深度学习小作业 2 报告

刘翰文 522030910109

1. 图像分类问题

图像分类问题可以表述为：给定一批图片$x_1, x_2, \ldots, x_n$以及它们对应的标签$y_1, y_2, \ldots, y_n$，希望可以学习到一种输入数据和标签的映射f，使得$f(x_i)$尽可能等于y_i。目前常用的方法是借助判别式神经网络，利用网络学习这种映射，以最大化或最小化某个目标函数，从而实现对图片的分类。在本次作业中，将主要使用 CNN 和 RNN 两种常见的神经网络对图片进行分类。

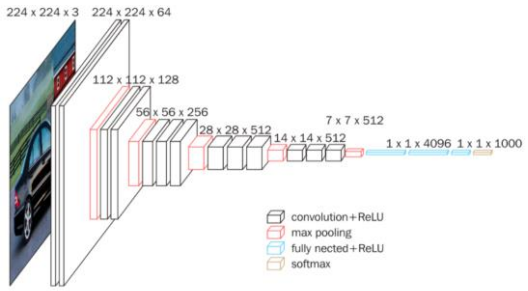
2. 数据集

本次作业使用的数据集是 CIFAR-10 数据集，包含 50000 张训练照片和 10000 张测试照片总计 60000 张照片，每张照片有着$32 \times 32 \times 3$的大小，并且有 10 种类别，每种类别的图像都是 6000 张。

3. CNN 卷积神经网络

3.1 网络介绍

CNN 卷积网络由卷积层、池化层、激活层、线性层等组成，可以很好地提取输入图像的特征，然后利用线性层进行分类任务。一个经典的 CNN 卷积网络 VGG 如下图所示



在本次对 CIFAR-10 进行分类的作业中，我构建的 CNN 网络参考了 VGG 以及其他一些网络，主要包含了 6 个卷积层 nn.Conv，在每层的卷积操作过后进行归一化操作 nn.BatchNorm 和用 nn.Relu 函数激活，在经过卷积层后输出通道会相应地增加，所以增加了 nn.MaxPool2d 的池化层降低输入图像的分辨率，在经过一系列的卷积层和 3 个池化层后得到$128 \times 4 \times 4$的输出，将此输出最终送入用 BN 层和激活函数相连接的两个 FC 全连接层得到一个1×10的输出，对其进行取 10 个类别中得到的最大值即可将其作为分类结果。

3.2 数据预处理

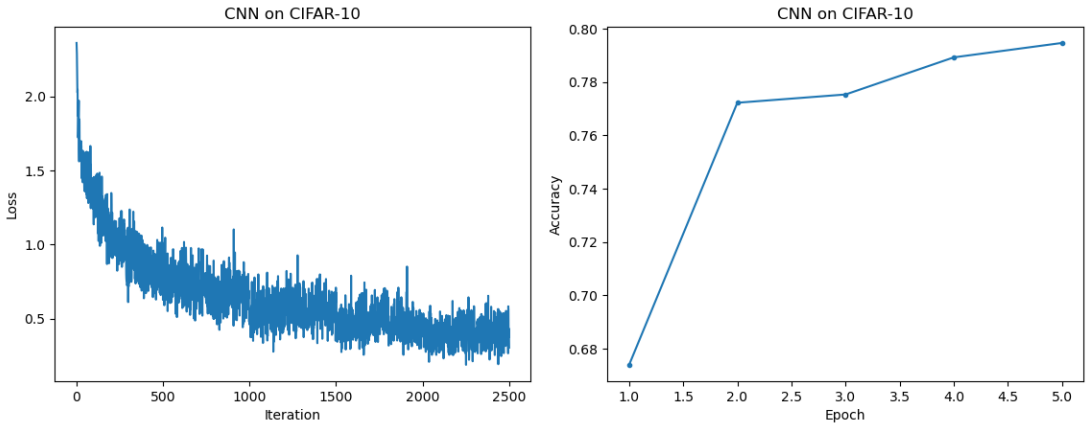
本次对 CIFAR-10 的分类前，首先对输入数据进行数据的预处理，包括随机翻转，归一化，将原本每个通道$(0,255)$的数据归一化到均值为 0 的小数。同时在 Dataloader 中设置 batch_size=100，并对数据进行打乱分别得到训练的数据集和测试的训练集。

