

机器学习 cifar-10 上机报告

邵智轩

1400012141

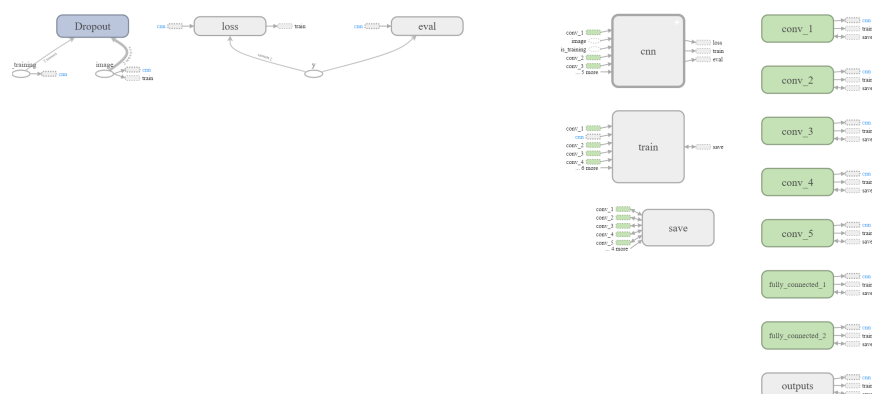
物理学院

目标

实现对 `cifar-10` 数据集的图片分类

网络结构

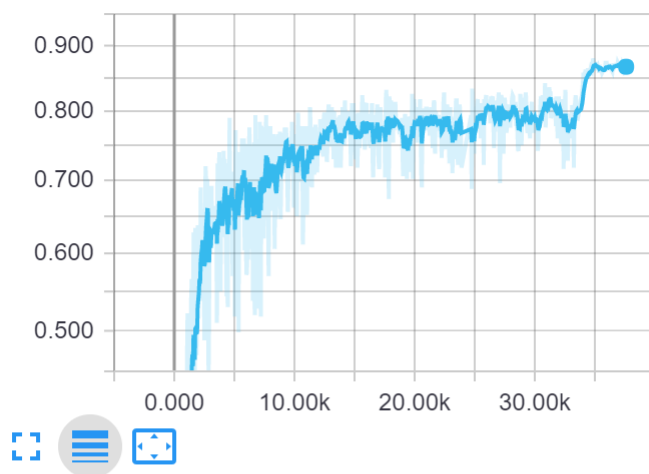
- 总的结构: 5 层 CNN+3 层 fully-connected
- activation function: leaky relu
- loss: cross entropy
- optimizer: AdamOptimizer
- regularization: no
- normalization: batch_norm
- pooling: after conv1, conv2 and conv5
- dropout: keep_prob=0.6
- batch_size=100
- weight initializer: xavier_initializer()



learning rate

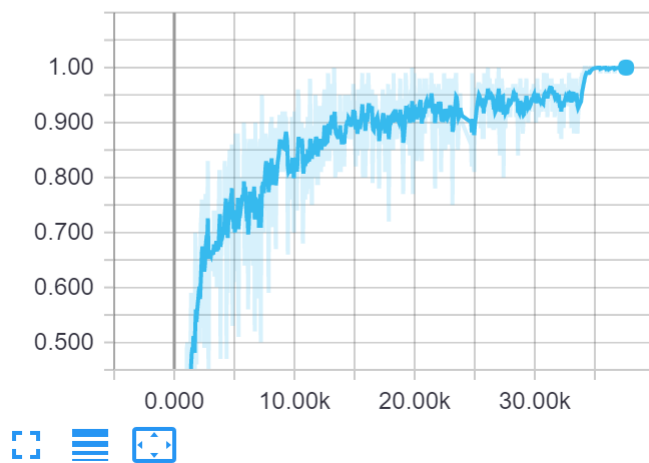
由于采用的是自适应的 Optimizer，一定程度上减少了手动调节 learning rate 的需求。我的 learning rate 一开始设在 0.005；当 training accuracy 开始在 0.6 附近震荡时，下调到 0.001；当 training accuracy 在 0.9 附近震荡时，下调到 1e-4；最后 training accuracy 基本接近于 1，下调到 1e-5。

Test_Accuracy



Train_Accuracy

Train_Accuracy



结果分析

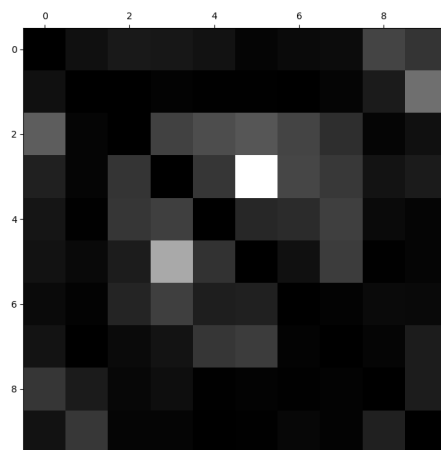
1. 正确率

a) training accuracy: ~ 1.0

b) test accuracy: ~ 0.87

2. 混淆矩阵下面显示了用模型对 `test_data` 做预测的混淆矩阵。

	飞机	轿车	鸟	猫	鹿	狗	蛙	马	船	卡车
飞机	871	9	14	13	10	3	6	7	38	29
轿车	9	907	0	2	1	1	0	3	15	62
鸟	52	3	741	37	43	48	38	26	3	9
猫	18	3	29	681	30	142	39	32	11	15
鹿	12	1	30	35	832	22	24	35	6	3
狗	10	5	16	94	28	800	9	34	1	3
蛙	6	2	21	35	17	18	888	2	6	5
马	11	0	6	11	30	34	2	887	3	16
船	30	15	4	8	1	2	1	2	921	16
卡车	10	31	3	3	0	1	4	2	18	928



可以看到两个最亮的点是 (3,5) 和 (5,3)，也就是说猫和狗的混淆是最严重的。

其他尝试

l_1 or l_2 regularization

由于有了 batch-norm 层, 本身已经减少了对 regularization 的需求。我自己的试验结果是, 无论加入 l_1 还是 l_2 的 regularization, 都会使模型收敛速度下降, 而且最后的 accuracy 反而不如不加。

Data Augmentation

我尝试过 data augmentation, 利用 tensorflow 中以一系列函数如 `tf.image.random_flip_left_right()` 可以方便的实现。但我最终发现“data augmentation”对现有的 test accuracy 的提升并不显著 (不超过 1%), 而且图片预处理的过程会显著地降低训练速度。我认为, 没有显著提升的原因可能是对 input 的 dropout 多少已经起到了相似的泛化作用。