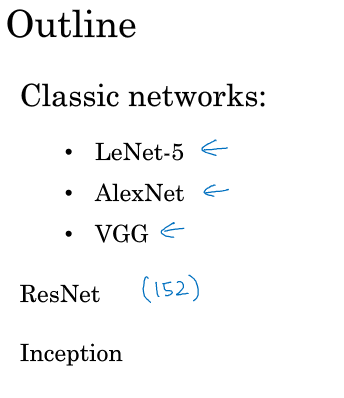
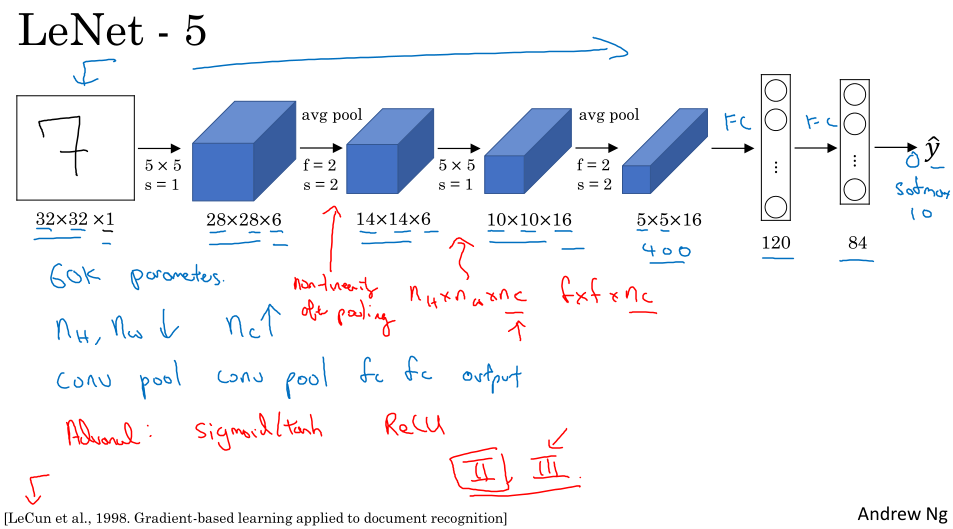
第二章 深度卷积网络

2.1 为什么要进行实例探究

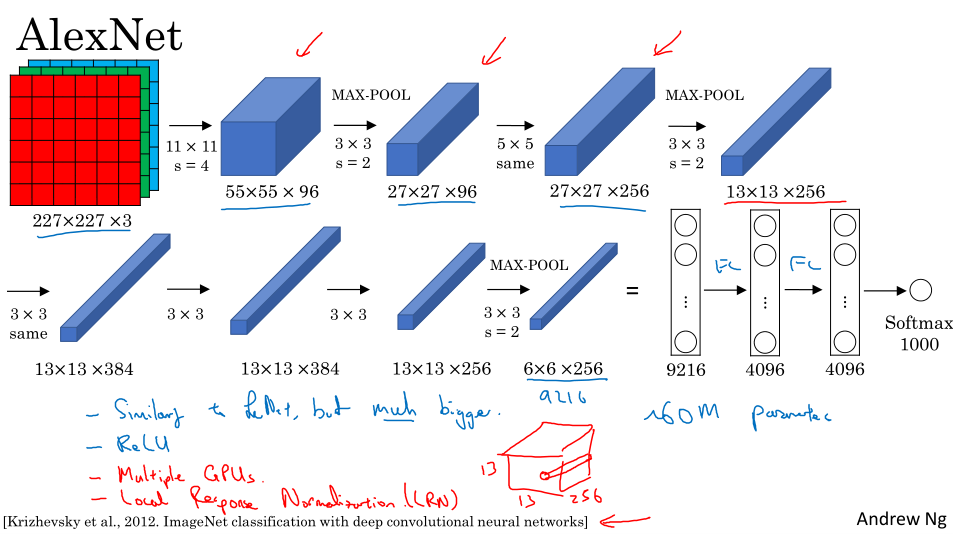


2.2 经典网络

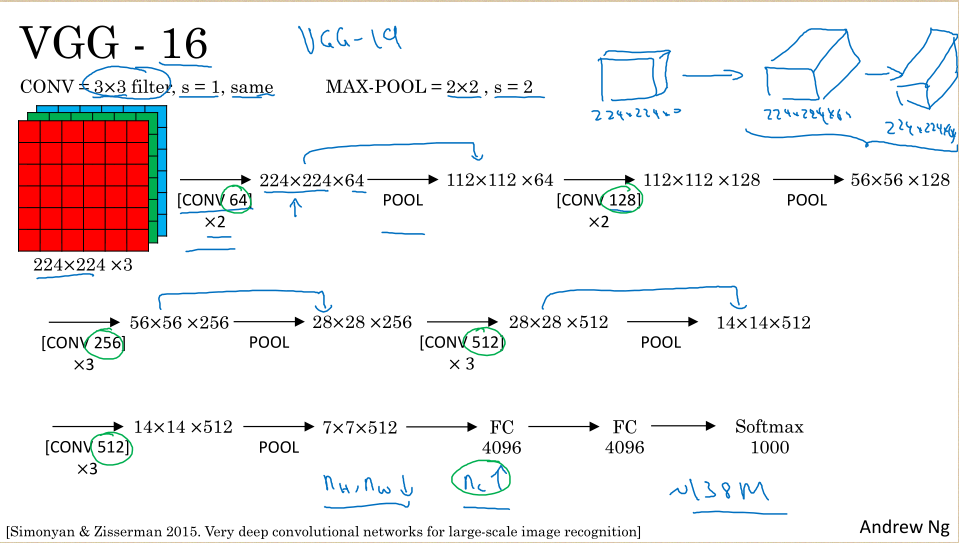
LeNet-5论文中使用平均池比较多，但是构建一个现代卷积神经网络时候，可以使用最大池代替。这个网络中，平均池的大小和步长都是2，所以减少到长度和宽度的一半,基本不使用填充，总是使用有效的卷积。这个网络是一个比较小的神经网络，大约有6000个参数，今天你看到的任何一个网络都比这个大。当时这篇论文使用的是sigmoid和tanh，还没有使用ReLU。



