

深度学习模型的开发、训练和推理实验：



摘要：

本次实验借助NKI深度学习平台，选择MNIST手写字符数据集和TensorFlow框架，分别构建了线性神经网络（LNN）和卷积神经网络（CNN），并通过反复调整超参数和模型结构的方式进行训练。通过观察不同参数结构下的训练趋势变化（loss、acc），初步理解了各个超参数对模型训练的影响，特别是在避免过拟合和欠拟合方面的平衡，以及神经网络层次架构的选择等方面有了初步的认识。

实验题目：深度学习模型的开发、训练和推理实验

1. 引言

本次实验借助NKI深度学习平台，使用MNIST手写字符数据集，tensorflow框架（经过多次测试，好像NKI的pytorch框架并不完善，模型训练很难成功），分别构建线性神经网络（Linear Neural Network）和卷积神经网络（Convolutional Neural Network, CNN）。通过调整不同超参数组合，以及模型的层次结构，观察训练趋势和模型准确率，对深度学习的流程有所体会。

2. 背景

线性神经网络（Linear Neural Network）和卷积神经网络（Convolutional Neural Network, CNN）是深度学习中两种常见的神经网络结构，它们模型结构比较简单，而且在图像分类（手写字符识别）任务中效果较好。

2.1 线性神经网络（LNN）：

- LNN是最基本的神经网络结构，包含输入层、隐藏层和输出层。每个节点与上一层的所有节点相连，通过权重进行连接。对于图像数据，LNN将图像像素展平成一维向量，并通过全连接层进行处理。然而，在处理图像时LNN可能难以捕捉到图像的空间结构和局部特征。

2.2 卷积神经网络（CNN）：

- CNN是专门设计用于处理图像数据的神经网络结构。它引入了卷积层和池化层，能够更有效地捕捉图像的局部特征，并通过权值共享减少参数量。CNN的结构更符合图像的空间层次结构，使其在图像分类任务中表现出色。在卷积层中，通过卷积操作可以提取图像的特征，而池化操作则有助于减小数据维度并保留重要信息。

