



华南理工大学
South China University of Technology

《小白学人工智能之机器学习》 课 程 论 文

(2019-2020 学年第 1 学期)

论文题目

基于数种神经网络的可转债发行上市首日价格研究
与基于 GRU 的关于北上资金的上证综指走势研究

摘要： 针对可转债发行上市首日价格的预测研究，本文构建了一系列的神经网络模型，包括全连接的 BP (back propagation) 神经网络、卷积神经网络 (CNN) 和长短期记忆网络 (LSTM) 模型；针对关于北上资金的上证综指走势预测研究，本文构建了门控制神经网络 (GRU) 模型。上述模型均根据各自的特点对所选特征、具体应用进行了优化，本着以学习为目的的精神，对不同的模型展开探究，旨在了解、掌握各种模型的结构、性与实际应用。

关键字： 可转债；BP 神经网络；卷积神经网络 (CNN) 长短期记忆网络 (LSTM)；门控制神经网络 (GRU)；北上资金；上证综指

(一) 基于数种神经网络的可转债发行上市首日价格研究

一、 背景

近些年来，可转债投资市场逐渐走近了人们的视野，因此兼具债权和期权特征的优良特性，不少投资者纷纷布局可转债。同时作为一种优良的融资工具，由于其有可以“债转股”的特性，可以让企业大大减轻偿还的后顾之忧，在如今经济下行压力大的金融大环境中颇受企业的青睐。在这种企业和投资者双赢的局面下，可转债的发行可谓是日趋火热。而普通老百姓作为一名小散户，值博率最高的可转债市场参与方式便是可转债“打新”，即以每张 100 的面值价格申购原始发行的可转债。随着日趋增多的选择首日卖出的“打新”族加入市场，许多套利与反套利投资策略逐渐流行起来，而这其中的核心要点便是可转债发行上市首日价格预测。本着学习的精神，下面本文分别研究全连接的 BP 神经网络、卷积神经网络、长短期记忆网络并应用于此问题中。

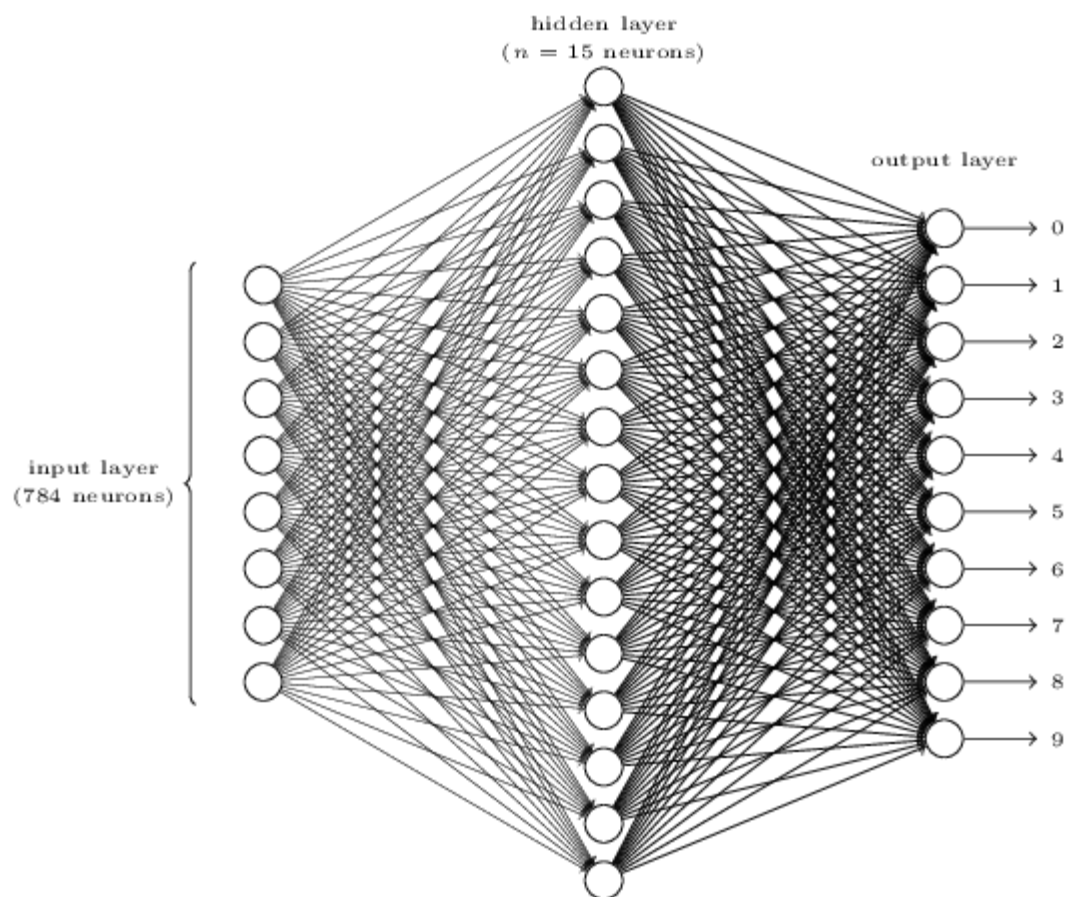
二、 算法简介

下面对这三种神经网络进行初步的介绍。

1. 全连接的 BP 神经网络

注：论文要求中并没有涉及 BP 神经网络，但由于课程视频中的介绍、笔者代码学习的接触，现作为学习过程的记录在此文中提及。

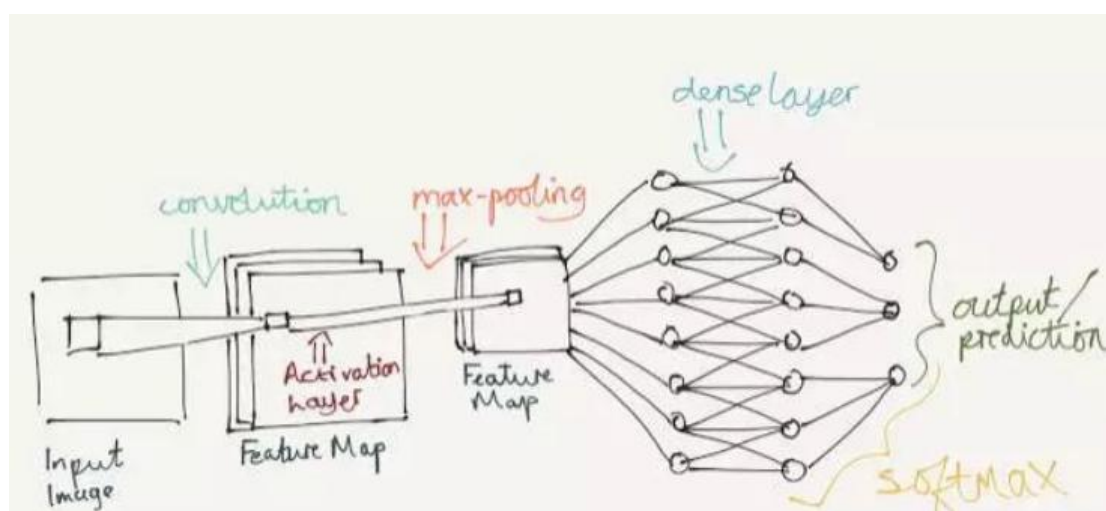
BP 神经网络可以说是最传统，也最容易理解的神经网络了。
其网络结构便是最简单的三层：输入层、隐藏层、输出层。



