

基于股价和评论的股票预测集成深度学习模型

刘文龙, 陈开枝, 陈飞洋, 刘金鑫, 赵嘉萱, 高新飞

ABSTRACT 各种深度学习模型已成为股票预测中的常用方法。然而, 股票价格会受到很多因素的影响, 各种深度学习模型的精度受其影响很大。现有的基于深度学习的股票预测通常直接将开盘价、最高价和最低价等常见的股票价格特征作为模型输入特征来直接学习得到预测模型, 这种方法可以在一定程度上预测股票走势, 但模型精度较低。此外, 已有学者尝试增加情感特征来提高模型的预测精度。为了探究不同深度学习模型在股票预测中的预测精度, 我们使用了不同的深度学习模型并融合情感特征来进行股票预测, 除此之外, 我们还尝试了利用集成学习来集成不同的深度学习模型以进一步提高模型预测精度。大量实验表明, 与一般的机器学习模型相比, 融合了情感特征的机器学习模型在预测精度上提高了约 3.5%, 集成学习使得预测精度提高了约 26%, 验证了情感融合和集成学习对股票预测模型精度提升的有效性。

INDEX TERMS 股票预测; 深度学习; 情感融合; 集成学习

I. 引言

许多因素会以各种方式影响股票价格。股票价格的变化受市场力量的影响, 这意味着股票价格的变化受股票市场供求关系的影响。供应因素包括公司发行股票(例如, 发行新股给投资者)、股票价格(例如, 发行新股给投资者)和股票价格(例如, 发行新股给投资者)。需求因素包括公司新闻(如新产品发布、未完成目标、业绩良好)、经济因素(如利率变化)、行业趋势(如行业蓬勃发展)。需求因素包括公司新闻(如新产品发布、未完成目标、业绩良好)、经济因素(如利率变化)、行业趋势(如行业蓬勃发展)、市场情绪(可能是心理和主观因素)和突发事件(如自然灾害或政府领导人去世)。通常, 我们可以从财经新闻、公司通讯或年报中获取这些供求因素。例如, 当苹果公司发布一款新产品时, 很多人都想购买, 而该产品的性能通常很快就会得到改善。因此, 更多的人对苹果股票感兴趣, 苹果股票的需求就会增加, 从而导致苹果股价上涨, 这就是需求定律。此外, 人们的情绪或信念也会对股价产生影响。政治局势或国际事务也会影响股票价格。因此, 这是一个股票供求和价格之间的复杂过程。不过, 影响股票供求的主要因素有几个, 如公司新闻、公司业绩、行业表现、投资者情绪(如是否处于牛市或熊市)以及其他主要经济因素。如果我们将注意力集中在这些主要因素上, 并追溯历史股价, 我们或许能相当准确地预测未来股价。人们对股票因素的记忆通常较短。因此, 确定合适的历史窗口大小对于正确预测股价非常重要。如果窗口大小与人的记忆力相比过大, 很多因素或新闻就会被投资者遗忘, 早已过时, 预测效果也不会好。因此, 对于计算机科学家来说, 利用人工智能预测未来的股票走势是一项非常具有挑战性的工作, 因为计算机很

难接收到最新信息并立即做出反应, 由于股价与许多因素相关, 仅考虑一两个因素无法提供足够的信息来预测股价走势。尽可能多地纳入相关的有用信息, 才能保证更好的预测效果

1. 我们采用情感分析法从推特评论中获取情感分数。
2. 我们使用多种现代深度学习架构, 根据情感分数和历史数据进行预测。
3. 我们利用集合神经网络来融合决策, 从而进一步改善预测结果。

主要步骤如下: 本文提出利用情感分析从已公开的数据源中提取有用信息, 并利用混合集合深度学习模型预测未来股票走势。混合集合模型包含两个层次。第一层包含两个递归神经网络, 第二层模型是一个全连接神经网络。RNN、LSTM、GRU 模型能有效捕捉输入数据中的时间序列事件, 而全连接神经网络则用于集合多个单独的预测结果, 以进一步提高预测精度。

II. 相关工作

本节从现有的深度学习和情感分析在股票预测中的应用出发, 概述了与本文相关的研究进展。

深度学习技术, 如LSTM和CNN, 已在股票价格预测中得到广泛应用, 这一点在多篇研究中得到了证实。^[1]董玲展示了深度学习如何通过分析大量历史数据来识别股价趋势。类似地, 廖畅^[2]和Xinyi Li^[3]这两位将情感分析融入深度学习模型中, 以提高对市场情绪变化的预测能力。

在股票市场预测中, 结合情感分析已成为提高模型准确度重要方法。例如, 廖畅^[2]和Xinyi Li^[3]这两位论文强调了如何利用新闻和社交媒体中的情绪信息。这种融合方法不仅考虑了传统的股价数据, 还包括了能够反映公众情绪和市场趋势的非结构化数据。通过分析这些数据, 模型能

够更全面地理解市场动态，从而在预测股票价格时提供更高的准确性和可靠性。

集成模型在股票价格预测方面的创新主要体现在结合多种预测模型和数据源，以提高预测的准确性和鲁棒性。Ghosh^[4]和Frank^[5]等人展示了通过融合不同的深度学习技术，如LSTM和随机森林，以及结合多源数据，包括市场情绪分析，能够显著提高预测模型的综合性能。这种集成方法优于单一模型，因为它能更全面地捕捉市场的复杂性和动态变化，从而为股票预测提供更精确的视角。

本文研究基于现有研究，结合深度学习、情感分析以及多元数据集成方法，为股票价格预测提供了全新视角。我们研究了市场情绪在资产配置问题中的作用。这种综合性方法充分考虑了市场的动态性和复杂性，通过深度学习模型处理结构化的股价数据，同时融入来自社交媒体和新闻评论的情感分析，增强了对市场情绪的洞察力。这些创新的集成策略显著提升了预测的准确性和可靠性，展示了在复杂金融市场环境中，多技术和多数据源结合的强大潜力。

III. 方法

A. LSTM网络

在股票市场预测领域，LSTM展现了处理长期时间序列数据的独特优势。Hochreiter和Schmidhuber首次提出的LSTM模型成功解决了传统递归神经网络中的梯度消失问题，为复杂时间序列的处理提供了理论基础^[6]。Fischer和Krauss进一步证实了LSTM在金融市场预测中的有效性，特别是在日交易预测方面^[7]。Nelson等人通过实证研究展示了LSTM在金融时间序列分析中的应用价值^[8]。同时，Selvin等人强调了模型结构和超参数在股市预测中的重要性^[9]。

LSTM是一种特别设计的循环神经网络，用于解决标准RNN在处理长期依赖时的梯度消失问题。它通过特殊的结构设计，使网络能够在长时间间隔内保留信息，从而更有效地学习和预测时间序列数据中的长期依赖关系。我们通过分析它的内部结构可以更加清晰的了解LSTM的特点。

1) LSTM的总体架构

LSTM网络的内部结构和细胞状态更新过程是其处理长期依赖性问题的关键。LSTM通过一系列门控制机制（遗忘门、输入门和输出门）来精确地调节信息流，这样不仅可以决定在每个时间步骤中保留或遗忘的信息，还能决定要加入到细胞状态的新信息量。这种动态更新细胞状态的能力使得LSTM在诸如股票市场预测等涉及复杂时间序列分析的任务中表现出色^[6]。

LSTM网络单元结构如图1所示，接下来我们会分别从三个门及细胞状态更新四个方面对LSTM进行分析。

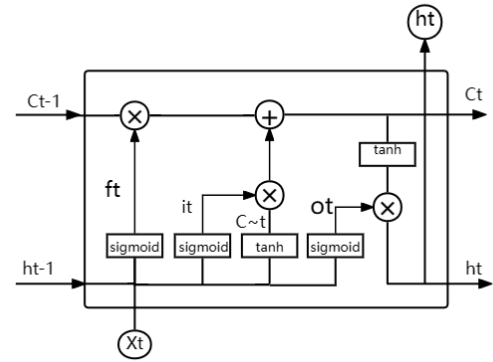


FIGURE 1. LSTM网络单元结构

2) 遗忘门

LSTM网络的关键创新之一是引入了遗忘门机制。这个机制允许模型决定何时丢弃或保留细胞状态中的信息。遗忘门通过评估当前输入和前一隐藏状态来生成一个0到1之间的值，这个值代表保留每个细胞状态信息的程度。值接近1意味着保留该信息，而接近0则表示丢弃，使得LSTM能够根据新输入动态更新其内部记忆。这一特性显著增强了LSTM处理长期依赖性问题的能力^[6]。

遗忘门的结构如图2所示。

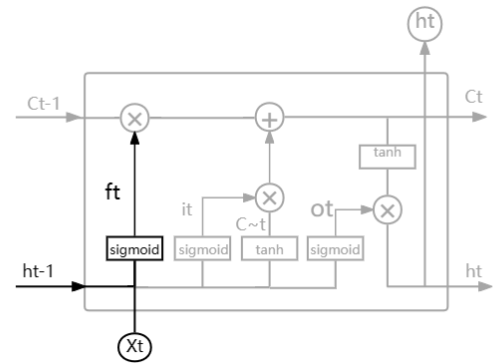


FIGURE 2. 遗忘门结构

相关公式

$$[W_f] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} = [W_{fh} \quad W_{fx}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} = W_{fh} h_{t-1} + W_{fx} x_t \quad (1)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

