Tensorflow 框架搭建 vgg 网络实现对 CIFAR-10 的训练

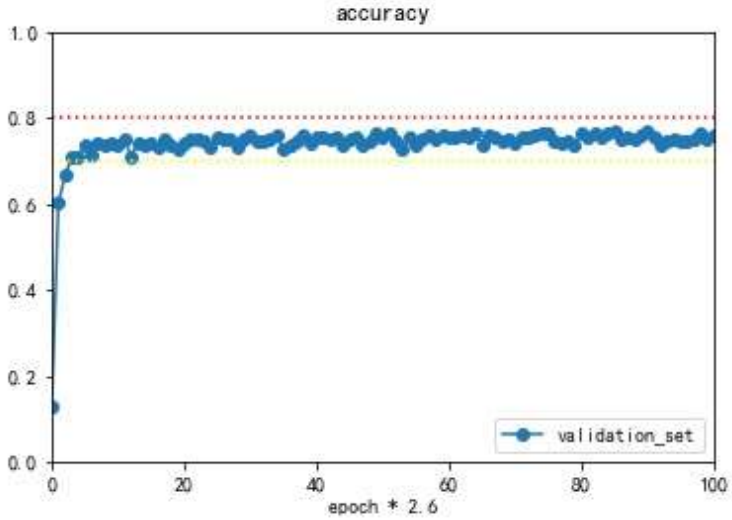
摘要

本次实验基于 Tensorflow 框架，并借助了华为云 ModelArts 的 GPU（1*p100）进行训练，构建的 Lenet 网络准确率可达 74.7%，VGG 网络准确率可达 84.5%。

实验过程

1. Lenet 网络（经过了调整，因为有 GPU，我提高了通道数）

LeNet Configuration			
	Activation shape	Activation Size	# parameters
Input:	(32, 32, 3)	3,072	0
CONV1 (f=5, s=1)	(32, 32, 32)	32,768	832
POOL1	(16, 16, 32)	8,192	0
CONV2 (f=5, s=1)	(16, 16, 64)	16,384	1664
POOL2	(8, 8, 64)	4096	0
FC3	(512, 1)	512	2,097,153
FC4	(128, 1)	128	65,536
Softmax	(10, 1)	10	1281



图表 1：LeNet 网络验证集准确率

```
After 160000 iterations, test_acc: 0.738
After 162000 iterations, loss: 0.0124437, val-acc: 0.752, time spent: 15 min, 41 sec
After 164000 iterations, loss: 0.0405536, val-acc: 0.763, time spent: 15 min, 53 sec
After 166000 iterations, loss: 0.0155908, val-acc: 0.752, time spent: 16 min, 4 sec
After 168000 iterations, loss: 0.0131255, val-acc: 0.762, time spent: 16 min, 15 sec
After 170000 iterations, loss: 0.0115154, val-acc: 0.766, time spent: 16 min, 27 sec
After 170000 iterations, test_acc: 0.744
After 172000 iterations, loss: 0.013643, val-acc: 0.748, time spent: 16 min, 38 sec
After 174000 iterations, loss: 0.0193276, val-acc: 0.754, time spent: 16 min, 50 sec
After 176000 iterations, loss: 0.0156511, val-acc: 0.75, time spent: 17 min, 2 sec
After 178000 iterations, loss: 0.0112459, val-acc: 0.759, time spent: 17 min, 13 sec

After 180000 iterations, loss: 0.0152442, val-acc: 0.768, time spent: 17 min, 25 sec
After 180000 iterations, test_acc: 0.743
After 182000 iterations, loss: 0.0120351, val-acc: 0.756, time spent: 17 min, 36 sec
After 184000 iterations, loss: 0.0285467, val-acc: 0.738, time spent: 17 min, 48 sec
After 186000 iterations, loss: 0.0117467, val-acc: 0.744, time spent: 18 min, 0 sec
After 188000 iterations, loss: 0.0116366, val-acc: 0.749, time spent: 18 min, 11 sec
After 190000 iterations, loss: 0.012156, val-acc: 0.746, time spent: 18 min, 23 sec
After 190000 iterations, test_acc: 0.747
After 192000 iterations, loss: 0.0113215, val-acc: 0.744, time spent: 18 min, 35 sec
After 194000 iterations, loss: 0.0113481, val-acc: 0.75, time spent: 18 min, 46 sec
After 196000 iterations, loss: 0.0226298, val-acc: 0.765, time spent: 18 min, 58 sec
After 198000 iterations, loss: 0.0113382, val-acc: 0.75, time spent: 19 min, 9 sec
After 200000 iterations, loss: 0.0128412, val-acc: 0.759, time spent: 19 min, 21 sec
After 200000 iterations, test_acc: 0.743
```

