训练技巧

1.优化器的选择

自深度学习发展以来，就有很多关于优化器的研究者工作，优化器的目的是为了让损失函数尽可能的小，从而找到合适的参数来完成某项任务。目前业界主要用到的优化器有 SGD、RMSProp、Adam、AdaDelt 等，其中由于带 momentum 的 SGD 优化器广泛应用于学术界和工业界，所以我们发布的模型也大都使用该优化器来实现损失函数的梯度下降。带 momentum 的 SGD 优化器有两个劣势，其一是收敛速度慢，其二是初始学习率的设置需要依靠大量的经验，然而如果初始学习率设置得当并且迭代轮数充足，该优化器也会在众多的优化器中脱颖而出，使得其在验证集上获得更高的准确率。一些自适应学习率的优化器如 Adam、RMSProp 等，收敛速度往往比较快，但是最终的收敛精度会稍差一些。如果追求更快的收敛速度，我们推荐使用这些自适应学习率的优化器，如果追求更高的收敛精度，我们推荐使用带 momentum 的 SGD 优化器。

2.学习率以及学习率下降策略的选择

学习率的选择往往和优化器以及数据和任务有关系。这里主要介绍以 momentum+SGD 作为优化器训练 ImageNet-1k 的学习率以及学习率下降的选择。

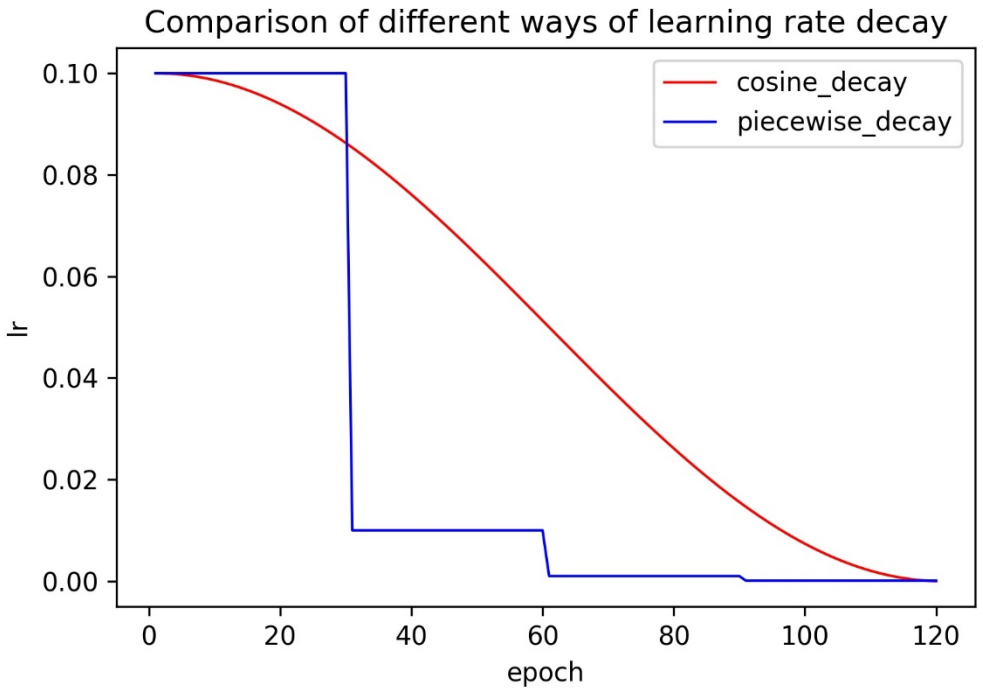
学习率的概念：

学习率是通过损失函数的梯度调整网络权重的超参数的速度。学习率越低，损失函数的变化速度就越慢。虽然使用低学习率可以确保不会错过任何局部极小值，但也意味着将花费更长的时间来进行收敛，特别是在被困在高原区域的情况下。

