人工智能导论大作业 (2)

猫狗识别

March 1, 2018

目录

1 概要

对于猫狗识别这个二分类问题，我选择用 CNN 来解决。

本文将写出我在使用不同 CNN 模型得到的数据和结果。

我的最优成果是：抽取所给图片的后 20% 来作为测试集（猫和狗分别 10%），其余为训练集，在不做什么厉害的数据预处理的情况下，测试集有 80% 左右的正确率。

2 各种模型

由于这几天我调参训练的模型实在太多，而且没有很好的记录，所以在这里只说一下我调这些模型的一些参数，就不具体展开每个模型了：

调节的参数有：

1. conv 层数。最终我选择了 3 个 conv 层。
2. conv 层中的 filter 大小和步长还有层数。最终我选择了 3×3，步长为 1，且用 0 补边界，每个 conv 层的特征层分别为 32, 32, 64。
3. max pooling 的大小和步长。最终我选择了 2×2，步长为 2。
4. fc 层个数和大小。最终我选择了 1 个 fc 层，且大小为 128。
5. 图像的处理。一开始我是直接拉伸、压缩到相同大小，但这样感觉会损失一些结构信息，所以我选择同比例放大缩小然后用 0 填充。
6. 初始学习率。最终选择的初始学习率为 1e-4。
7. batch 的大小。最终使用的大小为 32。

基本相同的参数有：

1. 都使用 ReLU 作为激活函数。
2. 在 fc 层后 dropout（训练时概率为 0.5）。
3. 都使用 softmax 的交叉熵来作为 loss。
4. 都使用 tf 中的 AdamOptimizer。
5. 图片大小统一缩放成 200×200，且只使用给定的数据集，且未做过数据增强。

3 最终模型

在自己调参陷入一脸懵逼的情形下，我最终还是用了别人的模型 [7]（其实之前自己调出来的和这个也差不多，但我已经分不清是哪份代码了...）

200 × 200 的 RGB 图片 → conv1-ReLU（3 × 3, 32 个特征图）→ max-pooling（2 × 2, 步长 2）→ conv2-ReLU（3 × 3, 32 个特征图）→ max-pooling（2 × 2, 步长 2）→ conv3-ReLU（3 × 3, 64 个特征图）→ dense1（128 个神经元）→ softmax（2 个输出）