无论在哪一层，每一个神经元的任务都是三个：输入、输出和判断。每个神经元中，包含了激活函数和偏置（bias）。每一个神经元都与处于自己下一层和上一层的有连接，此即所谓“全连接”。而所谓的“BP”算法，则是在经过前向传播将输入层的值通过隐含层传播到输出层后，不断地计算输出层的预期值与真实值之间的误差，然后将误差经过隐藏层，传输到输入层，在一次又一次传播中不断地修改模型的参数。此谓之反向传播算法。

2. 卷积神经网络（CNN）

卷积神经网络的网络结构也离不开“输入层、隐藏层、输出层”的架构，而其中的隐藏层的结构则是其结构核心。CNN 的隐藏层包括了卷积层、全连接层和池化层三类层次结构，下面对卷积层、池化层展开介绍。



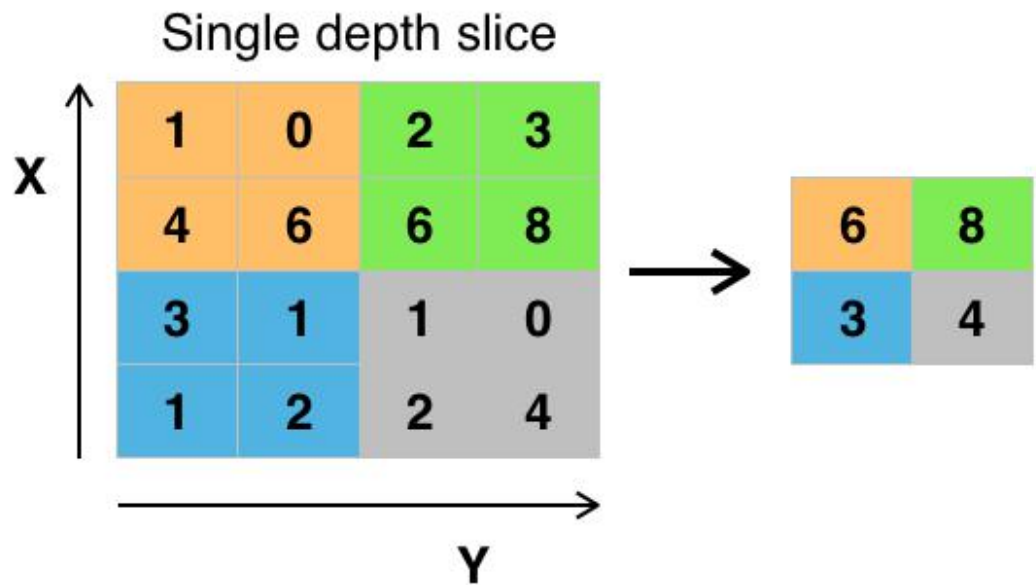
注：图片来自网络

2.1 卷积层（Convolutional layer）

卷积层所做的事情，顾名思义，就是进行卷积运算。曾经在数学分析课上听过一个很形象的比喻：像把一张二维的地毯从角沿着对角线卷起来。而在这里的具体操作，则是对输入图像的区域和进行区域图像特征检测的滤波器进行点积的操作，即所谓“加权叠加”。卷积层可以提取图像的特征。而卷积神经单元的

参数如上面的全连接 BP 神经网络一般，也是通过反向传播得到的。

2.2 池化层(Pooling Layer)

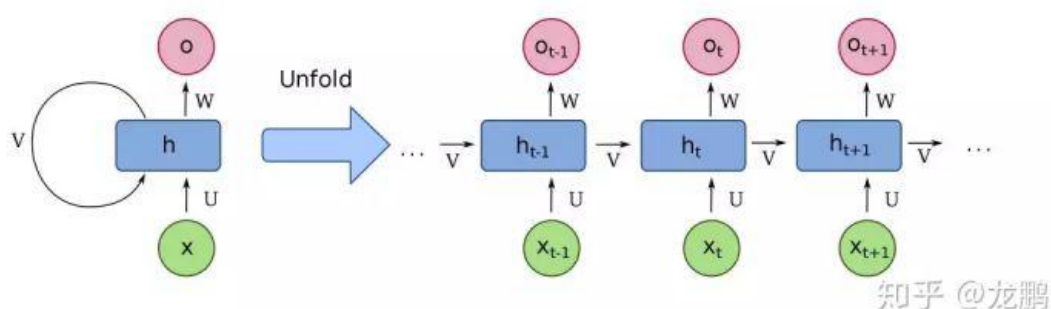


池化层可以将样本的尺寸缩小，从而减少全连接层的神经元数量。而其深层次的原因则是防止过拟合。本文采用的池化类型是最大池化（MaxPooling），看似很深奥其实很简单，就是取不同区域里面的最大的值作为新样本的值罢了。

