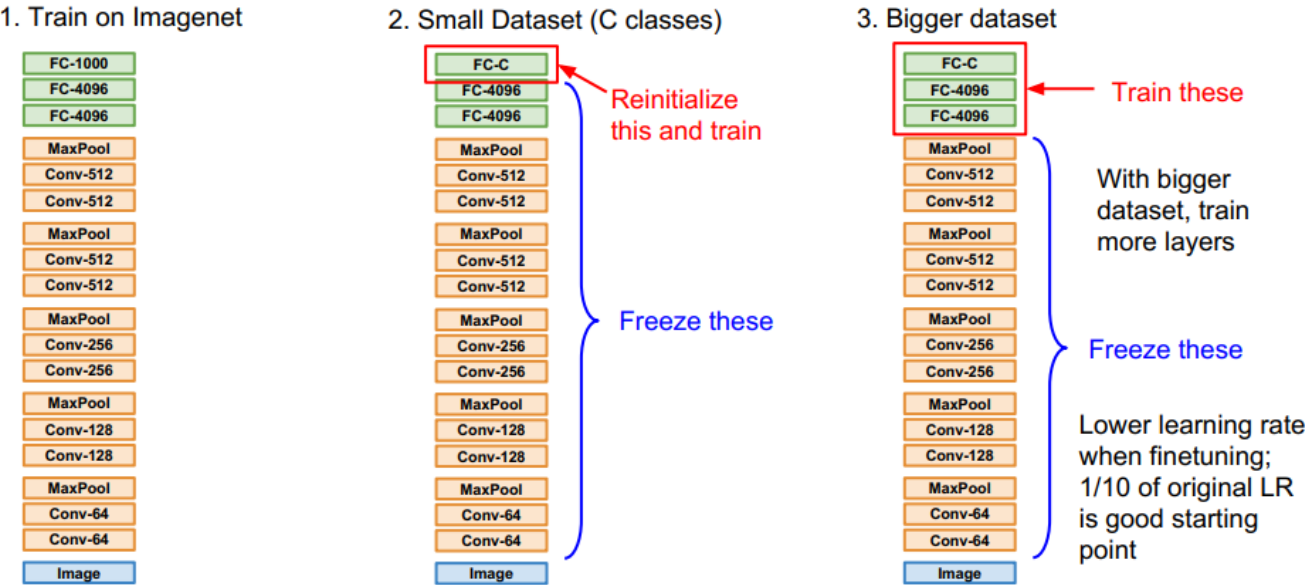


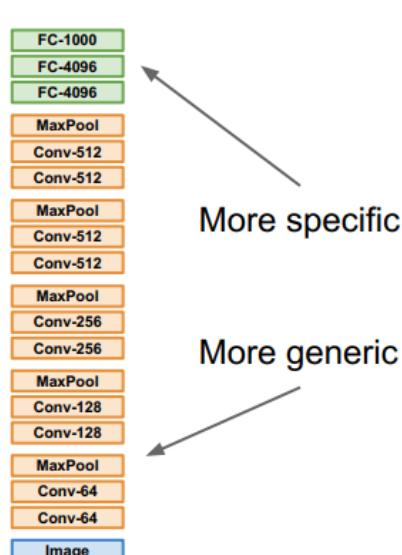
一个简单的介绍，迁移学习的思想，我们看到使用正则化，加入不同正则策略可以帮助减小训练误差和测试误差的间隙。过拟合的一个问题是，有时候过拟合是由于数据不够，你希望得到一个功能大的模型，一个功能强大的网络在你使用小数据集时 很容易过拟合，正则化是一种处理它的方法。

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014



另一种方法是使用迁移学习，迁移学习能够打破这种神话，你不需要超大的样本集，也能训练卷积神经网络。它的思想很简单，你首先找到一些卷积神经网络，这是VGG架构的网络。你首先使用你的卷积神经网络在一个非常大的数据集训练，例如ImageNet，这里你有足够的数据去训练整个网络，现在你想尝试的想法，是把从这个数据集中训练出的提取特征的能力，用到你更感兴趣的小的数据集上，可能不需要像ImageNet分1000类，你可能只关注10个狗的品种分类或者类似的分类，你只需要一个小的数据集，这里我们的数据集只有C个类别，接着你一般的做法是修改，从最后一层的特征到最后的分类输出之间的全连接层，你需要重新随机初始化这部分矩阵，对于ImageNet，它是4096乘以1000维的矩阵，对于你新的分类，矩阵大小变为4096乘以C，例如10或者任何一个数，重新随机初始化最后的矩阵，冻结前面层的权重。现在只需要训练一个线性分类器，只需要训练最后这层，让它在你的数据上收敛，当你只处理一个小的数据集的时候，这会让你的工作很完美。如果你的数据稍微充裕一点，另一件你可以尝试的事情是微调整个网络，在最后一层收敛，在数据集上充分训练之后，你可以试着更新，整个网络的权值，如果你有更多的数据，你可以更新网络的更大一部分，一个通用的策略是，你更新网络时，将学习率调低，因为最初的网络参数可能是在ImageNet上收敛的，泛化能力已经很强了，你只是希望让它们有微小的调整来适应你的数据集。



| | very similar dataset | very different dataset |
|---------------------|------------------------------------|--|
| very little data | Use Linear Classifier on top layer | You're in trouble... Try linear classifier from different stages |
| quite a lot of data | Finetune a few layers | Finetune a larger number of layers |

