



课程/2023 深度学习 学号/SA23229086 学生/郎文翀
大数据学院计算机技术班
2023 年 11 月 27 日星期六

使用 pytorch 或者 tensorflow 实现卷积神经网络，在 ImageNet 数据集上进行图片分类。研究 dropout, normalization, learning rate decay, residual connection, network depth 等超参数对分类性能的影响。

2.1 设计需求

本次实验选择 Pytorch 实现，并使用 GPU 版本进行训练，最后采用 matplotlib 完成绘图与实验结果分析，具体程序设计需求如下：

数据集处理：数据集制作完备方便训练与测试。本次实验使用 ImageNet200 小数据集，并且训练集与验证集都基于 ImageNet 中的 train data 制作并自动划分，测试集使用 ImageNet200 中的验证集数据。

模型参数设置：可以手动设置网络模型深度，是否加入数据归一化，残差链接，dropout, 学习率自动退化调节等。

分批次训练：每 Epoch 数据分为 64 一组进行训练。

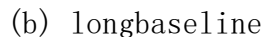
对比实验：对比不同网络模型在测试集的表现并分析各优化模块对实验结果的影响。

2.2 网络模型图

本次实验我共进行了 13 组测试，分别是仅仅基于浅层卷积神经网络的 shortbaseline 模型(具体网络模型图如下)以及其变种：

各仅加入数据归一化(shortbaseline+normalization), dropout(shortbaseline+dropout), 学习率调节(shortbaseline+learningratedecay), 残差连接(shortbaseline+residualblock)情况以及全部优化整合(shortbaseline+all)。

同时深度学习网络(相较于浅层神经网络分别各层加入两次双卷积的情况 longbaseline)及其变种(+dropout, +learningratedecay, +residualblock, +normalization, +all)以及介于浅层与深层网络之间的网络模型(各层之间仅加入一层双卷积神经网络层 midbaseline)。如下图是 shortbaseline, longbaseline, midbaseline 的三种情况网络模型图:

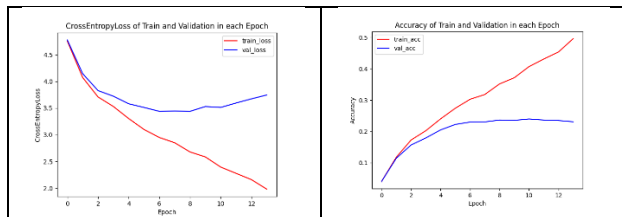


3 实验分析

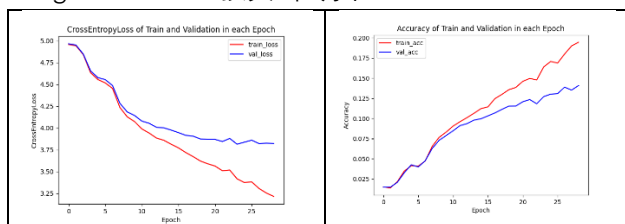
3.1 模型深度对比

在模型深度对比实验中，我通过选取 shortbaseline+all, midbaseline+all, longbaseline+all 三个网络模型的训练结果以及测试集上的结果进行对比研究。

