Project Report for Image Categorization

Chaoyue Wu
Tsinghua University
2017213865
East Annex Building 320, Tsinghua University
wucy17@mails.tsinghua.edu.cn

Yao Luo
Tsinghua University
2017213866
East Annex Building 201, Tsinghua University
luo-y17@mails.tsinghua.edu.cn

Jiecheng Wu
Tsinghua University
2017213851
East Annex Building 320, Tsinghua University
jasonwood2031@gmail.com

摘要

本文是针对图像分类任务的项目报告，介绍了我们对于方法选取的思考、任务整体流程及使用到的相关技术，利用这些方法和技术进行实验并得到最终结论。 实验获得的最好结果是使用 ResNet50，在课程提供的测试数据集上得到了 95.38%的准确率。

关键词

AlexNet ResNet50 Inception-Resnet-v2 数据增广 fine tune

1. 简介

在本次图像分类任务中，训练数据集包含 32000 多张属于 20 个不同类别的图片，其中包括一些噪声数据。我们需要首先识别并去除这些噪声数据，再在清洗后的训练数据集上训练我们的分类模型，并最终用于预测给定的图片所属的类别。

在深度学习出现之前，我们必须借助 SIFT、HoG 等算法提取具有良好区分性的特征，再结合 SVM 等机器学习算法进行图像识别和分类。卷积神经网络提取的特征则可以达到更好的效果，同时它不需要将特征提取和分类训练两个过程分开，而是在训练时就自动提取了最有效的特征。从时间上考虑，传统的机器学习算法如 KNN，在训练时不需要花太多时间，而在测试时则需要花费大量时间进行计算；相反的，卷积神经网络在训练时需要较多的时间，而在测试时则相对耗时较少。此外，通过调研近几年的 ImageNet 竞赛结果，我们发现当前图像分类的主流方法、同时也是效果最好的方法均来自于深度学习模型，如下图所示：

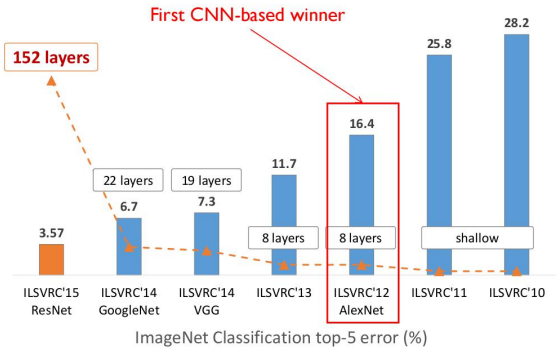

图 1 ImageNet Classification top-5 error (%).

2. 方法

2.1 图像分类任务框架

本次图像分类任务的总体流程如图 2 所示，在获得训练数据集后首先对其进行简单去噪，再进行数据增广，接着利用去噪和增广后的训练数据集在三种卷积神经网络上进行训练，得到三个不同的模型，之后尝试将上述三个模型进行集成，采用多数投票的的方法获得最终的预测类标结果。下面对我们在任务中所使用到的技术进行简单介绍。

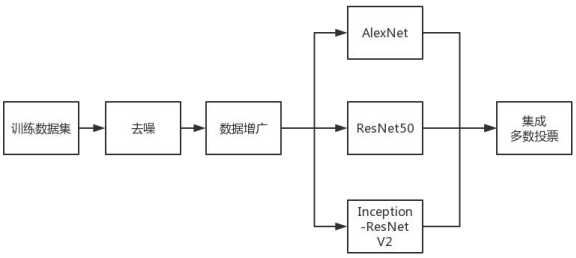

图 2 图像分类任务总体流程图

2.2 训练数据集去噪

通过对原始训练数据集进行观察后我们发现，数据集的噪声主要包括两个方面，一是黑白噪声，属于这种类型的两个典型的噪声数据如下图所示，针对这种噪声数据我们编写了相关程序对训练数据集进行去噪处理，即对类似于图 3 这样的全黑或全白图片进行过滤。

图 3 黑白噪声数据示例

存在于训练数据集中的另外一种噪声是图片类标打错，如图 4 所示，在给定的 label 文件中均分类为飞机。出于时间和精力等方面的考虑，我们并没有对这种类型的噪声数据做系统的处理，但是在最终对模型进行测试的时候，我们会抽取一个将噪声数据清洗后的干净测试集，并利用此测试集来衡量我们模型的准确率。

图 4 类标错误的噪声数据示例

2.3 数据增广

在深度学习中，为了避免出现过拟合，通常需要输入充足的数据量。为了得到更加充足的数据，我们需要对原有的图像数据进行几何变换，改变图像像素的位置并保证特征不变。

通过对训练数据集进行分类统计我们发现，给定的数据集在不同类别上的分布有较大差异，如表 1 所示，类别 15 包含 13268 张图片，而类别 6 仅包含 571 张图片，因此我们采用数据增广的手段，在不改变图像类别的情况下，增加数据量，提高模型的泛化能力。

