

尝试了多种加载预训练模型的方法，找到一种效果比较明显的一种。

```
net = models.resnet50(pretrained=False)
model_weight_path = "./resnet50-19c8e357.pth"
net.load_state_dict(torch.load(model_weight_path))
```

Tips:因为之前没有有意识的保存，所以前面的几次实验的图都是被覆盖了但是记录了一些关键的结果，所以可能有些实验有图有些没有图。

调参前期: 恒定学习率

Experiment_1

Parameter	Learning rate	Weight decay	Momentum	Batch size
	0.001	0.00005	Null	16
Result	上升速度缓慢，20-40 代上升缓慢到到过最高 81%，之后也是维持 81%到 81.5%左右，维持稳定的阶段震荡比较明显			
Think	Lr=1e-3 容易引起震荡，且训练后期应该降低学习率。然后上网去搜找到了可以通过增加动量的方法来加快学习同时也可以让让 sgd 避开奇点的可能性加大，提高准确率的可能。			

Experiment_2

Parameter	Learning rate	Weight decay	Momentum	Batch size
	0.001	0.00005	0.9	16
Result	准确率没有很明确的提升，最后稳定大概在 83.2%左右			
Think	可能 Momentum 不是关键因素可能。关键因素还是学习率的问题，实验 3 就探索一下各个学习率对结果的影响			

Experiment_3

Parameter	Learning rate	Weight decay	Momentum	Batch size
	[2e-3,1e-3,5e-4,1e-4,5e-5,1e-5]	0.00005	0.9	16
Result	那天刚好有事，为了方便，在每一个 epoch 大迭代的基础上再加一层循环，每层循环设置不同的学习率，然后就出去了，没看实验过程，然后因为我忽略了每次大循环的时候要清空梯度，导致只有第一次 2e-3 学习率的实验结果是有效的，后面几个学习率曲线都被干扰了，基本就浪费了 24 个小时了。其中 2e-3 的实验结果跟 0.001 的趋势大致相同，但是，前 20 代的上升速度明显提升，但是最后稳定的准确率却降低为 81.8%。			
Think	在每次训练前，记得清空梯度。。。学习率应该在 1e-3 左右一点点能取得比较高的准确率			

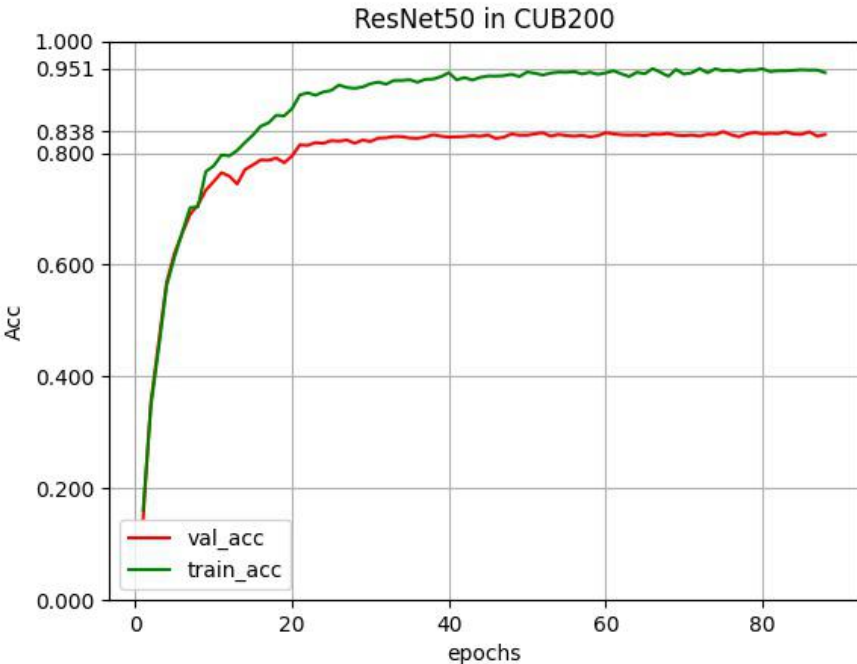
Experiment_4

Parameter	Learning rate					Weight decay	Momentum	Batch size
	epoch	0-19	20-29	30~		0.00005	0.9	16
	lr	2e-3	1e-3	5e-4				
Result	0-19 代:上升速率跟实验 2 稍微快一点, 且最高能达到 80% 20-29 代:缓慢上升到 83.2% 30 代以后: 维持在 83.5%左右							
Think	目前为止收敛效果不错的实验基本结果就是 83.5%, 或许可以跳出学习率, 观察一下其他参数。							