3. 模型设计

3.1 线性神经网络（LNN）：

线性神经网络包含输入层784节点和输出层10节点，采用不同激活函数、学习率、权重初始化和网络层数进行调参。

1 Input Layer:

2 - 多个输入（可能是多个特征）

3

4 Hidden Layer 1 (dense):

5 - 多个神经元

6 - 使用激活函数（可以选择ReLU、Sigmoid等）

7

8 Hidden Layer 2 (dense):

9 - 多个神经元

10 - 使用激活函数（可以选择ReLU、Sigmoid等）

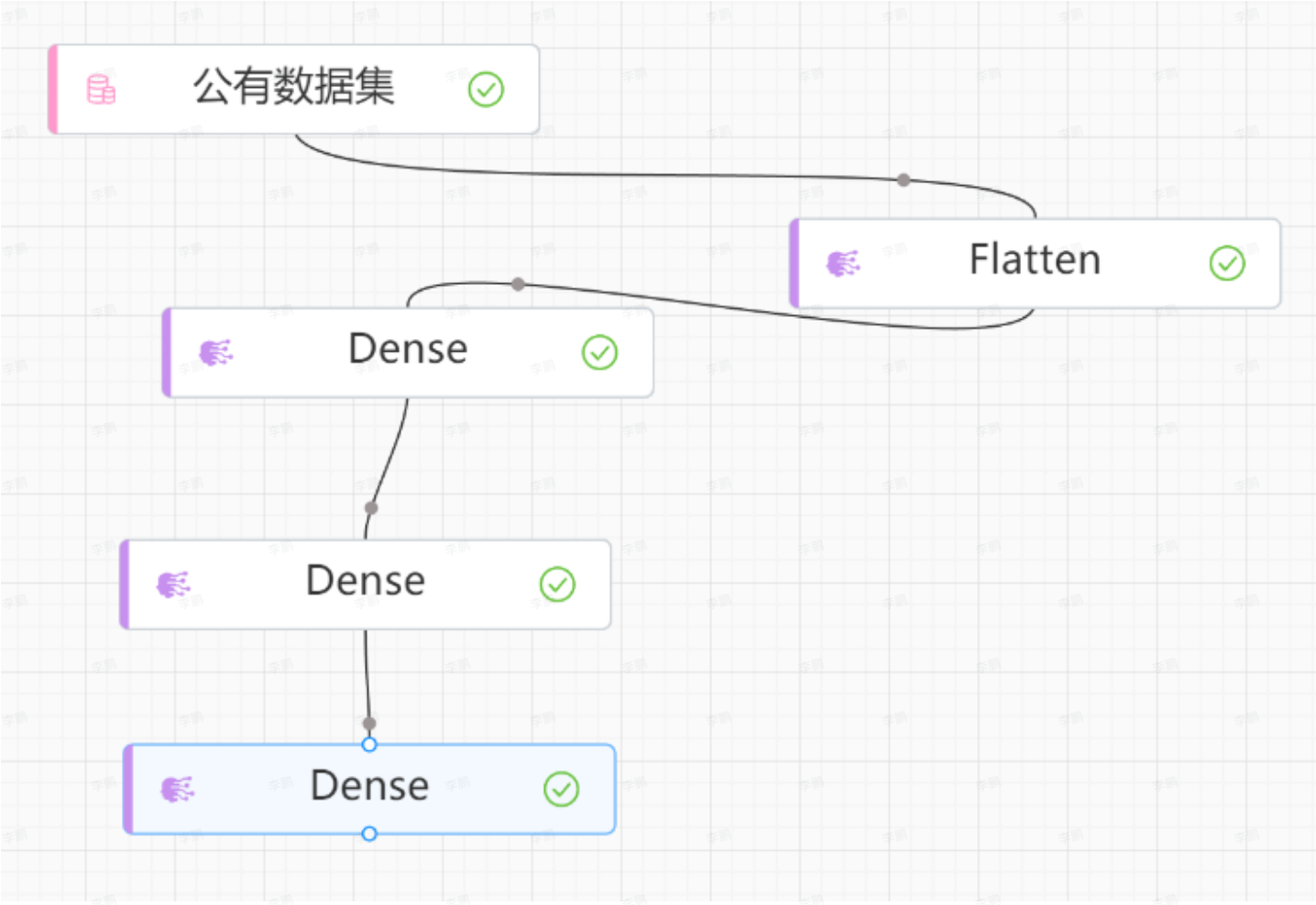
11

12 Output Layer (dense_3):

13 - 多个神经元，对应于多个类别或输出

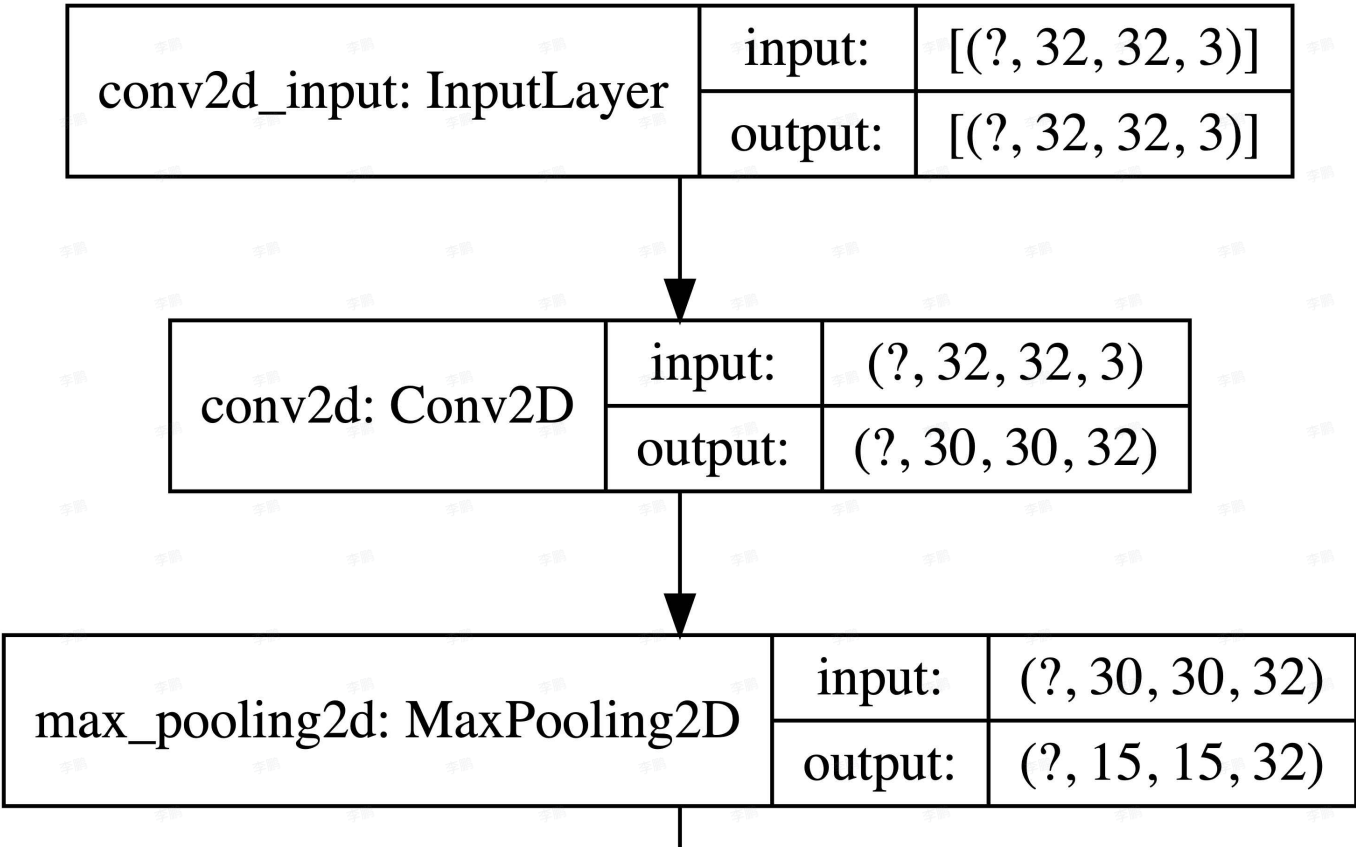
14 - 使用激活函数（可以选择ReLU、Sigmoid等）

