

神经网络与深度学习

课程设计报告

适用专业：智能科学与技术

姓名：***

学号：***

班级：*****

**大学
计算机学院
2024 年 5 月

目 录

一、课程设计目的	1
二、课程设计基本要求	1
三、课程设计内容及基本步骤	1
(一) 设计课题.....	1
(二) 基本步骤.....	2
四、课程设计报告内容	3
(一) 设计任务.....	3
1. 选题 1 分析	3
2. 选题 2 分析	4
(二) 具体处理过程.....	5
1. 选题 1.....	5
2. 选题 2.....	15
(三) 在调试中的方法技巧，遇到的问题、原因及解决方法	22
1. 调试技巧	22
2. 遇到的问题	23
3. 解决方案	24
五、结束语	26
(一) 总结设计的特点和方案的优缺点	26
1. 总体设计特点	26
2. 方案优缺点	26
(二) 设计收获体会.....	27
六、课程设计考核方法	28
参考文献.....	29

一、课程设计目的

《神经网络与深度学习》是计算机专业高年级本科生的选修课程之一，是相关课程的一个重要的实践教学环节，它与理论教学和实验教学相结合，培养学生综合运用所学的基础理论和掌握的基本技能来解决实际问题的能力。

课程设计通过完成一个课题的理论设计和实际调试工作，即能加深对所学知识的理解，又能培养综合的实践技能，从而提高分析问题和解决问题的能力。训练学生综合运用学过的深度学习各类网络基本框架，了解模型运行的基本流程。通过实践教学引导学生在理论指导下有所创新，为日后工程实践和未来继续深造奠定基础。

二、课程设计基本要求

1. 根据设计的任务和指标，初步设计实验方案。
2. 通过查阅资料，确定实验方案，绘制流程图（或伪代码）。
3. 选择模型框架，利用 Python 编写代码进行测试，并通过调试改进方案。
4. 分析设计结果，撰写课程设计报告。

三、课程设计及基本步骤

（一）设计课题

选题 1（选项 2）：采用深度卷积神经网络（VGG）在 Fashion-MNIST 数据集上实施训练，并对其 epoch 与收敛性间的关系进行分析。

选题 2（选项 4）：采用长短时记忆网络（LSTM）对股票数据集中的 label 变量进行预测，并分析训练/测试比例对预测结果的影响。

（二）基本步骤

I. 初步分析问题。

首先对问题进行深入的理解，确定其主要性能指标。这可能涉及到对问题的深入研究，以便了解其核心需求和预期的结果。例如，如果问题是关于开发一个新的手机应用程序，那么主要性能指标可能包括用户界面的友好性、应用程序的稳定性和效率等。下面的步骤分析仍使用开发一个新的手机应用程序的例子，以便更好地进行阐述。

II. 需求分析。

第二个步骤是明确项目的具体需求。这可能涉及到与项目的利益相关者进行讨论，以了解他们的具体需求和期望，包括确定应用程序需要哪些功能，以及这些功能应该如何工作。

III. 总体设计。

第三个步骤是对整个项目进行高层次的设计。这涉及到确定项目的架构，以及各个部分如何协同工作，包括确定应用程序的用户界面应该如何设计，以及后端服务应该如何实现。

IV. 详细设计。

第四个步骤是对每个部分进行详细的设计。这涉及到编写代码，或者创建详细的设计文档，包括编写实现每个功能的代码，以及创建详细的设计文档来描述这些功能的工作原理。

V. 调试并分析结果。

第五个步骤是对项目进行测试，以确保它能够正确地工作。此外，还需要分析测试结果，以确定项目是否满足需求，以及是否存在任何需要改进的地方。

四、课程设计报告内容

（一）设计任务

1. 选题 1 分析

（1）设计任务简介&主要性能指标

选题 1 的主题是采用深度卷积神经网络（VGG）在 Fashion-MNIST 数据集上实施训练，并对其 epoch 与收敛性间的关系进行分析。

主要设计任务如下：

- 搭建 Pytorch 环境。
- 构建 VGG 网络。
- 下载并处理 Fashion-MNIST 数据集。
- 使用 VGG 对 Fashion-MNIST 数据集进行训练。
- 分析 epoch 与收敛性间的关系：
 - 绘制 epoch 与准确率的关系曲线。

通过绘制 epoch 与准确率的关系图，可以直观地看到模型随着训练轮数的增加，其准确率的变化趋势。如果模型在训练集上的准确率持续提高，而在验证集上的准确率开始下降，这可能表明模型出现了过拟合。相反，如果模型的准确率在训练集上提高缓慢或不再提高，可能意味着模型存在欠拟合的风险。

- 绘制 epoch 与损失值的关系曲线。

损失值是衡量模型预测值与真实值之间差异的指标。通过绘制 epoch 与损失值的关系图，可以观察到模型在训练过程中损失值的变化情况，从而评估模型的学习效果。理想情况下，随着 epoch 的增加，训练损失和验证损失都应该逐渐减小。如果训练损失持续减小而验证损失开始增大，这可能是模型过拟合的信号。反之，如果两者都减小，但减小的速度放缓，可能意味着模型正在逐渐收敛到最佳状态。通过分析损失值的变化，可以对模型的超参数进行调整，如学习率、批量大小等，以期达到更快的收敛速度和更低的损失值。

选题 1 的主要性能指标是 epoch 与准确率的关系图及与损失值的关系图。

(2) 需求分析

在深度学习领域，使用深度卷积神经网络（如 VGG）对图像数据集进行训练并分析其性能是一个常见且重要的任务。

下面我将从问题描述，要求与限制条件，设计目标三个方面展开需求分析。

I. 问题描述

本选题需要解决的主要问题是探究 VGG 网络在 Fashion-MNIST 数据集上训练时，不同数量的 epoch 如何影响模型的收敛性。这包括理解模型在不同训练阶段的表现、识别最佳 epoch 以避免过拟合或欠拟合，以及优化模型的整体性能。

II. 要求与限制条件

- ✧ 必须使用 VGG 网络结构进行实验。
- ✧ 数据集应为 Fashion-MNIST，不得更换。
- ✧ 需要对不同的 epoch 进行实验，以观察和分析模型的收敛性。
- ✧ 实验应在有限的计算资源下进行，考虑到时间和成本效率。

III. 设计目标

- ✧ 明确不同 epoch 对 VGG 网络性能的影响。
- ✧ 确定在 Fashion-MNIST 数据集上训练 VGG 网络时的最佳 epoch。
- ✧ 评估和比较不同 epoch 的模型性能。

2. 选题 2 分析

(1) 设计任务简介&主要性能指标

选题 2 的主题是采用长短时记忆网络（LSTM）对股票数据集中的 label 变量进行预测，并分析训练/测试比例对预测结果的影响。

主要设计任务如下：

- 数据准备与预处理
- LSTM 网络搭建
- 训练与调参
- 模型评估与预测

选题 2 的主要性能指标是调整训练/测试比例之后，预测结果的变化情况。

(2) 需求分析

下面我仍将从问题描述，要求与限制条件，设计目标三个方面展开需求分析。

I. 问题描述

本选题需要解决的主要问题是有效利用 LSTM 模型处理和预测时间序列数据，以及如何确定最优的训练与测试集划分比例来提高模型的预测性能。这包括理解 LSTM 模型的基本结构和工作原理，数据预处理和时间窗口划分，选择合适的技术指标和特征作为模型输入，性能评估和结果分析等问题。

II. 要求与限制条件

- ✧ 必须使用 LSTM 网络结构进行实验。
- ✧ 需要调整训练/测试比例进行实验，以观察预测结果的效果。
- ✧ 实验应在有限的计算资源下进行，考虑到时间和成本效率。

III. 设计目标

- ✧ 明确不同训练/测试比例对预测结果的影响。
- ✧ 提高预测准确性，优化超参数。
- ✧ 提升模型泛化能力，减少过拟合风险。
- ✧ 对预测的结果进行深入分析。

(二) 具体处理过程

1. 选题 1

(1) 所采用模型框架简介 (VGG-11)

VGG 网络是由牛津大学的 Visual Geometry Group (VGG) 于 2014 年发布。

VGG 网络诞生于深度学习快速发展的时期，它是为了参加 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 而设计的。在这个挑战赛中，VGG 取得了非常优异的成绩，在分类任务中获得了第二名，在定位任务中获得了第一名。

VGG 网络以其简洁的结构著称。该网络主要使用 3x3 的小卷积核，通过增加

网络的深度来提高性能。相较于之前的网络，如 AlexNet，VGG 避免了使用大尺寸的卷积核，而是通过串联多个 3x3 卷积层来实现功能复杂且参数更少的卷积操作。

VGG 的成功推动了深度学习领域的研究，尤其是在迁移学习方面的应用。由于其出色的图像特征提取能力，VGG 很快成为了众多计算机视觉任务的首选模型之一。随着时间的推移，尽管出现了更多先进的模型，VGG 仍然因其简单性和有效性而被广泛研究和使用的。其凭借其深度结构和小卷积核设计，在图像识别和分类领域展现了卓越性能。

不同深度的 VGG 网络，全连接层的数量相同，不同的是卷积层的数量。例如，VGG-16 包含 13 个卷积层和 3 个全连接层，VGG-19 则包含 16 个卷积层和 3 个全连接层。VGG-19 比 VGG-16 多了三个卷积层，两者并没有本质上的区别，只是网络深度不一样。

