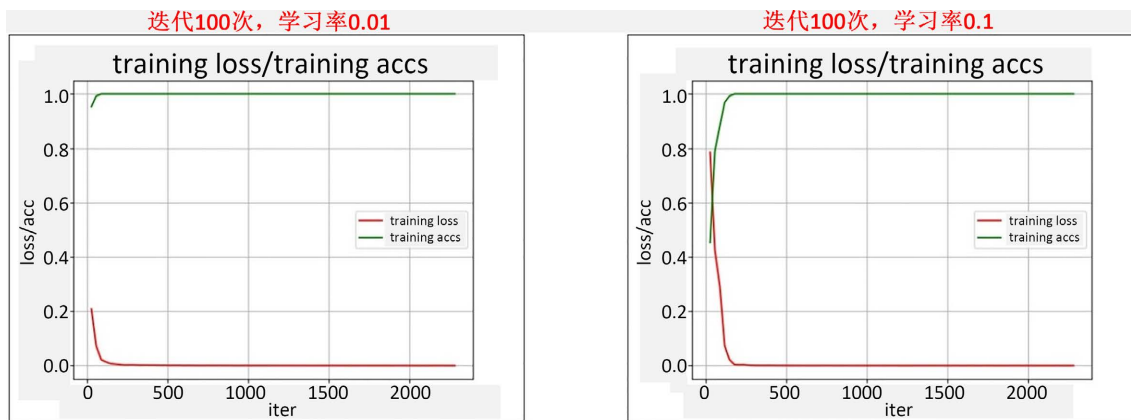


Figure 3. The Final Result

图 3. 最终结果

表 1 详细记录了各个超参数的调整与损失/精度之间的联系与区别。

Table 1. The relation and difference between the adjustment of each hyper-parameter and the loss/accuracy

表 1. 各个超参数的调整与损失/精度之间的联系与区别

训练轮次	学习率	训练最后一次损失/精度	实际损失/精度
50	0.001	0.121208/0.992188	0.545780/0.851562
50	0.002	0.013035/1.000000	0.360584/0.869792
50	0.01	0.000286/1.000000	0.329458/0.867188
50	0.1	0.000216/1.000000	0.541900/0.851562
100	0.001	0.033631/1.000000	0.535068/0.841146
100	0.002	0.003107/1.000000	0.347282/0.864583
100	0.01	0.000123/1.000000	0.304642/0.877604
100	0.1	0.000031/1.000000	0.573150/0.835938

由表 1 得，纵观全局，实际精度要比训练精度要低，但损失越少，精度越高。同时，并不是学习率越高、迭代次数越多，最后的精度越高。这是因为，当学习率很高但训练次数不多的情况下，训练期间计算机所学到的内容过少，这就导致了计算机处理数据时出错的概率大大增加，损失也随之增高，最后精度更低，反之亦反。

重新进行实验，以表 1 中训练次数 50 次，学习率 0.001 为标准，最终预测结果如图 4 所示。

```
Final validation result: ave loss: 0.549042, ave acc: 0.830729
预测结果为: 谣言
```

Figure 4. Final validation result

图 4. 最终验证结果

## 5. 结语

当今世界，随着互联网的发展与人们生活经济实力的显著提升，信息传播的速度急速增长，谣言大肆兴起，而这种类似于 RCNN 的算法几乎可以做到对谣言进行预测，实验结果与真实数据虽有些出入，但大方向上高度一致，说明 RNN 与 CNN 的结合模型对于谣言预测来说非常适合。本项目需要改进的地



方依旧很多,比如如何提高在实际预测中的精确度等。这些可以在以后的实验中进行进一步的讨论与研究。

## 基金项目

浙江省自然科学基金项目(LY19F020022);杭州师范大学钱江学院师生共研专项项目(2021QJXS01);2021年国家级大学生创新创业训练计划立项项目(基于表示学习的多社交平台谣言源发现模型研究);杭州师范大学2021年“星光计划”大学生创新创业项目(基于表示学习的多社交平台谣言源发现模型研究);杭州师范大学钱江学院科研项目(面向实时交通信息的行车轨迹分析与路径规划研究)。

## 参考文献

- [1] Castillo, C., Mendoza, M. and Poblete, B. (2011) Information Credibility on Twitter. *International Conference on World Wide Web (WWW'11)*, Association for Computing Machinery, New York, NY, March 2011, 675-684. <https://doi.org/10.1145/1963405.1963500>
- [2] Qazvinian, V., Rosengren, E., Radev, D.R., et al. (2011) Rumor Has It: Identifying Misinformation in Microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, Association for Computational Linguistics, USA, 1589-1599.
- [3] 尹鹏博, 彭成, 潘伟民. 基于集成学习的微博谣言早期检测[J]. 微电子学与计算机, 2021, 38(1): 83-88.
- [4] Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. *Neural Networks*, **61**, 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [5] Ng, A., Kian, K. and Younes, B. (2018) Sequence Models, Deep Learning. Coursera and Deeplearning.ai.
- [6] 广发证券股份有限公司. 一种基于音素联想及深度学习的语义理解方法及装置: CN202010919954.8 [P]. 2021-01-05.
- [7] Kim, Y., Jernite, Y., Sontag, D. and Rush, A.M. (2016) Character-Aware Neural Language Models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, AAAI Press, 2741-2749.
- [8] Miljanovic, M. (2012) Comparative Analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction. *Indian Journal of Computer Science and Engineering*, **3**, 180-191.
- [9] 邱锡鹏, 著. 神经网络与深度学习, 第六章 循环神经网络[M]. San Francisco, CA: Github Inc., 2020.