dense: Dense	input:	multiple	dense_1: Dense	input:	multiple	dense_2: Dense	input:	multiple	dense_3: Dense	input:	multiple
	output:	multiple		output:	multiple		output:	multiple		output:	multiple



3.2 卷积神经网络（CNN）：

CNN模型结构包括卷积层、池化层和全连接层，对比了不同激活函数、学习率、权重初始化和网络层数的效果。



conv2d_1: Conv2D	input:	(?, 15, 15, 32)
	output:	(?, 13, 13, 64)

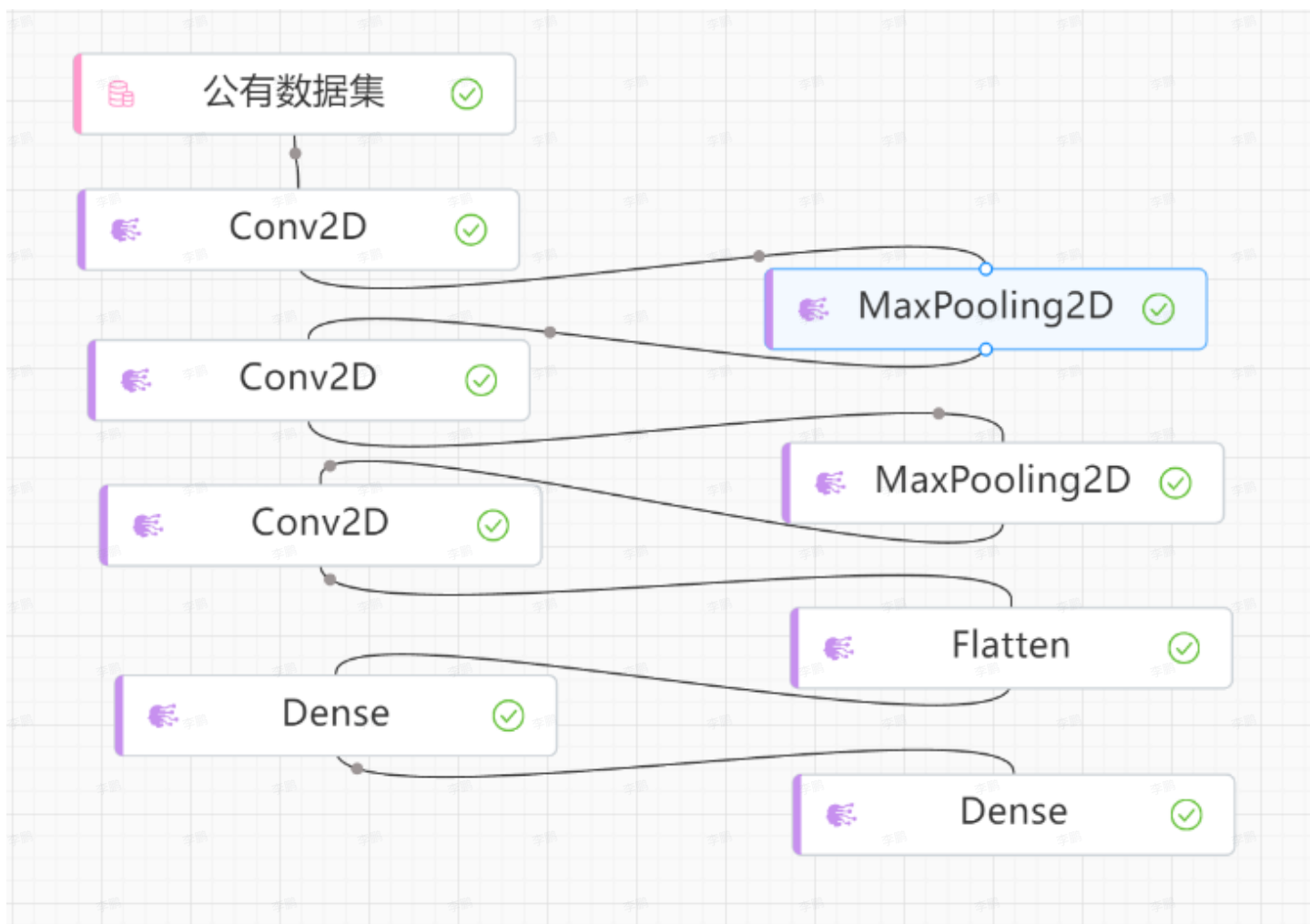
max_pooling2d_1: MaxPooling2D	input:	(?, 13, 13, 64)
	output:	(?, 6, 6, 64)