表 1 训练数据集不同类标上的图片个数

类标	个数	类标	个数
1	797	11	659
2	744	12	1265
3	1106	13	720
4	863	14	715
5	1286	15	13268
6	571	16	1054
7	2425	17	826
8	988	18	753
9	2626	19	614
10	673	20	752

具体来说，我们对图像的增广主要包括下述几个方面[1]：

旋转/反射变换(Rotation/reflection)；翻转变换(flip)；平移变换(shift)；尺度变换(scale)；对比度变换(contrast)；噪声扰动(noise)；颜色变换(color)。

2.4 AlexNet

AlexNet[1]有 6000 万个参数，650,000 个神经元，5 个卷积层和 3 个全连接层（1000 类的 softmax 分类器），采用 ReLU 作为激活函数，使用重叠池化和 dropout 方法来减少过拟合。在本实验中我们基于一个已有的 AlexNet 的 TensorFlow 实现[2]来进行模型的改造和学习。

2.5 ResNet50

ResNet[2]，即深度残差网络，提出 residual block 结构来解决所谓的“退化”问题，即当模型的层次加深时，错误率却提高了。在我们的分类任务中选择的是 ResNet50，并使用 Keras[3]，一个高层神经网络 API 进行实现。

2.6 Inception-Resnet-v2

Inception-Resnet-v2[3]是 ResNet 与 GoogleNet 的结合，在本实验中我们基于一个已有的 TensorFlow 实现[4]来进行模型的改造和学习。

2.7 Fine tune

ImageNet 数据集有 1400 多万幅图片，涵盖 2 万多个类别[5]，关于图像分类、定位、检测等研究工作大多基于此数据集展开。考虑到训练数据集和时间非常有限，如今大多数主流神经网络都提供基于 ImageNet 进行训练得到的预训练模型，为了进一步提高模型的泛化能力并防止过拟合，在本次的图像分类任务中，我们并不是利用给定的训练数据集从头开始训练网络，而是在获得基于 ImageNet 的预训练模型后，固定网络的前几层参数，修改最后的全连接层输出类别，并利用我们的训练数据集逐步向前进行网络权重的微调（即 fine tune）。在 fine tune 的过程中，学习速率、步长和迭代次数都要适当减小。

2.8 集成学习

在我们的前期实验中，基于三个不同的网络获得了三个不同的学习模型，在最后的预测阶段，我们希望将这三个学习模型进行集成，并使用多数投票的方式对给定的图片进行分类。具体来说，给定一张测试图片 x，三个学习模型给出的预测类标分别为 $h_1(x)$，$h_2(x)$，$h_3(x)$，针对这三个结果进行少数服从多数的原则来选择最终类标输出；如果三个模型给出的预测结果均不相同，则选取在验证阶段具有最高准确率的模型给出的类标作为最后的输出。

3. 实验和评估

3.1 实验环境

程序的运行环境如下：

操作系统：Ubuntu 16.04

深度学习平台：tensorflow 1.8.0

GPU：GeForce GTX 1080 Ti/PCIe/SSE2

CPU：Intel® Core™ i7-7700K CPU @ 4.20GHz × 8

3.2 训练数据准备

我们在数据集中随机选出 3000 张图片作为测试集，利用剩余图片进行训练。

3.3 基于 AlexNet 模型的实验

在利用 AlexNet 进行 fine-tune 的过程中，通过对学习速率和 batch size 的各种尝试，我们发现该模型相较其他两个模型能达到的精度上限较低，且在训练的过程出现了 loss 上升的现象，下图展示了使用该网络进行训练的最好情况对应的训练曲线，其中学习率设为 0.00005，batch size 为 120，当 epoch 为 129 时，测试集准确率达到了 79.02%。

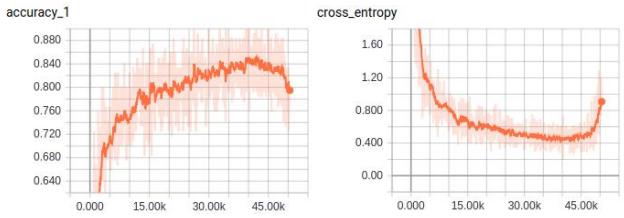

图 5 AlexNet 训练曲线

---

[1] 相关项目参考：

https://absentm.github.io/2016/06/14/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E4%B8%AD%E7%9A%84Data-Augmentation%E6%96%B9%E6%B3%95%E5%92%8C%E4%BB%A3%E7%A0%81%E5%AE%9E%E7%8E%B0/

[2] https://github.com/kratzert/finetune_alexnet_with_tensorflow

[3] https://keras.io/

[4] https://github.com/kwotsin/transfer_learning_tutorial

[5] http://www.image-net.org/

值得注意的是，在训练后期 loss 不降反升，经过查阅相关资料了解到这有可能是是因为学习速率在后期依然较大，相关解决方法包括用指数衰减的学习率或者直接调用 Rmsprop，Adam 等现成的优化算法。但是考虑到 AlexNet 提出较早，以及我们的实验资源和时间有限，因此关于 AlexNet 的实验止步于此。

