# 期末算法报告

和媛媛

## 一、基于 RNN 的股票预测

### 1. ****简介****

本部分详细描述了开发，改进和评估一个基于 RNN（循环神经网络）模型的过程，该模型旨在使用五年期的历史收盘价数据预测目标公司的股票价格（收盘价）。

### 2. ****单只股票单个数据列输入预测****

#### 2.1 数据准备

首先我们需要提取并归一化数据：数据集 all\_stocks\_5yr.csv 包含历史股票数据。提取 AAPL 的数据，并仅保留 date 和 close 列进行分析，我们将 date 列转换为日期时间格式，并按日期排序，使用 MinMaxScaler 将收盘价归一化到 [0, 1] 范围内。这种归一化对于改善神经网络的收敛性至关重要。

然后为了进行时间序列上的预测和训练，我们需要以时间为序列整理数据集，为了准备 RNN 模型的数据，创建了固定长度（60）的序列。每个序列是用于预测后续价格的归一化收盘价的时间窗口。

数据集分为训练集（80%）和测试集（20%）。然后将数据转换为 PyTorch 张量，并重塑以匹配 RNN 模型的输入要求。

#### 2.2 ****RNN 模型架构定义****

RNN 模型由一个 RNN 层和一个全连接（线性）层组成。RNN 层捕获数据中的时间依赖性，全连接层将 RNN 输出映射到所需的输出大小。

#### ****训练函数，评价函数和预测函数****

训练：在每个epoch，使用训练数据进行前向传播，将优化器的梯度缓存置零，计算损失，反向传播误差，计算梯度。最后，使用优化器更新模型参数。每10个epoch输出一次当前的损失值。

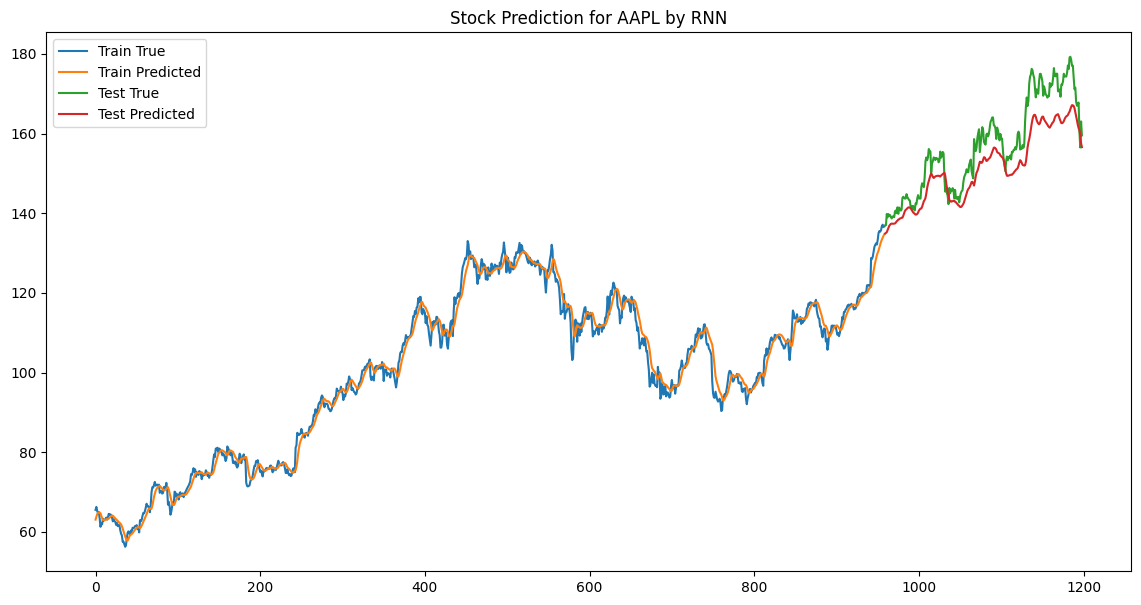
#### ****2.4 训练，评估，可视化****

使用均方误差（MSE）损失函数和 Adam 优化器训练模型。训练过程涉及迭代数据指定的周期数（100），并更新模型权重以最小化损失。

评估模型在测试集上的表现。计算预测值和实际值之间的均方误差（MSE），以评估模型的性能。

在训练和测试数据上生成预测值。使用缩放器将预测值和真实值反向变换回原始尺度。绘制训练和测试集的真实值和预测值，以可视化模型的性能。

RNN 模型在测试集上的 MSE 大约为 0.0040，但是训练集上的MSE 大约为 0.0004。绘制的结果显示模型能够捕捉到股票价格的总体趋势，但在准确预测短期波动方面可能存在局限性。



#### ****2.5 分析****

测试集MSE偏高，说明预测精度可以通过增加数据量等方式进一步提高，从股价预测图像上来看，训练集拟合较好，但是测试集拟合不够，训练集MSE显著低于测试集MSE，需要增加数据量和采取防止过拟合方法进行训练。针对此问题，有两种改进的办法，第一种是增加股票的数量，考虑股票之间的相互影响可以提供更多信息，另外一种是增加股票中不同维度的信息，比如虽然输出为收盘价，但是模型输入信息可以扩展到开盘价、最高价、最低价、收盘价、成交量等。

### 3. ****多只股票联合训练模型****

#### ****3.1**** 改进

相比于只对单只股票进行训练的模型，本模型通过同时考虑多只股票的数据，利用更多的市场信息，提高了预测的准确性和模型的泛化能力。

#### 3.2 数据准备

同样使用了包含5年股票数据的all\_stocks\_5yr.csv文件。该文件包括日期、开盘价、最高价、最低价、收盘价、成交量等信息。

归一化：normalize\_stock\_data函数用于对股票数据进行归一化处理，使得不同股票的数据具有相同的范围。这有助于提高模型训练的稳定性和收敛速度。

序列建立：为了将时间序列数据转换为RNN模型可接受的输入格式，定义了create\_sequences函数。该函数通过滑动窗口技术，将数据分割成长度为seq\_length的序列，每个序列都对应一个预测目标值。