总的来说，根据卷积层和池化层对样本进行压缩、放大、特征筛选的特点，显然，卷积神经网络（CNN）所擅长处理的一般是图像问题。

3. 长短期记忆网络（LSTM）

3.1. 循环神经网络（RNN）

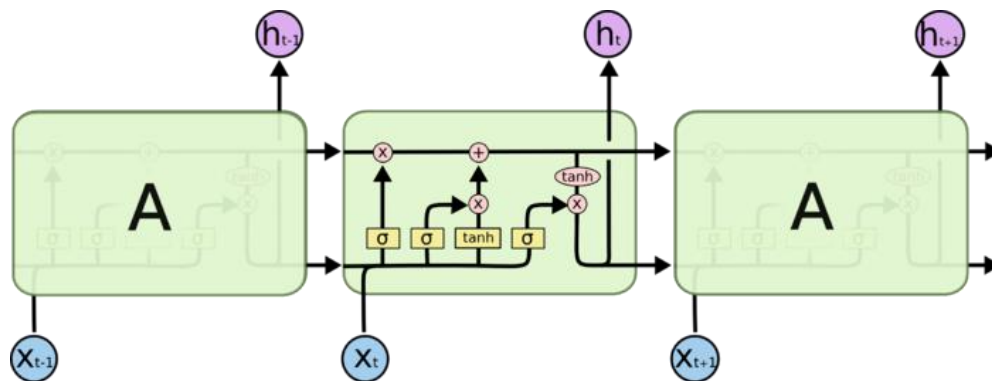


注：图片来自网络

首先我们来说循环神经网络（RNN）。上面提到的 BP 全连接神经网络模型和卷积神经网络模型从时间序列的角度来看都是静止的，但是在实际应用中，显然有不少的研究对象都是“序列信息”，例如语言（语音、文字）、视频等等，前面出现过的内容对我们理解当前这句话、这个词的意思有着很大的影响。这时 RNN 就大派用场了。其与 CNN 只把上一时间点的输出作用在本节点不同，RNN 让以往所有历史连同上一时间点的输出共同作用在本节点中，以此产生了“记忆”的效果。除了图中所示的从前往后的正向 RNN，也有双向的，“结合上下文”型的双向 RNN (Bidirectional RNN)。但是，由于 RNN 也是基于反向传播算法的，且在重复的模块中结构简单（与 LSTM 对比），若序列过长、层数过多再加上 RNN 本身层层之间的相互依赖，很容易产生梯度爆炸和梯度消失的问题。就如老师所说的，“成也循环，败

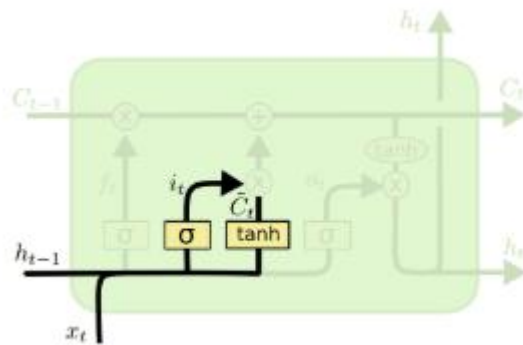
也循环”，因此，RNN 并不是一种实用的模型。接下来，本文介绍 RNN 中被广泛使用的特例长短期记忆网络（LSTM）。

3.2 长短期记忆网络（LSTM）

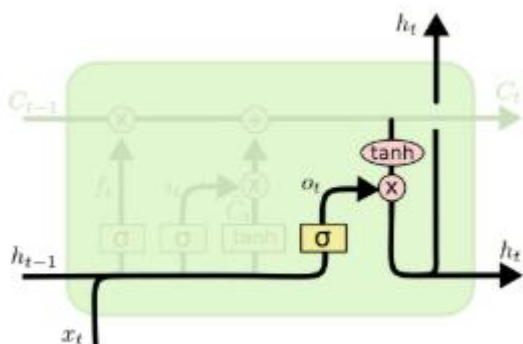


LSTM 是一种特殊的 RNN，其使用了三个“门”以控制信息的记忆，分别为输入门、输出门和遗忘门，“三思”而“三门定状态”，以解决上面提到的各层联系过多导致的梯度消失和梯度爆炸问题。

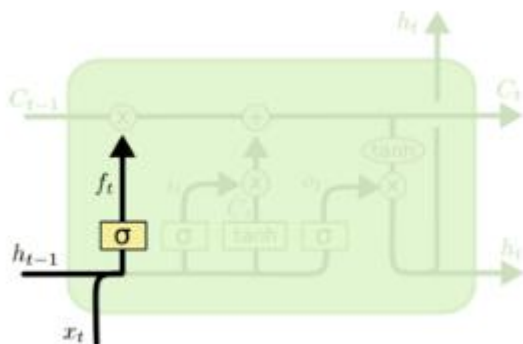
输入门用于控制新数据，取舍使用多少新数据，来更新长期状态。



输出门用于控制输出，取舍输出多少长期状态到短期状态中。



遗忘门用于控制长期状态内容，取舍放下和保留多少长期状



态内容。

三、可转债预测（指标选择与数据预处理）

首先，通过东方财富 CHOICE 平台获取可转债的相关数据。在众多指标中，本文选择了‘可转债代码’，‘开盘价’，‘上市日期’，‘网上中签率’，‘第一持有人持有比例’，‘上市转股溢价率’作为数据集，其中‘开盘价’作为模型的输出值的真实值，其他则作为输入值。下面一一介绍选取各指标的理由。

1. 上市转股溢价率

首先介绍上市转股溢价率。这是预测可转债发行首日开盘价的核心。转股溢价率是指可转债市价相对于其转换后价值的溢价水平。^[1]这是衡量可转债价值的核心指标，但就像股票市场上有的股票市净率高达几十倍几百倍，但也有的股票破净（即市净率低于一倍）一样，可转债首日开盘价并不会仅仅以此而确定。

2. 上市日期

前文曾经提过，由于“盘子小”等各种缘故，可转债上市价格与市场情绪、大盘行情紧密相关。而我们可以基于训练集中某段时间的可转债表现来判断该段时间的情绪与大盘行情。

3. 网上中签率、第一持有人持有比例

与机构、大户、私募、企业账户参与网下打新不同，散户绝大多数都是参与网上打新，因而从网上中签率我们就可以大致看出这只可转债的受欢迎程度。一般情况下，网上中签率越低，市场给出的开盘价格会越高，反之更低。而第一持有人一般是该可转债正股的大股东配售或者是一些战略投资者配售，大多数情况下上市首日并不会进行过多的交易，有助于可转债开盘价的抬升。但需要注意的是，近些时候（2019 年~2020 年初）可转债市场逐渐升温，市场给出的溢价较高，而同时宏观经济下行压力大使得企业资金更为紧张，大股东首日就抛售的情况渐渐发生，该项指标可能会逐渐失灵。

4. 可转债代码

从可转债代码中也可以对可转债首日开盘价一窥一二。可转债代码前三或四位为固定的代码编号，从中可以分辨出该转债正股所在的板块以及是否多次发行可转债。一般多次发行可转债的公司再次发行的可转债市场会给予较高的价格，理由是这种公司先前发售的可转债大多已经顺利地转股或者强赎，可以说明公司对于可转债的重视程度以及操作熟练