3.3 模型训练

定义损失函数为交叉熵损失函数，定义优化器为随机梯度下降优化算法，以较小的 epoch=5，学习率为 0.01 对构建的 CNN 模型进行训练，在每次 epoch 的训练后对模型进行测试，同时记录每次训练过程中的损失下降结果和测试准确率结果。

3.4 结果分析

训练过程中生成的损失变化图和测试准确率结果图如下

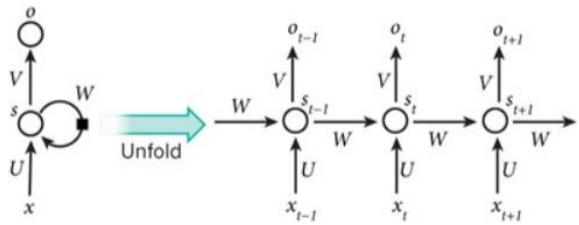


从结果可以看出：训练损失随训练的轮数整体成下降趋势并且最终逐渐趋于稳定在 0.5 的损失值，在测试集上的准确率随 epoch 的增加不断上升，在 5 个 epoch 后可以达到 0.79 左右的水平。

4. RNN 循环神经网络

4.1 网络介绍

RNN 网络相比全连接神经网络，在它的基础上增加了前后时序上的关系，在解决存在着许多序列关系的问题上得到了较好的应用，如语音、文本、视频等，此外，RNN 还有着许多的变体，包括 LSTM、GRU 等。基本的 RNN 循环网络结构包括了输入层、隐藏层和输出层，其结构可以由下图表示



RNN 网络隐藏层的值不仅取决于当前的输入，还取决于上次隐藏层的值，也得以具备了解决序列问题的能力。本次作业所使用的 RNN 网络为其变体 LSTM，网络层数设置为 2，所使用的隐藏层特征维度为 256，在经过 LSTM 后再通过两个由 BN 层和 Relu 激活函数的全连接层得到一个输出的1×10的输出，对其进行取 10 个类别中得到的最大值即可将其作为分类结果。

4.2 数据预处理

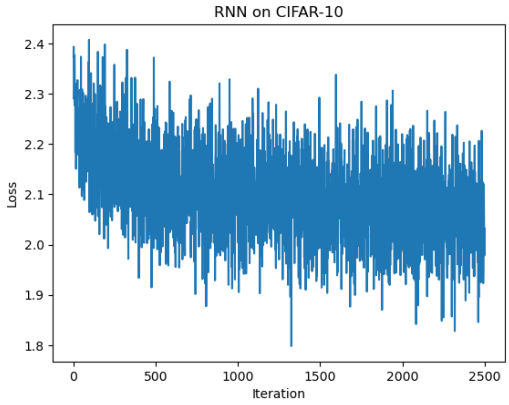
与 CNN 部分的数据预处理相似，首先对输入图片进行了随机翻转和归一化的操作，在 Dataloader 中设置 batch_size=100，并对数据进行打乱分别得到训练的数据集和测试的训练集。不同的是，LSTM 要求把图片转化为一串序列，所以对于一张$3 \times 32 \times 32$的图片，需要将其转化为$(32,3 \times 32)$的 32 个 96 维的张量，再把这些张量作为模型输入。