数据集分割和标准化：prepare\_data函数用于处理多只股票的数据，包括数据归一化、创建序列和分割训练集与测试集。对于每只股票，我们使用MinMaxScaler进行标准化处理，以确保不同股票的数据具有相同的范围。而prepare\_single\_stock\_data函数用于为特定股票准备数据，主要用于模型评估和可视化。

#### 3.3 模型设计

定义了一个名为SimpleRNN的RNN模型。包括以下组件：

一个RNN层，用于处理时间序列数据。

一个Dropout层，用于防止过拟合。

一个全连接层，将RNN层的输出转换为预测值。

#### 3.4 ****训练，评估，可视化****

train\_model函数用于训练RNN模型。我们使用均方误差（MSE）作为损失函数，并使用Adam优化器进行梯度下降。训练过程中记录每个epoch的训练损失。

evaluate\_model函数用于在测试集上评估模型，计算并返回预测值和MSE。

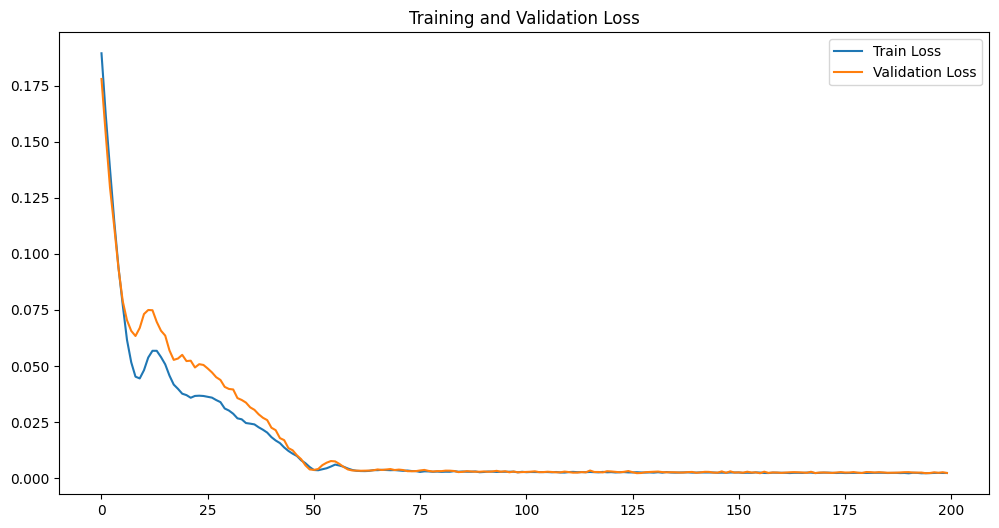
plot\_results函数用于可视化训练集和测试集上的预测结果，直观展示模型的预测能力。

inverse\_transform函数用于对标准化后的数据进行逆变换，以便更好地理解预测结果。evaluate\_and\_plot函数用于对单只股票进行评估并绘制预测结果。

#### 3.5 ****实验和分析****

选择四只股票（AAPL、MSFT、GOOG、AMZN），使用prepare\_data函数准备训练和测试数据。

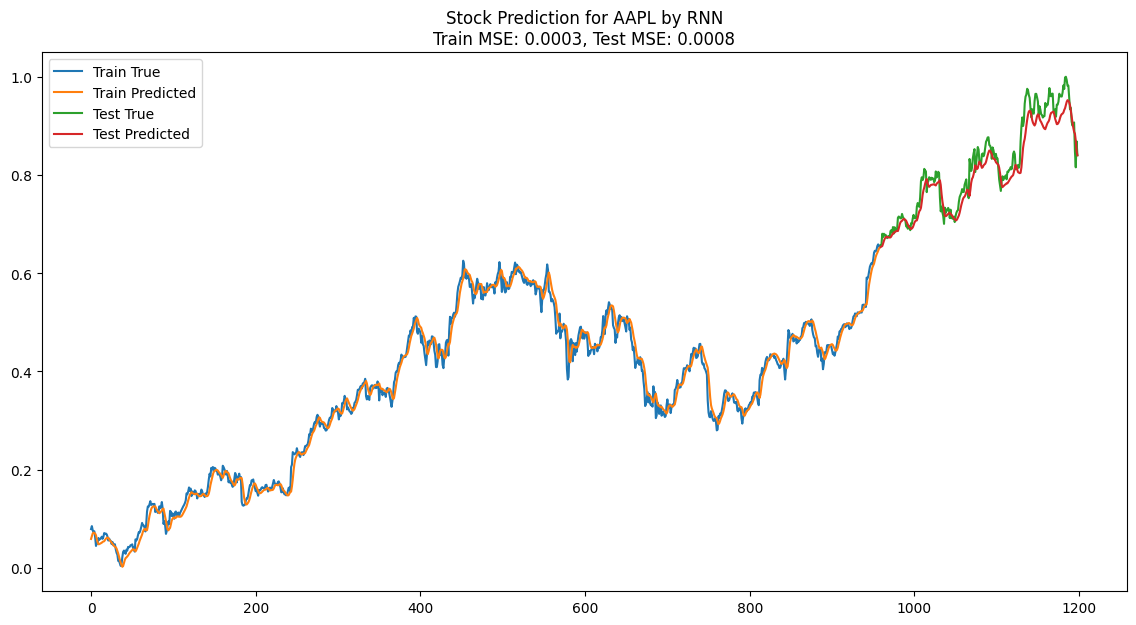
我们使用train\_model函数训练RNN模型，训练过程中的损失曲线如下图所示。