相关图像（Accuracy、Loss、ROC）如下（蓝线为测试集、橙线为训练集，不包括 ROC 图）：

图 1 - 最终模型 Accuracy（Smoothing=0.945）

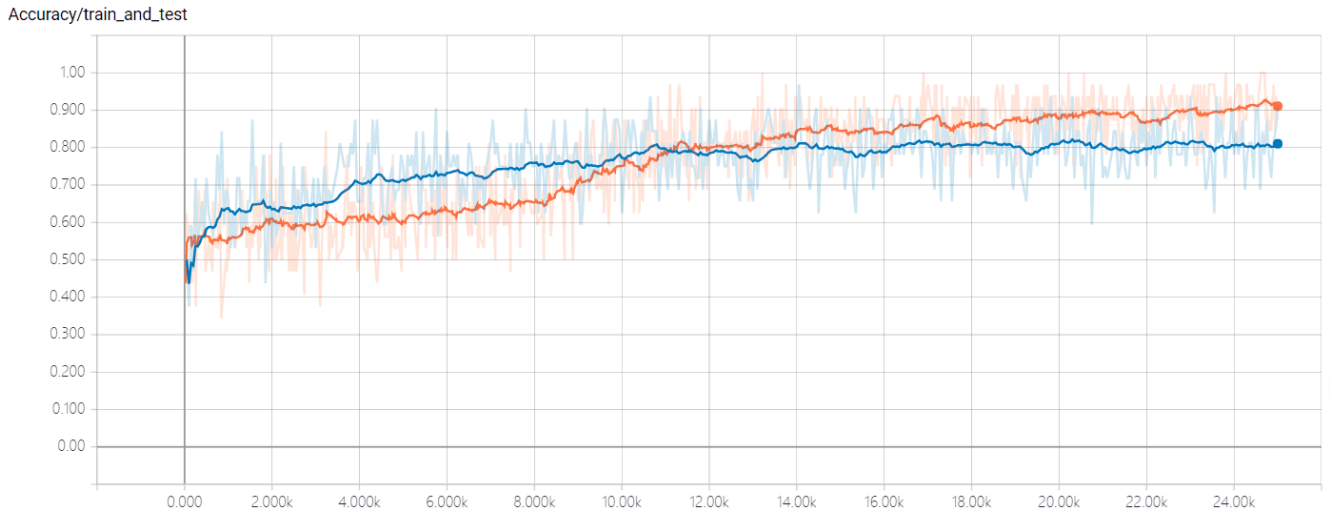
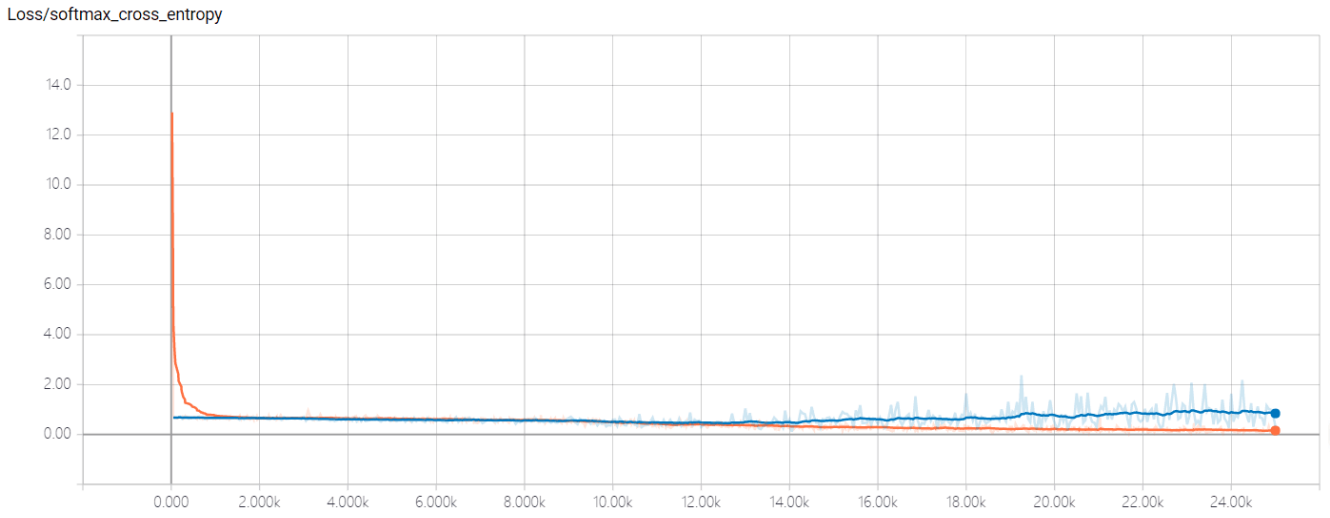