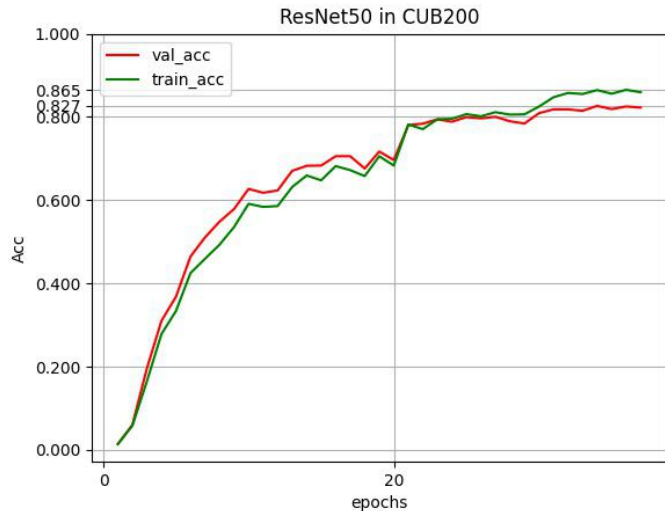
调参后期: 修改 batch size

因为 `wd` 不会直接影响准确率,主要作用是防止过拟合,因此就剩下 `batch size`。调整 `batch size` 有两个方向,增大 `bs` 可以增大训练效率,但是准确率可能减小,减小 `bs` 可能可以增大准确率,速度减慢。关键就是要找到最合适的 `bs`。上面已经跑了 `bs=16` 了,下面在相同参数的情况下爱再跑 `bs=4,bs=8,bs=32` 三种情况,考虑到 `bs=32` 需要用两块 `gpu` 内存进行分布式训练才够,但是另一块在跑着其他实验,所以用 `bs=24` 代替。

Experiment_1

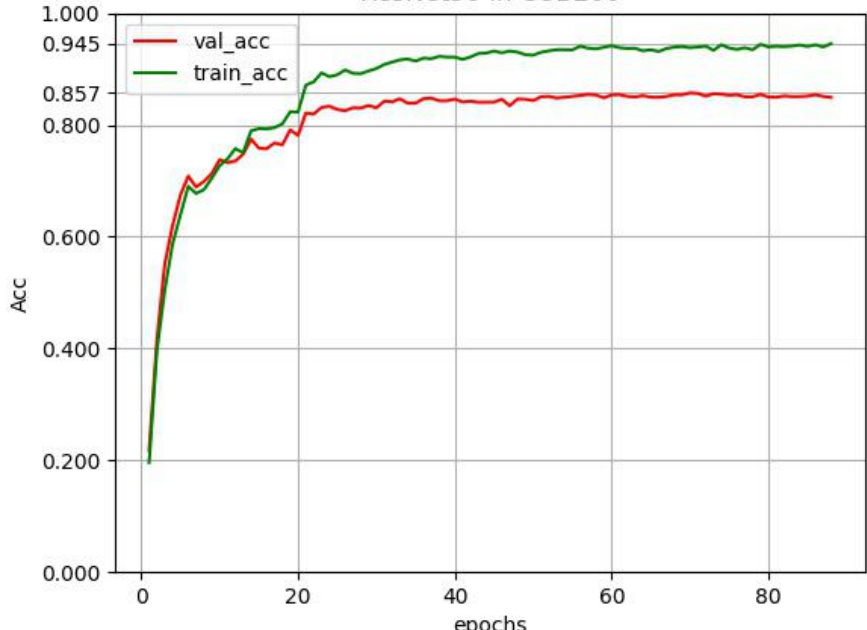
Parameter	Learning rate					Weight decay	Momentum	Batch size
	epoch	0-19	20-29	30~49	50~	0.00005	0.9	24
	lr	2e-3	1e-3	5e-4	1e-4			
Result	 <p>ResNet50 in CUB200</p> <p>Acc</p> <p>epochs</p> <p>val_acc</p> <p>train_acc</p>							
Think	收敛速度可以，但是最后准确率只有 83.8%							

Experiment_2

Parameter	Learning rate					Weight decay	Momentum	Batch size																																				
	epoch	0-19	20-29	30~49	50~	0.00005	0.9	4																																				
	lr	2e-3	1e-3	5e-4	1e-4																																							
Result	<div><p>ResNet50 in CUB200</p><table border="1"><caption>Approximate data points from the ResNet50 in CUB200 graph</caption><thead><tr><th>Epochs</th><th>train_acc</th><th>val_acc</th></tr></thead><tbody><tr><td>0</td><td>0.000</td><td>0.000</td></tr><tr><td>5</td><td>0.400</td><td>0.450</td></tr><tr><td>10</td><td>0.550</td><td>0.600</td></tr><tr><td>15</td><td>0.650</td><td>0.680</td></tr><tr><td>20</td><td>0.750</td><td>0.780</td></tr><tr><td>25</td><td>0.800</td><td>0.800</td></tr><tr><td>30</td><td>0.820</td><td>0.810</td></tr><tr><td>35</td><td>0.840</td><td>0.820</td></tr><tr><td>40</td><td>0.850</td><td>0.827</td></tr><tr><td>45</td><td>0.850</td><td>0.827</td></tr><tr><td>50</td><td>0.850</td><td>0.827</td></tr></tbody></table></div>								Epochs	train_acc	val_acc	0	0.000	0.000	5	0.400	0.450	10	0.550	0.600	15	0.650	0.680	20	0.750	0.780	25	0.800	0.800	30	0.820	0.810	35	0.840	0.820	40	0.850	0.827	45	0.850	0.827	50	0.850	0.827
Epochs	train_acc	val_acc																																										
0	0.000	0.000																																										
5	0.400	0.450																																										
10	0.550	0.600																																										
15	0.650	0.680																																										
20	0.750	0.780																																										
25	0.800	0.800																																										
30	0.820	0.810																																										
35	0.840	0.820																																										
40	0.850	0.827																																										
45	0.850	0.827																																										
50	0.850	0.827																																										
Think	Batch size 为 4 的情况下，收敛速度明显降低，且最后的准确率只有 82.7%左右																																											