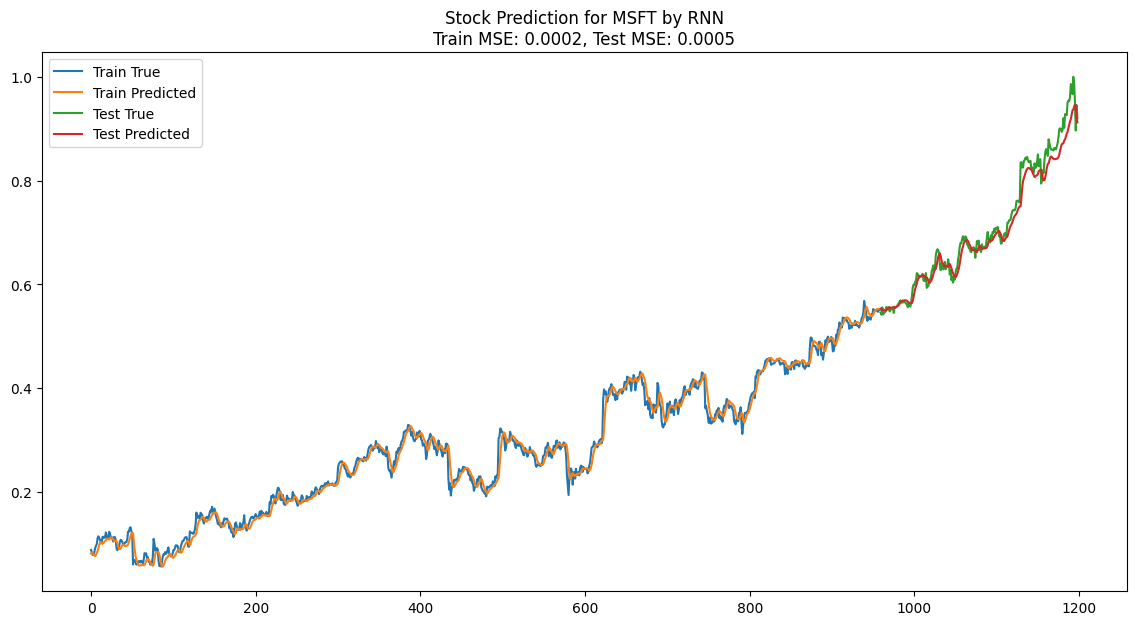
**模型评估：**评估结果表明，在测试集上的MSE：RNN Test MSE: 0.0012

**股票评估：**通过对AAPL、MSFT、GOOG和AMZN股票的评估和绘图，我们展示了模型在训练集和测试集上的预测结果，并计算了MSE：

AAPL:



MSFT:



（另外两只股票的曲线已在源代码文件中给出）

**优势和分析**

**a.**利用更多市场信息**：**多只股票的数据提供了更多的市场信息，这有助于模型更好地捕捉市场趋势和模式，提高预测的准确性。

b.提高泛化能力:多只股票的数据可以防止模型过拟合单一股票的特定模式，从而提高模型的泛化能力，使其在不同股票上的表现更加稳健。

c.降低数据波动影响:单只股票的数据可能会受到一些异常事件的影响，而多只股票的数据能够平滑这些波动，提高模型的鲁棒性。

结论

通过同时考虑多只股票的数据，我们的RNN模型在预测股票价格方面表现出更高的准确性和更强的泛化能力。可以进一步优化模型结构，尝试更多的特征工程。

### 4. ****多维度数据训练模型（单股和多股训练对比）****

#### ****4.1**** 改进

与仅使用单一特征（如收盘价）进行预测的模型相比，本模型引入了多维度数据（开盘价、最高价、最低价、收盘价、成交量），从而提高了预测的准确性和模型的泛化能力。首先每只股票均训练一个独立的模型，利用其多维度数据进行预测。然后最终模型我们采用多维度数据，**多只股票联合训练的模型。**

#### ****4.2**** 数据准备

选择包括日期、开盘价、最高价、最低价、收盘价和成交量在内的列。对日期列进行日期类型转换并按照日期排序，使用MinMaxScaler将所有选定列归一化到0到1之间的范围，返回归一化后的数据和scaler对象。

调用normalize\_stock\_data函数获取归一化后的数据和scaler，调用create\_sequences函数生成输入序列和目标值。将数据分为训练集和测试集，比例为8:2。将数据转换为PyTorch的张量格式。