其中， f_t 表示遗忘门的输出； σ 是sigmoid激活函数； W_f 和 b_f 分别是遗忘门的权重和偏置； x_t 为当前时间步输入的信息； h_{t-1} 为前一时间步的隐藏状态。

3) 输入门

输入门负责控制新信息的进入。它通过评估当前输入和之前的隐藏状态，决定保留多少新信息并添加到细胞状态中。这个机制使得LSTM不仅能够记忆过去的信息，还能够根据新数据动态更新其状态。对于LSTM来讲它得到的是当前的细胞状态，而不是像经典RNN一样得到的是隐含状态。输入门结构如图3所示。

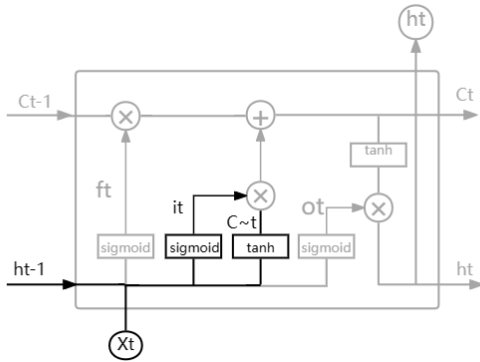


FIGURE 3. 输入门结构

其公式为:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \text{TANH}(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

其中 W_i 是输入门的权重; b_i 是输入门的偏置; i_t 是输入门的输出, 决定要添加到细胞状态的新信息量; \tilde{C}_t 新信息的候选值, 由TANH激活函数生成。

4) 输出门

LSTM网络的输出门在决定何时将网络的内部状态传递到下一个时间步骤中起着关键作用。输出门根据当前的输入和先前的隐藏状态来评估, 决定细胞状态中哪些信息应该输出并用于生成当前的隐藏状态。这使得LSTM网络能够基于过去和当前的信息进行有效的决策和预测, 整个输出门的过程就是为了产生隐含状态。

输出门结构如图FIG.4

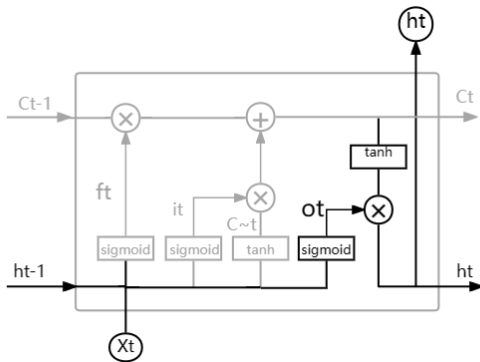


FIGURE 4. 输出门结构

其公式为:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

其中 o_t : 输出门的输出, 决定隐藏状态的哪部分信息要被输出; h_t : 当前时间步的隐藏状态, 由输出门和当前细胞状态共同决定; W_o 、 b_o : 分别为输出门的权重和偏置。

5) 细胞状态更新

细胞状态更新涉及遗忘门、输入门和输出门的协同作用。这些门控制信息流, 使网络能够根据当前输入和之前的状态信息来更新其细胞状态。这一机制确保LSTM在处理时间序列数据时, 如股票市场分析, 能够有效地记住重要信息并忽略不相关信息。

细胞更新的结构是将当前时间步得到的遗忘门门值与上一个时间步得到的 $C(t-1)$ 相乘, 再加上输入门门值与当前时间步得到的未更新 $C(t)$ 相乘的结果。最终得到更新后的 $C(t)$ 作为下一个时间步输入的一部分。整个细胞状态更新过程就是对遗忘门和输入门的应用如图Fig.5。

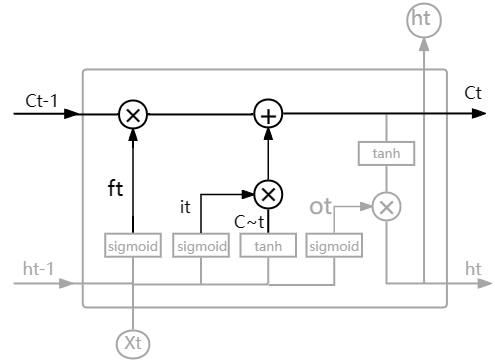


FIGURE 5. 细胞状态更新结构

其公式为:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (7)$$

其中, C_t 为当前时间步的细胞状态, 会参与到下一时间步的计算中

B. CNN

卷积神经网络 (CNN) 是一类用于处理图像和空间数据的神经网络。不同于传统的前馈神经网络, CNN通过使用卷积层和池化层来有效地捕捉输入数据中的局部空间结构。这种网络结构特别适用于处理图像、视频以及具有网格结构的数据。CNN的核心思想在于局部感知和参数共享, 通过卷积操作可以提取输入数据中的特征, 而池化层则用于降低空间维度和减少计算量。

卷积神经网络的关键优势在于它能够自动学习图像的空间层次特征, 而无需手动设计特征提取器。每个卷积层通过滑动卷积核来检测图像中的局部模式, 这种局部感知性使得CNN能够有效处理具有空间相关性的数据。此外, 卷积神经网络通过参数共享减少了需要训练的参数数量, 使得模型更具有泛化能力。

1) 输入层

卷积神经网络 (CNN) 的输入层包含两个主要部分, 空间步和特征表示。

空间步 (Spatial Step): CNN的输入数据是在空间中具有结构的图像或空间数据。在每个空间步, 网络接收图像中的一个局部区域 (通常是一个滤波器大小的窗口) 的信息, 以滑动窗口的方式在整个输入数据上进行操作。

特征表示 (Feature Representation): 每个空间步的输入数据 (局部区域) 会被转换成一个特征表示的向量。这个向量通常是通过卷积操作提取的局部特征, 用于表示图像

中的纹理、形状或其他视觉信息。这一步的目的是捕捉输入数据的局部模式。

2) 卷积层

卷积层通过使用卷积核对输入数据进行卷积操作。卷积核是一个小的窗口，通过滑动在输入数据上，与输入数据的局部区域进行点乘操作。卷积核的大小和权重参数是可以学习的，使得网络能够自动学习到数据中的特征。卷积操作的结果是生成一个特征图，也称为激活图。特征图表示了输入数据中的不同特征在空间上的分布。一个卷积层可以包含多个卷积核，每个卷积核生成一个独立的特征图，从而捕捉输入数据的不同方面。

在卷积神经网络（CNN）的卷积过程中，假设输入数据为二维矩阵 X ，卷积核为二维矩阵 K ，步幅为 S ，填充为 P ，输出特征图为 O ，则卷积操作的计算公式为：

$$O[i,j] = \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} X[i \times S + m, j \times S + n] \cdot K[m,n] + b \quad (8)$$

其中， i,j 是输出特征图的坐标； F 是卷积核的大小； S 是步幅； b 是偏置（可选）。

3) 激活层

卷积神经网络（CNN）的激活层在每个卷积层之后，负责引入非线性变换，使得卷积神经网络能够学习到更为复杂的模式和特征。激活函数通常被应用在卷积操作的输出上，提高网络的拟合能力。通过激活函数，网络中的神经元在某些位置上可以被激活，而在其他位置上则保持非激活状态。常见的激活函数有：

ReLU（Rectified Linear Unit）：

$$f(x) = \max(0, x) \quad (9)$$

ReLU是最常用的激活函数之一，它在输入为正时输出与输入相同，而在输入为负时输出为零，具有简单的计算和导数。

Sigmoid：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

Sigmoid函数将输入映射到0到1的范围内，常用于二分类问题的输出层。

Tanh（双曲正切）：

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (11)$$

Tanh函数将输入映射到-1到1的范围内，与Sigmoid类似，但具有更广的输出范围。

Leaky ReLU：

$$f(x) = \max(\alpha x, x) \quad (12)$$

其中 α 是小于1的小常数。

Leaky ReLU在输入为负时不直接输出零，而是乘以小的负斜率，以避免死亡神经元问题。

4) 池化层

池化层用于减小特征图的空间尺寸，降低计算复杂性，并且提取输入数据的关键信息。池化层通过对输入数据的局部区域进行池化操作，实现了下采样的效果，减少网络的参数数量。并且由于其具有平移不变性，有助于使网络对输入的某些变化更具鲁棒性。常见的池化方式有最大池化和平均池化。

池化操作的计算公式为：

$$O[i,j] = \max(X[i \times S : i \times S + F, j \times S : j \times S + F]) \quad (13)$$