表 1 VGG 系列网络层数

VGG 版本	CNN 层	FC 层
VGG-11	8	3
VGG-13	10	3
VGG-16	13	3
VGG-19	16	3

本次选题采用的网路结构是 VGG-11。

VGG-11 网络结构图如下：

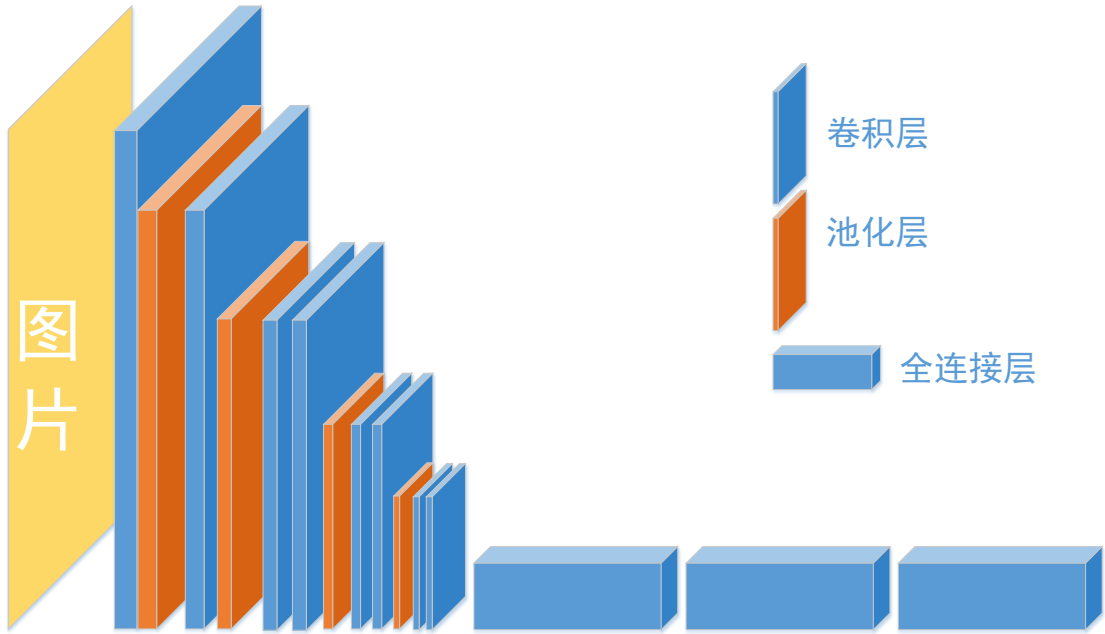


图 1 VGG-11 的网络结构图

本次选题采用 VGG-11，而不选择 VGG-16, VGG-19 等更深网络层结构的原因主要包括如下 3 点：

I. 计算资源有限。

更多的层数和参数意味着 VGG-16 需要更多的计算资源和存储空间，而我的电脑性能有限，需要借助云服务器进行模型训练。

II. VGG-11 的性能已经可以满足训练 Fashion-MNIST 的需要。

VGG-11 参数量较少，它在小数据集上可能表现得更好，可以降低过拟合的可能性。虽然 VGG-11 的网络层次没有 VGG-16, VGG-19 深，但是 VGG-11 在图像分类领域表现出来的性能优异，完全可以满足本课题中训练 Fashion-MNIST 数据集的需求。

III. 应用场景不同。

对于中小型数据集或者计算资源有限的场景，VGG-11 可能是一个更实用的选择，比如本选题。而 VGG-16 更适用于大规模数据集和复杂的图像识别任务。为充分发挥模型优势，综合考虑硬件设备性能，故使用 VGG-11 网络进行本次选题的训练。

(2) 总体设计

本次实验使用 PyTorch 框架。而要使用 VGG-11 网络对 Fashion-MNIST 数据集进行训练，必须进行下列步骤：



图 2 选题 1 总体设计流程图

(3) 详细设计

各个模块间的调用信息如下图所示：

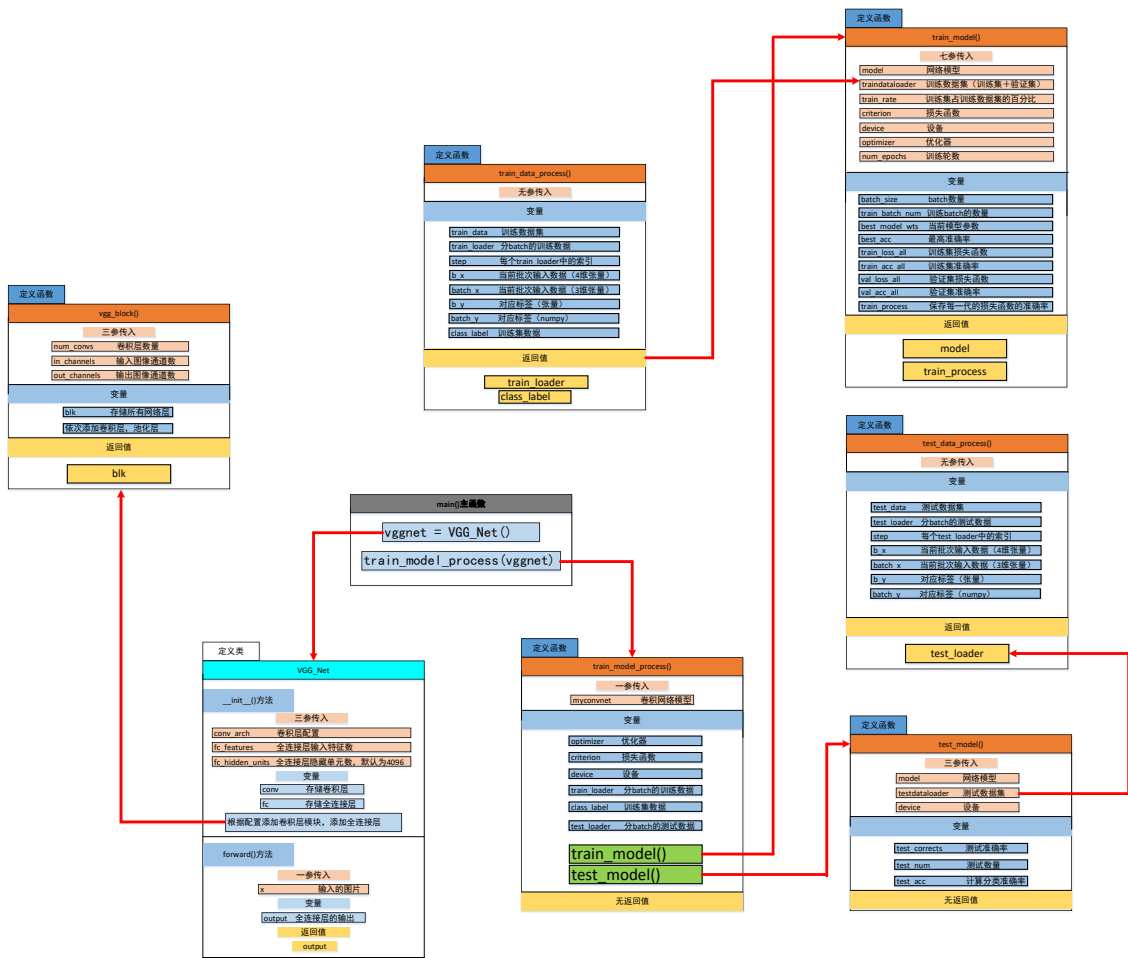


图3 选题1 各个模块间的调用信息

由于印刷清晰度有限, 现将各个模块详细设计信息展示如下:

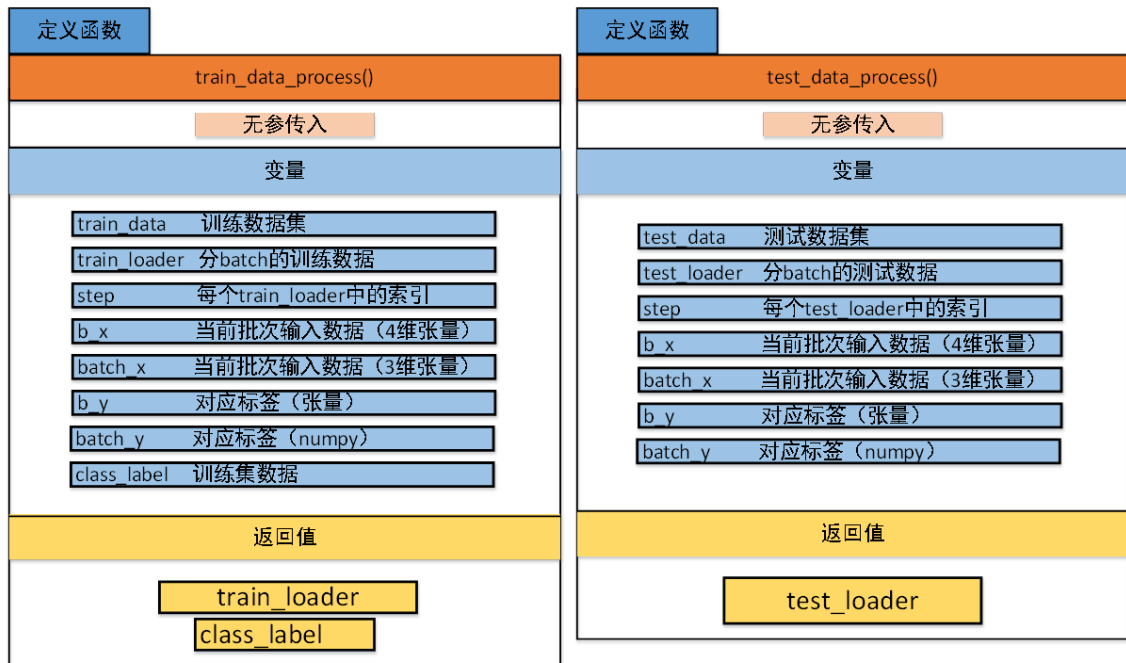


图4 模块详细设计信息 1

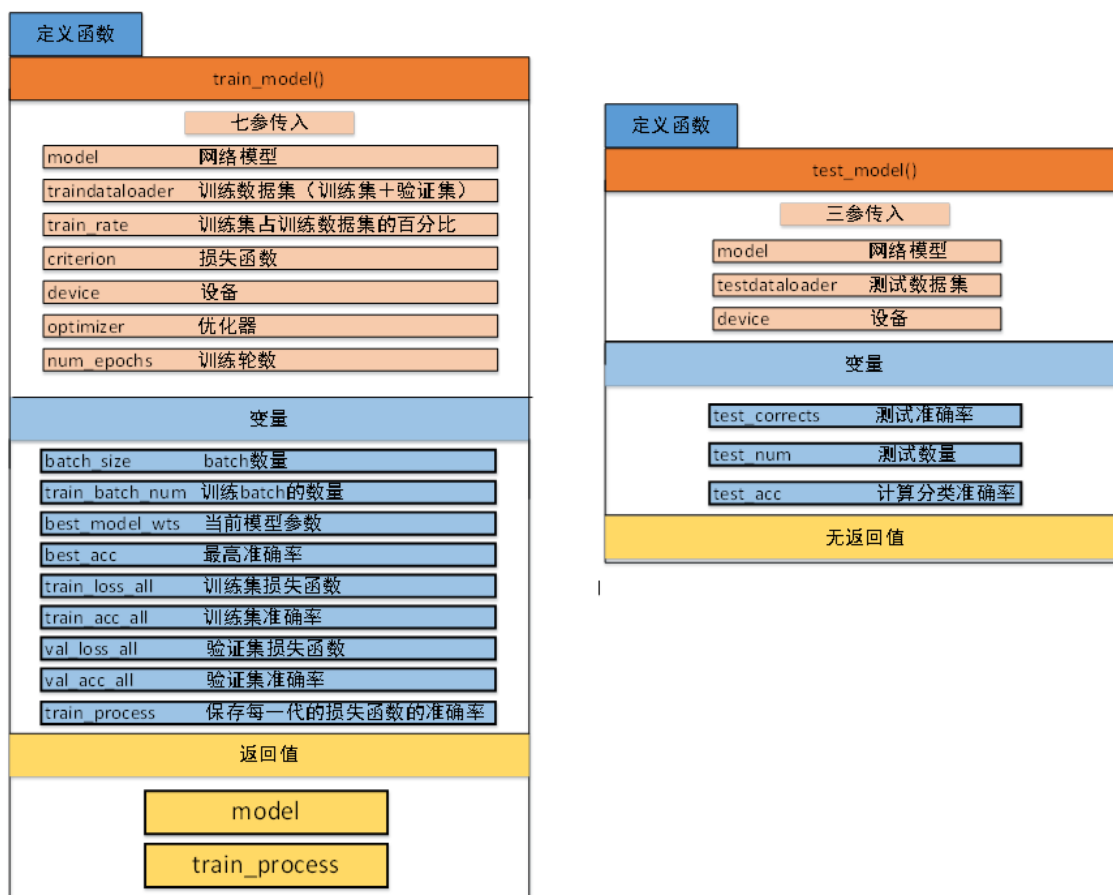


图 5 模块详细设计信息 2

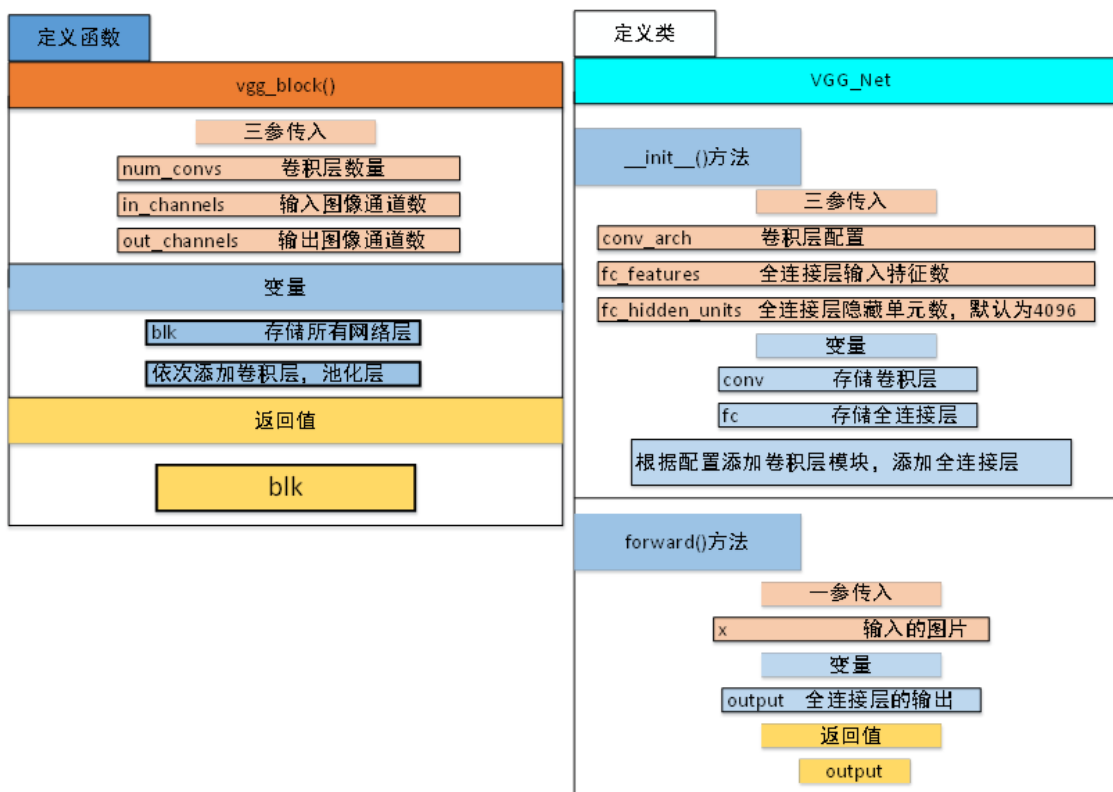


图 6 模块详细设计信息 3

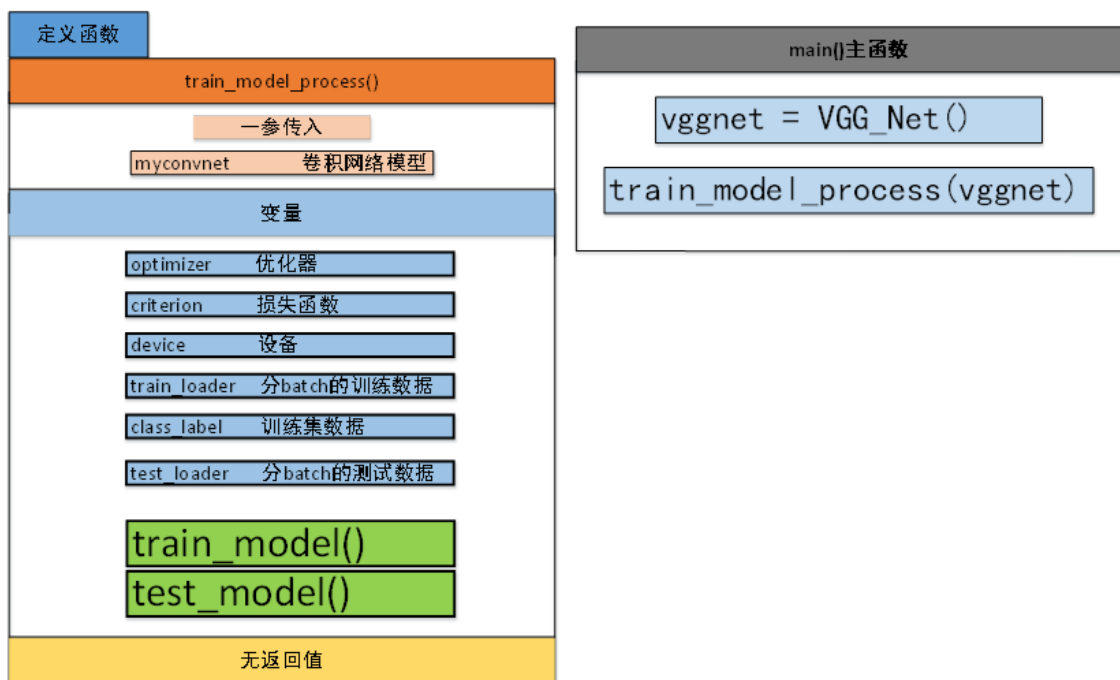


图 7 模块详细设计信息 4

(4) 程序设计结果

Loss 与 epoch 的关系如下图:

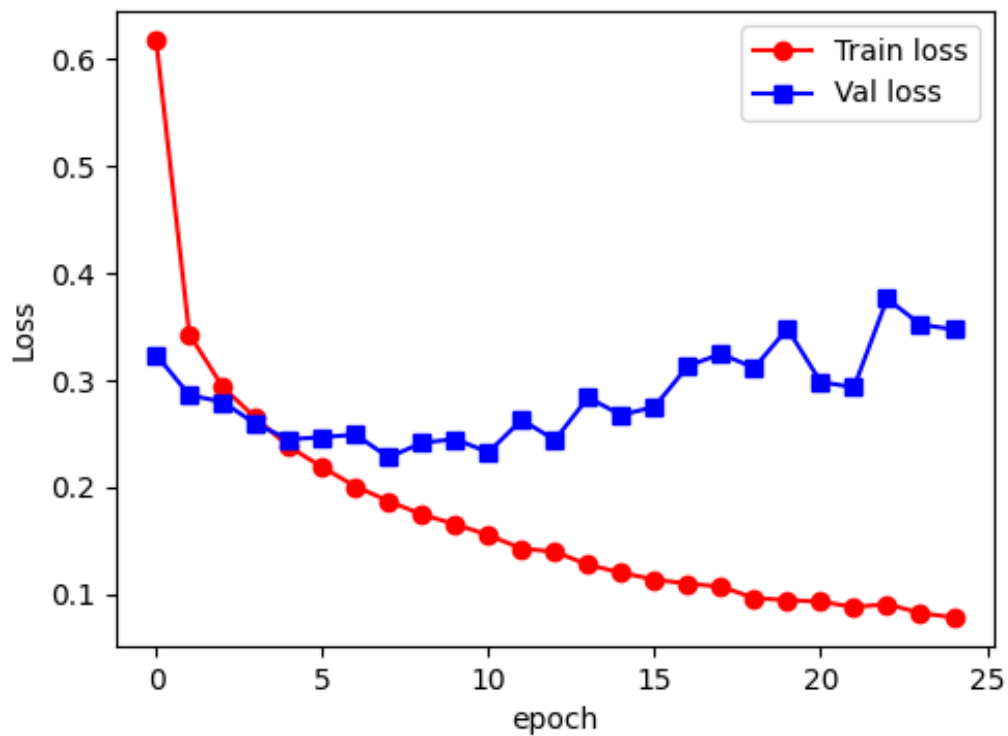


图 8 选题 1 中 Loss 与 epoch (epoch=25) 的关系结果图

acc 与 epoch 的关系如下图:

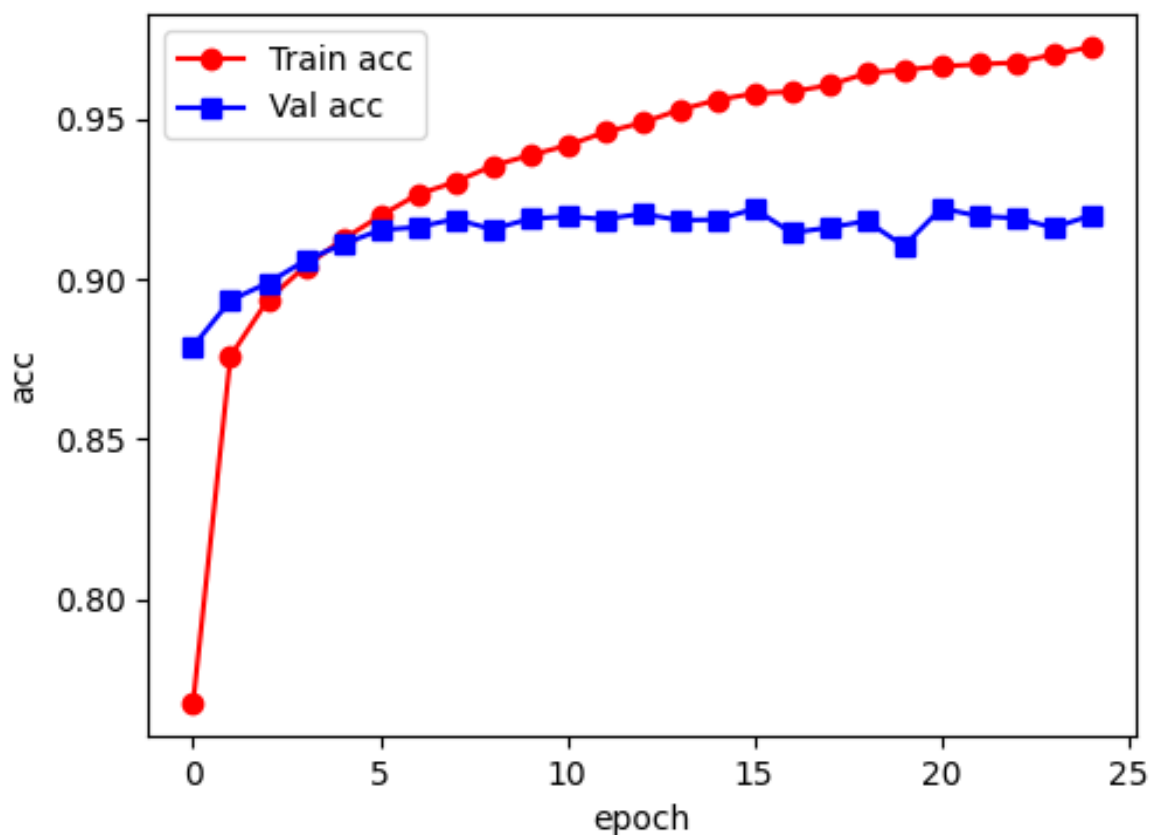


图9 选题1中 acc 与 epoch (epoch=25) 的关系结果图

这两张图表显示了训练损失 (Train loss) 和训练准确度 (Train acc), 以及验证损失 (Val loss) 和验证准确度 (Val acc) 随训练周期 (epoch) 的变化。

以下是对图表结果的分析:

训练损失从 0.6 以上开始逐渐减少至 0.1 以下, 这表明模型在训练数据上的性能随着训练的进行而不断提升。**训练准确度**逐渐增加至 0.90, 这表明模型在训练数据上的预测越来越准确。**验证损失**从 0.3 开始, 逐渐减少随后便曲折变化, 这表明模型在验证数据上的性能先提升随后在不断变化。**验证准确度**从 0.87 开始, 逐渐增加至 0.93, 这表明模型在验证数据上的预测也越来越准确。