这里仅预测收盘价，所以目标值y\_train和y\_test只保留第4列（即close列），并将其增加一个维度以符合模型的输入格式。

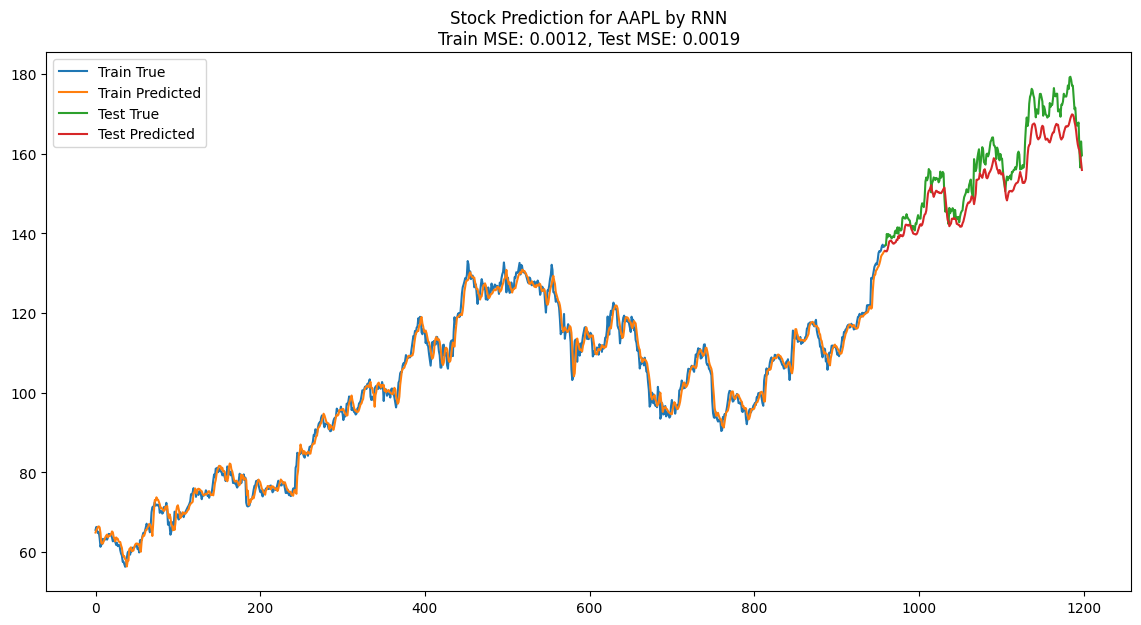
#### ****4.3**** 模型设计

模型设计同上述3中的RNN。

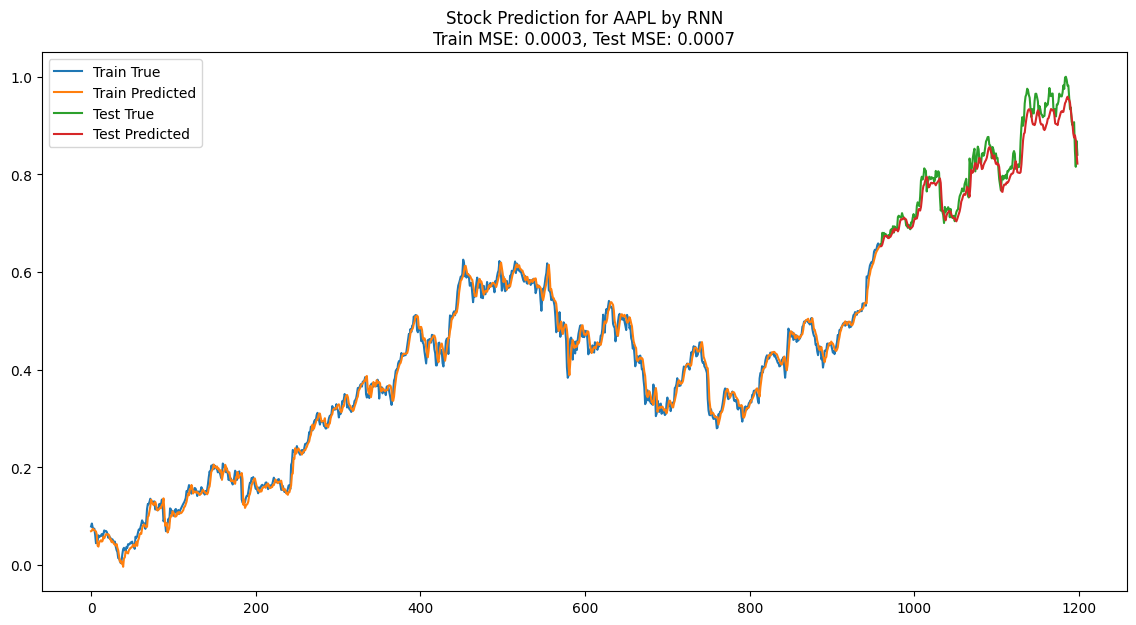
#### ****4.4 修改后实验结果****

以AAPL股为例（另外三只股票的曲线已在源代码文件中给出）:

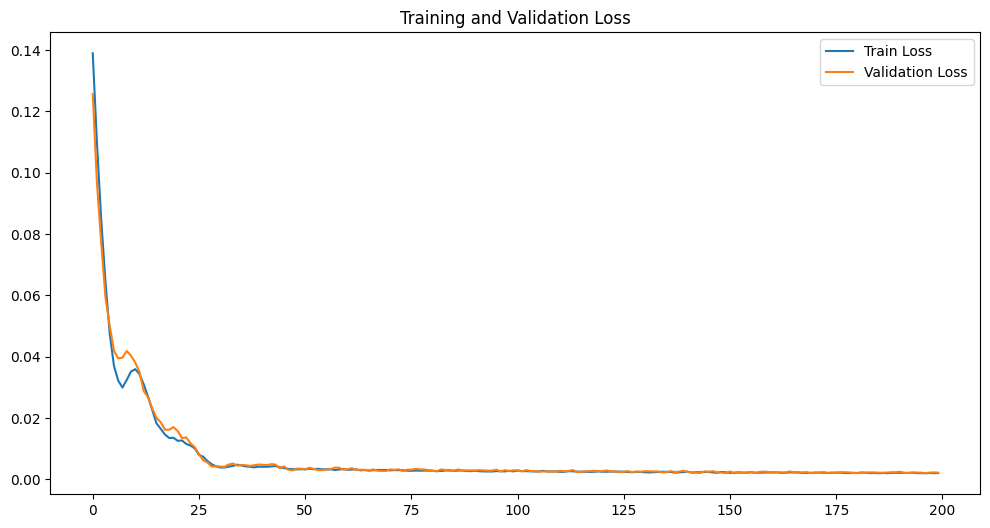
AAPL单股多维联合训练结果：



AAPL多股多维联合训练结果：



多股多维联合训练的Loss变化曲线如下：



### 5. ****对RNN股票预测实验的分析总结****

在本次分析实验中，我们针对一个简单的RNN网络设计了四种不同的训练方法，并进行了针对性的对比实验，我们首先建立对单个股票单个维度的RNN模型训练方法，之后我们将多只股票（4只）引入丰富数据集，最后我们将维度增大到多维度数据（开盘价、最高价、最低价、收盘价、成交量），获得的主要实验结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 基本收敛epochs（AAPL股） | Train\_mse（AAPL股） | Test\_mse（AAPL股） |
| 单只股单维度数据RNN模型训练 | 大约90轮 | 0.0006 | 0.0081 |
| 多只股单维度数据RNN模型训练 | 大约180轮 | 0.0003 | 0.0008 |
| 单只股多维度数据RNN模型训练 | 大约100轮 | 0.0012 | 0.0019 |
| 多只股多维度数据RNN模型训练 | 大约150轮 | **0.0003** | **0.0007** |

我们可以得出，虽然多只股票增加了数据量数据量，延长了训练时间，但是，我们可以训练出拟合能力更强的RNN模型，数据在数据量和特征上的丰富对模型学习丰富的范式和预测准确的收盘价有很大作用。