3.4 基于 ResNet50 模型的实验

在利用 ResNet50 进行 fine-tune 时，我们采用下面三组训练参数进行实验。训练中采用从后往前逐步放开的形式，表中的数字代表我们放开对应层时的 epoch。学习率统一设为 0.0001，动量设为 0.9。

表 2 ResNet50 训练参数设置

	全联接层	Resnet5a（142）	Resnet4a（80）	Resnet3a（38）	全部网络（0）
1	0	5	未放开	未放开	未放开
2	0	5	30	60	未放开
3	0	5	15	30	50

下图是三组参数的训练曲线：橙色线代表参数一，红色线代表参数二，灰色线代表参数三。

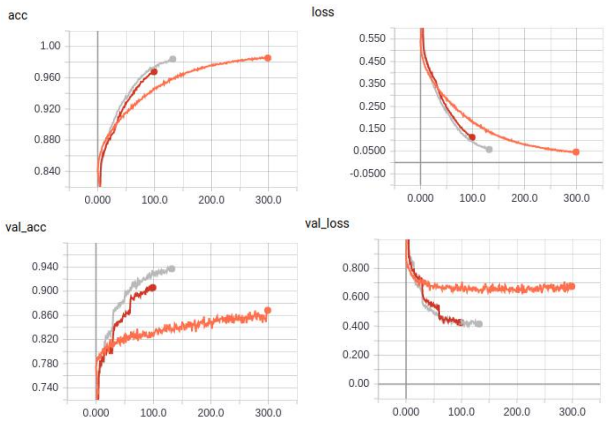

图 6 ResNet50 训练曲线

我们发现随着逐步放开更多的层数，我们在测试集上的准确率会产生显著的提升，同时测试集上 loss 显著下降。当我们只训练全联接层和 Resnet5a 之后的层时，我们发现训练集的准确率提升缓慢；同时在验证集上的 loss 在初始的 30 个 epoch 之后没有显著下降，始终大于 0.6；当训练结束时，测试集上的准确率达到了 98%，但是测试集上的准确率仅达到 85%。我们猜测此时网络倾向于过拟合。

与之相对的，使用参数二和三进行训练时，模型在测试集上的准确率显著提升，达到 90%以上，loss 降到 0.4 左右。我们觉得这是因为本数据集在底层特征方面和 imagenet 不一致的原因，比如 imagenet 中没有人这一身份，所以训练底层特征带来了准确率的上升。同时，利用参数二、三训练需要的时间也大幅度减少，仅用了参数一一半的训练时间。最终，我们利用参数三训练的模型在测试集上的准确率达到了 93.93%，在去除了噪声的测试集上的准确率达到了 94.84%。

3.5 基于 Inception-ReSnet-v2 模型的实验

Inception-ResNet-v2 模型的结构相比于前面所述的两个模型复杂许多，在 3.1 所述的硬件条件下，运行 50 个迭代需要花费近 12 小时。模型规模限制了训练迭代次数不能太多。另外，高于 32 的 batch size 值将导致显卡内存不足，这进一步拉长了训练时间。过少的迭代次数和过小的 batch size 值导致了该模型的准确率较低。

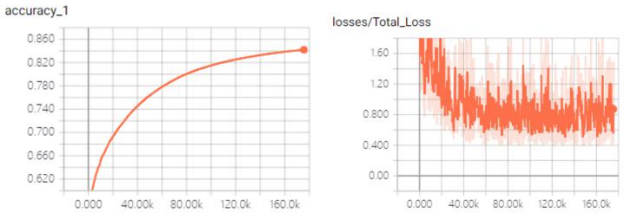

图 7 Inception-ResNet-v2 训练曲线

由于硬件吃紧以及训练时间成本较高，针对本模型的训练只进行了一次，结果如上所示。本次训练的 batch size 被设置为 8，迭代 50 次，耗时近 12 小时，在训练集上的准确率为 84.21%，在验证集的准确率为 89.58%，在测试集上的准确率为 72.08%。

3.6 模型集成

我们考虑利用三个已训练的模型进行集成学习，采用多数投票的方式产出最终的预测结果。然而在利用 2.8 中提出的方法进行模型集成后，在测试集上的准确率为 82.12%，比单独的 ResNet50 准确率要低，我们猜测这是因为 ResNet50 的准确率相较于其他两个要高，导致多数投票会将原来 ResNet50 分对的图片赋予一个错误的类标。考虑到最后要以准确率作为评价标准，因此虽然我们做了模型集成的实验，但是在最终的测试阶段仍然以 ResNet50 模型为准进行分类预测。

4. 结论

在本次图像分类任务中，我们通过对训练数据进行去噪和数据增广来获得扩充后的数据集，并分别在 AlexNet、ResNet50、Inception-ResNet-v2 三个网络上进行训练，采用基于 ImageNet 的预训练模型进行 fine tune，在获得三个训练好的模型后，我们尝试进行模型集成，然而集成效果并不理想。经过实验，我们获得的最好结果是使用 ResNet50 模型，在验证数据集上得到了 94.84%的准确率，在课程提供的测试数据集上得到了 95.38%的准确率。

参考文献

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[3] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//AAAI. 2017, 4: 12.