从图表中可以看出, 模型的性能在训练周期的早期阶段迅速提升, 并在大约第 10 个 epoch 左右达到一个峰值, 此时训练和验证的准确度都达到了 0.90。

这张图表中训练和验证的损失和准确度都显示出相似的趋势, 没有出现明显的过拟合现象。这表明模型在训练数据和验证数据上都保持了较好的性能。图表覆

盖了从第 0 到第 25 个训练周期，但关键的性能提升主要发生在前 10 个周期内。

综上，这张图表显示了模型在训练和验证数据上的性能随着训练周期的增加而不断提升，没有明显的过拟合现象。模型在大约第 10 个 epoch 时达到了最佳性能。

在设置 epoch 为 25 之后，我又改变 epoch 的数值为 10,15, 45, 70 重新进行了训练，结果如图所示：

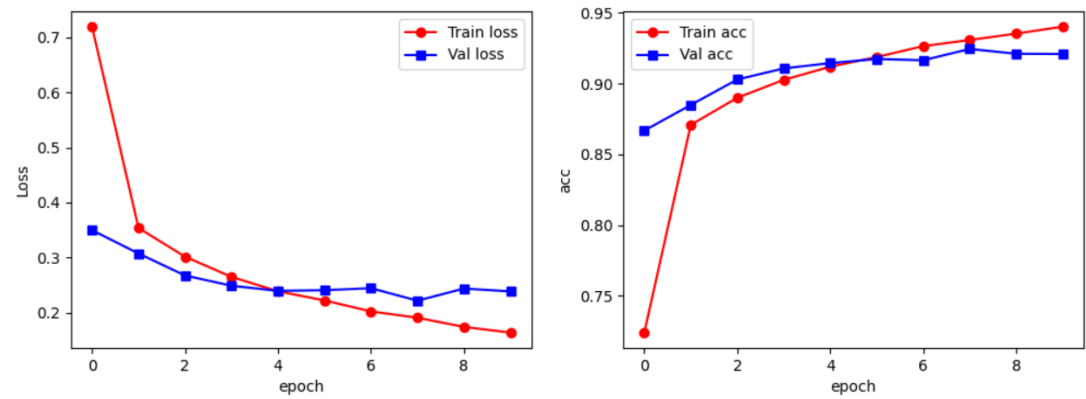


图 10 选题 1 中 acc、Loss 与 epoch (epoch=10) 的关系结果图

由图 10 可得，训练/测试损失值，准确度的平缓趋势均不明显。

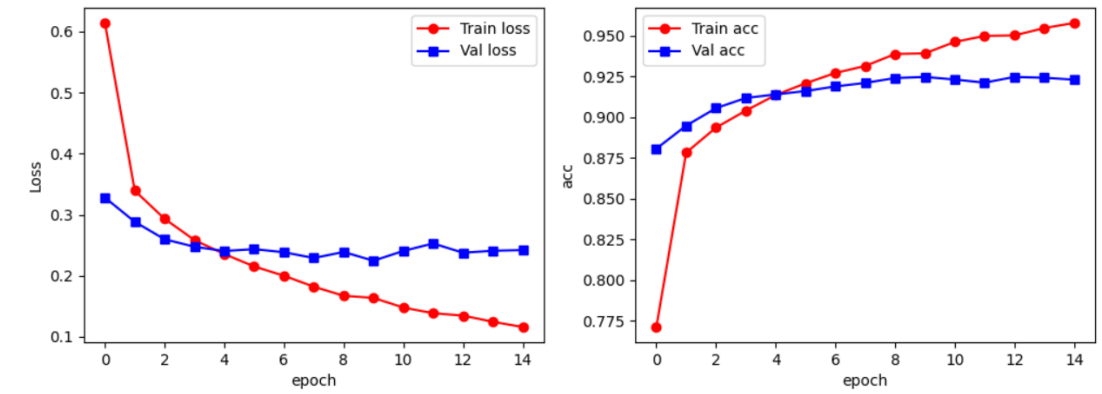


图 11 选题 1 中 acc、Loss 与 epoch (epoch=15) 的关系结果图

由图 11 可得，训练/测试损失值，准确度的平缓趋势开始显现。

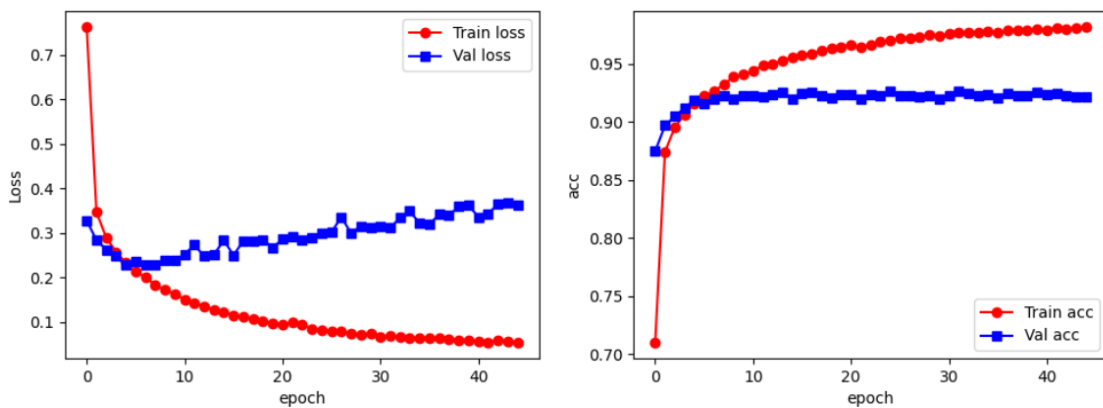


图 12 选题 1 中 acc、Loss 与 epoch (epoch=45) 的关系结果图

由图 12 可得，训练/测试损失值，准确度随 epoch 变化的曲线正在趋于稳定。

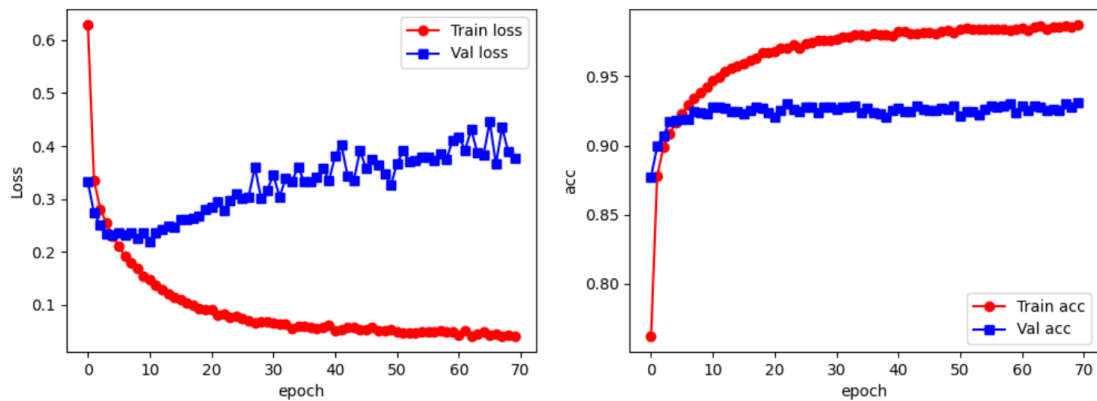


图 13 选题 1 中 acc、Loss 与 epoch (epoch=70) 的关系结果图

由图 13 可得，模型在 epoch 为 10 之后的测试损失值在曲线上升，故在 **epoch=10** 时已达到模型最佳性能。

这也可以看出 **epoch 与收敛性之间的关系**：在一定范围内，epoch 越大，模型收敛越快；超出这个范围，epoch 越大，模型收敛的速度变化不明显。

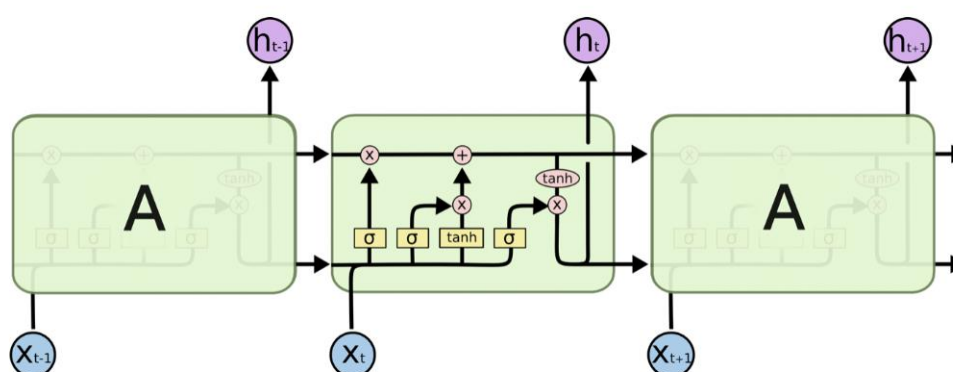
epoch 与收敛性之间的关系不是一成不变的，而是受到多种因素的影响。在往后的实践中，可以通过逐步增加 epoch 的方法来观察模型的性能变化，同时结合验证集的表现来选择最佳的停止点，以实现模型性能和资源使用的最优平衡。

2. 选题 2

(1) 所采用模型框架简介

LSTM 网络（Long Short-Term Memory networks）于 1997 年由 Sepp Hochreiter 和 Jürgen Schmidhuber 发布。这种特殊的循环神经网络（RNN）旨在解决标准 RNN 在处理长序列数据时遭遇的梯度消失问题。

LSTM 的设计主要是为了克服传统 RNN 在处理长序列时的主要缺陷，即梯度消失问题。通过引入“记忆单元”和“门控机制”，LSTM 能够在长序列中保持信息的流动，从而有效地捕捉和理解长序列中的复杂依赖关系。



The repeating module in an LSTM contains four interacting layers.

图 14 LSTM 中的重复模块四个相互作用的层

自从 LSTM 被提出以来，它的结构和算法经历了多次优化和变种，如添加窥视孔连接、耦合忘记机制等，这些都旨在进一步提升其性能和效率。同时，随着计算能力的提升和数据量的增加，LSTM 被广泛应用于语音识别、机器翻译、时间序列分析等领域，并取得了显著的成果。

LSTM 自 1997 年被提出以来，凭借其独特的结构和门控机制，在解决长期依赖问题上取得了重大突破。它的提出不仅解决了 RNN 的核心问题，也开启了许多先前无法解决的复杂序列学习任务的新篇章。