学习率下降策略：

在整个训练过程中，我们不能使用同样的学习率来更新权重，否则无法到达最优点，所以需要在训练过程中调整学习率的大小。在训练初始阶段，由于权重处于随机初始化的状态，损失函数相对容易进行梯度下降，所以可以设置一个较大的学习率。在训练后期，由于权重参数已经接近最优值，较大的学习率无

法进一步寻找最优值，所以需要设置一个较小的学习率。在训练整个过程中，很多研究者使用的学习率下降方式是 piecewise_decay，即阶梯式下降学习率，如在 ResNet50 标准的训练中，我们设置的初始学习率是 0.1，每 30epoch 学习率下降到原来的 1/10，一共迭代 120epoch。除了 piecewise_decay，很多研究者也提出了学习率的其他下降方式，如 polynomial_decay（多项式下降）、exponential_decay（指数下降）,cosine_decay（余弦下降）等，其中 cosine_decay 无需调整超参数，鲁棒性也比较高，所以成为现在提高模型精度首选的学习率下降方式。Cosine_decay 和 piecewise_decay 的学习率变化曲线如下图所示，容易观察到，在整个训练过程中，cosine_decay 都保持着较大的学习率，所以其收敛较为缓慢，但是最终的收敛效果较 peicewise_decay 更好一些。



另外，从图中我们也可以看到，cosine_decay 里学习率小的轮数较少，这样会影响到最终的精度，所以为了使得 cosine_decay 发挥更好的效果，建议迭代更多的轮数，如 200 轮。

warmup 策略

如果使用较大的 batch_size 训练神经网络时，我们建议您使用 warmup 策略。Warmup 策略顾名思义就是让学习率先预热一下，在训练初期我们不直接使用最大的学习率，而是用一个逐渐增大的学习率去训练网络，当学习率增大到最

高点时，再使用学习率下降策略中提到的学习率下降方式衰减学习率的值。实验表明，在 batch_size 较大时，warmup 可以稳定提升模型的精度。在训练 MobileNetV3 等 batch_size 较大的实验中，我们默认将 warmup 中的 epoch 设置为 5，即先用 5epoch 将学习率从 0 增加到最大值，再去做相应的学习率衰减。

3.batch_size 的选择

batch_size 是训练神经网络中的一个重要的超参数，该值决定了一次将多少数据送入神经网络参与训练。在论文[1]中，作者通过实验发现，当 batch_size 的值与学习率的值呈线性关系时，收敛精度几乎不受影响。在训练 ImageNet 数据时，大部分的神经网络选择的初始学习率为 0.1，batch_size 是 256，所以根据实际的模型大小和显存情况，可以将学习率设置为 0.1*k,batch_size 设置为 256*k。

4.weight_decay 的选择

过拟合是机器学习中常见的一个名词，简单理解即为模型在训练数据上表现很好，但在测试数据上表现较差，在卷积神经网络中，同样存在过拟合的问题，为了避免过拟合，很多正则方式被提出，其中，weight_decay 是其中一个广泛使用的避免过拟合的方式。Weight_decay 等价于在最终的损失函数后添加 L2 正则化，L2 正则化使得网络的权重倾向于选择更小的值，最终整个网络中的参数值更趋向于 0，模型的泛化性能相应提高。在各大深度学习框架的实现中，该值表达的含义是 L2 正则前的系数，在 paddle 框架中，该值的名称是 l2_decay，所以以下都称其为 l2_decay。该系数越大，表示加入的正则越强，模型越趋于欠拟合状态。在训练 ImageNet 的任务中，大多数的网络将该参数值设置为 1e-4，在一些小的网络如 MobileNet 系列网络中，为了避免网络欠拟合，该值设置为 1e-5~4e-5 之间。当然，该值的设置也和具体的数据集有关系，当任务的数据集较大时，网络本身趋向于欠拟合状态，可以将该值适当减小，当任务的数据集较小时，网络本身趋向于过拟合状态，可以将该值适当增大。下表展示了 MobileNetV1_x0_25 在 ImageNet-1k 上使用不同 l2_decay 的精度情况。由于 MobileNetV1_x0_25 是一个比较小的网络，所以 l2_decay 过大会使网络趋向于欠拟合状态，所以在该网络中，相对 1e-4，3e-5 是更好的选择。