conv2d_2: Conv2D	input:	(?, 6, 6, 64)
	output:	(?, 4, 4, 64)

flatten: Flatten	input:	(?, 4, 4, 64)
	output:	(?, 1024)

dense: Dense	input:	(?, 1024)
	output:	(?, 64)

dense_1: Dense	input:	(?, 64)
	output:	(?, 10)



4. 调参过程

4.1 线性神经网络（LNN）：

选择激活函数：

在处理手写字符集MNIST时，尝试了不同的激活函数，包括ReLU、Softmax等。在隐藏层使用ReLU通常是一个不错的选择，而输出层的激活函数取决于任务。在本实验中，经过尝试，使用Softmax作为输出层的激活函数效果略好一点点，但区别不大。

确定神经元数量：

在每个密集层中调整神经元的数量。过多的神经元可能导致过拟合，而过少可能导致欠拟合。通过交叉验证等方法选择适当的神经元数量。经过尝试，不同的神经元数量配置对最终模型的准确率影响较大。最终采用的神经元数量配置如下：

- 输入层到 Flatten 层：28 * 28 -> (Flatten) -> 784
- Flatten 层到 隐藏层1：784 -> 256
- 隐藏层1 到 隐藏层2：256 -> 128
- 隐藏层2 到 输出层：128 -> 10

这个配置在验证集上表现较好，既不过拟合也不欠拟合。

学习率：

调整学习率以影响模型的收敛速度和稳定性。过大的学习率可能导致不稳定的训练，而过小可能导致收敛缓慢。经过尝试，学习率在0.01至0.1之间，当模型收敛后准确率差异不大。最终选择学习率为0.01，保证了相对稳定的训练过程。

批量大小：

调整批量大小，观察对模型性能的影响。在本实验中，尝试了不同的批量大小，发现在128的批量大小下模型表现较好，既保证了相对快速的收敛速度，又控制了训练时间。因此，最终选择批量大小为128。

最终线性神经网络（LNN）参数配置：

- 激活函数：ReLU（隐藏层），Softmax（输出层）
- 神经元数量配置：
 - Flatten 层到 隐藏层1：784 -> 256
 - 隐藏层1 到 隐藏层2：256 -> 128
 - 隐藏层2 到 输出层：128 -> 10
- 学习率：0.01
- 批量大小：128