其中： $O[i,j]$ 是池化后的输出特征图的元素； F 是池化窗口的大小； S 是步幅。

平均池化操作的过程是在每个池化窗口中计算局部区域内的平均值。下面是平均池化的具体步骤：

假设我们有一个输入矩阵 X ，平均池化窗口的大小为 $F \times F$ ，步幅为 S 。那么，对于输出特征图 O 中的每个元素 $O[i,j]$ ，计算方式如下：

$$O[i,j] = \frac{1}{F \times F} \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} X[i \times S + m, j \times S + n] \quad (14)$$

其中： $O[i,j]$ 是平均池化后的输出特征图的元素； F 是平均池化窗口的大小； S 是步幅。（与上相同）

5) 全连接层

全连接层，也称为密集连接层。它的主要特点是每个神经元与前一层的所有神经元都有连接，形成全连接的关系。对于第 i 个神经元，全连接层引入一个权重矩阵 W_i ，其中每一列对应于前一层的一个神经元。偏置项 b_i 用于每个神经元的线性变换。通常在全连接层的输出上应用非线性激活函数，如ReLU、Sigmoid或Tanh，以引入非线性变换。将所有神经元的输出组合成一个向量，即可得到全连接层的最终输出。

其中，对于第 i 个神经元，进行线性变换的公式如下：

$$z_i = \sum_{j=1}^N W_{i,j} \cdot x_j + b_i \quad (15)$$

z_i 是第 i 个神经元的加权和。 $W_{i,j}$ 是权重矩阵的元素； x_j 是前一层第 j 个神经元的输出； b_i 是偏置项。

将线性变换的结果 z_i 应用激活函数 f ，得到第 i 个神经元的输出 y_i ：

$$y_i = f(z_i) \quad (16)$$

C. RNN

循环神经网络（RNN）是一类用于处理序列数据的神经网络。与传统的前馈神经网络不同，RNN具有内部状态（隐藏状态），能够对输入数据中的序列进行建模。这种网络特别适合于处理时间序列数据或任何形式的有序数据集，例如文本、语音、视频帧等。RNN的核心思想在于网络中存在循环，这使得网络在模型内部保持一个状态，从而可以捕捉到数据中随时间变化的动态特性。这种内部状态的概念是RNN能够处理序列数据的关键，它可以被看作是一种“记忆”，存储了之前时间步的信息。

循环神经网络（RNN）的基本架构设计用于处理序列数据，并且具有处理输入数据的时间动态性的能力。RNN的核心特性是网络中的循环连接，这使得网络可以维持一个内部状态，随着序列的进展而更新。RNN由简单的输入层、隐藏层、循环结构和输出层构成，而循环结构是它的核心部分^[10]。接下来分析RNN的各层结构。

1) 输入层

传统RNN的输入层包括两个部分，时间步和特征表示。时间步是指输入层在每个时间步接收一个输入，这些输入按照时间顺序形成序列。而输入层每个时间步接收的序列数

据，比如自然语言处理中的单词序列，都会被转换成一个向量来表示，通常通过词嵌入等方式进行实现。

2) 隐藏层

隐藏层由循环神经元和隐藏状态组成。其中，循环神经元包含了循环结构，允许网络在不同时间步之间传递信息。每个神经元在处理当前时间步的输入时，会考虑之前时间步的输出，并将其与当前输入结合。而隐藏层在每个时间步都会输出一个隐藏状态，这些状态包含了网络在之前时间步所学到的信息。对于这种传统的RNN，其隐藏层的计算可通过以下公式来表示：

隐藏状态的更新：

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (17)$$

其中： h_t 是时间步 t 的隐藏状态。 x_t 是时间步 t 的输入。 W_{xh} 是输入到隐藏层的权重矩阵。 W_{hh} 是隐藏层到自身的权重矩阵（循环权重）。 b_h 是隐藏层的偏置。 ϕ 是激活函数，通常是非线性的，如tanh或ReLU。

3) 循环结构

循环结构是RNN的核心，它使得网络能够保留并传递先前时间步的信息。在数学上，这通过将上一个时间步的隐藏状态作为当前时间步的输入之一来实现，并且是有一定权重来进行传递的。循环结构中的权重在不同时间步共享，这意味着网络可以使用相同的参数来处理不同时间步的输入，从而减少模型的参数数量。

4) 输出层

输出层的构成是输出预测和激活函数。输出层生成模型的最终输出，这可以是分类任务中的类别概率，回归任务中的预测值，或其他任务中相应的输出。而输出层通常会使用适当的激活函数，如softmax（用于多分类问题）或线性激活（用于回归问题），以将输出映射到适当的范围^[11]。在每个时间步，基于当前的隐藏状态，输出层可以计算出当前时间步的输出 y_t 。对于许多RNN应用来说，输出层的计算可以表示为：

$$y_t = W_{hy}h_t + b_y \quad (18)$$

其中： y_t 是时间步 t 的输出； h_t 是时间步 t 的隐藏状态； W_{hy} 是隐藏层到输出层的权重矩阵； b_y 是输出层的偏置。

D. GRU

1) GRU的架构

GRU类似LSTM，也是为了解决RNN长期记忆的梯度消失问题。

LSTM有三个不同的门，参数较多，训练困难。GRU只含有两个门控结构，调优后相比LSTM效果相差无几，且结构简单，更容易训练，所以很多时候会更倾向于使用GRU^[12]。

GRU门控结构包括重置门（reset gate）和更新门（update gate），重置门用于决定保留多少之前的隐状态，更新门用于决定更新隐藏状态，它们都是向量，GRU的思想就是通过重置门与更新门与原输入或者候选隐状态之间运算得到状态的更新^[1]。

GRU的内部结构图如图Fig.6所示

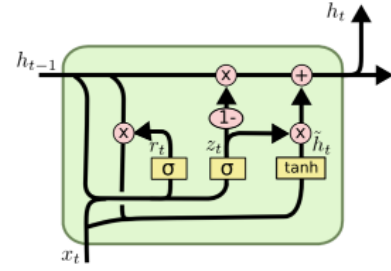


FIGURE 6. GRU内部结构图

重置门与更新门的计算公式如下：

$$R_t = \sigma(X_t W_{XR} + H_{t-1} W_{HR} + B_R) \quad (19)$$

$$Z_t = \sigma(X_t W_{XZ} + H_{t-1} W_{HZ} + B_Z) \quad (20)$$

其中 X_t 表示 t 时刻输入， W_{xr} 表示权重矩阵， H_{t-1} 表示上一时刻的隐藏状态， b_r 表示偏置项，下面一个类似， σ 为sigmoid变换，将向量所有元素控制在0-1范围。从上面公式看出，重置门与更新门由输入与隐状态决定，并且会不断变化，长度与隐状态长度相同，如图Fig.7。

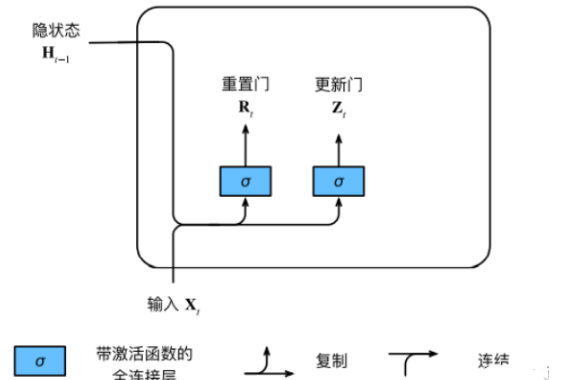


FIGURE 7. 重置门与输入门

2) 重置门

候选隐藏状态：利用重置门计算保留的隐状态，称作候选隐状态：

$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \quad (21)$$

\odot 可以发现候选隐状态由输入、上一时刻隐状态与重置门决定， \odot 表示Hadamard积（向量对应元素相乘），特别的，当重置门的元素全为0时， $R_t \odot H_{t-1}$ 等于0，此时表示丢弃所有之前的隐状态。如图Fig.8：

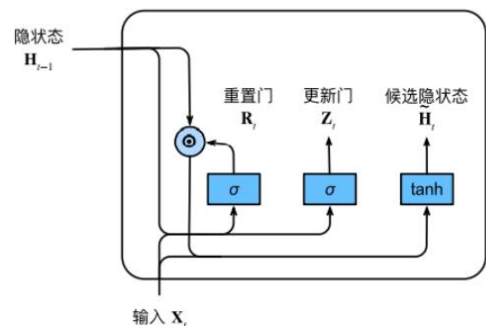


FIGURE 8. 候选隐藏状态

3) 更新门

更新隐状态: 得到候选隐状态后, 通过更新门更新隐状态, 更新后的隐状态由前一时刻隐状态与这一时刻候选隐状态决定。

特别的, 当 Z_t 全部元素都为1时, 更新的隐状态就是前一时刻的隐状态。

$$H_T = Z_T \odot H_{T-1} + (1 - Z_T) \odot \tilde{H}_T \quad (22)$$

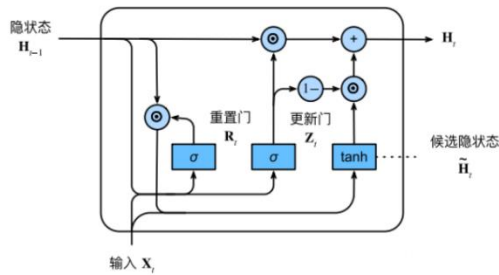


FIGURE 9. 更新隐藏状态

E. 集成学习

集成学习 (Ensemble Learning) 通过特定的结合策略, 对多个个体学习器进行结合以完成学习任务。具体而言, 先对多个个体学习器 (individual learner) 使用训练集进行训练, 再通过结合模块得到输出结果, 得到一个效果通常优于最优个体学习器的强学习器。