度，本次发行的可转债顺利地转股或者强赎的几率很高。

在东方财富 CHOICE 数据平台中，本文选择了自 2008 年以来成功发行上市的可转债作为样本，其中剔除掉数据记录不全、因为特殊情况而偏离正常值的样本后，共 256 只可转债按 1:9 的比例作为我们的验证集和训练集。

四、 模型实操（核心代码）

本文基于keras构建了以上三个神经网络模型并给与评估。

（完整代码及数据集另附）

4.1 BP 神经网络模型

```
#普通 BP

model = Sequential() # 初始化
model.add(Dense(units = 10, # 输入大小
                activation='relu', # 激励函数
                input_shape=(x_train_pd.shape[1],) # 输入大小，也就是列的大小
                )
        )

model.add(Dropout(0.2))

model.add(Dense(units=8,
                activation='relu' # 激励函数
                )
        )

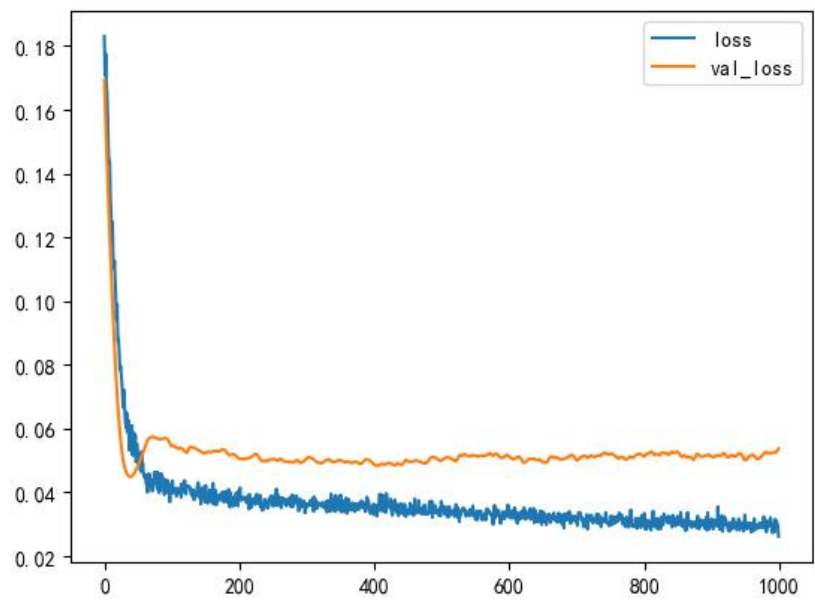
model.add(Dense(units=1,
                activation='linear' # 线性激励函数 回归一般在输出层用这个激励函数
                )
        )

print(model.summary()) # 打印网络层次结构

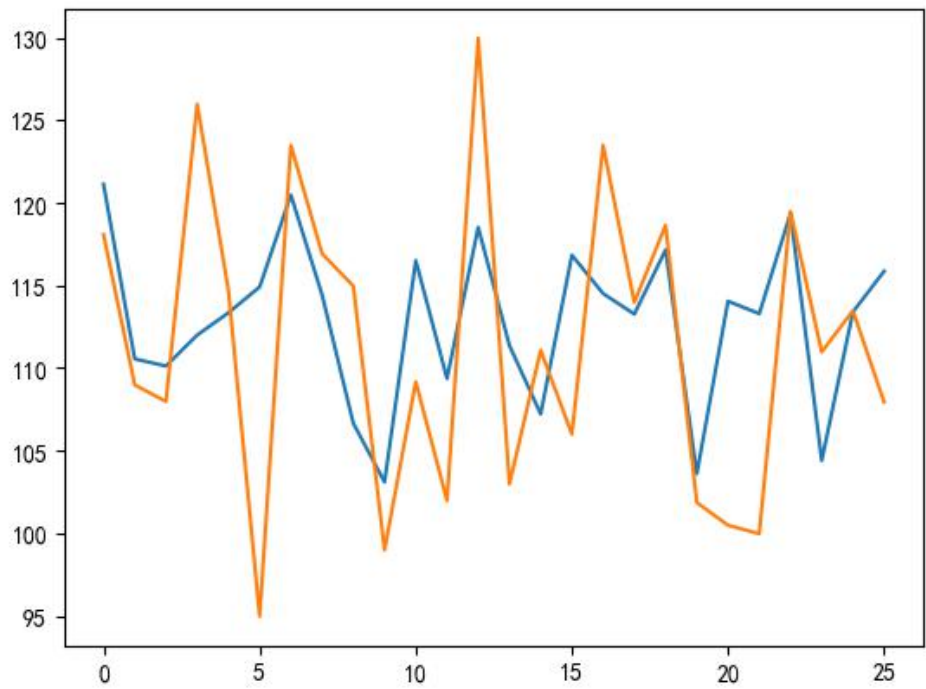
model.compile(loss='mse', # 损失均方误差
              optimizer='adam', # 优化器
              )

model.fit(x_train, y_train,
        epochs=1000, # 迭代次数
        batch_size=128, # 每次用来梯度下降的批处理数据大小
        shuffle=True,
        verbose=2,
        validation_data = (x_valid, y_valid) # 验证集
        )
```