Experiment_3

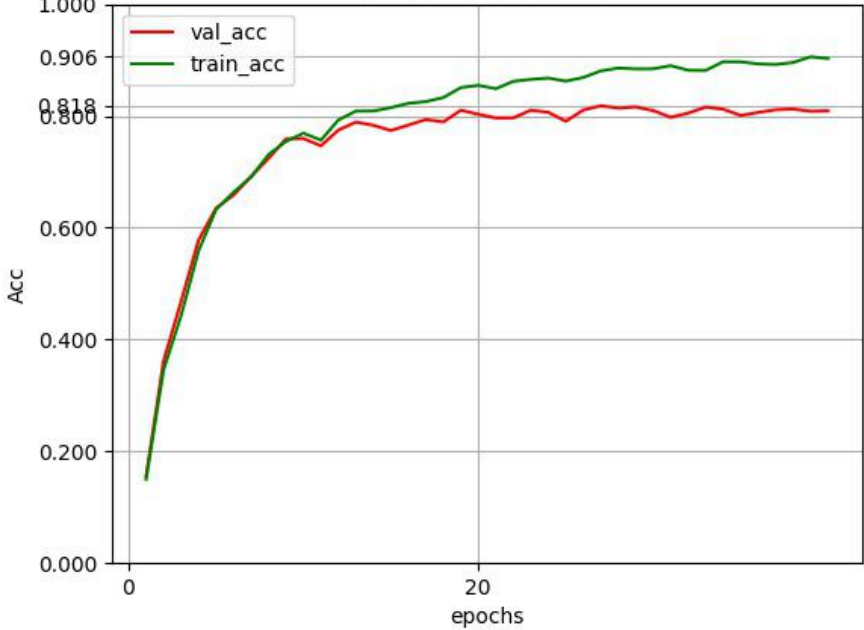
Parameter	Learning rate					Weight decay	Momentum	Batch size
	epoch	0-19	20-29	30-49	50~			
	lr	2e-3	1e-3	5e-4	1e-4			
						0.00005	0.9	8

Result	<div><p>ResNet50 in CUB200</p><table><caption>Approximate data points from the ResNet50 in CUB200 graph</caption><tr><th>Epochs</th><th>train_acc</th><th>val_acc</th></tr><tr><td>0</td><td>0.200</td><td>0.200</td></tr><tr><td>10</td><td>0.800</td><td>0.750</td></tr><tr><td>20</td><td>0.850</td><td>0.800</td></tr><tr><td>40</td><td>0.945</td><td>0.857</td></tr><tr><td>60</td><td>0.945</td><td>0.857</td></tr><tr><td>80</td><td>0.945</td><td>0.857</td></tr><tr><td>90</td><td>0.945</td><td>0.857</td></tr></table></div>	Epochs	train_acc	val_acc	0	0.200	0.200	10	0.800	0.750	20	0.850	0.800	40	0.945	0.857	60	0.945	0.857	80	0.945	0.857	90	0.945	0.857
Epochs	train_acc	val_acc																							
0	0.200	0.200																							
10	0.800	0.750																							
20	0.850	0.800																							
40	0.945	0.857																							
60	0.945	0.857																							
80	0.945	0.857																							
90	0.945	0.857																							
Think	Batch size 为 8 的情况下，收敛速度良好，且最后的准确率达到 了 85.7%																								

实验四用来对比不进行学习率衰减且其他参数相同的情况下进行训练进行训练。

Experiment_4

Parameter	Learning rate					Weight decay	Momentum	Batch size
	epoch	0-19	20-29	30~				
	lr	2e-3	1e-3	5e-4				
						0.00005	0.9	8

<p>Result</p>	<p style="text-align: center;">ResNet50 in CUB200</p>  <p>Batch size 为 8 的情况下，如果用恒定的 $lr=1e-3$ 去学习，准确率最后稳定在 81.8 左右。</p>
<p>Think</p>	<p>Batch size 为 8 的情况下，如果用恒定的 $lr=1e-3$ 去学习，准确率最后稳定在 81.8 左右。</p>

自我小结:

1. 首先得确定大概的学习率范围，然后这个范围左边一点右边一点试探。
2. 确定了大概的学习率范围之后，通过观察恒定学习率的曲线来设置学习率衰减变化。
3. 不同的数据量对应不同合适的 batch size，常用的 4,8,16,32,64 可以多试。
4. Momentum 可以一开始就用进去，可以加大学习速率，帮助更快找到学习率范围。
5. 记得每次做实验都要记录实验结果图。