这组参数组合在验证集上取得了令人满意的性能，使得模型在手写字符集MNIST上达到了相对较好的效果。

4.2 卷积神经网络（CNN）：

卷积核数量和大小：

- 尝试了不同数量和大小的卷积核，以观察对模型性能的影响。通过调整卷积核的数量和大小，寻找更好的特征提取。

尝试1：

- 第一个卷积层（conv2d）：64个3x3的卷积核
- 第二个卷积层（conv2d_1）：128个3x3的卷积核
- 第三个卷积层（conv2d_2）：128个3x3的卷积核

结果：模型出现了过拟合，训练集表现良好，但验证集性能下降。

尝试2：

- 第一个卷积层（conv2d）：32个3x3的卷积核
- 第二个卷积层（conv2d_1）：64个3x3的卷积核

- 第三个卷积层（conv2d_2）：64个3x3的卷积核

结果：模型表现较好，验证集性能有所提升，减小了过拟合的趋势。

激活函数的选择：

- 在卷积层中选择了ReLU作为激活函数。ReLU通常在卷积神经网络中表现较好，能够更好地学习非线性特征。这一选择在实验中取得了较好的性能。
- 使用Sigmoid作为激活函数

结果：模型训练缓慢，收敛速度较慢，验证集性能相对较差。

- 使用ReLU作为激活函数

结果：模型训练速度加快，收敛较快，验证集性能相对提高。

池化层的影响：

- 调整了池化层的大小，观察对模型的空间维度压缩和特征提取的影响。最终选择了2x2的最大池化层（MaxPooling2D），以减小空间维度并保留重要信息。
- 选择3x3的最大池化层

结果：模型性能有所下降，可能因为过大的池化窗口损失了部分特征。

- 选择2x2的最大池化层

结果：模型性能有所提升，验证集准确率较高。

全连接层神经元数量：

- 在全连接层中调整了神经元数量，以影响模型的拟合能力。通过实验观察模型在验证集上的性能，最终选择了以下配置：
 - Flatten 层到 全连接层1：256个神经元
 - 全连接层1 到 全连接层2：128个神经元

最终卷积神经网络（CNN）参数配置：

卷积核数量和大小：

- 第一个卷积层（conv2d）：32个3x3的卷积核
- 第二个卷积层（conv2d_1）：64个3x3的卷积核
- 第三个卷积层（conv2d_2）：64个3x3的卷积核

激活函数：

- 在卷积层中使用ReLU作为激活函数

池化层：

- 选择2x2的最大池化层（MaxPooling2D）

全连接层神经元数量：

- Flatten 层到 全连接层1：256个神经元
- 全连接层1 到 输出层：10个神经元

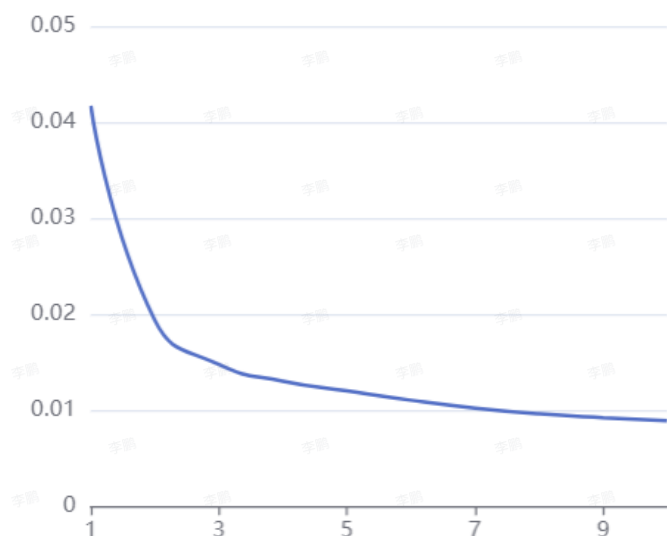
通过对这些参数的调整，得到了在手写字符集MNIST上表现较好的最终卷积神经网络模型。这组参数配置在维持性能的同时减小了过拟合的风险，提高了模型的泛化能力。