图 2 - 最终模型 Loss（Smoothing=0.945）



可以看出，大概在第 1 万批次后，测试集的 loss 开始波动变大，训练集的准确率逼近 1 了，说明已经开始过拟合了。

我们将第一个卷积层得到的 32 个特征图转换成灰度图（如图4）：

解释？讲道理我无法解释这种特征图... 感觉好像是加强了线条。

图 3 - 最终模型 ROC 曲线

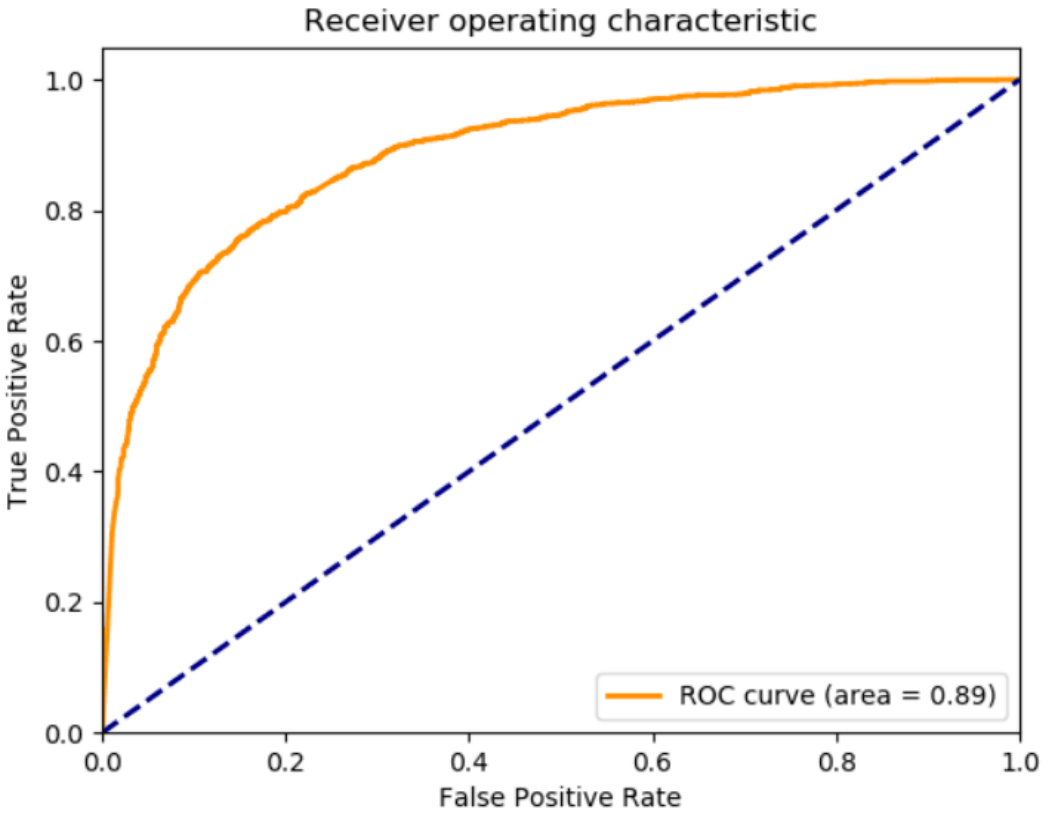
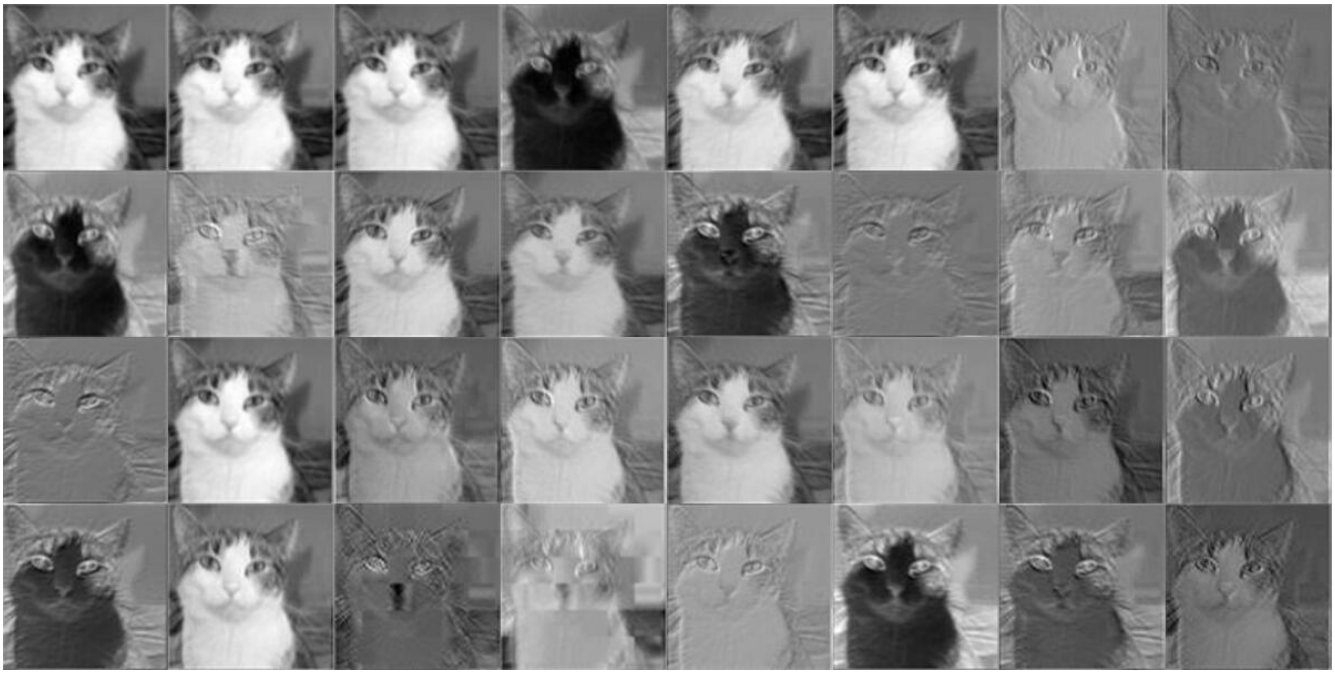