集成学习流程图如图Fig.10所示。首先由训练数据产生多个个体模型 (又叫个体学习器), 然后再利用某种策略, 将多个个体模型进行组合, 形成一个新的组合模型 [14]。

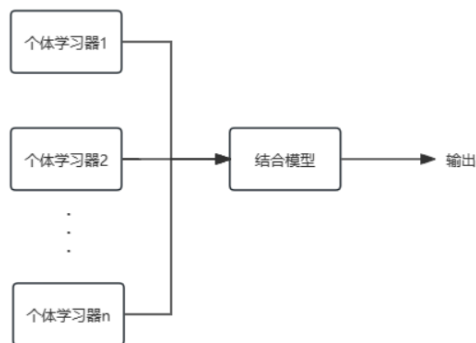


FIGURE 10. 集成学习流程图

1) BOOSTING算法

Boosting意为“提升”, 顾名思义是一类可将弱学习器提升为强学习器的算法。它的思想重点放在个体学习器的生成方式上, 并且个体学习器存在强依赖关系, 必须串行生成且将个体学习器序列化 [15]。

Boosting算法框架为: 首先, 对样本进行权重初始化 (第一个弱学习器a权重都相同)。然后, 在增大分类错误的样本权重的分配原则基础上, 使用弱学习器a对训练数据进行权重重新分配, 使其在训练弱学习器b时能够重点关

注分类错误样本。重复分配权重和训练, 不断生成新的弱学习器, 直到满足停止条件。然后将学习到的弱学习器组合成强学习器。其工作机制图如图Fig.11所示。

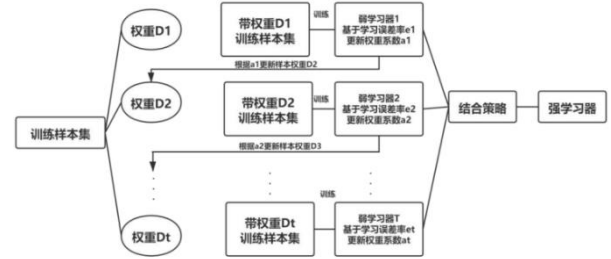


FIGURE 11. Boosting算法工作机制图

2) BAGGING算法

Bagging是一种典型的并行式集成学习的方法, 它的思想重点也是放在个体学习器的生成方式上。Bagging是BootstrapAGGREGATING的缩写。

Bootstrap指的是自助采样法, 它是一种有放回采样, 具体过程为: 给定样本量为N的数据集, 先随机从中抽取1个样本点, 放入采样集, 然后再把该样本点放回原始数据集, 再重新抽取1个样本点。如此重复N次, 就得到一个样本量为N的采样集。显然的, 初始训练集中的样本点有的在采样集中, 有的可能不在采样集中。经证明, 初始训练集中约有62.3%的样本点出现在采样集中。

Bagging的具体思路是: 首先利用自助采样法在训练集中采样T个样本量为N的采样集, 然后根据每个采样集训练出一个个体学习器, 最后将这些学习器进行结合。其工作机制如图Fig.12所示。

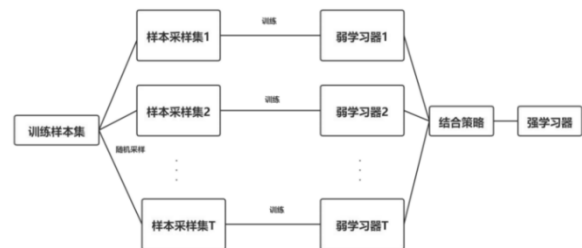


FIGURE 12. Bagging算法工作机制图

3) STACKING算法

Stacking也是一种个体学习器的组合策略思想。Fig.13为Stacking的模型组合思路图。第一步: 训练出多个个体学习器 {学习器1, 学习器2, ..., 学习器K}; 第二步: 以这些个体学习器的输出作为输入, 来训练一个新的学习器K+1, 学习器K+1的输出为最终的模型预测值。Stacking的特点是, 组合模型中的K+1个学习器可以是任意设定的, 不需要规定它们为同一种模型 [16]。

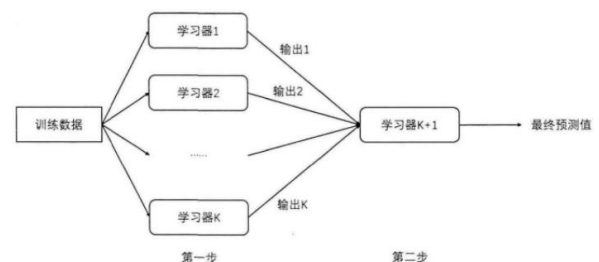


FIGURE 13. Stacking的模型组合思路图

图4Stacking的模型组合思路图

4) VOTING算法

Voting是一种个体学习器的组合策略思想，它的思想重点是在生成个体学习器之后，如何对这些个体学习器进行组合，分为软投票（SoftVoting）和硬投票（HardVoting）两种方法。硬投票是根据少数服从多数来定最终结果，一般用于分类任务；软投票是将所有模型预测样本为某一类别的概率的平均值作为标准，概率最高的对应的类型为最终的预测结果，一般用于回归任务。

IV. 实验

在本节中，我们将通过实验来回答以下研究问题

- 问题1：各种机器学习模型的预测精度如何？
问题2：情感融合是否有助于提升模型精度？
问题3：集成学习是否有助于提升模型精度？

A. 数据集

为了评估我们的研究方法，我们主要在一个公开的带有大量已经过预处理的股票价格数据和对应的推特评论数据集stock-dataset^[17]中进行了实验。该数据集包括AAPL等多个股票数据集，每个股票数据集的特征包括日期、开盘价、最高价、最低价、收盘价、成交量和成交额，除此之外，每个股票都有对应的已进行分词的推特评论数据集。我们选用了AAPL和其对应的推特评论数据集作为我们的数据集，且为了进行情感融合，我们对推特评论数据集进行了词嵌入和聚类处理，将评论数据聚成两类，即好和坏作为单条评论的情感值，最后通过对当天所有的评论情感值作均值得到当天的情感评分作为情感特征以补充AAPL数据集。

B. 评估标准

为了评估基于机器学习模型的股票预测模型的预测精度，我们使用了RMSE和精确度作为评估指标进行衡量。对于RMSE来说，评估指标值越小，说明模型的性能越好，对于精确度来说，评估指标值越大，说明模型的性能越好。

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (23)$$

其中：n是样本数量； y_i 是实际观测值(真实值)； \hat{y}_i 是相应观测值的预测值。

$$Accuracy = \frac{\sum Correct_i}{n}, Correct_i = \begin{cases} 1, & PREDICT_i = TRUE_i \\ 0, & PREDICT_i \neq TRUE_i \end{cases} \quad (24)$$

其中：Correct_i是第i天的涨幅预测正确与否；Predict_i是第i天的涨幅预测；True_i是第i天的真实涨幅；n是总天数。

C. 实验设置

1) LSTM模型

a. 训练集和测试集设置

我们从原始数据按顺序选取 70% 的数据记录作为训练集，其余 30% 的数据集样本作为测试集，用于验证模型的性能。

b. 参数设置

我们的实验采用了经典的LSTM模型作为股票预测模型^[6]。对于LSTM模型，我们将隐藏层维度设为32，将学习率设为 0.0001，将抛弃率设为 0.1，并将优化函数设为Adam，将LSTM层数设为2层，将训练轮次设为100次。

2) CNN模型

a. 训练集和测试集设置

我们从原始数据按顺序选取 70% 的数据记录作为训练集，其余 30% 的数据集样本作为测试集，用于验证模型的性能。

b. 参数设置

我们的实验采用了经典的CNN模型作为股票预测模型^[18]。对于CNN模型，我们将隐藏层维度设为32，将学习率设为 0.0005，将抛弃率设为 0.1，并将优化函数设为Adam^[19]，将输入通道数设置为8，输出大小设置为1。将训练轮次设为100次。

3) RNN模型

a. 训练集和测试集设置

我们从原始数据按顺序选取 70% 的数据记录作为训练集，其余 30% 的数据集样本作为测试集，用于验证模型的性能。

b. 参数设置

对于RNN模型^[10]，我们将隐藏层维度设为32，将学习率设为 0.0001，将抛弃率设为 0.1，并将优化函数设为Adam，将LSTM层数设为1层，将训练轮次设为100次。