5. 实验结果：

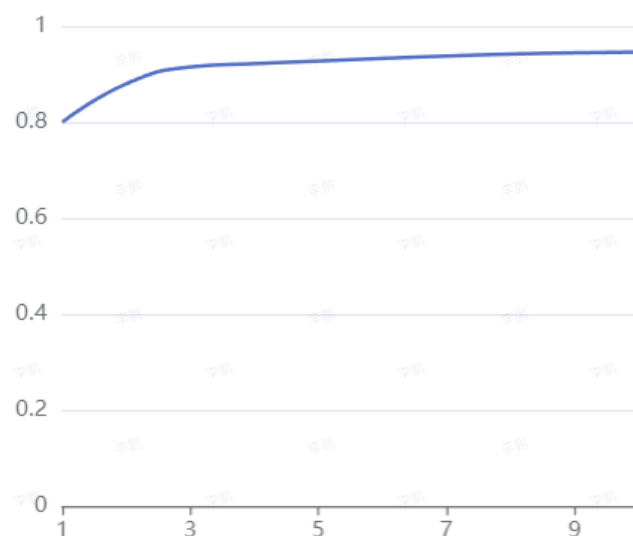
5.1 线性神经网络（LNN）：

- 训练误差

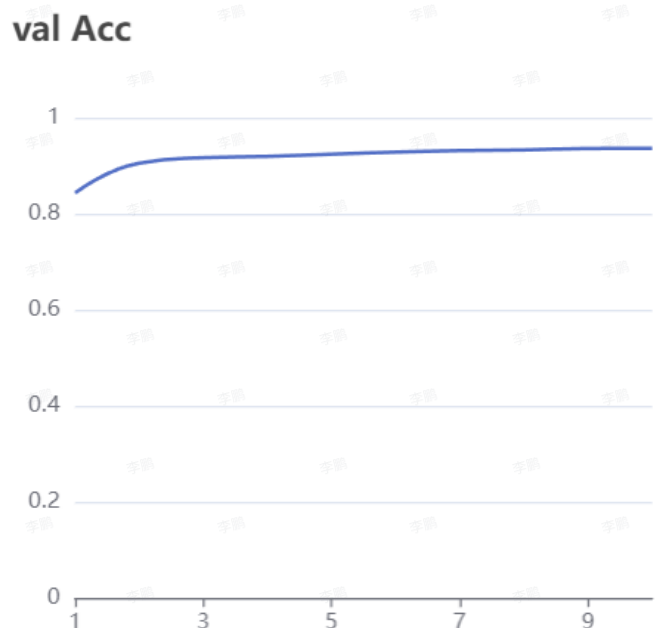
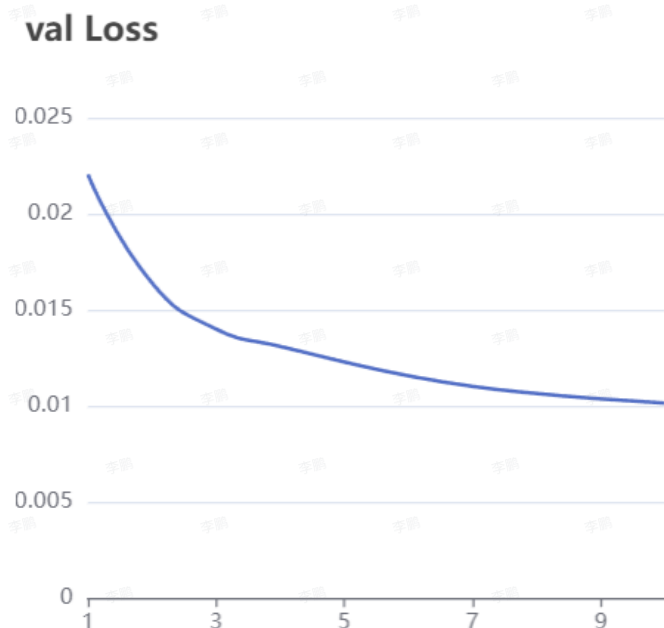
train Loss