(2) 总体设计

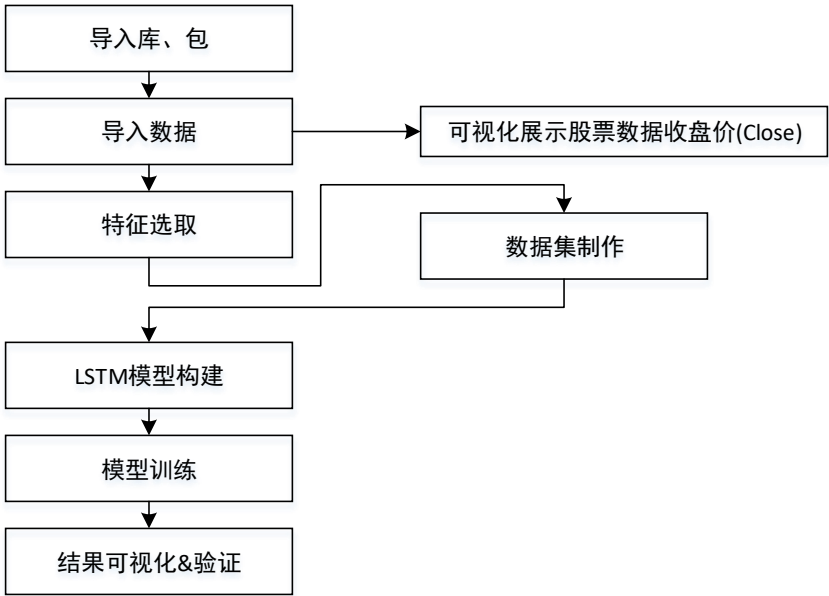


图 15 选题 2 总体设计流程图

(3) 详细设计

初始实验的训练/测试比例为 8:2。

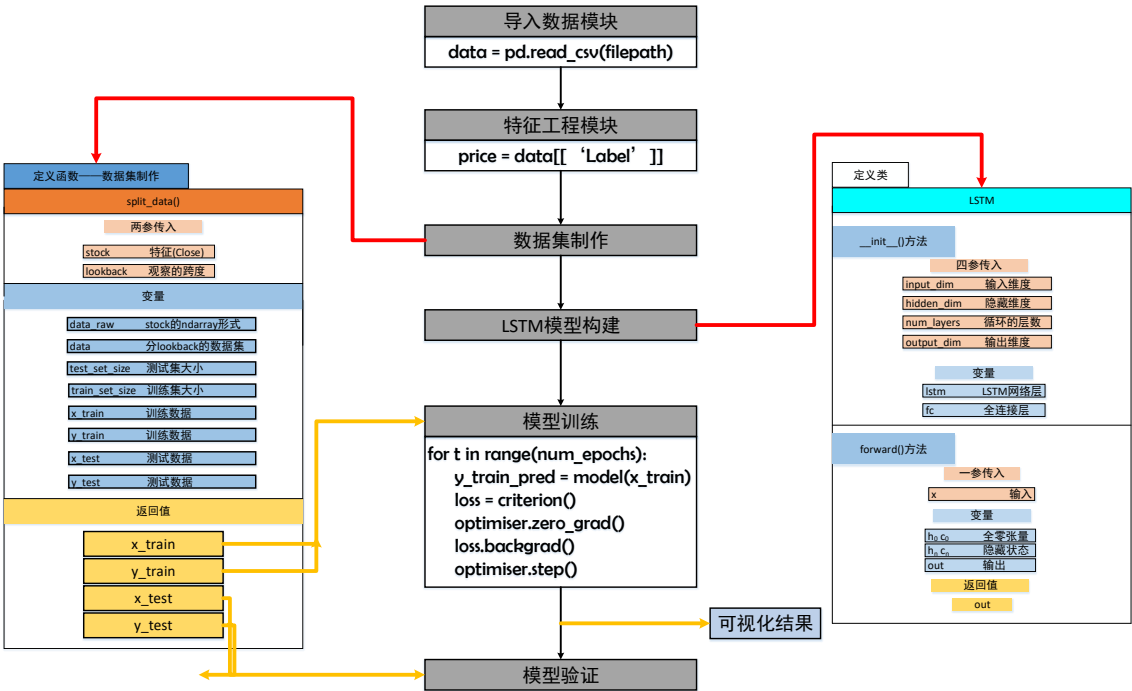


图 16 选题 2 各个模块间的调用信息

由于印刷清晰度有限，现将各个模块详细设计信息展示如下：

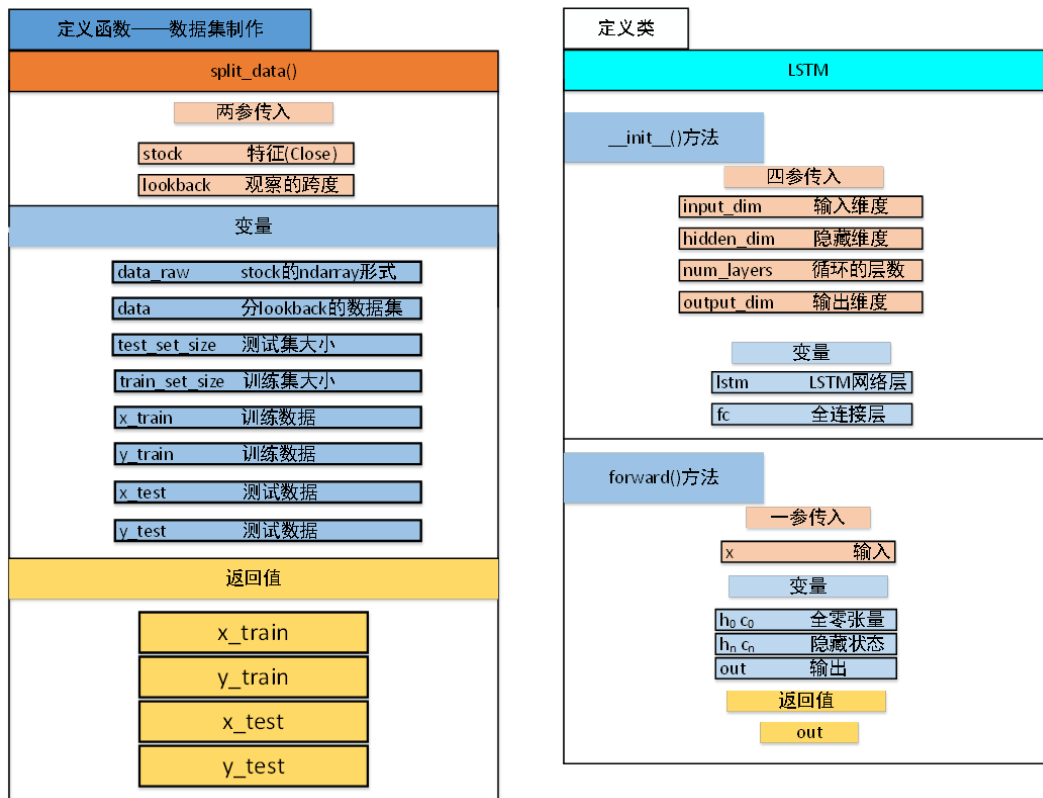


图 17 模块详细设计信息

(4) 程序设计结果



图 18 选题 2 运行结果图 1

```
main x
Epoch 82 MSE: 0.0005904278950765729
Epoch 83 MSE: 0.0005015587667003274
Epoch 84 MSE: 0.0005190395168028772
Epoch 85 MSE: 0.0005284988437779248
Epoch 86 MSE: 0.000561375287361443
Epoch 87 MSE: 0.0006022729794494808
Epoch 88 MSE: 0.00057318335166201
Epoch 89 MSE: 0.0005612420500256121
Epoch 90 MSE: 0.0005041632684879005
Epoch 91 MSE: 0.00047375785652548075
Epoch 92 MSE: 0.00046316225780174136
Epoch 93 MSE: 0.0004537075001280755
Epoch 94 MSE: 0.0004764854966197163
Epoch 95 MSE: 0.0004799015005119145
Epoch 96 MSE: 0.00048423215048387647
Epoch 97 MSE: 0.00048219208838418126
Epoch 98 MSE: 0.00046030920930206776
Epoch 99 MSE: 0.00044951957534067333
Training time: 9.222731590270996
predict data:
```

图 19 选题 2 运行结果图 2

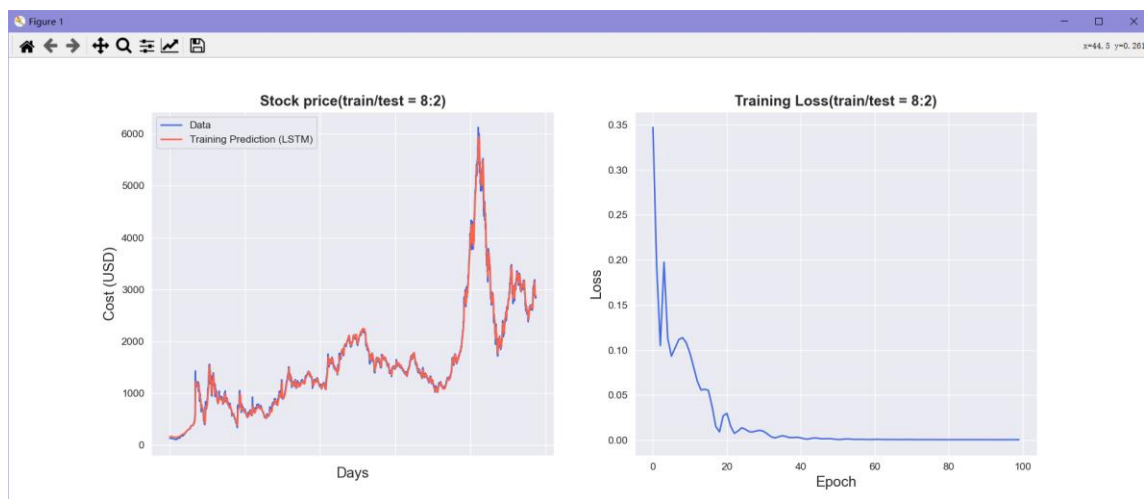


图 20 训练预测曲线拟合图及训练损失变化图 (train/test = 8:2)

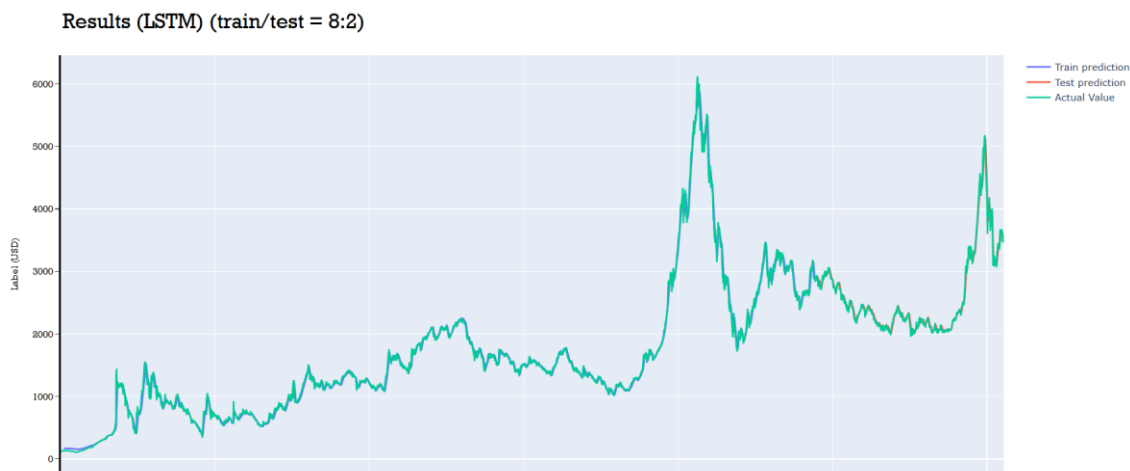


图 21 训练预测曲线及测试预测拟合图 (train/test = 8:2)

图 20 左图显示了两条曲线，蓝色曲线是随天数的收盘价的变化，红色曲线是随天数的预测值的变化，右图显示了 100 个 epoch 的训练损失折线。

从右图可以看出，训练损失随着训练的进行而逐渐减小，这表明模型在训练过程中逐渐学习并改善其预测能力。曲线大致在 **epoch 为 20** 时趋于稳定。

接下来改变训练/测试的比例，将训练/测试比为 8:2 分别改为 6:4 和 9:1 进行进一步的对比实验。

具体改变参数：

```
data = np.array(data)
test_set_size = int(np.round(0.2 * data.shape[0])) # 将20%比例的数据作为测试集
train_set_size = data.shape[0] - (test_set_size)

data = np.array(data)
test_set_size = int(np.round(0.4 * data.shape[0])) # 将40%比例的数据作为测试集
train_set_size = data.shape[0] - (test_set_size)
```

图 22 改变参数示意图

训练/测试比为 6:4 运行结果:



图 23 训练预测曲线拟合图及训练损失变化图 (train/test = 6:4)



图 24 训练预测曲线及测试预测拟合图 (train/test = 6:4)