**二、基于 LSTM 的股票预测**

**1. 简介**

详细描述了开发，改进和评估一个基于LSTM（长短期记忆）神经网络模型的过程，该模型旨在使用五年期的历史收盘价数据预测目标公司的股票价格（收盘价）。

### 2. ****单只股票单个数据列输入预测****

#### 2.1 数据准备

类似于RNN网络的数据准备，使用包含五年股票数据的all\_stocks\_5yr.csv文件，选择AAPL的收盘价，并将其标准化到[0, 1]区间，将标准化后的数据转换为固定长度的输入序列和目标值。

#### 2.2 模型设计

LSTM网络实现包含了LSTM网络中的四个门：遗忘门、输入门、输入总信息门和输出门。

a.遗忘门参数：Wxf是输入到隐藏状态的线性变换，Whf是隐藏状态到隐藏状态的线性变换，bf是遗忘门的偏置。

b.输入门参数：Wxi是输入到隐藏状态的线性变换，Whi是隐藏状态到隐藏状态的线性变换，bi是输入门的偏置。

c.输入总信息参数：Wxc是输入到隐藏状态的线性变换，Whc是隐藏状态到隐藏状态的线性变换，Bc是输入总信息的偏置。

d.输出门参数：Wxo是输入到隐藏状态的线性变换，Who是隐藏状态到隐藏状态的线性变换，bo是输出门的偏置。

初始化中，h\_t是隐藏状态，初始化为零，c\_t是细胞状态，初始化为零。

在某一个时间步中，获取当前输入x\_t，计算遗忘门f\_t（决定哪些信息需要遗忘），计算输入门i\_t（决定哪些新信息需要存储），计算输入总信息c\_tilde\_t（生成新的候选细胞状态），更新细胞状态c\_t，计算输出门o\_t（决定哪些信息需要输出）。生成新的隐藏状态h\_t。

这个LSTM模型在时间步循环中依次更新每个时间步的隐藏状态和细胞状态，并在最后一步生成输出。

#### 2.3 ****训练，评估，可视化****

训练循环num\_epochs次，每个周期进行以下步骤：

a.通过模型计算输出outputs。

b.清除优化器的梯度缓存，使用optimizer.zero\_grad()。

c.使用损失函数计算预测值与真实值之间的损失。

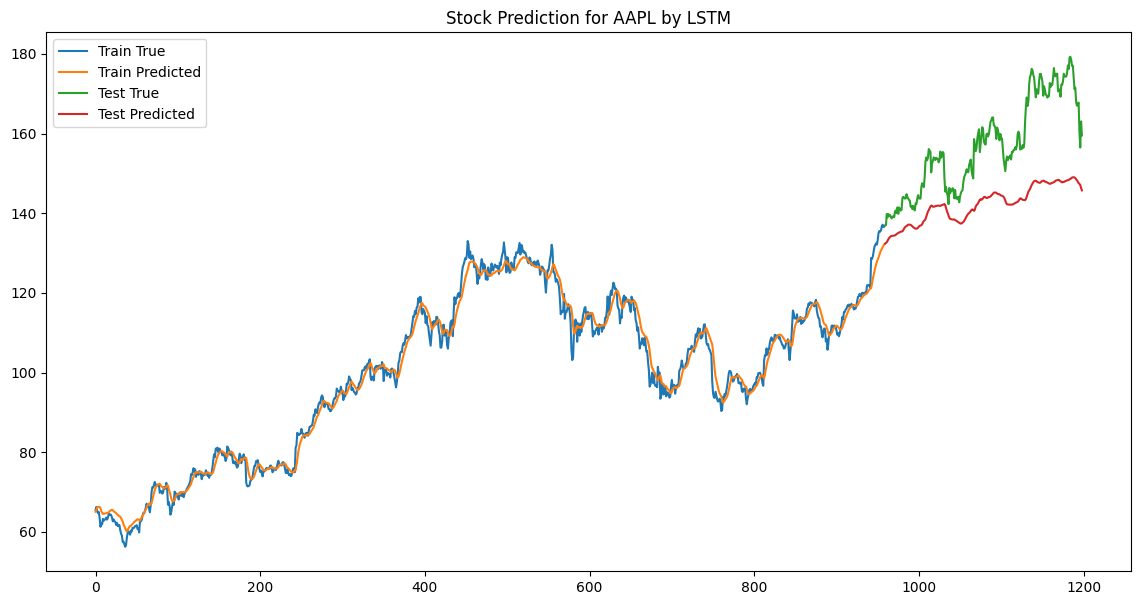
d.通过loss.backward()计算梯度。

e.使用optimizer.step()更新模型参数。

评估模型函数分析使用model.eval()将模型设置为评估模式，禁用Dropout和BatchNorm等训练时特有的操作。使用with torch.no\_grad()上下文管理器禁用梯度计算，减少内存使用，加速计算。通过模型计算测试集的预测值predictions。使用MSE损失函数计算预测值与真实值之间的均方误差mse。

将标准化后的预测值和真实值反向变换回原始尺度，以便进行结果分析和可视化，将训练集和测试集的真实值和预测值存储在两个DataFrame中，便于后续的可视化，使用 plot\_results 函数绘制训练集和测试集的真实值和预测值，以直观地展示模型的预测性能。

LSTM模型在测试集上的 MSE 大约为 0.0174，但是训练集上的MSE 大约为 0.0032。在数据量少的情况下，LSTM的预测效果不佳。难以反映数据趋势和细节。