训练后以验证集的真实值与预测值数据、以及训练时的 loss 画



图。



4.2 卷积神经网络模型

为了更好地适应卷积神经网络模型的特性，本文将应用的场景略作修改为“参与收购这只可转债首日卖出是否会盈利”，即把问题变成一个“是否盈利”的分类问题。接下来，将 6 个不同的输入值指标 reshape 成 2*3 大小的输入值，以此“卷积”。

```
x_train = x_train.reshape(x_train.shape[0], 3, 2, 1)
x_test = x_test.reshape(x_test.shape[0], 3, 2, 1)

model = Sequential()
model.add(Conv2D(64,(2,2),strides=(1,1),input_shape=x_train.shape[1:],padding='same',activation='relu',kernel_initializer='uniform'))

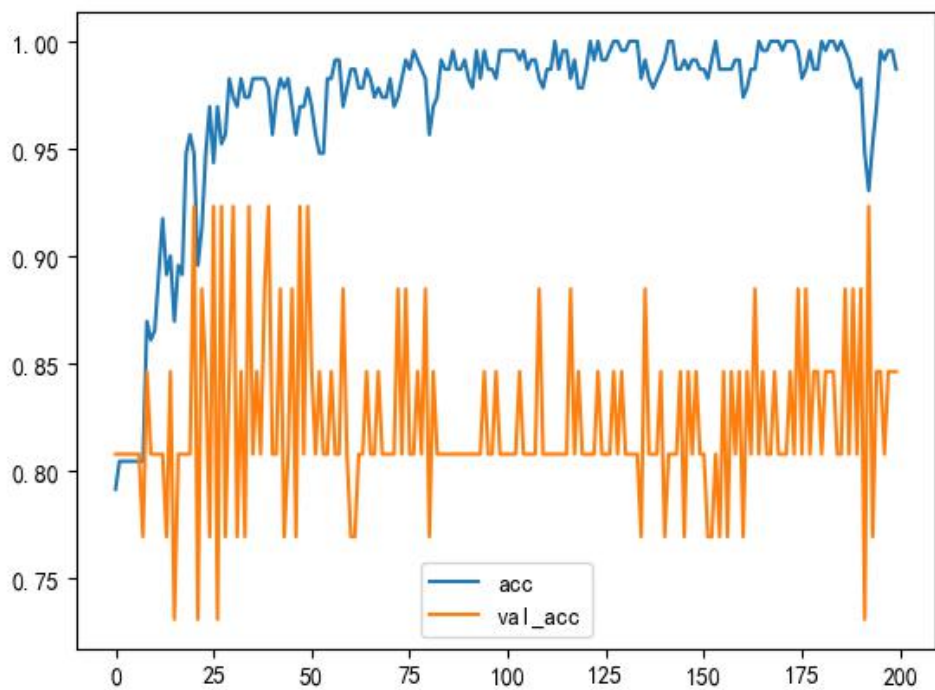
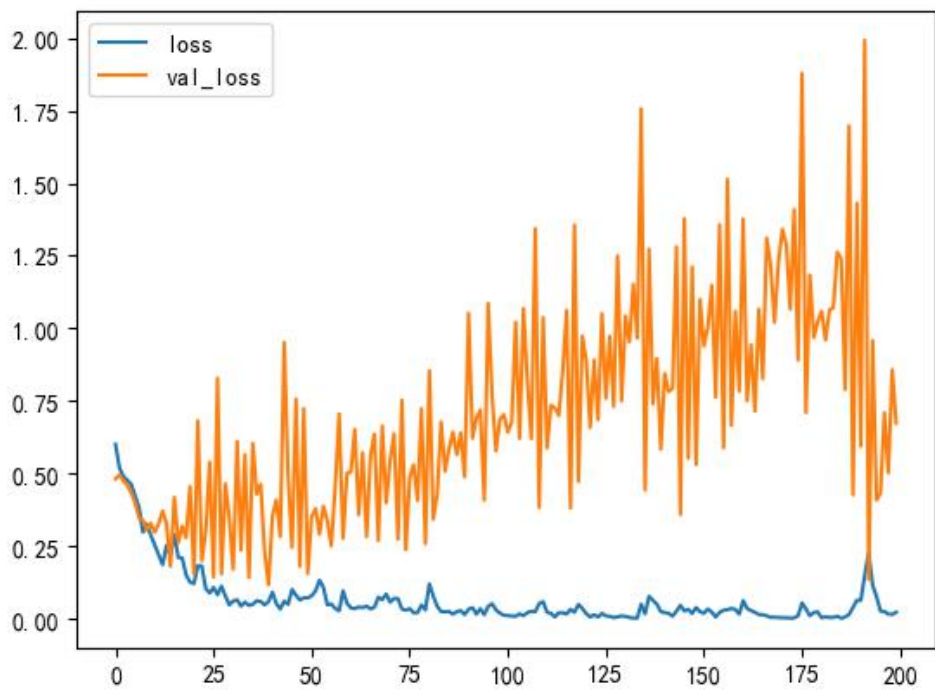
model.add(Conv2D(64,(2,2),strides=(1,1),padding='same',activation='relu',kernel_initializer='uniform'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.add(Dense(256,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(2,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
model.summary()

# train the model 训练模型
history=model.fit(x_train, y_train,
                  batch_size=32,
                  epochs=200,
                  verbose=1,
                  validation_data=(x_vaild, y_vaild))
```

将训练过程中的误差与正确率绘图。



从图中我们可以看到，模型对训练集的拟合良好。训练过程中，训练集正确率可以上升并稳定在 0.95 以上。而对于验证集正确

率则在 0.73~0.93 之间震荡。

4.3 LSTM 神经网络模型

在 LSTM 神经网络中，“序列”可谓是其中的灵魂。可惜的是，各支分别上市的可转债并没有像某支特定股票一样特别明显的“序列”概念，但正如本文前面所说，可转债市场情绪因素影响明显，而情绪总是随着时间的变化而变化，因此，“时间序列”这一概念在此应用中也许可能会有一定的参考价值。因此，本文抱着尝试的态度将可转债发行数据重新按时间顺序排列，进行 LSTM 模型训练。

```
model = Sequential()

model.add(LSTM(128,input_shape=(timesteps,1),return_sequences= True))

# 防止过拟合 丢弃
model.add(
    Dropout(0.2)
)

model.add(
    LSTM(
        200,
        return_sequences=False
    )
)

model.add(
    Dropout(0.2)
)

model.add(Dense(output_dim=1))

model.add(Activation('linear'))
```