图 23 中左右两图都发生了明显变化。左图中训练预测数据与原始数据的拟合程度明显降低，而右图中，当 epoch 为 60 时，训练损失才趋于平缓。

由图 24 可以看出，该模型在测试集上的表现极差，较测试集为 20%的比例时所表现出来的性能相差较远。

训练/测试比为 9:1 运行结果：

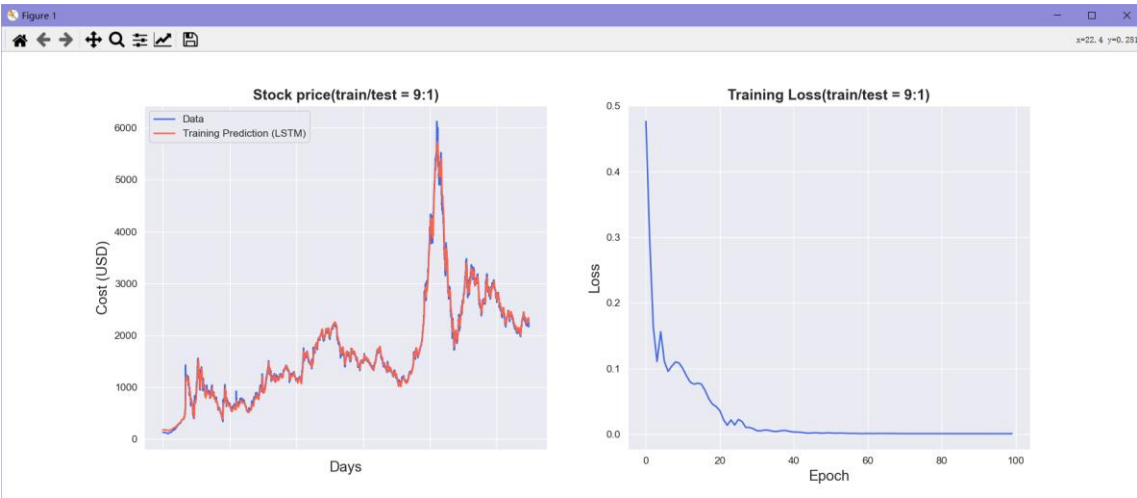


图 25 训练预测曲线拟合图及训练损失变化图（train/test = 9:1）

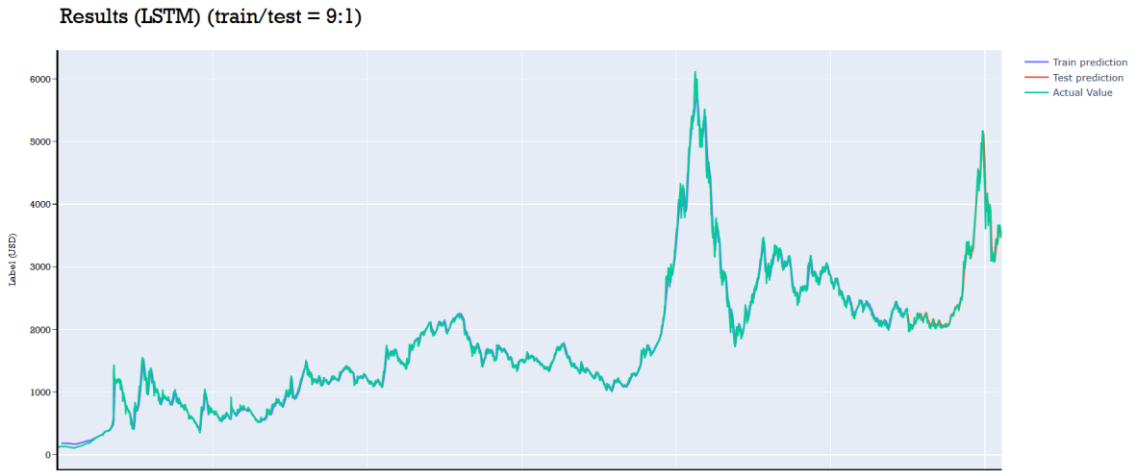


图 26 训练预测曲线及测试预测拟合图（train/test = 9:1）

而将训练/测试比调至 9:1 后，可以发现模型的拟合程度再一次变好，从图 26 也可以看出其在测试集上的准确度也明显变高。

由上述实验可以得出**训练/测试比例对预测结果的影响：**

训练/测试比例过大或过小都会对预测结果造成一定的影响。首先训练数据比例过小可能会导致欠拟合的问题，导致模型训练得不够，因此其在测试和预测应用中的表现肯定会欠佳。而训练数据比例过大，可能会出现过拟合的问题。

因此，为了优化模型性能并确保模型具有良好的泛化能力，应当合理设置训练集、验证集和测试集的比例。同时，应考虑到数据集的大小、问题的复杂性以及模型类型等因素，灵活调整划分比例。在实践中，还可以采用交叉验证、网格搜

索等高级评估方法，以及正则化、数据增强等优化策略来进一步提升模型的性能。通过这些方法，可以确保模型在实际应用中的准确性和稳定性，为解决实际问题提供更大的价值。

（三）在调试中的方法技巧，遇到的问题、原因及解决方法

1. 调试技巧

技巧 1：可视化一个 batch 的图像，检验数据集是否加载成功。

```
# 可视化一个Batch的图像
plt.figure(figsize=(12, 5)) # 创建一个图形窗口，设置图形的大小为12x5英寸。
for ii in np.arange(len(batch_y)): # 遍历batch_y的长度，batch_y是一个包含类别标签的数组。
    plt.subplot(4, 16, ii + 1) # 创建一个4行16列的子图网格，并在第ii+1个子图中绘制图像。
    plt.imshow(batch_x[ii, :, :], cmap=plt.cm.gray) # 在当前子图中显示batch_x的第ii个图像，使用灰度颜色。
    plt.title(class_label[batch_y[ii]], size=9) # 为当前子图设置标题，标题为class_label数组中对应batch_y[ii]的类别名称。
    plt.axis("off") # 关闭当前子图的坐标轴。
    plt.subplots_adjust(wspace=0.05) # 调整子图之间的水平间距为0.05英寸。
# plt.show()
plt.savefig('/root/VGG-11-Project/train_fig_output/one batch figure.png')
```

图 27 可视化 batch 图像代码展示图



图 28 batch 图像图

技巧 2：在每一模块的代码编写完成之后进行步骤完成的文字打印，用于检验这一模块的代码运行是否完成。


```

main.py M • test.py M test.png U
main.py > ...
1, #采用深度卷积神经网络（VGG）在 Fashion-MNIST 数据集上实施训练，并对其 epoch 与收敛性间的关系进行分析。
2, > import numpy as np ...
15 print("第1步: 导入库包成功!\n-----")
16
17 # 处理训练集数据
18 > def train_data_process(): ...
58 print("第2步: 训练数据集准备成功!\n-----")
59
60 |
61 # 处理测试集数据
62 > def test_data_process(): ...
81 print("第3步: 测试集准备成功!\n-----")
82
83 # 定义一个VGG模块
84 > def vgg_block(num_convs, in_channels, out_channels): ...
104 print("第4步: VGG模块创建成功!\n-----")
105
106 # 定义VGG-11网络结构
107 > class VGG_Net(nn.Module): ...
133 print("第5步: VGG-11网络搭建成功!\n-----")
134
135 # 定义网络的训练过程
136 > def train_model(model, traindata_loader, train_rate, criterion, device, optimizer, num_epochs=45): ...
235 # 测试模型
236 > def test_model(model, testdata_loader, device): ...
260 # 模型的训练和测试
261 > def train_model_process(myconvnet): ...
271 print("第6步: 定义训练过程和预测过程成功!\n-----")

```

图 29 技巧 2 打印图

技巧 3：使用远程服务器进行模型训练。

-----AutoDL-----

目录说明：

目录	名称	速度	说明
/	系统盘	一般	实例关机数据不会丢失，可存放代码等。会随保存镜像一起保存。
/root/autodl-tmp	数据盘	快	实例关机数据不会丢失，可存放读写IO要求高的数据。但不会随保存镜像一起保存

CPU：15 核心
内存：80 GB
/root/miniconda3/bin/python /root/VGG-11-Project/main.py
GPU：NVIDIA GeForce RTX 4090 D, 1
存储：
系统盘/ : 5% 1.3G/30G
数据盘/root/autodl-tmp: 1% 116K/50G

*注意：
1.系统盘较小请将大的数据存放于数据盘或网盘中，重置系统时数据盘和网盘中的数据不受影响
2.清理系统盘请参考：<https://www.autodl.com/docs/qa/>

图 30 服务器连接成功示意图

2. 遇到的问题

- 问题 1：远程服务器上运行的图像结果不能传回本地 PC 显示。
- 问题 2：服务器上的文件不能直接使用本地文件库，故服务器无法使用本地文件。
- 问题 3：安装 sklearn 库时，使用 `pip install sklearn` 安装失败，如下图所示。

```
root@autodl-container-00274fbfbc-a2d54c12:~/2024 NN Curriculum Design# /root/miniconda3/bin/python "/root/2024 NN Curriculum Design/main.py"
Traceback (most recent call last):
  File "/root/2024 NN Curriculum Design/main.py", line 4, in <module>
    from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
ModuleNotFoundError: No module named 'sklearn'
root@autodl-container-00274fbfbc-a2d54c12:~/2024 NN Curriculum Design# pip install sklearn
Looking in indexes: http://mirrors.aliyun.com/pypi/simple
Collecting sklearn
  Downloading http://mirrors.aliyun.com/pypi/packages/46/1c/395a83ee7b2d2ad7a05b453872053d41449564477c81dc356f720b16eac4/sklearn-0.0.post12.tar.gz (2.6 kB)
  Preparing metadata (setup.py) ... error
error: subprocess-exited-with-error

x python setup.py egg_info did not run successfully.
  exit code: 1
  > [15 lines of output]
The 'sklearn' PyPI package is deprecated, use 'scikit-learn'
rather than 'sklearn' for pip commands.

Here is how to fix this error in the main use cases:
- use 'pip install scikit-learn' rather than 'pip install sklearn'
- replace 'sklearn' by 'scikit-learn' in your pip requirements files
  (requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
- if the 'sklearn' package is used by one of your dependencies,
  it would be great if you take some time to track which package uses
  'sklearn' instead of 'scikit-learn' and report it to their issue tracker
- as a last resort, set the environment variable
  SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

More information is available at
```

图 31 安装 sklearn 报错示意图

问题 4：在进行第二个实验，对 Close 列的数据进行缩放时，报错如下：

```
F:\Interests\Postgraduate Exam\Project\2024 NN Curriculum Design\LSTM\main.py:40: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#r
price['Close'] = scaler.fit_transform(price['Close'].values.reshape(-1, 1)) # 将Close列的数据缩放到-1,1之间
```

图 32 Close 列报错示意图

3. 解决方案

问题 1 解决方案：

使用 `plt.show()` 函数不能直接在 PC 端显示出图，故采用 `plt.savefig()` 函数先将图保存到文件夹中，之后进行查看。

如下图语句所示：

```
plt.savefig('/root/VGG-11-Project/train_fig_output/one batch figure.png')
```