同样的，我们也引入多只股票多维度来改善训练效果。

### 3. ****多只股票联合训练模型****

#### ****数据标准化和序列创建****

同上，首先对每只股票的数据进行标准化处理，创建时间序列数据，并生成训练和测试集，将多只股票的数据合并成一个大的训练集和测试集。

#### ****模型训练，模型评估****

训练：使用合并后的训练集训练LSTM模型，在训练过程中，模型可以学习到多只股票的价格变化模式，并通过模式共享提高预测性能。

评估：使用合并后的测试集评估模型的性能，通过计算均方误差（MSE）来衡量模型的预测准确性。

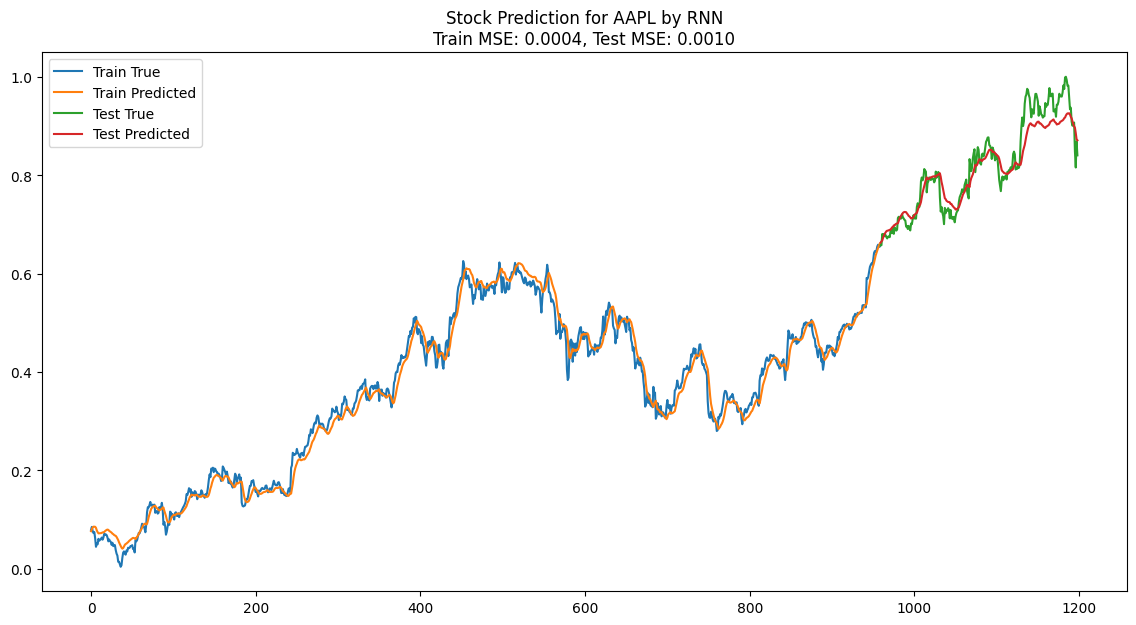
#### ****可视化和分析****

在多只股票联合训练中，我们首先为每只股票创建时间序列数据，并将这些数据标准化。通过为每只股票单独创建序列，然后将所有序列组合起来，可以使模型同时学习多只股票的历史价格模式。

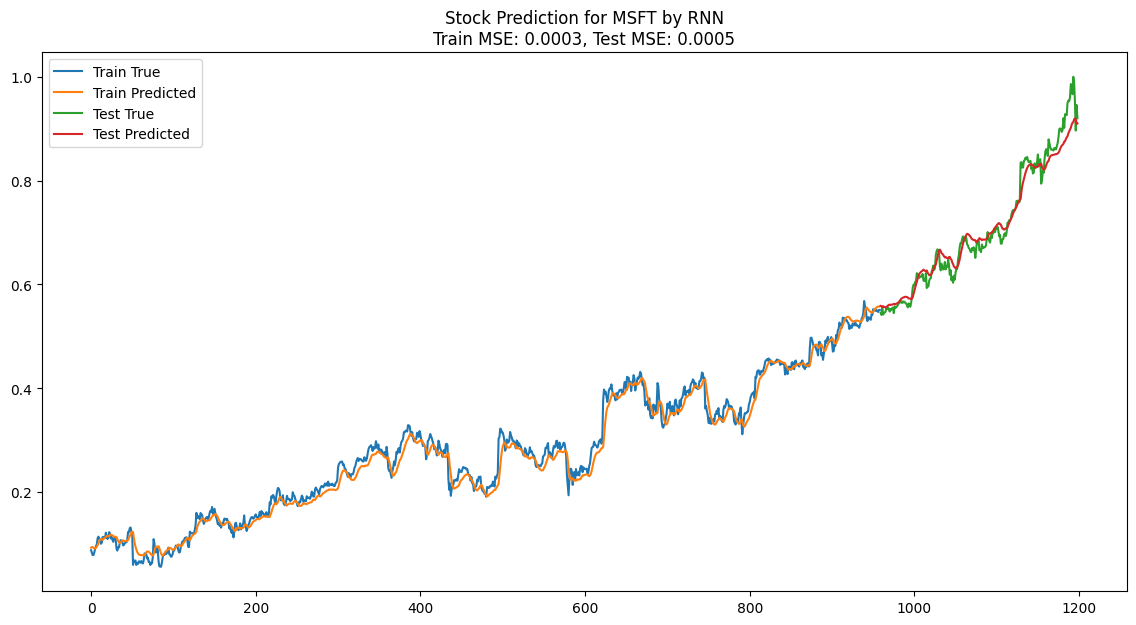
绘制训练集和测试集的真实值和预测值，以直观地展示模型的预测性能，接着为单只股票评估模型并绘制结果。

股票预测结果：

AAPL预测和MSE:



MSFT预测和MSE:



联合训练的优点：

首先是数据量增加，多只股票的数据合并后，训练集的数据量显著增加，有助于模型的训练。

其次是模式共享，不同股票的价格变化模式可能具有相似性，联合训练可以帮助模型捕捉这些共享模式，从而提高整体预测性能。

### 4. ****多维度数据训练模型（单股和多股训练对比）****

#### ****4.1数据标准化和序列创建****

对每只股票的多维度数据进行标准化处理，创建时间序列数据，并生成训练和测试集，将多只股票的数据合并成一个大的训练集和测试集。

#### ****4.2模型训练和评估****

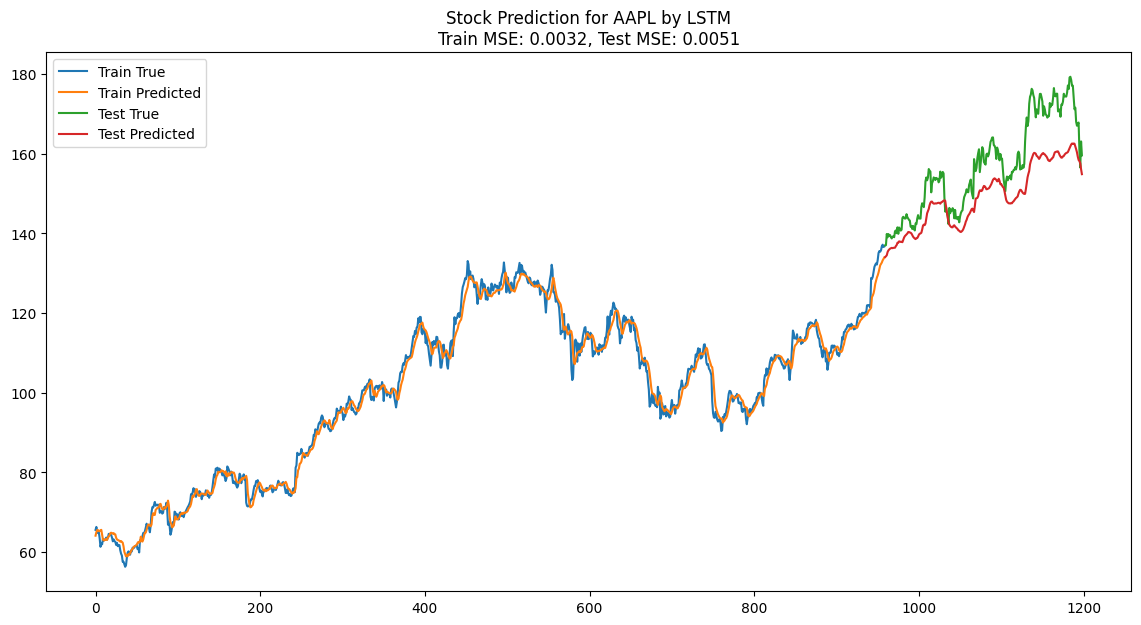
使用合并后的训练集训练LSTM模型。在训练过程中，模型可以学习到多只股票的多维度价格变化模式，并通过模式共享提高预测性能。评估方法和之前提过的模型相同。

#### ****4.3可视化和分析****

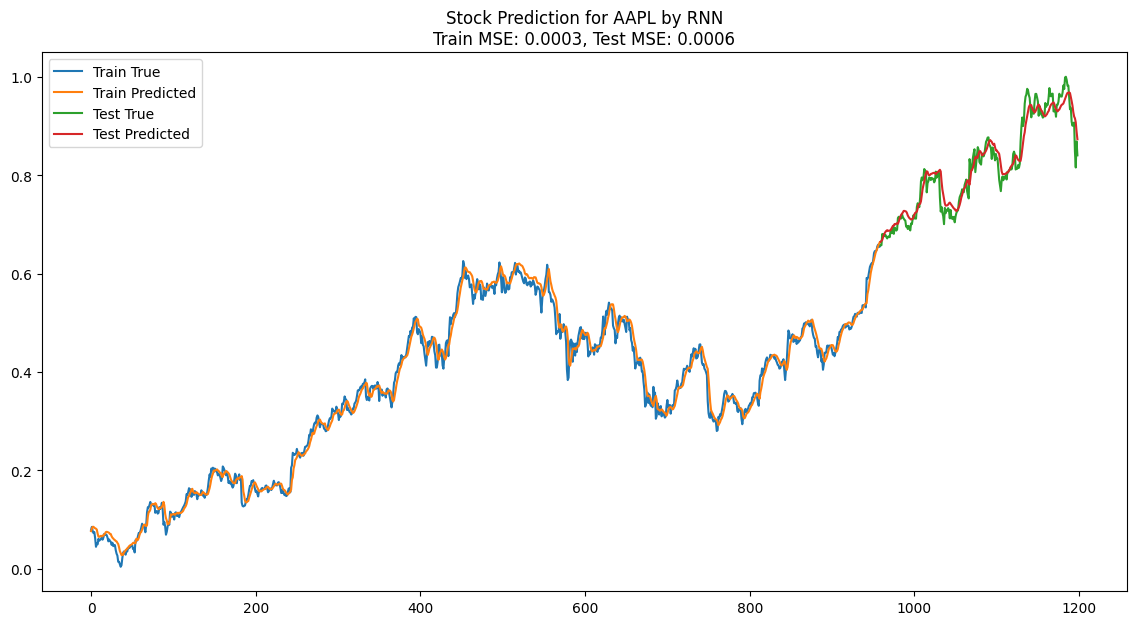
绘制训练集和测试集的真实值和预测值，以直观地展示模型的预测性能，接着为单只股票评估模型并绘制结果。

以AAPL股为例（另外三只股票的曲线已在源代码文件中给出）:

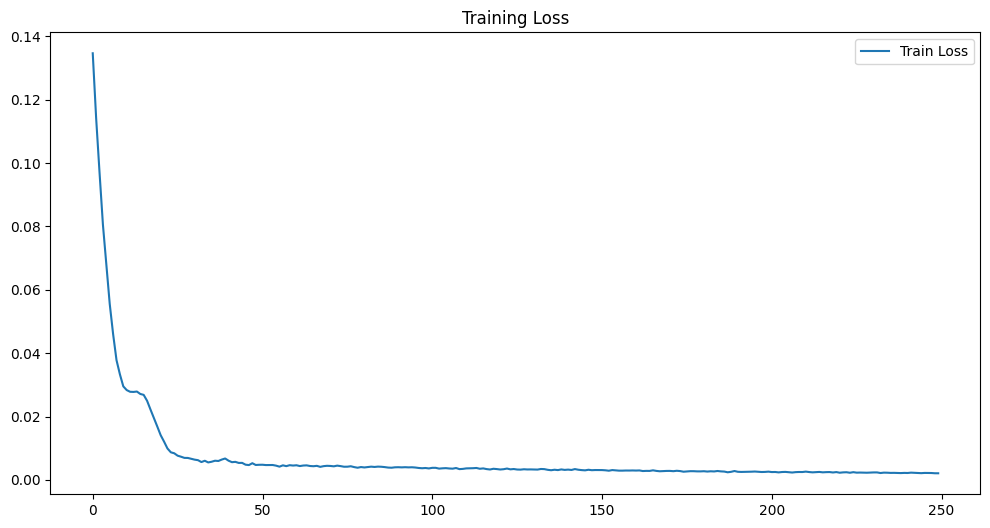
AAPL单股多维联合训练结果：



AAPL多股多维联合训练结果：



多股多维联合训练的Loss变化曲线如下：



在多维度数据参与训练中，我们首先为每只股票创建时间序列数据，并将这些数据标准化。这里，数据维度包括open、high、low、close和volume。通过为每只股票单独创建序列，然后将所有序列组合起来，可以使模型同时学习多维度的历史价格模式。