图 4 - conv1 提取出的 32 个特征图（原图为 cat.9967.jpg）



4 十倍交叉验证

一个模型都要花 5 个小时来训练了.... 十个....

我的代码是很容易修改参数（只需要修改几个数字）来得到这十个模型的... 但由于实在没有时间来训出这十个模型... 因此这一步我还是放弃吧...

5 优化

在搜索了相关资料和询问了一些人后，我认为可以这样做来提高正确率：

1. 使用更复杂的模型，比如 AlexNet、VGG 等，但硬件开销很大。
2. 数据增强，将一张图片做一些变换（剪裁、旋转、加入噪声等）得到更多图片。一些现有的库有 imgaug 等。
3. ~~换战略核显卡？~~

6 最后

项目主页：https://github.com/xalanq/rgzndl

协议: LGPL v3.0

6.1 运行环境

- Python 3
- pip 安装 tensorflow、tensorflow-gpu、pillow、scipy

6.2 使用方法

命令行里输入 python judger.py [picture_names] 可判断一张或多张图片（会输出概率和判断结果）。若要简化或使用输出（即对每张图片只输出 0、1 代表猫、狗或代表猫、狗的概率），请使用代码中 judge 函数的返回值。

7 附录

一些调参时得到的各种图片（前方高能，慎入）：

图 5 - 模型二 Accuracy

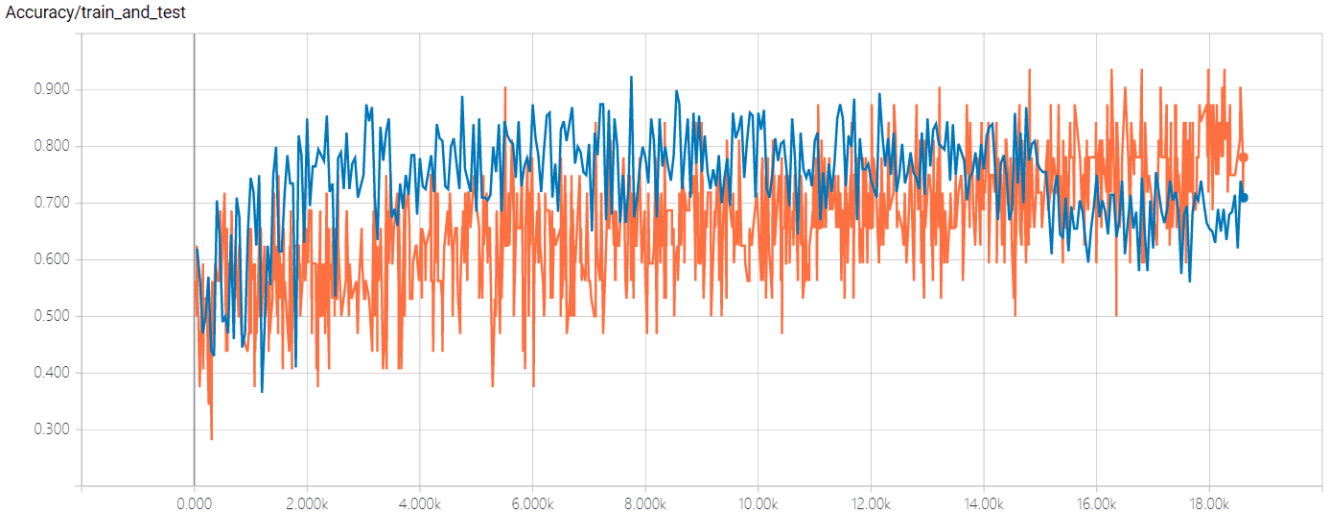


图 6 - 模型二 Loss

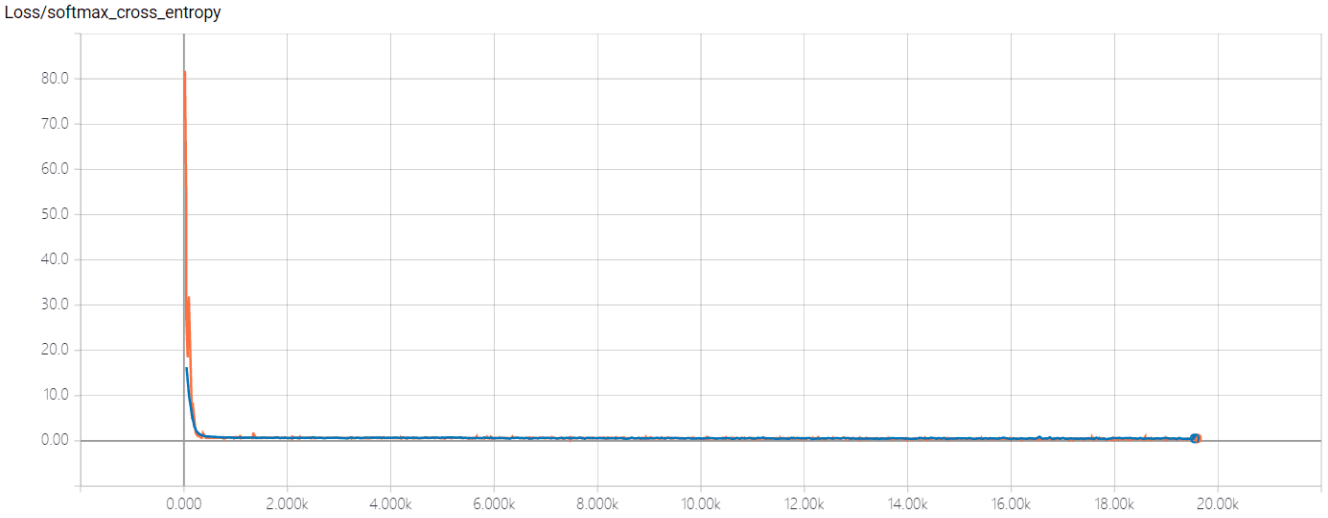


图 7 - 模型三 Accuracy

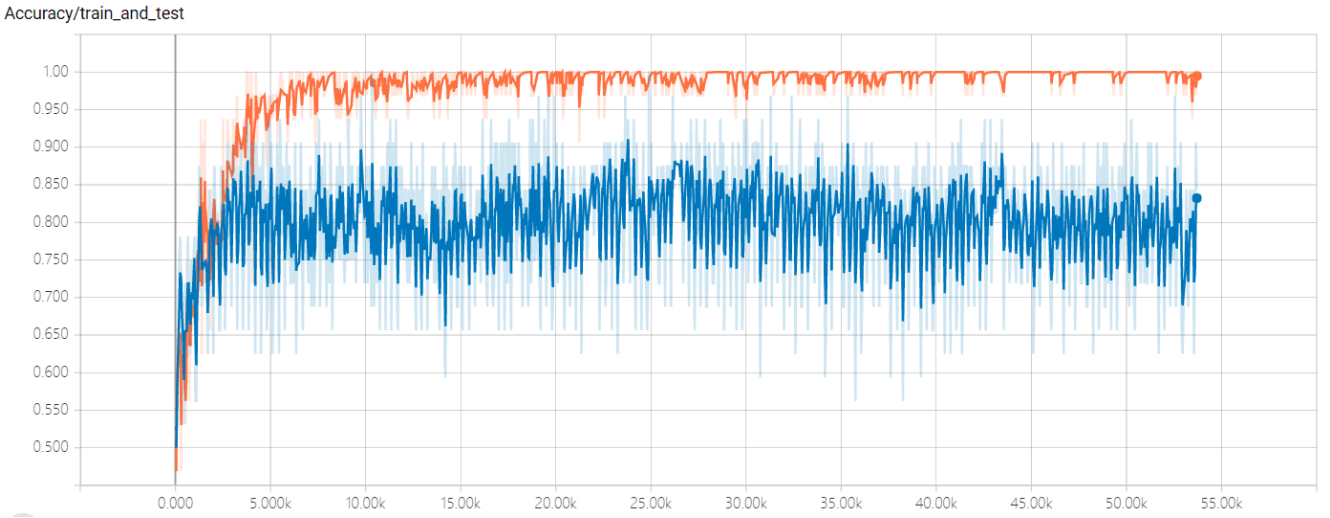


图 8 - 模型三 Loss

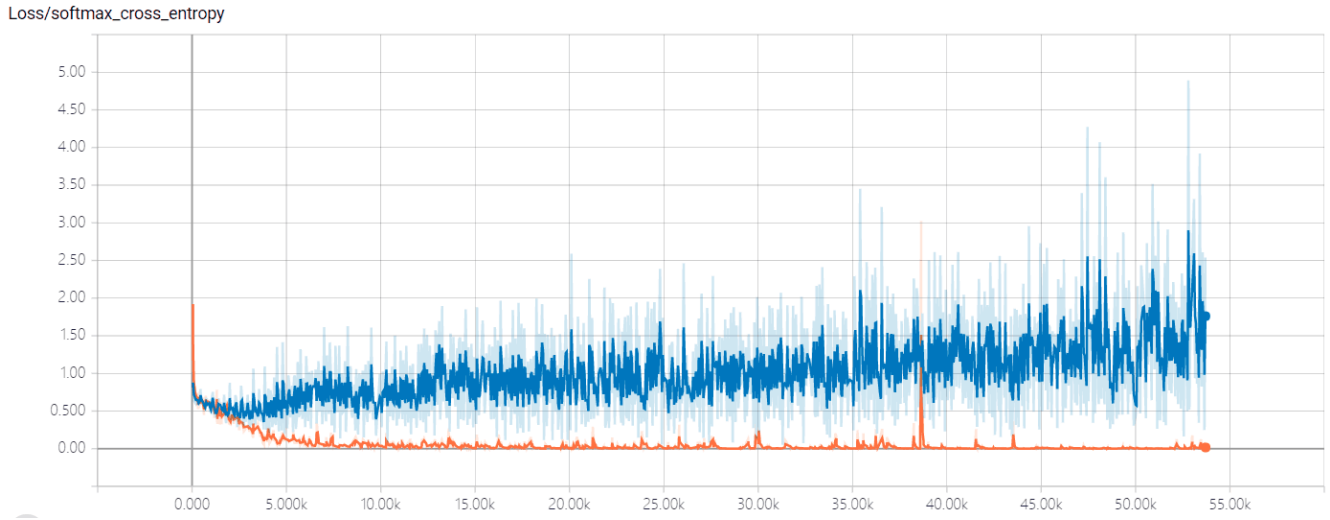


图 9 - 模型四 Accuracy

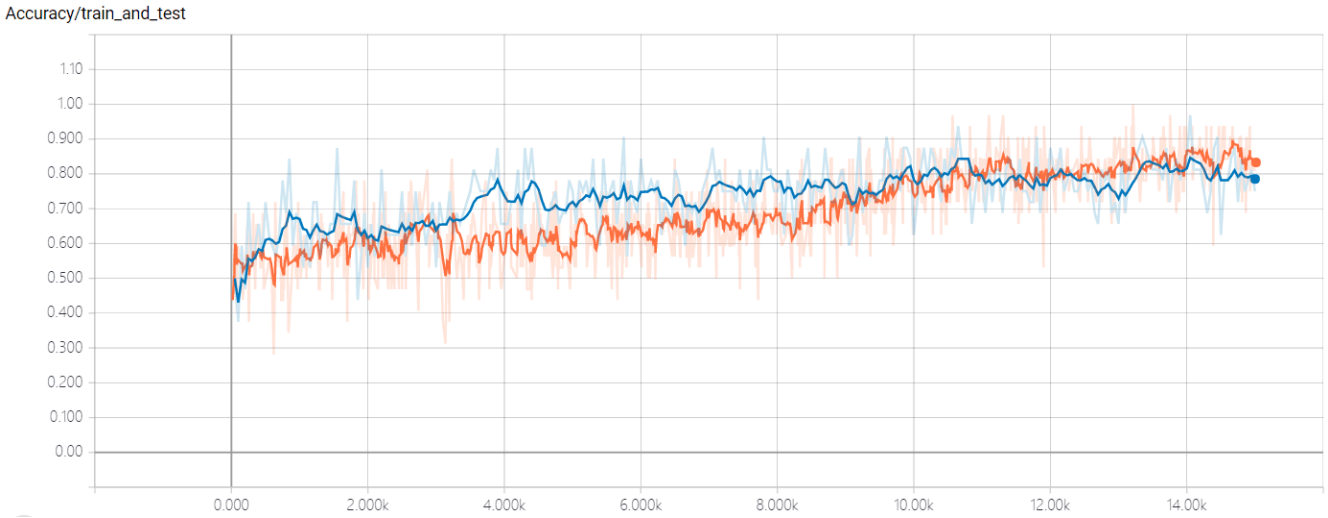