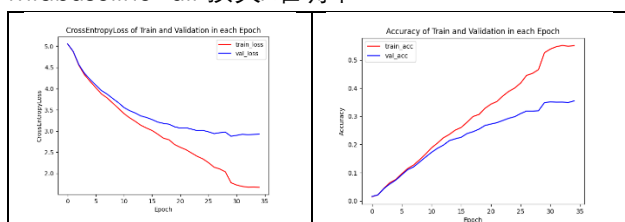
Shortbaseline+all 损失/准确率



Longbaseline+all 损失/准确率



Midbaseline+all 损失/准确率

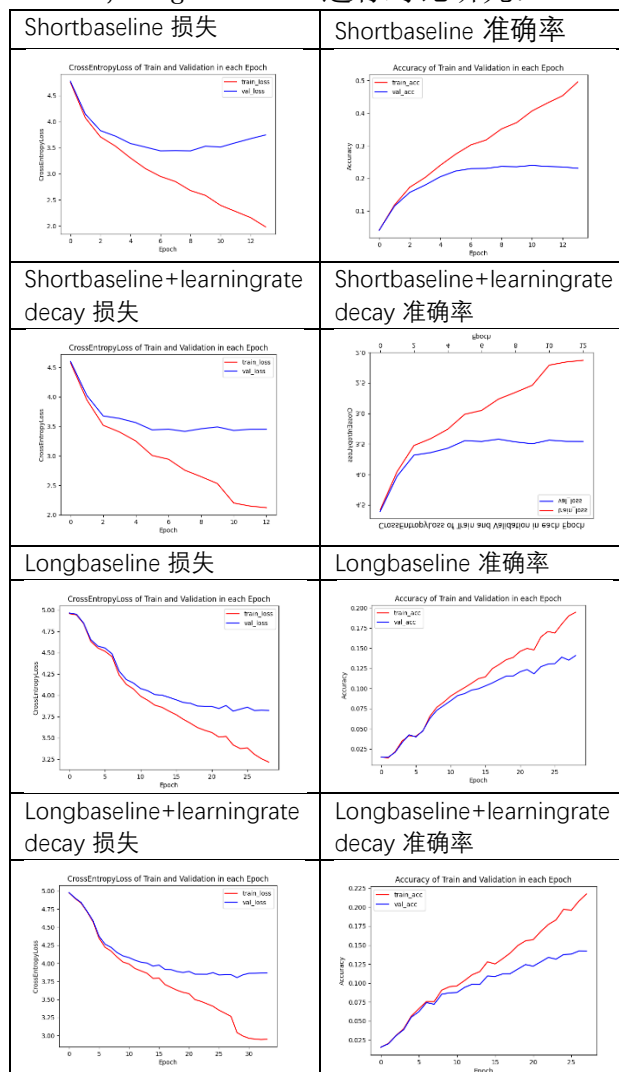


观察上图发现 shortbaseline+all 在第 15 轮即达到最佳, 训练集的准确率已高达 82%, 验证集准确率 43.7%, 在测试集的准确率为 43%, 首先分析 shortbaseline+all 在训练集的准确率表明其已经充分学习训练数据的分布情况, 之所以无法在测试集表现更佳是因为训练集分布无法充分模拟测试集, 因此此时 shortbaseline 在最佳训练模型参数情况下只能在测试集得到 43% 的推理分类图像准确率。而 longbaseline+all 模型在经过 67 轮充分的训练迭代后也仅能在训练集达到 41.7% 准确率, 且在验证集和测试集的准确率均在 25% 左右, 分析其原因是因为模型深度过深导致的梯度退化无法充分学习调整参数因此最终收敛后对图像的分类能力反而不如 shortbaseline, 这初步说明模型深度并非越深越好, 为了进一步验证结论, 我又对 midbaseline+all 模型的结果进行分析, midbaseline+all 模型在 34 轮训练迭代后停止达到最佳, 此时在测试集的准确率为 36.1% 介于 shortbaseline+all 与 longbaseline+all 两者模型之间, 这验证了模型深度过深会导致模型学习能力退化的结论。

3.2 学习率调节对比

网络模型的学习能力与学习率的超参数设置关系密切, 当学习率超参数设置为固定的一个较大数值后可能会导致模型由于每轮迭代更新幅度过大而导致模型无法收敛, 为了避免这个情况, 提出了动态调整学习率的方法, 即根据每轮训练后模型在验证集的准

准确率表现进行评估, 当连续多轮验证集优化幅度不佳时尝试降低学习率。为了验证学习率调节对模型具有正反馈优化作用, 我在 shortbaseline 与 longbaseline 中分别仅仅加入了学习率调节优化器, 即生成 shortbaseline+learningratedecay 与 longbaseline+learningratedecay 两个模型并于基线 shortbaseline, longbaseline 进行对比研究:

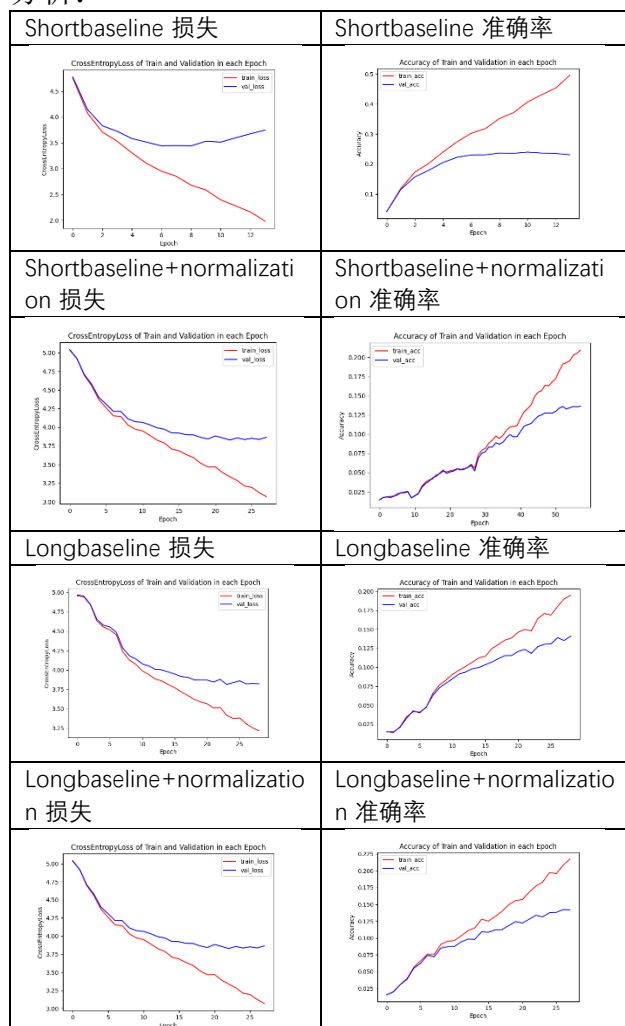


对比上图不难发现加入 learning rate decay 学习率衰减调节后相较于基线 baseline 模型迭代轮数更多, 可学习参数可以进一步充分学习从而使得 shortbaseline+learningratedecay 与 longbaseline+learningratedecay 相较于 shortbaseline 和 longbaseline 模型。

3.3 数据归一化对比

在深度学习中, 归一化对输入数据进行预处理, 以便更好地训练模型。这有助于加

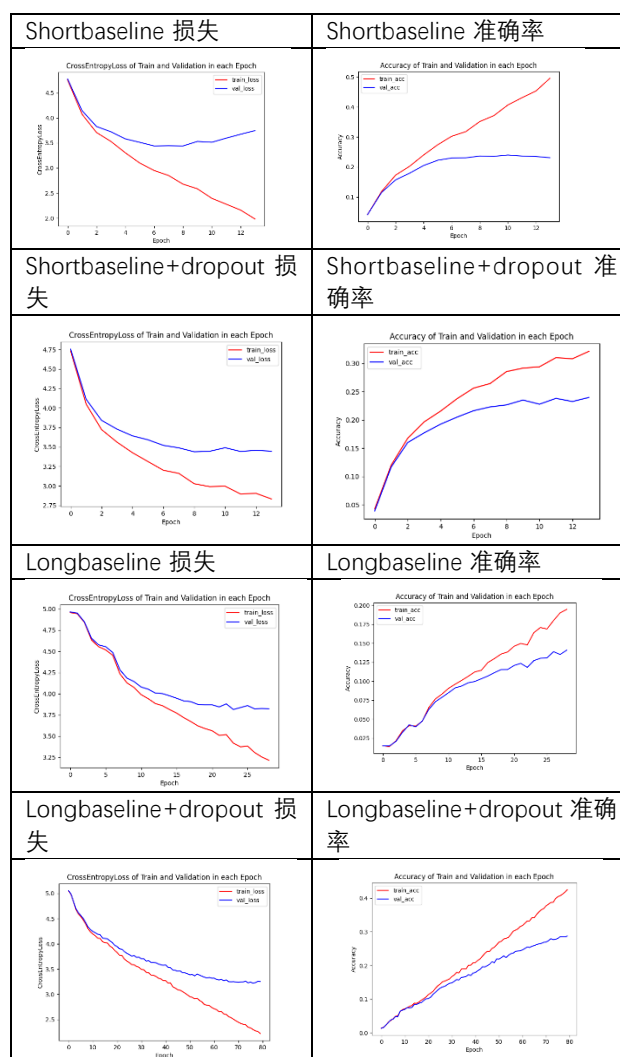
速训练过程，允许使用更高的学习率，并提高模型的泛化能力。因此我将接下来对比 shortbaseline, longbaseline 与 shortbaseline+normalization, longbaseline+normalization 模型训练后的测试集表现结果进行分析：