train Acc



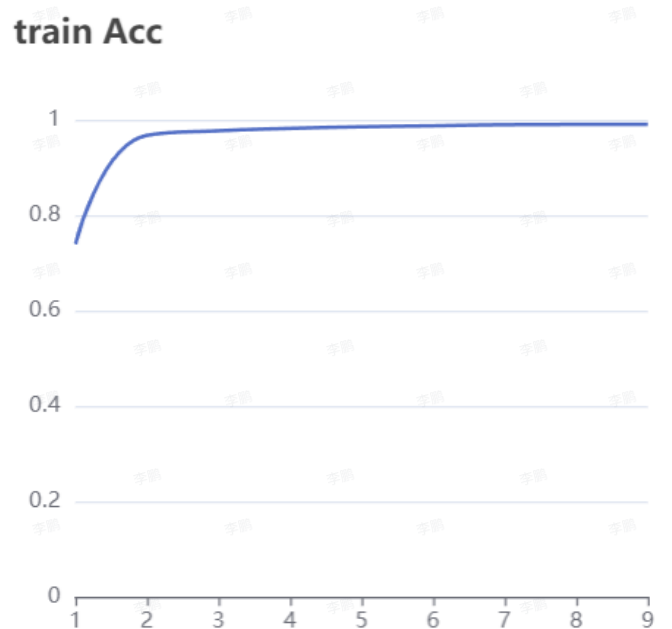
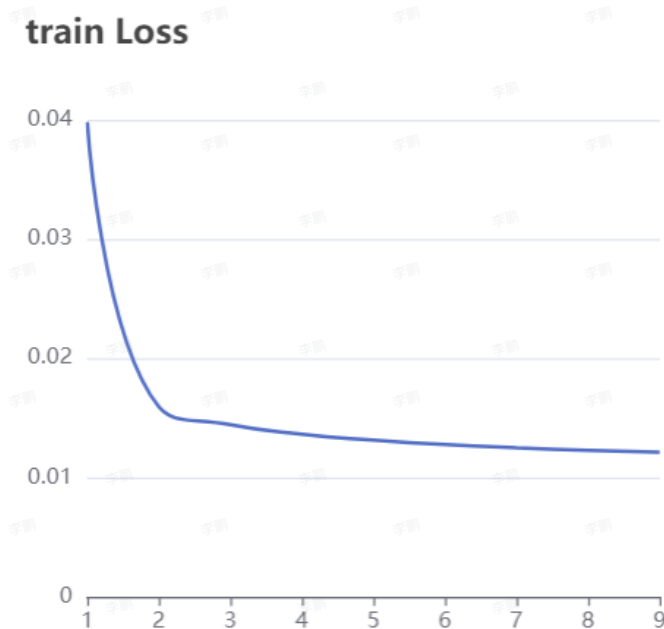
- 测试误差



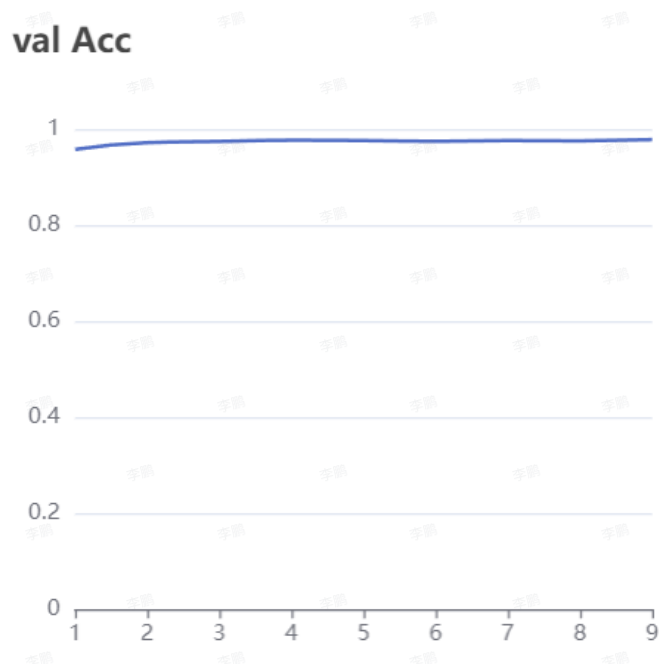
- 训练误差：随着训练次数增加，训练误差逐渐降低，模型在训练集上表现良好。
- 测试误差：测试误差在合理范围内波动，说明模型在测试集上保持一定的泛化性能。

5.2 卷积神经网络（RNN）：

- 训练误差：



- 测试误差：



6. 实验总结

- 通过对不同超参数的调整，进一步理解了深度学习神经网络中的各种超参数的含义，包括激活函数、学习率、权重的初始化、网络层数等对模型性能的影响。
- 通过观察和分析训练趋势（loss值、准确率等），根据变化趋势对症下药，对模型进行简单的改进，
- 在调参过程中，需要在训练集和验证集、稳定性和性能之间找到平衡，避免过拟合和欠拟合的问题。
- 卷积神经网络在图像分类任务中表现出色，通过卷积和池化操作能更好地捕捉图像的空间特征。

通过本次实验，不仅学会了一些调参的具体方法，还体验到了深度学习模型设计的迭代过程，加深了对深度学习流程的理解，同时提高了对实验结果进行深入分析，并据此优化模型结构的能力。