图表 2：LeNet 网络准确率记录

由于网络结构基本没有变化，相关参数我基本保持了两周前不使用 Tensorflow 以及 GPU 情况下的参数不变（稍调低了学习率），效果还可以，相比于使用 CPU 的准确率（68%）有了明显提高（74.7%）

2. VGG 网络

紧接着，我基于 VGG 网络结构进一步加深网络。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

图表 3：VGG 网络结构

VGG 网络的最佳准确率（结合效率）是 16 层的网络，但此网络是用于分类 224*224*3 的图片，与 CIFAR-10 的 32*32*3 图片有很大差异，因此，我去掉了最后的几层卷积（我也尝试过用与论文完全一样的网络训练过，准确率并不是最高的，因为最后图像从 32*32 被压缩成 2*2 的矩阵的确很不合理）。

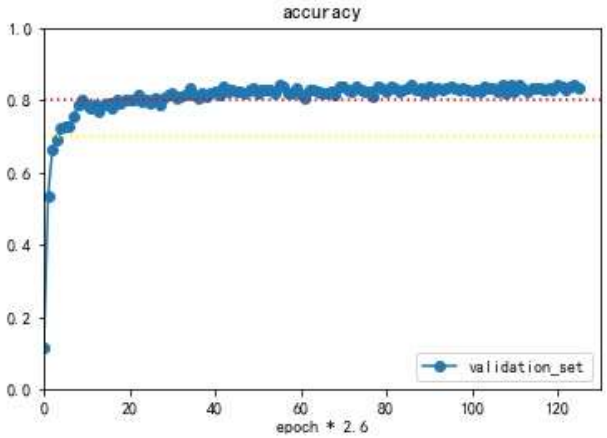
当我基于 Lenet 网络调整为 VGG 网络后，原先可以迅速收敛的 loss 不再收敛，且调整学习率不能解决问题。接着我将网络还原成 Lenet 网络，然后一层一层重新增加网络，发现是初始化参数的问题，原先网络的参数 W 在初始化时标准差设置为 0.01 非常有效，但增加了全连接层后需要稍稍提高一些才能拟合（提高到 0.05 后 loss=nan）

ConvNet Configuration		
A		B
10 weight layers		10 weight layers
input (32 x 32 x 3 RGB image)		
conv3-32 (32 x 32 x 32)		conv3-64 (32 x 32 x 64)
conv3-32 (32 x 32 x 32)		conv3-64 (32 x 32 x 64)
(16 x 16 x 32)	maxpool	(16 x 16 x 64)
conv3-64 (16 x 16 x 64)		conv3-128 (16 x 16 x 128)
conv3-64 (16 x 16 x 64)		conv3-128 (16 x 16 x 128)
conv3-64 (16 x 16 x 64)		conv3-128 (16 x 16 x 128)
(8 x 8 x 64)	maxpool	(8 x 8 x 128)
conv3-128 (8 x 8 x 128)		conv3-256 (8 x 8 x 256)
conv3-128 (8 x 8 x 128)		conv3-256 (8 x 8 x 256)
conv3-256 (8 x 8 x 256)		conv3-512 (8 x 8 x 512)
(4 x 4 x 256)	maxpool	(4 x 4 x 512)
FC-1024		
FC-128		
softmax		

经过调整的 11 层 VGG 网络结构

主要参数	数值
batch_size	64
learning_rate, learning_rate_decay	7e-3, 0.995
regularization_rate	1e-4
iteration	200001