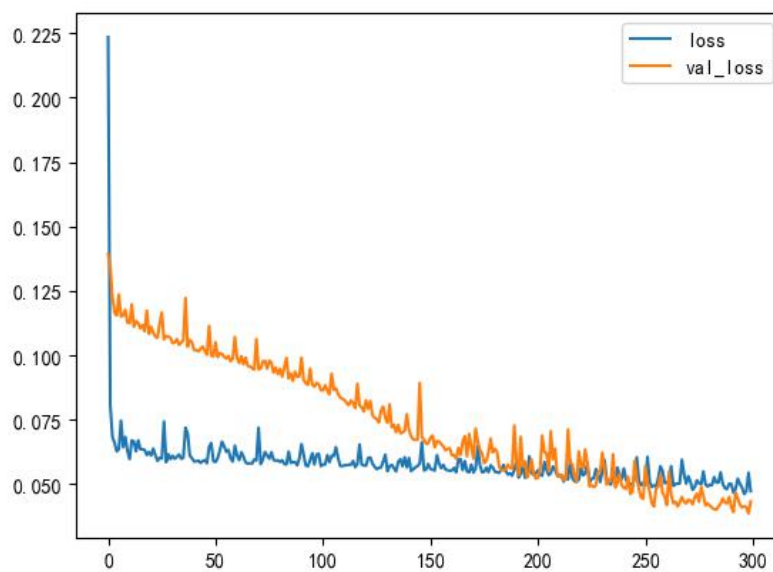
```
model.compile(loss='mse', optimizer='rmsprop')

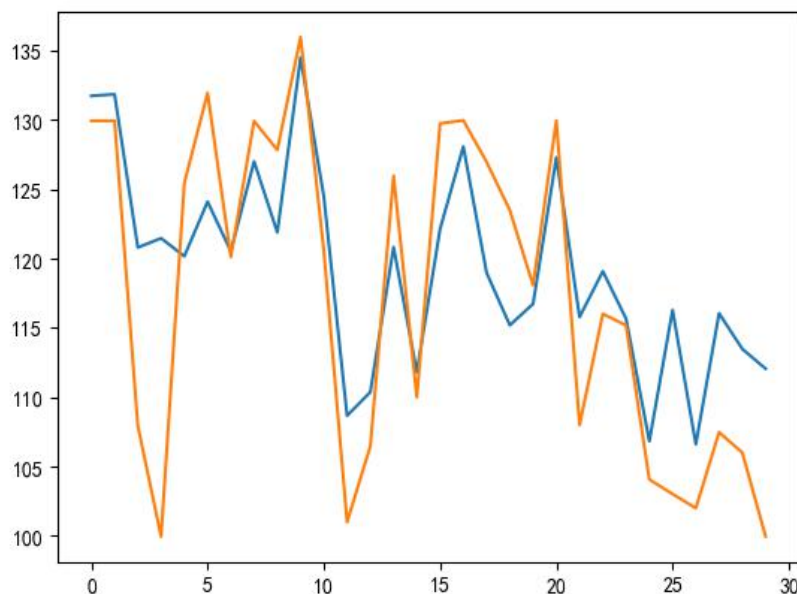
# 3 训练模型

history=model.fit(x_train, y_train,
                  batch_size=120,
                  epochs=300,

                  validation_data = (x_valid, y_valid) # 验证集
)
```

训练后以验证集的真实值与预测值数据、以及训练时的 loss 画图。





效果与上面的 BP 模型相比似乎略差，看来可转债发行的“时间序列感”并不强，效果总体差强人意。

为了发挥出 RNN 神经网络处理“序列”的能力，在下一部分中本文将会对 RNN 的另一变种 GRU 寻找更为合适的应用场景。

(二) 基于 GRU 的关于北上资金的上证综指走势研究

一、背景

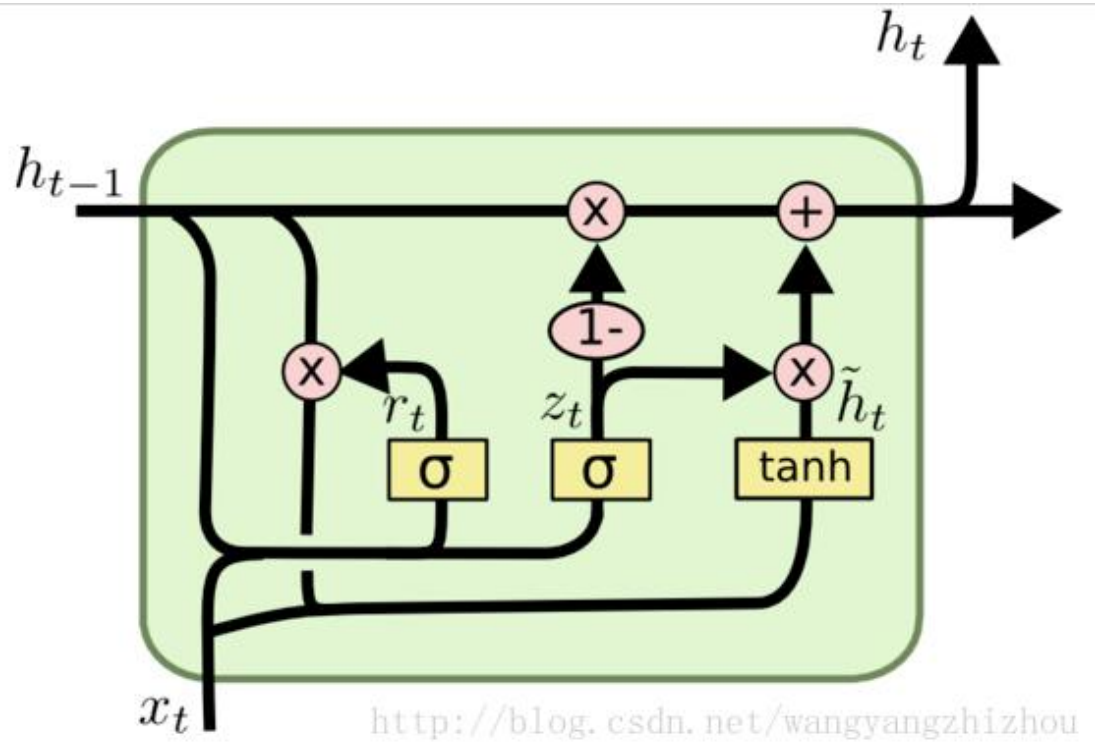
近些年来，在加强金融市场对外开放的政策背景下，“北上资金”在我国金融市场的话语权和受关注度都得到了明显的提升。甚至有人说，“过去 5 年来股市格局最大的变化，就是北上资金的崛起。”北上资金所指的是 A 股市场中的香港资金和国际资金。北上资金曾在 A 股市场创造出不少辉煌战绩。在 2019 年初北上资

金曾在 2018 年股市低迷的情况下，爆买以中国人保为首的蓝筹股，千金买马“股”引领了 2019 年的一波春节行情。2020 年 2 月 3 日，农历新年后第一个交易日，市场笼罩在新型冠状病毒的阴影下“千股跌停”，上证综指一日跳空暴跌 7 个点。可就在许多机构因害怕“新冠”病毒对经济的冲击而仓皇逃命之时，北上资金却大笔流入，与主流资金逆向而行。谁对谁错呢？在接下来的一周，A 股迎来了绝地反击，不仅收复了当天暴跌的缺口，还重新站上了 3000 点大关。北上资金的“聪明资金”头衔又一次被坐实。笔者不禁产生了一个想法：如果以北上资金为影响因子，利用深度学习模型，是否能尝试着预测上证综指的走势呢？

二、 算法简介

GRU (Gate Recurrent Unit) 是 LSTM 的变体，也是 RNN 的一种。关于 RNN、LSTM 在上文已有简介，在此不再赘述。下面对 GRU 进行简要介绍。