模型	L2_decay	Train acc1/acc5	Test acc1/acc5
MobileNetV1_x0_25	1e-4	43.79%/67.61%	50.41%/74.70%
MobileNetV1_x0_25	3e-5	47.38%/70.83%	51.45%/75.45%

另外，该值的设置也和训练过程中是否使用其他正则化有关系。如果训练过程中的数据预处理比较复杂，相当于训练任务变的更难，可以将该值适当减小，下表展示了在 ImageNet-1k 上，ResNet50 在使用 randaugment 预处理方式后使用不同 l2_decay 的精度。容易观察到，在任务变难后，使用更小的 l2_decay 有助于模型精度的提升。

模型	L2_decay	Train acc1/acc5	Test acc1/acc5
ResNet50	1e-4	75.13%/90.42%	77.65%/93.79%
ResNet50	7e-5	75.56%/90.55%	78.04%/93.74%

综上所述，l2_decay 可以根据具体的任务和模型去做相应的调整，通常简单的任务或者较大的模型，推荐使用较大的 l2_decay,复杂的任务或者较小的模型，推荐使用较小的 l2_decay。

5.label_smoothing 的选择

Label_smoothing 是深度学习中的一种正则化方法，其全称是 Label Smoothing Regularization(LSR)，即标签平滑正则化。在传统的分类任务计算损失函数时，是将真实的 one hot 标签与神经网络的输出做相应的交叉熵计算，而 label_smoothing 是将真实的 one hot 标签做一个标签平滑的处理，使得网络学习的标签不再是一个 hard label，而是一个有概率值的 soft label，其中在类别对应的位置的概率最大，其他位置概率是一个非常小的数。具体的计算方式参见论文[2]。在 label_smoothing 里，有一个 epsilon 的参数值，该值描述了将标签软化的程度，该值越大，经过 label smoothing 后的标签向量的标签概率值越小，标签越平滑，反之，标签越趋向于 hard label，在训练 ImageNet-1k 的实验里通常将该值设置为 0.1。 在训练 ImageNet-1k 的实验中，我们发现，ResNet50 大小级别及其以上的模型在使用 label_smooting 后，精度有稳定的提升。下表展示了 ResNet50_vd 在使用 label_smoothing 前后的精度指标。

模型	Use_label_smoothing	Test acc1
ResNet50_vd	0	77.9%
ResNet50_vd	1	78.4%

同时，由于 label_smoohing 相当于一种正则方式，在相对较小的模型上，精度提升不明显甚至会有所下降，下表展示了 ResNet18 在 ImageNet-1k 上使用 label_smoothing 前后的精度指标。可以明显看到，在使用 label_smoothing 后，精度有所下降。

模型	Use_label_smoohing	Train acc1/acc5	Test acc1/acc5
ResNet18	0	69.81%/87.70%	70.98%/89.92%
ResNet18	1	68.00%/86.56%	70.81%/89.89%

综上所述，较大的模型使用 label_smoohing 可以有效提升模型的精度，较小的模型使用 label_smoohing 可能会降低模型的精度，所以在决定是否使用 label_smoohing 前，需要评估模型的大小和任务的难易程度。