当你使用迁移学习时，你可以想成是一个2乘以2的情景网络，在一侧的，你可能会有很小的数据集或者很大的数据集，可能你的数据和一些大数据集的图片很相似，例如ImageNet有很多动物，植物之类的图片，如果那只是想分类动物植物或者其他类别的图片就比较好办了，接下来要做的就是，如果你的数据和ImageNet很像，但你的数据量很少，你可以在ImageNet预训练模型的基础上，只训练最后一层线性分类器，如果你有更多的数据，你可以精调你的模型，但是，如果你的数据和ImageNet不相似，情况就不乐观了，例如，你可能处理的是X光或者CT图像时或者一些和ImageNet的图片大相径庭的图片时，可能需要一些创造力，这种方法有时候可能还行得通，但最后一层提取特征可能没有太多信息量，你可以考虑是重新初始化大部分的网络，多做一些实验，在你的数据集数据量比较大的时候，这种情况会多少得到缓解，因为你可以精调大部分网络。

Transfer learning with CNNs is pervasive... (it's the norm, not an exception)

Object Detection (Fast R-CNN)

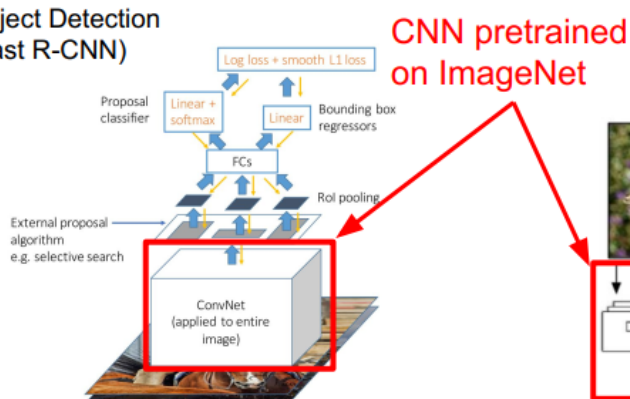
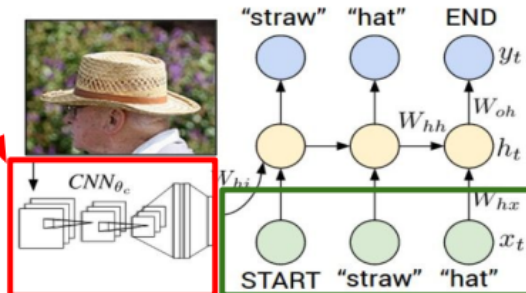


Image Captioning: CNN + RNN



Word vectors pretrained with word2vec

Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

另一点我想说的是转移学习的思想是超级普遍的，非常普遍。如果你读机器视觉的论文，你经常看类似这样的系统图，针对不同的任务，左边是目标检测，右边是图像加标，所有的模型都有一个卷积神经网络处理图像模块。目前无论是计算机视觉的哪个方面应用，大多数人都不会从头训练这些东西，大多数情况下，卷积神经网络在ImageNet上预训练，然后根据任务精调，同样的，在图像加标的环境下，有时候可以预选训练一些和一些语言相关的词向量，你可以在ImageNet上预训练CNN，在一些文本字典上预训练一些词向量，然后，针对你的网络精调，在加标任务中预训练词向量的方法不太常见，也不是很关键。

Takeaway for your projects and beyond:

Have some dataset of interest but it has $< \sim 1\text{M}$ images?

1. Find a very large dataset that has similar data, train a big ConvNet there
2. Transfer learn to your dataset

Deep learning frameworks provide a “Model Zoo” of pretrained models so you don’t need to train your own

Caffe: <https://github.com/BVLC/caffe/wiki/Model-Zoo>

TensorFlow: <https://github.com/tensorflow/models>

PyTorch: <https://github.com/pytorch/vision>

要记住的是，不管是课程的项目还是以后遇到的各种模型，无论什么时候，对于你处理的问题，你没有大数据集，你应该做的事是下载一些相关的预训练的模型，然后，要么重新初始化部分模型或者在你的数据上精调模型，即使你的训练数据有限，这种方法也能行得通。因为这是一个普遍的策略在所有不同的深度学习软件包上都提供了模型库，你可以下载不同模型的预训练版本。

Summary

- Optimization
 - Momentum, RMSProp, Adam, etc
- Regularization
 - Dropout, etc
- Transfer learning
 - Use this for your projects!

总结今天的内容，我们讲了最优化，它是用来改进训练效果的，还讲了正则化，改变在测试集上的性能，集成模型也算其中一种。我们还讲了迁移学习，它可以，让你在小样本的时候训练的一样好，这都是非常有用的策略。你应该在你的工程中应用它们。下一节，我们更具体的讲解一些不同的深度学习软件包。