4) GRU模型

a. 训练集和测试集设置

我们从原始数据按顺序选取 70% 的数据记录作为训练集，其余 30% 的数据集样本作为测试集，用于验证模型的性能。

b. 参数设置

丢弃率为0.1，有情感时输入层维度为7，无情感时输入层维度为6，学习率为0.0001，用前五天的数据预测下一天，输入特征为8，隐藏层32，输出特征为1，1层GRU。

D. 问题1的效果

为了评估不同机器学习模型对股票预测的性能，我们选择了LSTM^[6]、CNN、RNN和GRU模型进行比较。

1) 收敛效率

Fig.14, Fig.15, Fig.16,和Fig.17体现了不同机器学习模型的收敛效率，其中LSTM在第40轮训练时已经收敛，RNN在第20轮训练时已经收敛，而CNN经过100轮训练仍未收敛（排除学习率的问题，此外，推测应该是三个卷积层后跟全连接层的网络结构可能过于复杂。或许可以尝试简化网络结构，减少层数，或者考虑调整卷积核大小和池化窗口大小。另外，可能原因在于没有将时间序列数据有效地表示为图像，因为CNN相对来说会在图像领域表现更为出色。）

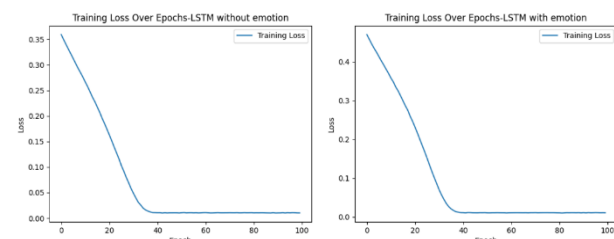


FIGURE 14. LSTM模型收敛效率

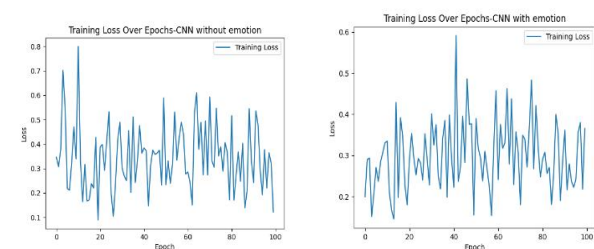


FIGURE 15. CNN模型收敛效率

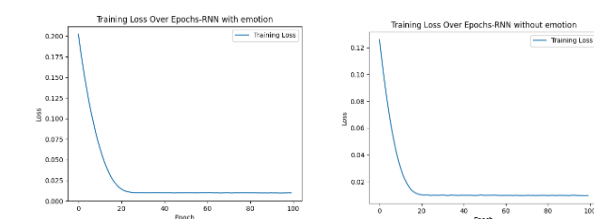


FIGURE 16. RNN模型收敛效率

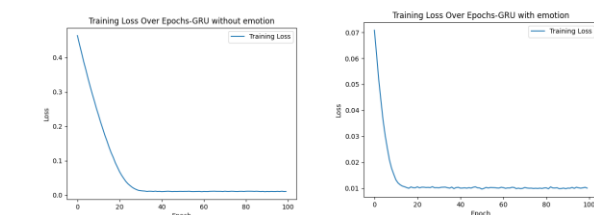


FIGURE 17. GRU模型收敛效率

2) 预测准确性

TABLE I列出了APPL数据集预测准确率的比较结果。我们发现四个预测模型中，LSTM的预测精确率最高，然后依次是RNN、CNN和GRU。虽然LSTM的预测精确率最高，但是LSTM的RMSE在四个模型中是比较高的，不过在股票预测的场景中，精确率的优先级要高于RMSE，因此我们还是认为LSTM的预测性能是很高的，其他模型的预测精确率都在50%上下浮动，而LSTM的预测精确度能将近达到90%。

TABLE I
四种模型预测数据的 RMSE 与精确率

模型	LSTM	GRU	RNN	CNN
RMSE	0.09	0.02	0.02	0.13
精确率	0.89	0.49	0.56	0.54

Fig.18为不同机器模型的预测价格-实际价格图，我们发现LSTM模型的预测价格和实际价格有明显的差异，但是预测价格和实际价格的涨幅情况类似，为了进一步查看涨幅情感，Fig.19为预测涨幅和实际涨幅，可以看到预测价

格和实际价格的涨幅情况相当，在371条涨幅情况中，只有28条与实际情况不符。

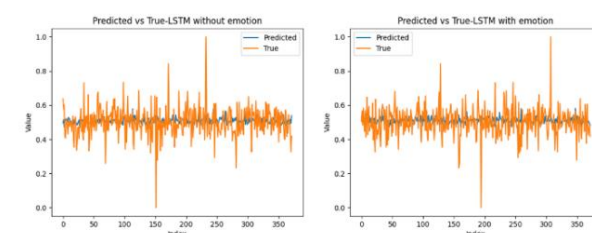


FIGURE 18. LSTM预测价格-实际价格

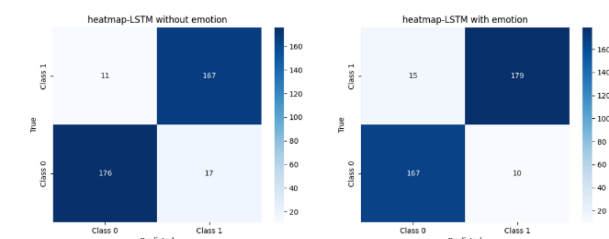


FIGURE 19. LSTM预测价格涨幅情况-实际价格涨幅情况

CNN模型的预测价格和实际价格有明显较大差异，并且预测价格和实际价格的涨幅情况相差较大，为了进一步查看涨幅情感，Fig.21为预测涨幅和实际涨幅，可以看到预测价格和实际价格的涨幅情况差异很大，甚至连大体的趋势都预测错误。

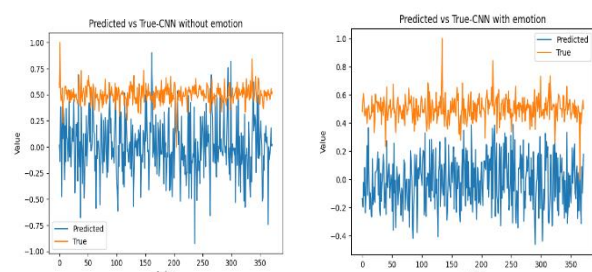


FIGURE 20. CNN预测价格-实际价格

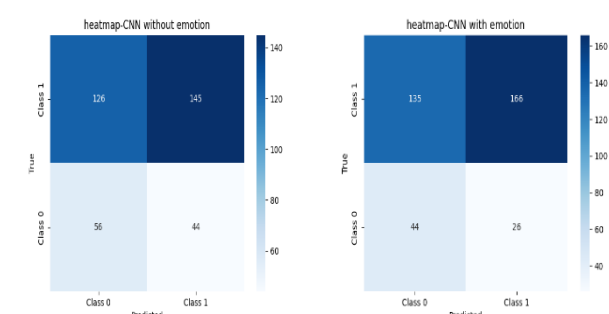


FIGURE 21. CNN预测价格涨幅情况-实际价格涨幅情况

RNN模型的预测价格和实际价格也有明显的差异，并且预测价格和实际价格的涨幅情况也有所差异，为了进一步查看涨幅情感，Fig.23为预测涨幅和实际涨幅，可以看到预测价格和实际价格的涨幅情况差异较大，甚至连大体的趋势都预测错误。

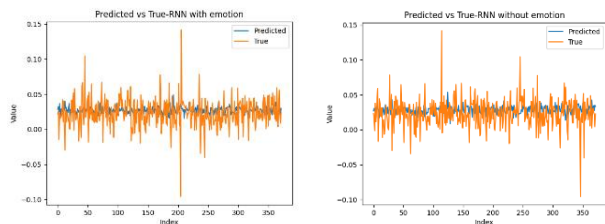


FIGURE 22. RNN预测价格-实际价格

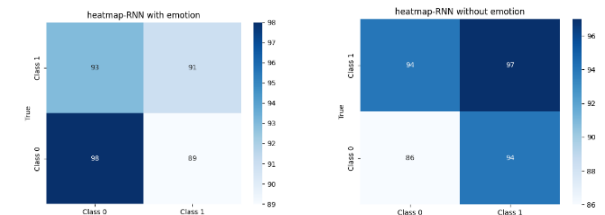


FIGURE 23. RNN预测价格涨幅情况-实际价格涨幅情况

我们发现GRU模型的预测价格和实际价格也有很大差异,预测价格和实际价格的涨幅情况类似,为了进一步查看涨幅情感, Fig.25为预测涨幅和实际涨幅,可以看到预测价格和实际价格的涨幅情况相当,在371条涨幅情况中,有高达177条与实际情况不符。