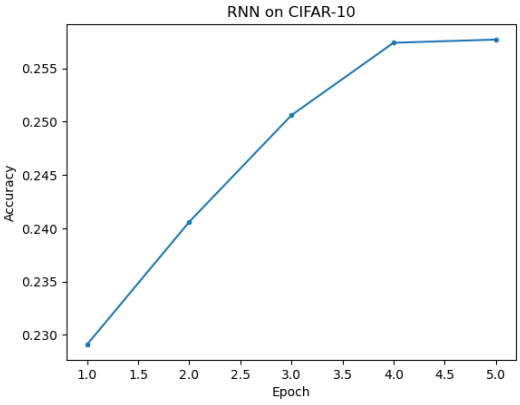
4.3 模型训练

与 CNN 的模型训练一样，定义损失函数为交叉熵损失函数，定义优化器为随机梯度下降优化算法，以较小的 epoch=5，学习率为 0.01 对构建的 CNN 模型进行训练，在每次 epoch 的训练后对模型进行测试，同时记录每次训练过程中的损失下降结果和测试准确率结果。

4.4 结果分析

训练过程中生成的损失变化图和测试准确率结果图如下





从结果可以看出：训练损失随训练的轮数整体成下降趋势并且最终逐渐趋于稳定在 2.1 的损失值，在测试集上的准确率随 epoch 的增加不断上升，但在 5 个 epoch 后只能达到 0.26 左右的水平。通过两个网络的比较发现在图像分类问题上，CNN 网络表现显著优于 RNN 网络。

5. 划分后数据集的分类

5.1 数据集介绍

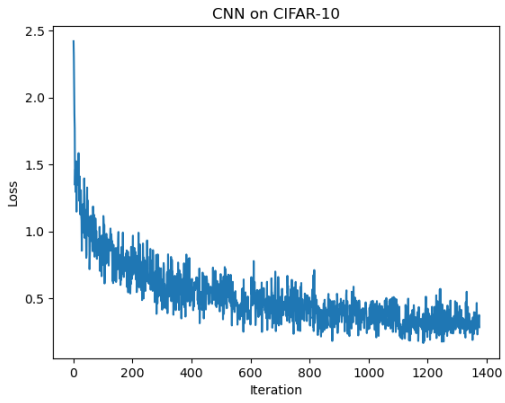
相较于之前所使用的数据集，在原先包含 60000 张图像的数据集中每个标签对应的图像数量相同，均为 6000 张。而在这个问题中，标签为 0-4 的图像减少了 90%，每个标签对应的图像数量仅剩下 600 张，而标签为 5-9 的图像数量保持为原先的 6000 张。数据集的变化使数据分布不在满足独立同分布，对模型的泛化能力带来了极大的挑战，要求提高模型的鲁棒性以具备一定的分布外泛化能力。

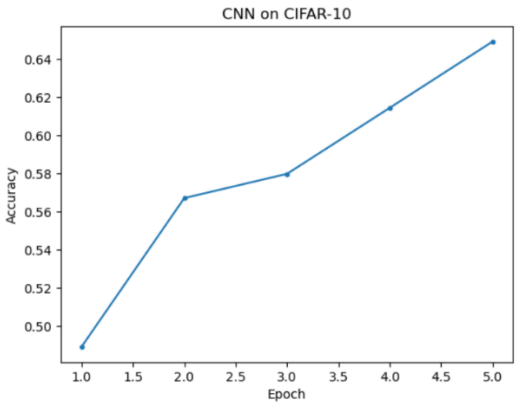
5.2 改进策略

首先不难想到可以从数据集上对数据集进行改进，例如遍历数据集，将缺少的那类图片进行复制，并对复制后的图像进行更多的预处理比如翻转、旋转、裁剪、添加噪声等。但考虑到此类问题直接改动数据集并非研究目的，所以本次作业改动的主要地方是 weight_decay 权重衰减参数。权重衰减参数可以在一定程度上抑制模型的过拟合，减小模型的复杂度来提高模型的泛化能力。通过将权重衰减参数 weight_decay=1e-4 引入 CNN 和 RNN，相当于在损失函数后面增加了一个模型参数的 2 范数平方$L = L_0 + \frac{\lambda}{2}||W||^2$，然后再分别引入 CNN 和 RNN 的训练过程中，和上文一样，用相同的参数对构建的 CNN 和 RNN 模型进行训练，在每次 epoch 的训练后对模型进行测试，同时记录每次训练过程中的损失下降结果和测试准确率结果。