对比以上实验结果图发现加入数据归一化后 shortbaseline+normalization, longbaseline+normalization 相较于 shortbaseline 与 longbaseline 基线模型有小幅准确率提升，且训练时间减少，收敛速度更快。

3.4 Dropout 结果对比

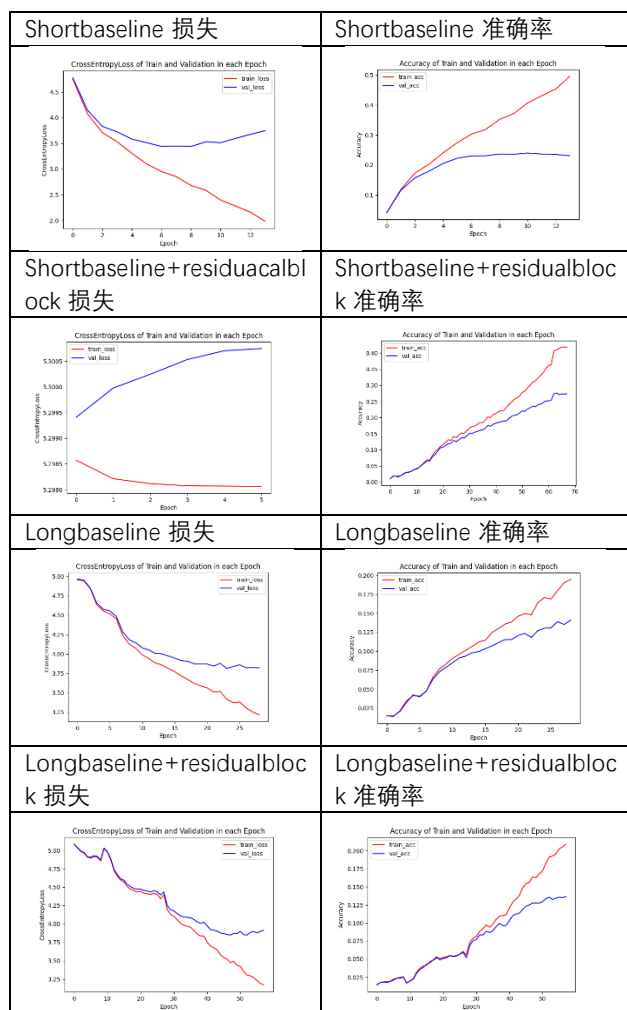
Dropout 是深度学习中一种常用的正则化技术，其主要目的是减少神经网络的过拟合。过拟合是指模型在训练数据上表现得很好，但在未见过的数据上表现较差的情况。Dropout 通过在训练过程中随机关闭一些神经元的连接来防止过拟合。以下是实验结果：



在加入 Dropout 模块后会随机按一定概率来抑制部分层神经元的激活从而减少模型的过拟合现象，结合上图实验结果可以看出在加入 dropout 后无论是 shortbaseline+dropout 还是 longbaseline+dropout 相较于基线模型都在测试集的准确率表现有明显的提升。

3.5 残差连接

残差连接是一种深度神经网络结构中的一项技术，用于解决深度网络训练过程中的梯度消失和梯度爆炸问题。在本次图片分类模型中，shortbaseline 和 longbaseline 的卷积层都较多，加入残差连接可以降低梯度消失的风险，同时避免由于深度过深导致梯度回传数值逐渐减少而无法有效训练模型参数造成模型学习速度极慢，无法在一定迭代次数后达到最佳的分类性能。以下是加入残差连接前后模型的效果图：