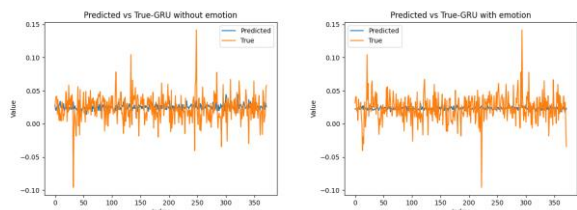


FIGURE 24. GRU预测价格-实际价格

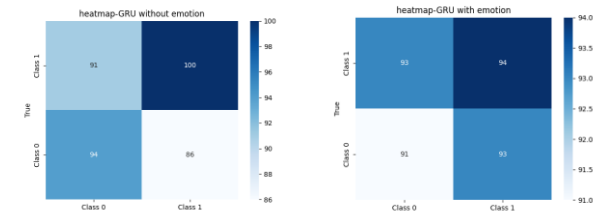


FIGURE 25. GRU预测价格涨幅情况-实际价格涨幅情况

E. 问题2的效果

为了评估情感融合对股票预测的性能影响,对于不同机器学习模型,我们直接新增了一个情感特征,模型的收敛效率和精度由图xxx-图xxx可见。

LSTM模型的收敛效率并没有因为情感融合而受到影响,模型还是在第40轮训练时收敛。LSTM模型的预测价格和实际价格的差异与没有情感融合的情况相当,还是与实际价格有很大的差异。LSTM模型的预测涨幅情况和实际涨幅情况在加入情感融合后有所改进,在371条涨幅情况中,只有25条与实际情况不符,相比情感融合前模型精度提高了约10.71%。

CNN模型的收敛效率同样没有因为情感融合而受到影响,模型在100轮训练后仍未收敛。CNN模型的预测价格和实际价格的差异与没有情感融合的情况相当,还是与实际价格有非常大的差异。CNN模型的预测涨幅情况和实

际涨幅情况在加入情感融合后略有降低,但总体并不明显。相比情感融合前模型精度降低了约3.6%。

RNN模型的收敛效率同样没有因为情感融合而受到影响,模型还是在第20轮训练时收敛。RNN模型的预测价格和实际价格的差异与没有情感融合的情况相当,还是与实际价格有很大的差异。RNN模型的预测涨幅情况和实际涨幅情况在加入情感融合后并没有改进,甚至变得更差了。相比情感融合前模型精度降低了约6%。

TABLE.2汇总了所有模型在情感融合前后的RMSE和精确度变化情况。

TABLE II
模型在情感融合前后的 RMSE 和精确度

	无情感融合		有情感融合	
模型	RMSE	精确度	RMSE	精确度
LSTM	0.09	0.89	0.08	0.94
CNN	0.13	0.54	0.13	0.56
RNN	0.02	0.56	0.02	0.50
GRU	0.02	0.49	0.02	0.48

F. 问题3的效果

为了评估集成学习对股票预测的性能影响,我们集成了以下不同模型来探究预测精度的变化情况。利用集成学习的思想,对多个时间序列神经网络模型进行组合,构造出组合模型,以提升效果。

LSTM-CNN
LSTM-RNN
LSTM-GRU
CNN-RNN
CNN-GRU
RNN-GRU

1) LSTM-CNN组合模型

在本次实验中,我们设计并训练了一个融合了长短时记忆网络(LongShort-TermMemory,LSTM)与卷积神经网络(ConvolutionalNeuralNetwork,CNN)的混合模型,用于处理特定的文本分类任务。该模型旨在通过提取文本序列的时间依赖性以及局部特征以提升分类性能。

LSTM-CNN组合模型构造流程图如图Fig.26。首先独立训练并优化LSTM和CNN模型以捕获文本数据的长期依赖性和局部特征,然后将两者在测试集上的预测结果进行融合(例如通过加权平均或其他融合策略),生成综合预测数值;再将两个模型的预测数值进行加权平均,得到新的数值,再将新生成的数值传到神经网络模型中进行训练,得到得出融合后的预测结果,并将最终预测结果与真实结果比对,进行运算获得准确率。



FIGURE 26. LSTM-CNN架构概述

我们进行了两组不同的二分类神经网络模型训练:一组是有情感评分,另一组是无情感评分。

2) LSTM-GRU组合模型

在本次实验中,我们设计并实现了一个结合了长短时记忆网络(Long Short-Term Memory, LSTM)与门控循环单元

网络（Gated Recurrent Unit, GRU）的混合模型，应用于特定文本分类任务。该模型核心理念是融合LSTM对序列数据长距离依赖关系的强大捕捉能力以及GRU结构简洁高效的特性，在提取文本序列的时间动态特征和局部上下文信息方面达到更优的效果。

实验流程如所示，首先，我们分别在训练集上独立训练LSTM模型和GRU模型，确保每个个体模型能够学习到文本序列的独特表征。接着，在测试集上应用这两个预训练模型进行预测，得到各自的预测数值输出。为了进一步整合两种模型的优势，我们将LSTM和GRU模型的预测结果进行加权平均，生成新的综合预测特征向量，这一过程旨在结合两者对于复杂文本模式的不同理解维度。随后，将这个综合预测数值输入到一个后续的神经网络层中进行联合训练以优化整体性能，从而得出最终的分类预测结果。最后，通过比较模型预测结果与实际标注的真实类别，计算出相应的准确率、损失值等评价指标，评估LSTM-GRU组合模型在给定文本分类任务上的表现及有效性。



FIGURE 27. LSTM-GRU架构概述

我们进行了两组不同的二分类神经网络模型训练：一组是有情感评分，另一组是无情感评分。

3) LSTM-RNN组合模型

在本次实验中，我们设计并实施了一个结合了长短时记忆网络（Long Short-Term Memory, LSTM）与循环神经网络（Recurrent Neural Network, RNN）的混合模型，用以解决特定文本分类问题。该模型旨在结合LSTM对序列数据长期依赖性强大的捕捉能力和RNN处理序列信息的基础特性，在挖掘文本序列的时间动态特征和局部上下文信息方面实现更优的表现。

实验过程如下：首先，我们在训练集上分别单独训练了LSTM模型和RNN模型，确保每个个体模型都能有效学习到文本序列的关键特征表示。然后，在测试阶段，我们将这两个预训练好的模型应用于测试集进行预测，并分别获取它们各自的预测数值输出。为了充分利用两种模型的优势互补，我们将LSTM和RNN模型的预测结果进行了加权平均处理，生成一个融合了两者洞察力的综合预测特征向量，这一操作意在结合两者从不同视角对复杂文本模式的理解。紧接着，将这个整合后的预测值输入至后续的神经网络层中进行联合微调优化，以追求最优的整体性能表现。最后，通过对模型预测结果与实际标注的真实类别进行比较，计算出相应的准确率、损失值等评估指标，以此来全面评价LSTM-RNN组合模型在所设定文本分类任务上的性能及有效性。



FIGURE 28. LSTM-RNN架构概述

我们进行了两组不同的二分类神经网络模型训练：一组是有情感评分，另一组是无情感评分。

4) CNN-GRU组合模型

在本次实验中，我们设计并实施了一个融合了卷积神经网络（Convolutional Neural Network, CNN）与门控循环单元网络（Gated Recurrent Unit, GRU）的混合模型，以解决特定文本分类问题。该模型利用CNN对局部特征进行高效抽取的优势结合GRU捕捉序列数据长期依赖性的能力，在提取文本序列的空间局部结构特征以及时间动态上下文信息方面追求更优的表现。

实验流程如下所述：首先，我们在训练集上分别独立训练CNN模型和GRU模型，确保每个个体模型都能够有效学习到文本序列的核心特征表示。随后，在测试阶段，我们将这两个预先训练好的模型应用到测试集上进行预测，并分别获取它们各自的预测数值输出。为了充分融合两种模型的不同优势，我们将CNN与GRU模型的预测结果进行了加权平均处理，生成一个整合了两者识别智慧的综合预测特征向量，这一操作旨在结合两者从不同视角对复杂文本模式的深层次理解。紧接着，将这个综合预测值输入至额外的神经网络层进行联合微调优化，以追求最佳的整体性能表现。最后，通过对模型预测结果与实际标注的真实类别进行比较，计算出相应的准确率、损失值等评估指标，从而全面评估CNN-GRU组合模型在所设定文本分类任务上的性能及实用性。



FIGURE 29. CNN-GRU架构概述

我们进行了两组不同的二分类神经网络模型训练：一组是有情感评分，另一组是无情感评分。

5) CNN-RNN组合模型

在本次实验中，我们设计并实施了一个结合了卷积神经网络（Convolutional Neural Network, CNN）与循环神经网络（Recurrent Neural Network, RNN）的混合模型，旨在解决特定文本分类问题。该模型充分利用CNN对局部特征的强大抽取能力以及RNN处理序列数据长期依赖性的特性，在提取文本序列的空间局部结构特征和时间动态上下文信息方面寻求最优解。