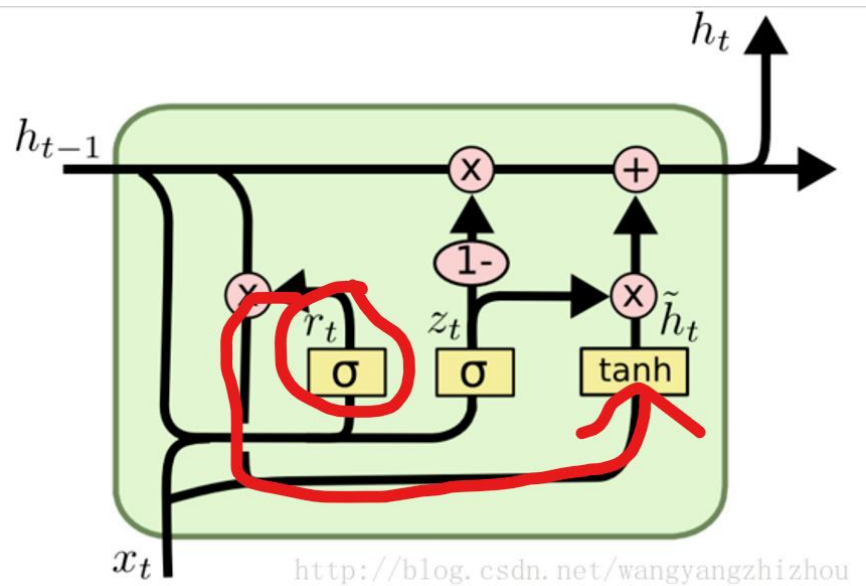
先来看看 GRU 的结构图：



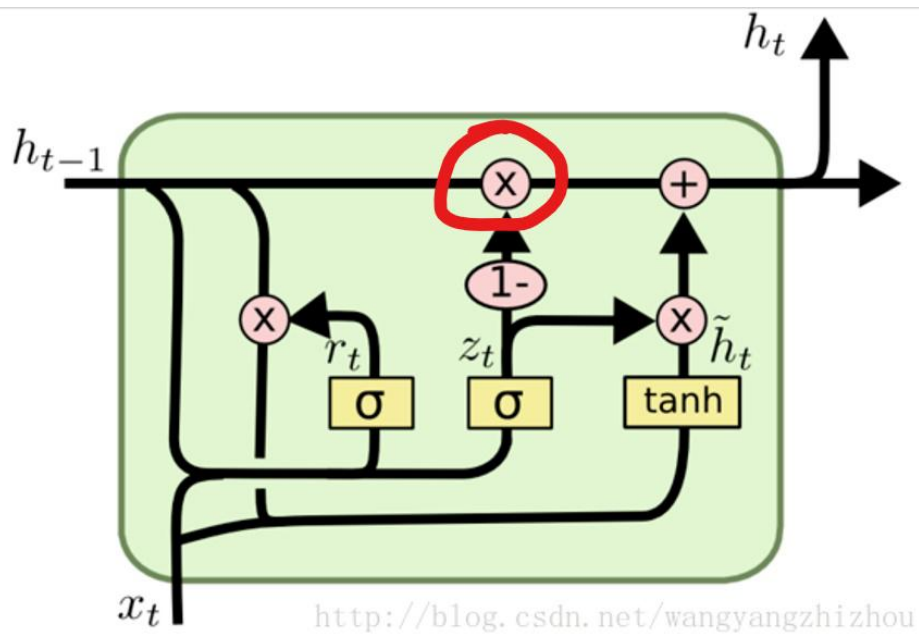
注：图片来自网络

我们可以看到，其与上文 LSTM 的结构图非常类似。LSTM 中有三门，分别是输入门、输出门和遗忘门。而 GRU 则做了精简，只有两个门：更新门和重置门。

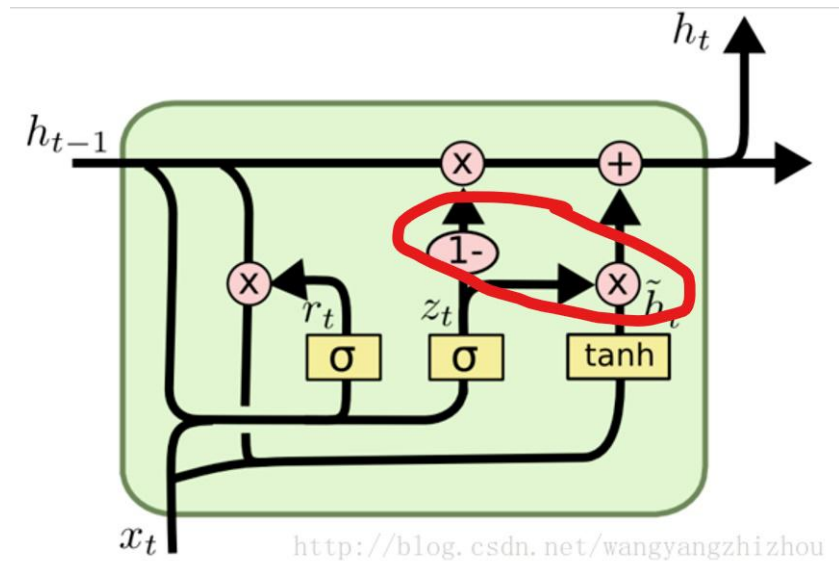
在 GRU 中，重置门任务较轻，模型在此用它获得 \tilde{h} 。



更新门则任务重，同时肩负了遗忘和选择记忆的任务。



此处表示对 h_{t-1} 中不重要的性状的“遗忘”，可以类比 LSTM 中的遗忘门。



此处 $(1-z_t) * \tilde{h}_t$ 表示的是对本节点的 \tilde{h}_t 的“记忆”过程，可以理解为对该维度中的信息的选择。

最后两者结合构成本节点输出。

三、 上证综指涨跌预测（指标选择与数据预处理）

鉴于准确预测上证综指的点位难度较大，本文把应用问题设置为一个分类问题，即“上证综指该日收盘是涨还是跌”。基于真实交易日里盘中可获得的数据，经权衡比较后，本模型的输入选择了 6 项参考值，分别是‘当日开盘价’，‘当日成交金额’，‘日期’，‘上日涨跌幅’，‘当日沪港通成交净买额’，‘昨日收盘价’。通过东方财富 CHOICE 获得了自 2014 年 11 月陆港通开通以来到 2020 年 2 月 19 日的相关数据。剔除部分因香港假期陆股通暂停交易的数据后，将 1221 天的历史记录切割作为我们的训练集和验证集。