开始阶段我使用了较少的通道数（32-256），因为相较于论文中的结构，228*228*3 的图片在第一次卷积就构造了 64 个通道，但 CIFAR10 的图片很小，应该不需要过多通道。



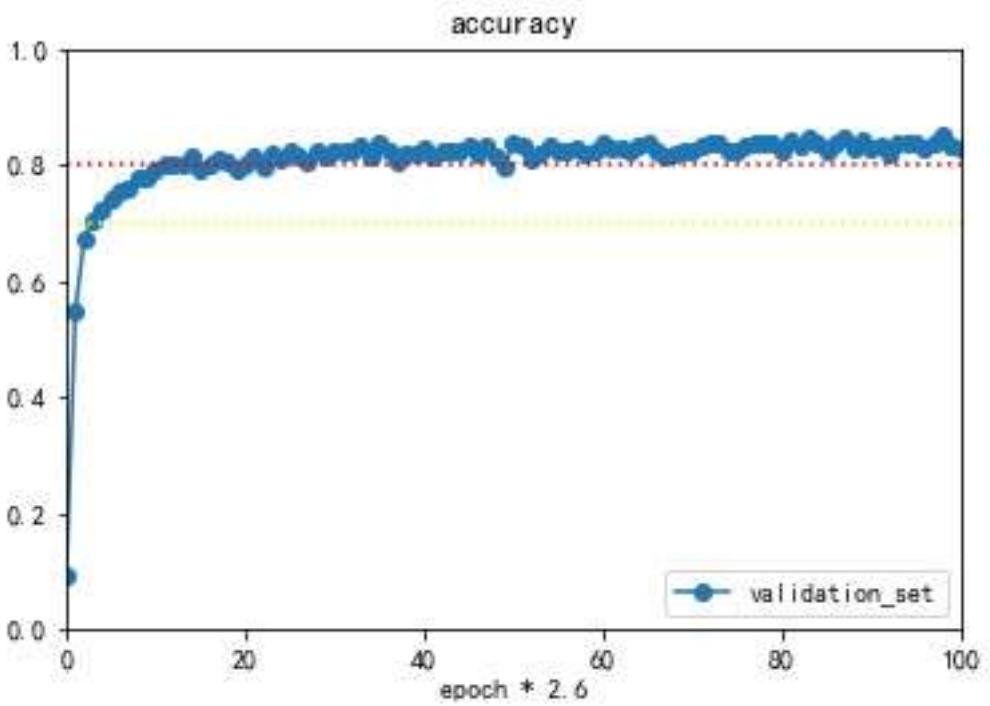
图表 4：通道数 32-256 准确率曲线

```
After 246000 iterations, loss: 0.133209, val-acc: 0.821, time spent: 33 min, 47 sec
After 248000 iterations, loss: 0.132692, val-acc: 0.837, time spent: 34 min, 3 sec
After 250000 iterations, loss: 0.132656, val-acc: 0.829, time spent: 34 min, 20 sec
After 250000 iterations, test_acc: 0.835
After 252000 iterations, loss: 0.133261, val-acc: 0.821, time spent: 34 min, 36 sec
After 254000 iterations, loss: 0.132668, val-acc: 0.82, time spent: 34 min, 52 sec
After 256000 iterations, loss: 0.13331, val-acc: 0.827, time spent: 35 min, 9 sec
After 258000 iterations, loss: 0.132399, val-acc: 0.825, time spent: 35 min, 25 sec
After 260000 iterations, loss: 0.132373, val-acc: 0.83, time spent: 35 min, 42 sec
After 260000 iterations, test_acc: 0.828
After 262000 iterations, loss: 0.132379, val-acc: 0.815, time spent: 35 min, 58 sec
After 264000 iterations, loss: 0.132231, val-acc: 0.825, time spent: 36 min, 15 sec
After 266000 iterations, loss: 0.132227, val-acc: 0.82, time spent: 36 min, 31 sec
After 268000 iterations, loss: 0.132306, val-acc: 0.819, time spent: 36 min, 47 sec
After 270000 iterations, loss: 0.132078, val-acc: 0.839, time spent: 37 min, 3 sec
After 270000 iterations, test_acc: 0.823
After 272000 iterations, loss: 0.132127, val-acc: 0.821, time spent: 37 min, 20 sec
After 274000 iterations, loss: 0.131992, val-acc: 0.832, time spent: 37 min, 37 sec
After 276000 iterations, loss: 0.13198, val-acc: 0.824, time spent: 37 min, 53 sec
After 278000 iterations, loss: 0.132077, val-acc: 0.832, time spent: 38 min, 10 sec
After 280000 iterations, loss: 0.131837, val-acc: 0.835, time spent: 38 min, 26 sec
After 280000 iterations, test_acc: 0.823
After 282000 iterations, loss: 0.131822, val-acc: 0.826, time spent: 38 min, 43 sec
After 284000 iterations, loss: 0.131748, val-acc: 0.824, time spent: 38 min, 59 sec
After 286000 iterations, loss: 0.132547, val-acc: 0.823, time spent: 39 min, 15 sec
After 288000 iterations, loss: 0.131803, val-acc: 0.818, time spent: 39 min, 31 sec
After 290000 iterations, loss: 0.131619, val-acc: 0.827, time spent: 39 min, 48 sec
After 290000 iterations, test_acc: 0.834
```