实验步骤如下：首先，我们在训练集上分别独立训练CNN模型和RNN模型，确保每个单独模型都能有效地学习到文本序列的核心特征表达。随后，在测试阶段，我们将这两个预先训练完备的模型应用于测试集进行预测，并分别获得它们各自的预测数值输出。为了充分发挥两种模型的不同优势互补作用，我们将CNN与RNN模型的预测结果进行了加权平均运算，生成一个融合两者智能识别的综合预测特征向量，这一操作目标在于整合二者从不同角度对复杂文本模式的深层次理解。紧接着，我们将这个综合预测值输入至额外的神经网络层进行联合微调优化，以追求整体性能的最大化。最后，通过对比模型预测结果与实际标注的真实类别，计算出相应的准确率、损失值等评估指标，从而全面评价CNN-RNN组合模型在所设定文本分类任务上的性能及实用性。



FIGURE 30. GRU-RNN架构概述

我们进行了两组不同的二分类神经网络模型训练：一组是有情感评分，另一组是无情感评分。

6) GRU-RNN组合模型

在本次实验中，我们设计并实施了一个结合了门控循环单元网络（Gated Recurrent Unit, GRU）与循环神经网络（Recurrent Neural Network, RNN）的混合模型，旨在解决特定文本分类任务。该模型利用GRU对序列数据长期依赖性高效捕捉的优势以及RNN处理序列信息的基础能力，在揭示文本序列的时间动态特征和局部上下文信息方面寻求更优的表现。

实验步骤如下：首先，我们在训练集上分别独立训练了GRU模型和RNN模型，确保每个个体模型都能有效学习到文本序列的核心特征表达。随后，在测试阶段，我们将这两个预先训练完成的模型应用到测试集进行预测，并分别获得它们各自的预测数值输出。为了充分整合两种模型的不同优势，我们将GRU与RNN模型的预测结果进行了加权平均运算，生成一个融合两者识别智慧的综合预测特征向量，这一过程目的在于结合二者从不同角度对复杂文本模式的理解深度。紧随其后，将这个综合预测值输入至额外的神经网络层进行联合微调优化，以期达到最佳的整体性能表现。最后，通过对比模型预测结果与实际标注的真实类别，计算出相应的准确率、损失值等评估指标，从而全面评判GRU-RNN组合模型在给定文本分类任务上的性能及实用性。



FIGURE 31. GRU-RNN架构概述

我们进行了两组不同的二分类神经网络模型训练：一组是有情感评分，另一组是无情感评分。

7) 组合模型分析

下面我们将对实验结果进行详细解读，其中实验结果指Fig.32和TABLE.3。

a. LSTM-CNN:

从实验数据可以看出，在整个训练过程中，模型的Loss值呈现逐步下降的趋势，表明模型正在不断学习和优化，对训练集的理解和拟合能力逐渐增强。

然而，最终在测试集上的准确率仅为56%和52%，这一结果并未达到理想的分类效果。同时通过比较我们可发现有情感评分的准确率较无情感评分的准确率较高。

b. LSTM-GRU:

实验结果显示，该LSTM-GRU混合模型相较于单一模型在保持较低计算复杂度的同时，有效提升了文本分类任务的性能，验证了我们提出的融合策略的有效性和可行性。在对LSTM-GRU组合模型进行训练的过程中，观察到Loss值随着Epoch数的增长呈现出逐渐减小的趋势，这清晰地展示了模型正在逐步学习和优化其对训练集数据的理解

及拟合能力。然而，在测试阶段，该模型分别在包含情感评分数据集上取得了56%的准确率，在不包含情感评分数据集上则获得了54.67%的准确率。尽管相较于之前的结果（或期望目标），这一分类性能并未达到理想状态，但值得注意的是，对于含有情感评分信息的数据集，模型的表现稍优于无情感评分的数据集，表明了该模型在处理具有情感倾向信息的文本时具有一定优势。未来的工作将继续聚焦于提升整体模型性能，尤其是改进在无特定情感线索情况下的分类效果。

c. LSTM-RNN:

实验结果显示，所构建的LSTM-RNN混合模型在保持相对较低计算复杂度的同时，相较于单一模型，在文本分类任务上取得了明显的性能提升，从而验证了我们提出的将LSTM与RNN相结合策略的有效性和实用性。

在对LSTM-RNN组合模型进行训练时，我们观察到随着Epoch数的增加，Loss值持续降低，这一现象有力地展示了模型正在不断优化学习过程，并逐步增强其对训练集数据的理解及拟合能力。然而，在测试阶段，该模型在面包含情感评分信息的数据集中取得了57.58%的准确率，而在无情感评分信息的数据集上则获得了56.25%的准确率。尽管相比先前的结果或期望目标，当前分类性能尚未达到理想状态，但值得一提的是，对于含有情感元素的文本数据，LSTM-RNN混合模型的表现略优于不含情感线索的数据集，这说明在处理具有情感倾向特征的文本时，该模型具备一定的优势。未来的研究工作将持续关注并致力于提升整体模型性能，特别是针对缺乏特定情感线索条件下的文本分类效果改进。

d. CNN-GRU:

实验结果显示，我们设计的CNN-GRU融合模型在保证计算效率相对较低的前提下，在文本分类任务上取得了相较于单独使用任一模型更为显著的性能提升，从而证实了将CNN与GRU相结合策略的有效性和实用性。

在训练CNN-GRU复合模型的过程中，随着Epoch数的增加，Loss值呈现稳步下降的趋势，这有力地验证了该模型正在不断优化学习过程，并逐步增强其对训练数据集的理解能力和模式匹配精准度。在测试环节，针对包含情感评分信息的数据集，模型实现了56%的准确率；而在缺乏明确情感评分信息的数据集中，则达到了57.33%的准确率。尽管相比其他同类组合模型，这一准确率表现尚可，但值得注意的是，对于不含有明显情感倾向特征的文本数据，CNN-GRU混合模型反而表现出更高的准确率。这种现象可能是由于在处理无明显情感色彩的文本时，模型能够更专注于捕捉和理解文本的结构和语义信息，避免了情感分析中可能引入的额外复杂性和不确定性，因此在非情感类别的文本分类任务上获得了更优的表现。

e. CNN-RNN:

实验结果显示，我们构建的CNN-RNN联合模型在维持相对较低计算复杂度的同时，在文本分类任务上相较于独立使用任一模型获得了明显的性能提升，从而证实了将CNN结构与RNN机制相结合策略的有效性和实用性。

在对CNN-RNN复合模型进行训练时，随着Epoch次数的增长，观察到Loss值呈现出逐步下降的趋势，这一现象强有力地证明了该模型正在持续优化学习过程，并不断深化其对训练数据集内在模式的理解和适应性。然而，在实际

测试环节中,此模型在处理含有情感评分信息的数据集中取得了56.25%的准确率,而在未提供情感评分信息的数据集中,准确率则达到了53.12%。尽管当前模型的分准确度尚未达到最优状态,但值得注意的是,在包含情感内容的文本分析场景下,CNN-RNN混合模型的表现略胜于那些不包含情感线索的数据集,这揭示了该模型在处理蕴含情感特征的文本任务时具有一定的优势。

f. GRU-RNN:

实验结果显示,我们构建的GRU-RNN混合模型在保持相对较低计算复杂度的同时,在文本分类任务上相较于单一模型取得了显著的性能提升,从而验证了提出的将GRU与RNN结合策略的有效性和实用性。

在对GRU-RNN组合模型进行训练时,随着Epoch数的增长,Loss值持续减小的现象有力地证明了模型正不断优化学习过程,并逐步提升其对训练集数据的理解和拟合能力。在测试阶段,该模型对于包含情感评分信息的数据集实现了62.5%的准确率,而在缺乏情感评分信息的数据集中则达到了59.38%的准确率。相较于其他组合模型,该模型的准确率较高。值得注意的是,对于包含情感成分的文本数据,GRU-RNN混合模型的表现明显优于那些不含情感线索的数据集,这揭示了在处理具有情感倾向特性的文本时,此模型具备一定的优势。未来的研究工作将持续关注并致力于改进整体模型性能,特别是针对在没有特定情感线索条件下的文本分类效果提升。

具体来说,在训练过程中,这些融合模型随着Epoch数的增长,Loss值呈现稳步下降态势,充分体现了模型通过不断迭代优化,逐步提升了对复杂文本数据集的理解能力和模式识别精准度。尤其在含有情感评分信息的数据集中。

8) 组合模型总结

在对组合模型进行深入细致的训练之后,我们对比了训练前后的模型性能表现。实验结果显示,在纳入比较的各种组合模型中,除了长短期记忆网络(LSTM)外,其他诸如RNN、GRU以及RNN等模型在文本分类任务上的准确率均实现了显著提升。

GRU-RNN模型的表现尤为突出,相较于单一模型,其准确率达到了62.5%,而对于无情感评分信息的数据集,也取得了59.38%的良好准确率。

然而,尽管LSTM在本次实验中的提升效果不明显,但值得注意的是,与其他组合模型相比,LSTM在网络结构上对于长期依赖关系的捕捉具有天然优势。因此,进一步研究如何有效结合LSTM与其他神经网络结构,以期在保持其处理序列信息优势的同时,提高整体模型在文本分类任务上的准确率,是我们未来探索的重要方向。