5.3 结果分析

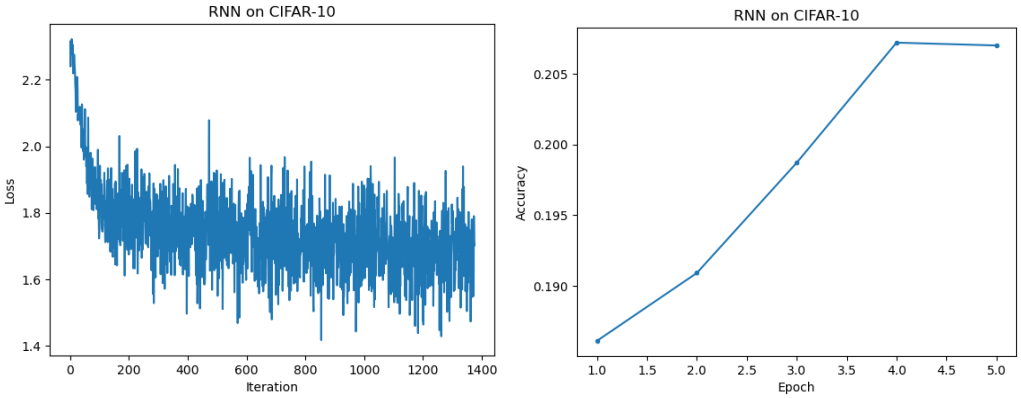
首先对 CNN 进行训练和测试，得到损失变化结果和测试结果如下





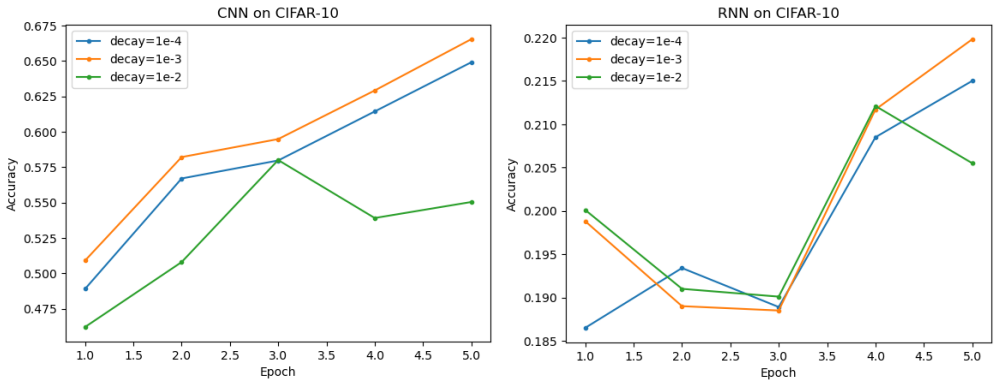
与原先在未改变的数据集上进行训练的 CNN 模型相比，新模型的训练损失与原模型都在 0.5 附近趋于稳定，但在测试集上的准确率却相较原模型从 0.79 降至 0.65。

对 RNN 进行训练和测试，得到结果如下



和原本结果对比，在损失减小的同时测试准确率却从 0.25 降至 0.20，不难想到是训练图像标签的集中使得损失的减小。

. 在对 CNN 和 RNN 引入权重衰减参数的同时，也研究了参数大小对模型的影响，分别取 weight_decay=1e-4,1e-3 和 1e-2 进行测试，在 CNN 和 RNN 上得到的结果如图



从两张比较图都可以发现，在参数值为 1e-3 时，模型的泛化能力最强，预测的准确率最高。同时，与未引入参数的两个模型相比，准确率都有了 0.01 到 0.02 的提升。

6. 结论

本次小作业实现了通过 CNN 和 RNN 网络对 CIFAR-10 数据集的分类，并且提出了改进 CNN 和 RNN 的方法：引入权重衰减参数，以此来改进模型在不平衡的 CIFAR-10 上的表现。最终结果显示，改进后了模型与原先相比有了微小的提升，并且在参数值为 1e-3 时表现最好。