图表 5：通道数 32-256 准确率记录

通道数 32-256 的网络准确率可达 83.4%。

由于计算资源充足，我又尝试了通道数 64-512 的结构，准确率略有上升（84.5%）



图表 6：64-512 通道数准确率曲线

```
After 130000 iterations, test_acc: 0.829
After 132000 iterations, loss: 0.276667, val-acc: 0.828, time spent: 32 min, 52 sec
After 134000 iterations, loss: 0.276737, val-acc: 0.842, time spent: 33 min, 22 sec
After 136000 iterations, loss: 0.276921, val-acc: 0.841, time spent: 33 min, 51 sec
After 138000 iterations, loss: 0.276614, val-acc: 0.83, time spent: 34 min, 21 sec
After 140000 iterations, loss: 0.285351, val-acc: 0.832, time spent: 34 min, 50 sec
After 140000 iterations, test_acc: 0.841
After 142000 iterations, loss: 0.276272, val-acc: 0.841, time spent: 35 min, 20 sec
After 144000 iterations, loss: 0.276078, val-acc: 0.839, time spent: 35 min, 50 sec
After 146000 iterations, loss: 0.276285, val-acc: 0.829, time spent: 36 min, 19 sec
After 148000 iterations, loss: 0.277777, val-acc: 0.839, time spent: 36 min, 49 sec
After 150000 iterations, loss: 0.275848, val-acc: 0.84, time spent: 37 min, 18 sec
After 150000 iterations, test_acc: 0.842
After 152000 iterations, loss: 0.275716, val-acc: 0.837, time spent: 37 min, 48 sec
After 154000 iterations, loss: 0.275645, val-acc: 0.828, time spent: 38 min, 18 sec
After 156000 iterations, loss: 0.275728, val-acc: 0.848, time spent: 38 min, 47 sec
After 158000 iterations, loss: 0.276388, val-acc: 0.827, time spent: 39 min, 17 sec
After 160000 iterations, loss: 0.278631, val-acc: 0.83, time spent: 39 min, 46 sec
After 160000 iterations, test_acc: 0.841
After 162000 iterations, loss: 0.275282, val-acc: 0.835, time spent: 40 min, 16 sec
After 164000 iterations, loss: 0.275366, val-acc: 0.839, time spent: 40 min, 46 sec
After 166000 iterations, loss: 0.275107, val-acc: 0.838, time spent: 41 min, 15 sec
After 168000 iterations, loss: 0.275087, val-acc: 0.839, time spent: 41 min, 45 sec
After 170000 iterations, loss: 0.276837, val-acc: 0.831, time spent: 42 min, 14 sec
After 170000 iterations, test_acc: 0.845
```

图表 7：通道数 64-512 准确率记录

经过近 50 分钟的训练，测试集的准确率可以达到 84.5%。

参考资料

[1] 郑泽宇、梁博文、顾思宇《Tensorflow 实战 Google 深度学习框架》第二版

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR, 2015.