综上所述,融合多种深度学习单元构建的融合模型在文本分类任务上展现出了强大的适应性和优越性,而针对不同模型在特定场景下的表现差异,我们将持续深化研究,致力于发掘并优化各类模型在处理不同类型文本数据时的潜力与效能。

TABLE III
模型准确率汇总表

模型	准确率	
	有情感评分	无情感评分
LSTM	94.00%	89.00%
CNN	53.00%	52.00%
GRU	48.52%	49.33%
RNN	48.00%	47.00%
LSTM-CNN	56.00%	52.00%
LSTM-GRU	56.00%	54.67%
LSTM-RNN	57.58%	56.25%
CNN-GRU	56.00%	57.33%
CNN-RNN	56.25%	53.12%
GRU-RNN	62.50%	59.38%

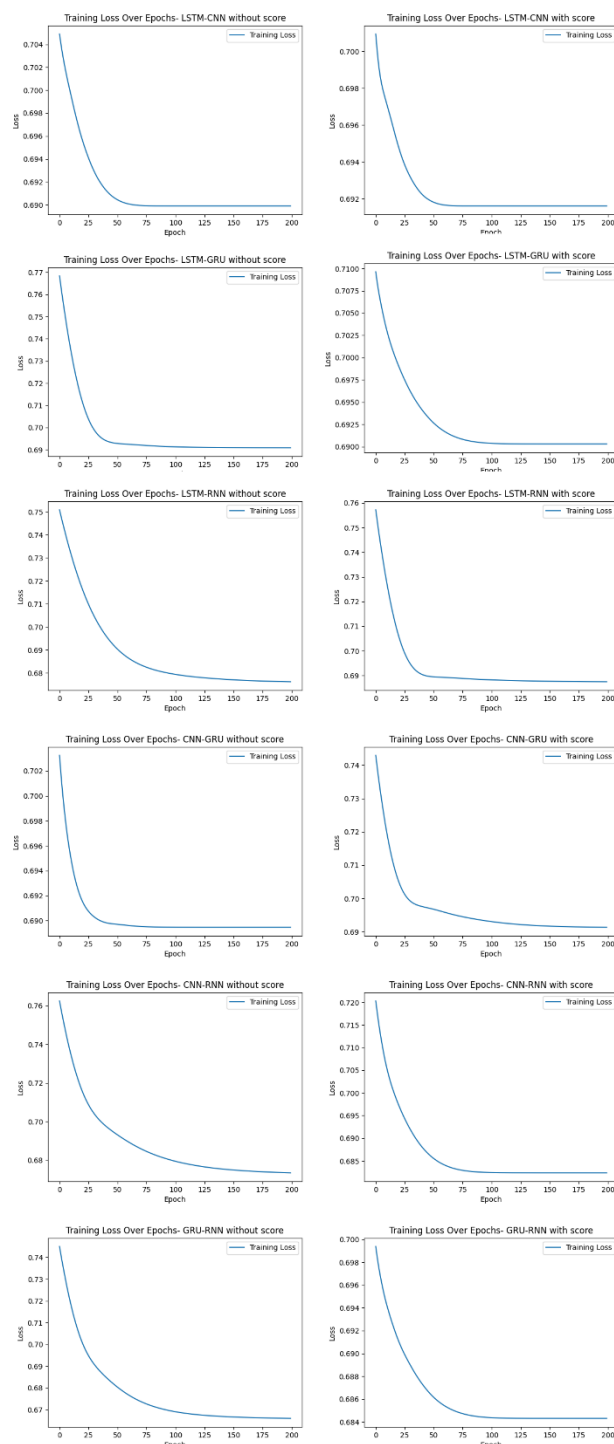


FIGURE 32. 集成模型学习损失变化图

V. 总结

在本研究中，我们探究了LSTM、CNN、RNN和GRU四种深度学习模型对股票预测的精度，在此基础上我们还进行了消融实验，通过引入情感评分与否和引用集成学习与否来探究股票预测模型的精度。我们的研究表明，在四种深度学习模型中，LSTM的预测精度最高，为89%~94%，其余深度学习模型的精度在50%左右；引入情感评分提升了

LSTM约10% 预测精度而对其他深度学习模型的预测精度没有明显的提升效果；在集成学习中，除了LSTM与其他深度学习模型集成相比单个模型的预测精度还低，其余深度学习模型的集成比单个模型的预测精度有xxx%的提升。

VI. 未来工作

我们的工作探究了不同深度学习模型的股票预测精度以及情感融合和集成学习对模型精度的影响。然而我们研究的模型仅限于四种，情感评分也仅取自于推特评论，还没涉及到财经新闻和政策新闻。除此之外，我们还没有研究不同测试集和训练集占比对模型精度的影响，对预测精度较差的模型也没有进行模型参数调整以尝试提高预测精度。因此该课题还有许多值得研究的问题。未来，我们将首先考虑对预测精度较差的模型调整，找到这些模型的精度上限。

REFERENCES

- [1] 董玲, "基于深度学习的股票价格预测研究," 山东工商学院, 山东, 中国, 2023, pp. 80, DOI: 10.27903/d.cnki.gsdsg.2023.000129.
- [2] 廖畅, "基于新闻情感量化和 LSTM 网络的股票预测模型设计与实现," 华中科技大学, 湖北, 中国, 2020, pp. 55, DOI: 10.27157/d.cnki.ghzku.2020.001004.
- [3] X. Li, Y. Li, H. Yang, L. Yang, and X.-Y. Liu, 'Dp-lstm: differential privacy-inspired lstm for stock prediction using financial news', *arXiv*, Dec. 19, 2019. Accessed: Jan. 05, 2024. [Online]. Available: <http://arxiv.org/abs/1912.10806>
- [4] P. Ghosh, A. Neufeld, and J. K. Sahoo, 'Forecasting directional movements of stock prices for intraday trading using LSTM and random forests'. *arXiv*, Jun. 30, 2021. Accessed: Jan. 05, 2024. [Online]. Available: <http://arxiv.org/abs/2004.10178>
- [5] F. Z. Xing, E. Cambria, and R. E. Welsch, 'Intelligent asset allocation via market sentiment views', *IEEE Comput. Intell. Mag.*, vol. 13, no. 4, pp. 25–34, Nov. 2018, doi: 10.1109/MCI.2018.2866727.
- [6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [7] T. Fischer and C. Krauss, "Deep Learning with Long short-term Memory Networks for Financial Market Predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: <https://doi.org/10.1016/j.ejor.2017.11.054>.
- [8] D. L. Nelson, A. P. Pereira, and R. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. IJCNN*, 2017.
- [9] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. ICACCI*, 2017.
- [10] Britz, D. (2015). Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs. WildML. <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>
- [11] 张尧, "激活函数导向的 RNN 算法优化," 浙江大学, 浙江, 中国, 2017, pp. 60.
- [12] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *EMNLP 2014*, 2014, doi: 10.48550/arXiv.1406.1078.

- [13] J. Chung, Çağlar Gülçehre, K. Cho, and Yoshua Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *ArXiv* (Cornell University), vol. Neural and Evolutionary Computing, 2014, Dec. 2014.
- [14] 宋林泽, "股票多因子挖掘的集成学习模型及系统实现," 电子科技大学, 中国, 2022, pp. 90, DOI: 10.27005/d.cnki.gdzku.2022.004572.
- [15] 王远楷, "股票趋势预测的集成学习方法的研究与实现," 西北大学, 中国, 2021, pp. 70, DOI: 10.27405/d.cnki.gxbdu.2021.001044.
- [16] 黄钰棠, "基于神经网络及其集成机器学习的股票预测对比研究," 中央民族大学, 北京, 中国, 2022, pp. 104, DOI: 10.27667/d.cnki.gzymu.2022.000272.
- [17] Y. Xu and S. B. Cohen, "Stock Movement Prediction from Tweets and Historical Prices," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), vol. 2018.
- [18] K. Bhardwaj, "Convolutional Neural Network(CNN/ConvNet) in Stock Price Movement Prediction.," *arXiv* (Cornell University), Jun. 2021.
- [19] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.



刘文龙: 概念, 方法, 数据整理, 写作, 可视化, 监督。目前, 正在中国汕头大学攻读计算机科学与技术学士学位。2023 年9月至今, 在汕头大学许建龙老师组内进行基于云边协同的Web服务个性化QoS预测隐私保护方法研究。研究兴趣包括推荐系统、数据挖掘与分析 and 人工智能。



陈开枝: RNN模型介绍以及相应实验结果整理; 中国汕头大学计算机科学与技术专业本科生。



陈飞洋: CNN模型介绍以及相应实验结果整理; 中国汕头大学计算机科学与技术专业本科生。



刘金鑫: GRU 模型介绍以及相应实验结果整理; 中国汕头大学计算机科学与技术专业本科生。



赵嘉萱: 集成学习模型介绍以及相应实验结果整理; 中国汕头大学计算机科学与技术专业本科生。



高新飞: LSTM 模型介绍、相关工作书写以及论文整合; 中国汕头大学计算机科学与技术专业本科生。