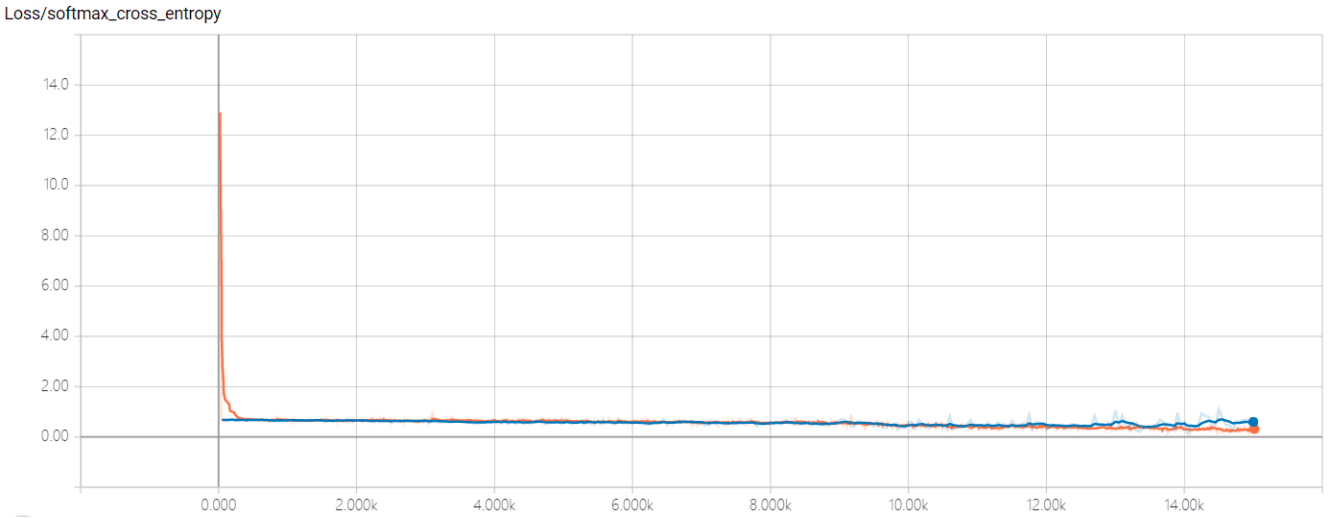


图 10 - 模型四 Loss



参考文献

[1] Stanford University CS231n Lecture 1-6
http://cs231n.stanford.edu/

[2] 卷积神经网络 CS231n 笔记
http://blog.csdn.net/column/details/cs231n.html

[3] TensorFlow 卷积神经网络之猫狗识别
http://blog.csdn.net/u012373815/article/details/78768727

[4] 详解卷积神经网络 (CNN)
http://blog.csdn.net/qq_25762497/article/details/51052861

[5] Deep Learning 回顾之 LeNet、AlexNet、GoogLeNet、VGG、ResNet
https://www.cnblogs.com/52machinelearning/p/5821591.html

[6] Cats and dogs and convolutional neural networks
http://www.subsubroutine.com/sub-subroutine/2016/9/30/cats-and-dogs-and-convolutional-neural-networks

[7] Tensorflow Tutorial 2: image classifier using convolutional neural network
http://cv-tricks.com/tensorflow-tutorial/training-convolutional-neural-network-for-image-classification/

[8] Receiver Operating Characteristic (ROC)
http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html