6.针对小模型更改图片的 crop 面积与拉伸变换程度

在 ImageNet-1k 数据的标准预处理中，random_crop 函数中定义了 scale 和 ratio 两个值，两个值分别确定了图片 crop 的大小和图片的拉伸程度，其中 scale 的默认取值范围是 0.08-1(lower_scale-upper_scale),ratio 的默认取值范围是 3/4-4/3(lower_ratio-upper_ratio)。在非常小的网络训练中，此类数据增强会使得网络欠拟合，导致精度有所下降。为了提升网络的精度，可以使其数据增强变的更弱，即增大图片的 crop 区域或者减弱图片的拉伸变换程度。我们可以分别通过增大 lower_scale 的值或缩小 lower_ratio 与 upper_scale 的差距来实现更弱的图片变换。下表列出了使用不同 lower_scale 训练 MobileNetV2_x0_25 的精度，可以看到，增大图片的 crop 区域面积后训练精度和验证精度均有提升。

模型	Scale 取值范围	Train_acc1/acc5	Test_acc1/acc5
MobileNetV2_x0_25	[0.08,1]	50.36%/72.98%	52.35%/75.65%

模型	Scale 取值范围	Train_acc1/acc5	Test_acc1/acc5
MobileNetV2_x0_25	[0.2,1]	54.39%/77.08%	53.18%/76.14%

7.使用数据增广方式提升精度

一般来说，数据集的规模对性能影响至关重要，但是图片的标注往往比较昂贵，所以有标注的图片数量往往比较稀少，在这种情况下，数据的增广尤为重要。在训练 ImageNet-1k 的标准数据增广中，主要使用了 random_crop 与 random_flip 两种数据增广方式，然而，近些年，越来越多的数据增广方式被提出，如 cutout、mixup、cutmix、AutoAugment 等。实验表明，这些数据的增广方式可以有效提升模型的精度，下表列出了 ResNet50 在 8 种不同的数据增广方式的表现，可以看出，相比 baseline，所有的数据增广方式均有收益，其中 cutmix 是目前最有效的数据增广。更多数据增广的介绍请参考数据增广章节。

模型	数据增广方式	Test top-1
ResNet50	标准变换	77.31%
ResNet50	Auto-Augment	77.95%
ResNet50	Mixup	78.28%
ResNet50	Cutmix	78.39%
ResNet50	Cutout	78.01%
ResNet50	Gridmask	77.85%
ResNet50	Random-Augment	77.70%
ResNet50	Random-Erasing	77.91%
ResNet50	Hide-and-Seek	77.43%

8. 通过 train_acc 和 test_acc 确定调优策略

在训练网络的过程中，通常会打印每一个 epoch 的训练集准确率和验证集准确率，二者刻画了该模型在两个数据集上的表现。通常来说，训练集的准确率比验证集准确率微高或者二者相当是比较不错的状态。如果发现训练集的准确率

比验证集高很多，说明在这个任务上已经过拟合，需要在训练过程中加入更多的正则，如增大 l2_decay 的值，加入更多的数据增广策略，加入 label_smoothing 策略等；如果发现训练集的准确率比验证集低一些，说明在这个任务上可能欠拟合，需要在训练过程中减弱正则效果，如减小 l2_decay 的值，减少数据增广方式，增大图片 crop 区域面积，减弱图片拉伸变换，去除 label_smoothing 等。

9.通过已有的预训练模型提升自己的数据集的精度

在现阶段计算机视觉领域中，加载预训练模型来训练自己的任务已成为普遍的做法，相比从随机初始化开始训练，加载预训练模型往往可以提升特定任务的精度。一般来说，业界广泛使用的预训练模型是通过训练 128 万张图片 1000 类的 ImageNet-1k 数据集得到的，该预训练模型的 fc 层权重是是一个 k*1000 的矩阵，其中 k 是 fc 层以前的神经元数，在加载预训练权重时，无需加载 fc 层的权重。在学习率方面，如果您的任务训练的数据集特别小（如小于 1 千张），我们建议你使用较小的初始学习率，如 0.001（batch_size:256,下同），以免较大的学习率破坏预训练权重。如果您的训练数据集规模相对较大（大于 10 万），我们建议你尝试更大的初始学习率，如 0.01 或者更大。