图 33 用 `savefig()` 函数替代 `show()` 函数示意图

如果需要在黑色窗口中实时看到图像的生成，可以使用 `Xming` 工具。

问题 2 解决方案：

使用 `FileZilla` 工具进行文件互传，如下图所示是本地 PC 与服务器连接成功之后的页面，可以直接进行拖拽完成文件互传。

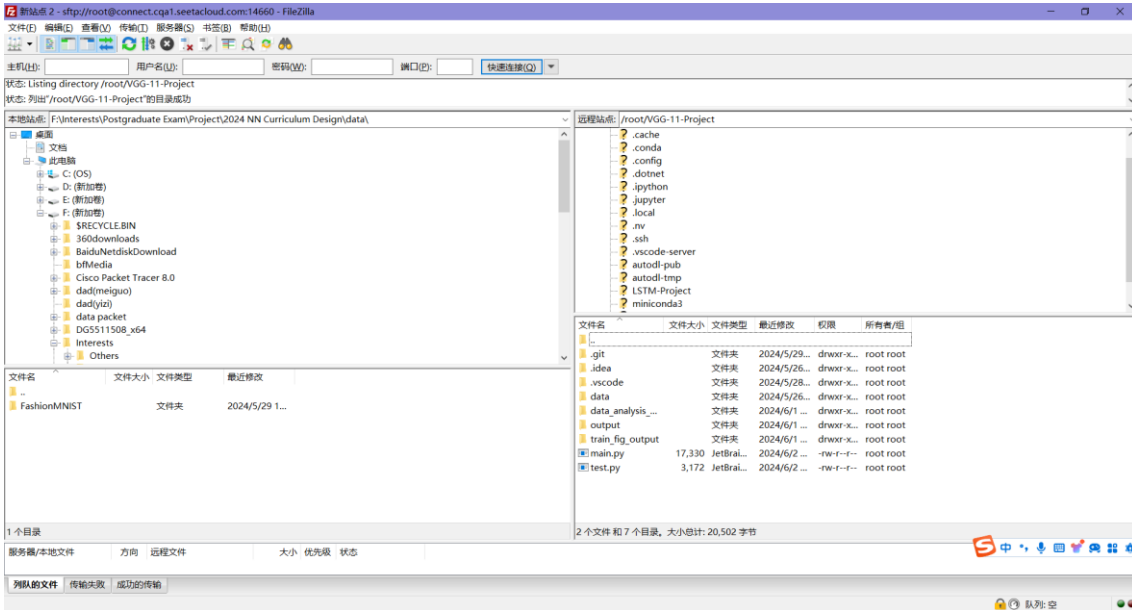


图 34 FileZilla 软件运行图

问题 3 解决方案:

查询资料可知, sklearn 库的标准名称是 scikit-learn, 使用命令 `pip install scikit-learn` 即可安装完成, 如下图所示。



图 35 成功安装 sklearn 库图

问题 4 解决方案:

查询资料可知, 报错的原因是由于 pandas 不允许在原始数据上直接修改, 需要先建立副本, 在副本上修改了之后再还原到原始数据上。改为如下图的代码后方可正确运行。

```
scaler = MinMaxScaler(feature_range=(-1, 1))
price_c = price.copy() # 先建立副本, 不直接在原始数据上进行操作
price_c['Close'] = scaler.fit_transform(price_c['Close'].values.reshape(-1, 1)) # 在副本上进行操作
price = price_c.copy() # 还原
print(price['Close'].shape)
```

图 36 解决 Close 列报错, 创建副本再赋值的代码示意图

五、结束语

（一）总结设计的特点和方案的优缺点

1. 总体设计特点

1. 流程清晰

设计图按照逻辑序列出了从数据准备到模型验证的各个步骤，使得整个工作流程清晰易懂。

2. 模块化设计

形成了模块化设计方案，便于管理和维护。

3. 数据可视化

对预测结果进行可视化展示，这有助于理解数据的基本趋势和特征。

2. 方案优缺点

优点：

逻辑清晰，步骤明了。

数据可视化使结果展现清晰。

模型灵活性高。模型可以根据需要调整参数和结构，以适应不同的预测任务。

缺点：

计算资源有限。

存在过拟合风险。

泛化能力不强。由于数据集的局限性，模型可能在更复杂或不同的图像数据集上表现不佳。

不足：

1. 模型的细节调参和调整仍有待提高。

2. X 服务工具配置失败，导致无法做出预期中的动图效果。

3. 整体实验可复制性较强，没有挖掘到独特的地方。

展望：

1. 优化器/损失函数可以多样选取进行对比。

2. LSTM/VGG 的优化算法。

3. LSTM 预测其他股票，检验可靠性。

（二）设计收获体会

实验阶段我遇到的问题以及解决办法已在四（三）节说明清楚。目前，两个实验的主要功能已经完成，包括数据集的处理，模型的构建，模型的训练，模型的测试以及可视化等等操作，能够满足选题的要求。

尽管选题的主要内容已经实现，但对于模型的细节调参和调整仍有待提高。例如模型的构建仍然可以更加复杂，选题 2 的可视化工作可以加上准确度等。

在之后的改进方案中，我会加上上面所述的几点不足。我认为在这次课设中我做的很好的一点是采用了模块化和分层的方法，尤其是在选题 1 的 VGG 实验中，这不仅提高了代码的可维护性，还便于后续的功能扩展。

在设计过程中，我们确保了所有数据处理过程符合数据保护法规，确保模型的开源性，同时将代码公开供有需要者使用。

通过在 Fashion-MNIST 数据集上应用 VGG 网络，可以深入理解深度卷积神经网络的工作原理和架构。研究 epoch 与模型收敛性之间的关系，有助于了解训练过程中模型性能的提升速度和稳定性，为确定合适的训练周期提供依据。

股票市场预测是金融领域的一个重要课题，LSTM 作为一种优秀的时间序列预测模型，可以捕捉股票价格的变化趋势。通过在股票数据集上的应用，可以加深对 LSTM 网络结构和记忆机制的理解，以及它是如何处理和预测时间序列数据的。分析不同的训练/测试数据比例对预测结果的影响，有助于找到最佳的模型评估和验证方法。通过实验可以探索不同的模型参数和训练策略，优化预测模型，提高预测的准确性和可靠性。

六、课程设计考核方法

1. 每位学生独立进行设计，在代码运行无错误后进行测试；若代码运行出现错误，分析其出现的原因。
2. 报告不允许任何形式的抄袭，各位同学尽可能用自己的话去阐述报告中的内容，如有公式采用 **Mathtype** 输入。
3. 每位同学独立上交一份课程设计报告，若出现两位同学报告高度雷同的情况，该门课程直接判定为不合格。

参考文献

- [1] Simonyan K ,Zisserman A .Very Deep Convolutional Networks for Large-Scale Image Recognition.[J].CoRR,2014,abs/1409.1556
- [2] Mpabulungi M ,Yu K ,Hong H .A CNN-LSTM model using elliptical constraints for temporally consistent sun position estimation[J].Heliyon,2024,10(10).
- [3] Li W ,Li X ,Yuan J , et al.Pressure prediction for air cyclone centrifugal classifier based on CNN-LSTM enhanced by attention mechanism[J].Chemical Engineering Research and Design,2024,205775-791.
- [4] Gao W ,Liu F ,Li C , et al.A portable terminal for acoustic monitoring and online recognition of bats with CNN-LSTM[J].Measurement Science and Technology,2024,35(7)
- [5] 冯国徽. 基于卷积神经网络 VGG 模型的小规模图像分类[D]. 兰州大学, 2018.
- 张天宝. 基于改进 VGG 网络的图像分类方法研究[D]. 安徽理工大学, 2024.
- [6] 刘勇. 改进 VGG16-LSTM 模型及其天气图像分类应用[D]. 兰州大学, 2024.
- [7] 孙瑞奇. 基于 LSTM 神经网络的美股股指价格趋势预测模型的研究[D]. 首都经济贸易大学, 2017.
- [8] 布客. PyTorch 中文翻译[EB/OL]. <https://pytorch.apachecn.org/>. 2024 (06)
- [9] 奔跑的 chanchanchan. 卷积神经网络 (VGG-11) 识别 Fashion-MNIST 数据集 (Pytorch 版) [EB/OL]. <https://blog.csdn.net/baoli8425/article/details/119977155>. 2021 (08)
- [10] SoloVersion. FileZilla 客户端的安装配置教程以及使用教程 (超级详细) [EB/OL]. <https://blog.csdn.net/SoloVersion/article/details/124579221>. 2022 (05)
- [11] AI_dataloads. 【深度学习】绘制模型训练曲线 (epoch-accuracy), 寻找最佳 epoch 值[EB/OL]. https://blog.csdn.net/AI_dataloads/article/details/133756763. 2023 (10)
- [12] 金渐层猫. VGG11、VGG13、VGG16、VGG19 网络结构图[EB/OL]. https://blog.csdn.net/weixin_43917574/article/details/113327681. 2021 (02)
- [13] 涌恒. VGG 特点总结[EB/OL]. https://blog.csdn.net/weixin_44835249/artic

- le/details/89441375#. 2019 (04)
- [14]圆 yay. VScode 的 python 使用 plt. show() 不显示图形[EB/OL]. <https://zhuanlan.zhihu.com/p/684381193>. 2024 (02)
- [15]难以触及的高度. Python 的 sklearn 库安装教程 (超简单) [EB/OL]. https://blog.csdn.net/2301_77836489/article/details/138615618. 2024 (05)
- [16]莫余. LSTM 时间序列预测+股票预测案例(Pytorch 版) [EB/OL]. https://blog.csdn.net/qq_44824148/article/details/126222872. 2023 (12)
- [17]Mark. LSTM - 长短期记忆递归神经网络[EB/OL]. <https://zhuanlan.zhihu.com/p/123857569>. 2020 (05)
- [18]程序猿林仔. 报错 | pandas 报 A value is trying to be set on a copy of a slice from a DataFrame 的解决[EB/OL]. <https://blog.csdn.net/xuzhongyi103/article/details/135051554>. 2023 (12)
- [19]新缸中之脑. Plotly 安装与使用方法[EB/OL]. <https://blog.csdn.net/shebao3333/article/details/102471674>. 2019 (10)
- [20]pandas. pandas documentation[EB/OL]. <https://pandas.pydata.org/pandas-docs/stable/>. 2024 (04)
- [21]colah. Understanding LSTM Networks[EB/OL]. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. 2015 (08)
- [22]Sepp Hochreiter, Jürgen Schmidhuber. Long Short-term Memory[EB/OL]. https://www.researchgate.net/publication/13853244_Long_Short-term_Memory. 2024 (05)
- [23]捌椒. 一文搞懂毕业论文格式规范【超详细!!!】 [EB/OL]. https://blog.csdn.net/qq_40926887/article/details/127475958. 2024 (04)
- [24]TitaHh. Word 文档中批量修改英文、中文或数字等的字体和格式的方法[EB/OL]. https://blog.csdn.net/qq_50981222/article/details/123283235. 2022 (06)