加入残差连接后发现在 longbaseline+residualblock 如之前所预测的可以避免梯度爆炸问题，只不过加入残差连接后训练时常更长这是因为需要额外进行其他分支的计算。但是在 shortbaseline+residualblock 后却无法收敛，模型不断震荡这可能是因为加入残差连接后梯度回传值过大，而浅层网络模型会由于回传的梯度值过大且过大的固定学习率在进行 $\omega' = \omega - \alpha * \text{grad}$ 造成可训练参数的变化值过大无法向最优值靠拢从而导致模型始终无法向 loss 最小值方向优化，因此 7 次迭代后即停止，无法在测试集实现有效的推理预测图片类别。

3.6 总结

以上各模块分别控制变量探讨研究了 shortbaseline, midbaseline, longbaseline 基线模型以及加入不同优化模块后的图片分类性能，综上所有不同的模型对比，发现 shortbaseline+all 是表现最佳的模型，可以使

得训练参数充分学习并且没有过拟合在测试集达到最高准确率，而 shortbaseline+residualblock 表现最差由于无法收敛基本无法实现图片分类功能。而整体上观察 longbaseline 与 midbaseline 由于深度过深相较于 shortbaseline 在测试集的准确率表现反而更低，这也验证了网络模型并不是无休止的越深越好，需要经过合理的构建与设计的网络模型与充分分布合理的数据集充分迭代训练后才能在未训练的测试集上表现出最佳的性能，这也启发我们以后在未来的科研工作中要着重注意网络模型设计的合理性。最后对以上左右使用的模型在训练集，验证集与测试集的准确率进行总结（每一个模型具体的训练过程请参考同文件夹下的 log 日志）：

网络模型	训练集准确率	验证集准确率	测试集准确率
Shortbaseline	49.60%	23.09%	22.71%
longbaseline	19.46%	14.09%	14.07%
Shortbaseline+learning rate decay	47.24%	26.41%	26.66%
Longbaseline+learning rate decay	23.69%	14.16%	15.1%
Shortbaseline+normalization	92.6%	34.43%	34.06%
Longbaseline+normalization	21.75%	14.20%	14.59%
Shortbaseline+dropout	32.11%	23.96%	24.14%
Longbaseline+dropout	42.49%	28.70%	29.01%
Shortbaseline+residual block	0.53%	0.37%	0.49%
Longbaseline+residualblock	20.93%	13.64%	13.93%
Shortbaseline+all	82.18%	43.07%	43.02%
Longbaseline+all	43.18%	27.43%	27.62%
Midbaseline+all	55.13%	35.50%	36.17%

4. Requirements.txt

```
abs1-py==2.0.0
autopep8==2.0.4
cachetools==5.3.1
certifi==2023.7.22
charset-normalizer==3.3.0
colorama==0.4.6
contourpy==1.1.1
cyclер==0.12.1
fonttools==4.43.1
google-auth==2.23.3
google-auth-oauthlib==1.1.0
grpcio==1.59.0
idna==3.4
importlib-metadata==6.8.0
importlib-resources==6.1.0
kiwisolver==1.4.5
Markdown==3.5
MarkupSafe==2.1.3
matplotlib==3.8.0
netron==7.2.5
numpy==1.26.1
oauthlib==3.2.2
packaging==23.2
Pillow==10.1.0
protobuf==4.23.4
pyasn1==0.5.0
pyasn1-modules==0.3.0
pycodestyle==2.11.1
pyparsing==3.1.1
python-dateutil==2.8.2
requests==2.31.0
requests-oauthlib==1.3.1
rsa==4.9
six==1.16.0
tensorboard==2.15.0
tensorboard-data-server==0.7.1
tomli==2.0.1
tqdm==4.66.1
typing_extensions==4.8.0
urllib3==2.0.7
Werkzeug==3.0.0
zipp==3.17.0
```