四、 模型实操（核心代码）

本文基于 keras 实现了上述模型并给予评价。（完整代码及数据集另附）

```
model = Sequential()

model.add(GRU(128,input_shape=(timesteps,1),return_sequences= True))

# 防止过拟合 丢弃
model.add(
    Dropout(0.2)
)

model.add(
    GRU(
        200,
        return_sequences=False
    )
)

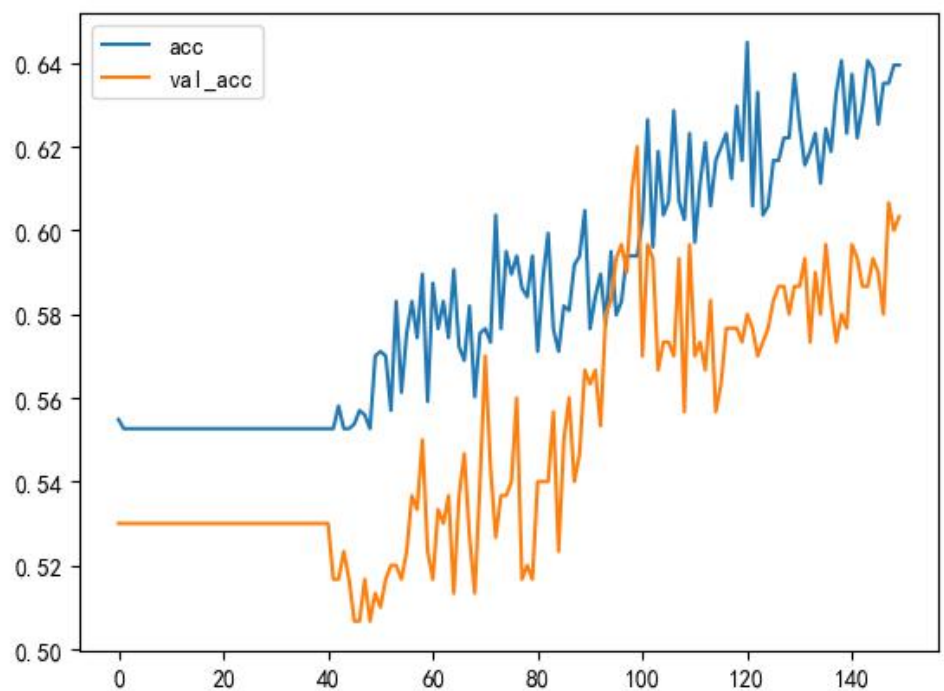
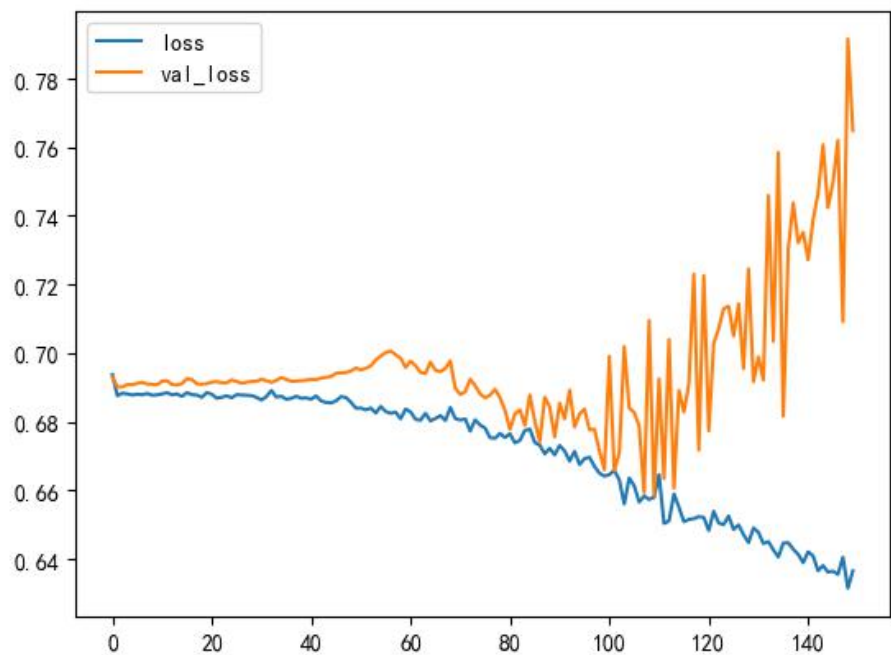
model.add(
    Dropout(0.2)
)

model.add(Dense(2,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
# 3 训练模型

history=model.fit(x_train, y_train,
                  batch_size=256,
                  epochs=150,
                  validation_data = (x_valid, y_valid) # 验证集

)
```

将模型训练过程作图。



从图中可以看到,模型在100个epochs处效果良好,但在之后就开始出现loss下降而val_loss反而上升的情况,或是“过拟合”的发生导致的。最后模型验

证集正确率在 0.6 左右。在这枪林弹雨的 A 股市场，
涨跌 60%的正确率尚可接受，也可为投资者提供一点点
帮助了。

● 感言

人工智能是结果，深度学习是方法。在这个寒假里，我通过老师的视频系统地了解了许多相关的知识，结合课上知识再自己通过网络资源学习相关的代码知识，掌握了许多深度学习模型，受益颇深。老师的慕课讲得很好，既有普及性也有深度性，笔者这种科班学生学习也不会觉得内容浅，反而觉得很有挑战性。拙作是笔者寒假依靠慕课和网络资料捣鼓出来的，实属闭门造车，若有纰漏敬请老师和助教不吝赐教，感谢！

● 参考资料

<https://zhuanlan.zhihu.com/p/32481747>

<https://www.jianshu.com/p/95d5c461924c>

<https://www.jianshu.com/p/1ea2949c0056>

<https://zhuanlan.zhihu.com/p/40601434>

<https://baike.baidu.com/item/%E5%8C%97%E4%B8%8A%E8%B5%84%E9%87%91/23548524?fr=aladdin>

肖琪. 人工神经网络在股票预测中的应用研究. 华南理工大学 2017

甄远涛. 基于深度学习 LSTM 神经网络的障碍可转债设计与定价. 兰州财经大学, 2019.

余强. 基于 GRU 神经网络研究不同证券市场对股票收益的影响——以恒生和上证指数为例 [J]. 经济研究导刊, 2019 (35): 117-120.