通过多维度数据参与训练方法，可以有效地提高模型的预测性能和稳定性。联合多只股票的数据提供了更丰富的训练集，使得模型能够更好地学习和泛化股票价格变化模式。

### ５. ****对LSTM股票预测实验的分析总结****

在本次分析实验中，我们针对一个简单的LSTM网络设计了四种不同的训练方法，并进行了针对性的对比实验，我们首先建立对单个股票单个维度的LSTM模型训练方法，之后我们将多只股票（4只）引入丰富数据集，最后我们将维度增大到多维度数据（开盘价、最高价、最低价、收盘价、成交量），获得的主要实验结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 基本收敛epochs（AAPL股） | Train\_mse（AAPL股） | Test\_mse（AAPL股） |
| 单只股单维度数据LSTM模型训练 | 大约260轮 | 0.0024 | 0.0067 |
| 多只股单维度数据LSTM模型训练 | 大约250轮 | 0.0004 | 0.0008 |
| 单只股多维度数据LSTM模型训练 | 大约170轮 | 0.0032 | 0.0051 |
| 多只股多维度数据LSTM模型训练 | 大约380轮 | **0.0003** | **0.0005** |

我们可以得出，虽然多只股票增加了数据量，但是，我们可以训练出拟合能力更强的LSTM模型，而且收敛的时间也没有很大差异，数据在数据量和特征上的丰富对模型学习丰富的范式和预测准确的收盘价有很大作用。

## 三、对比分析

### 1. 实验结果

我们分别取RNN模型和LSTM单股单维度数据和多股多维度数据的训练表现对比：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 基本收敛epochs（AAPL股） | Train\_mse（AAPL股） | Test\_mse（AAPL股） |
| 单只股单维度数据RNN模型训练 | 大约90轮 | 0.0006 | 0.0081 |
| 单只股单维度数据LSTM模型训练 | 大约260轮 | 0.0024 | 0.0067 |
| 多只股多维度数据RNN模型训练 | 大约150轮 | 0.0003 | 0.0007 |
| 多只股多维度数据LSTM模型训练 | 大约380轮 | **0.0003** | **0.0005** |

a.单只股单维度数据训练：RNN模型在90轮左右收敛，Train MSE为0.0006，Test MSE为0.0081,LSTM模型在260轮左右收敛，Train MSE为0.0024，Test MSE为0.0067。

b.多只股多维度数据训练：RNN模型在130轮左右收敛，Train MSE为0.0003，Test MSE为0.0007,LSTM模型在380轮左右收敛，Train MSE为0.0003，Test MSE为0.0005。

### 2. 优势劣势分析

在收敛速度上，RNN模型在单只股单维度和多只股多维度数据上都比LSTM模型收敛得更快。特别是在单只股单维度数据上，RNN模型收敛时间仅为LSTM模型的三分之一左右，而在多只股多维度数据上，RNN模型收敛时间也是LSTM模型的一半左右。

在测试测试误差上，单只股单维度数据上，LSTM模型的测试误差比RNN模型低，说明LSTM在测试集上的表现更好。多只股多维度数据上，LSTM模型的测试误差也比RNN模型低，显示出更好的泛化能力。

### 3. 总结

#### ****3.1**** ****RNN（Recurrent Neural Network）****

优势：

首先，RNN的结构相对简单，易于理解和实现。基本的RNN由一个循环神经元组成，这些神经元将前一时间步的输出作为当前时间步的输入。

其次RNN在时间步之间共享参数，这使得模型在处理不同长度的序列时具有一定的泛化能力。

最后，RNN能够有效地捕捉和利用短期依赖关系，对于一些不需要长时间记忆的任务表现良好。

劣势：

在长序列训练过程中，RNN容易出现梯度消失和梯度爆炸问题。这使得模型难以捕捉长时间步之间的依赖关系。此外RNN在处理长时间步依赖时表现不佳，因为它无法有效地记住很久以前的信息。在我们的实验中，RNN容易陷入过拟合，而且数据量少时，训练结果准确度不佳。

#### ****3.2**** LSTM（Long Short-Term Memory）

优势：

首先，LSTM通过引入输入门、遗忘门和输出门，能够有效地缓解梯度消失问题，使得模型在长序列训练过程中能够保留较长时间步的信息。

其次，LSTM的设计使其能够更好地捕捉和利用长时间步之间的依赖关系。这对于需要记住长时间信息的任务（如语言建模、时间序列预测等）非常有用。

最后，LSTM的门控机制使得它能够根据需要选择性地记住或忘记信息，增强了模型的灵活性和表达能力。

劣势：

与基本的RNN相比，LSTM的结构更加复杂，计算量也更大。这可能导致训练时间更长，模型调试更加困难。LSTM的复杂性使得它在训练和推理过程中消耗更多的计算资源，对于大规模数据集或实时应用可能是一个挑战。

#### ****3.3**** 对比

RNN适用于处理短期依赖关系和简单序列任务，由于其结构简单和参数共享的特性，具有较高的计算效率。

LSTM在处理长期依赖关系和复杂序列任务时表现优异，能够有效地记住和利用长时间步之间的信息，虽然计算复杂性较高，但其在解决梯度消失问题上的优势使得它在很多实际应用中得到广泛使用。