AlexNet，这个神经网络和LeNet有很多相似之处，但它是非常大的。事实上，有很多相似的基本构件块，但是有更多的隐藏单元和训练更多的数据，对数据集进行训练，使数据集具有非常出色的性能。传统的LeNet大约有60000个参数，而AlexNet大约有6000万个参数。这种结构比LeNet使用激活函数的功能更高。另外，那个时候GPU还是有点慢，所以是在两个GPU上训练，这些层实际上被拆分成两个不同的图形，在两个GPU上互相交流。另外原来的AlexNet也有一套Local Response Normalization，就是在13\*13的图像的每个位置，可能不希望有太多的神经元具有非常高的激活值，所以透过通道，将这个位置上的数进行归一化。



VGG-16网络，有16个层，它的特点是，每次卷积的时候都使用了same padding，然后卷积次数可能是2次或者3次，后面接上最大池化层，池化层采用的是f=2，s=2，所以每次经过池化层之后，高度和宽度减半，并且每次在池化之前的过滤器的数量随后都是递增的，从64到128到256到512,。VGG-16是一个比较深的网络。，我认为这种体系结构的相对统一性对研究人员来说颇具吸引力。主要的缺点是，它是一个相当大的网络，根据你必须训练的参数数量。VGG-19是一个更大的版本。

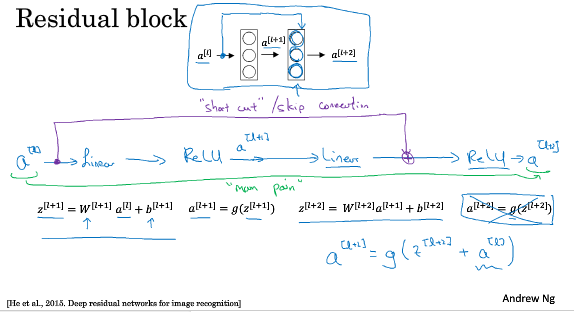


2.3 残差网络

非常非常深的网络是很难训练的，因为存在梯度消失和梯度爆炸的问题，这节课学习跳连接skip connection，可以从某一网络层获取激活，然后迅速反馈给另一层，甚至是神经网络的更深层，可以利用skip connection，构建能够训练深度网络的ResNets，有时候深度能超过100层。

ResNets网络是由残差块Residual block构成的，换句话说，信息从到

需要下列步骤，即这组网络层的主路径，在残差网络中有一点变化，将直接拷贝到神经网络的深层，在ReLU激活函数前输入，“short cut捷径（跳远连接）”。



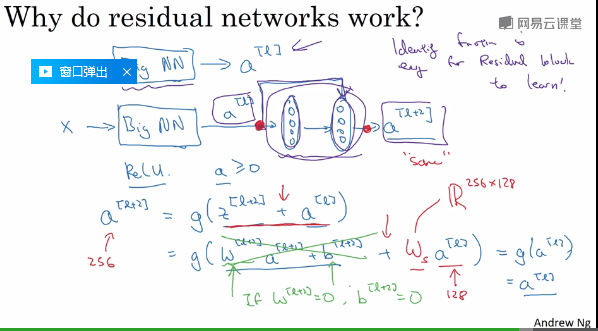
意味着到达的那一层公式发生改变，仍然要对进行处理。也就是说加上的产生了一个残差块。

所以skip connection的意思就是跳过一层或者好多层从而将信息传递到了更深层的地方。ResNets的发明者达县使用残差块能够训练更深的神经网络，所以构建一个ResNet网络就是通过将很多这样的残差块堆积在一起，构成一个神经网络。

Plain network普通神经网络，把普通神经网络变成残差神经网络就是加上很多的跳远连接，例如下图，每两层增加一个连接，构成5个残差块连接在一起，构成ResNets，如果使用标准优化算法训练一个plain network，并且没有多余的残差的话，你会发现，随着网络深度的加深，训练错误会先减少后增多，而理论上，随着深度的加深，应该越来越好，也就是说，理论上，网络深度越深越好，而事实上对于普通网络，深度加深，越难优化，训练错误越多，但有残差块的话，即使网络再深，训练的表现会很好。这种方法有助于解决梯度消失或者梯度爆炸，让我们在训练更深的网络的同时，又能保证良好的性能。

2.4 残差网络为什么对训练深度网络有用

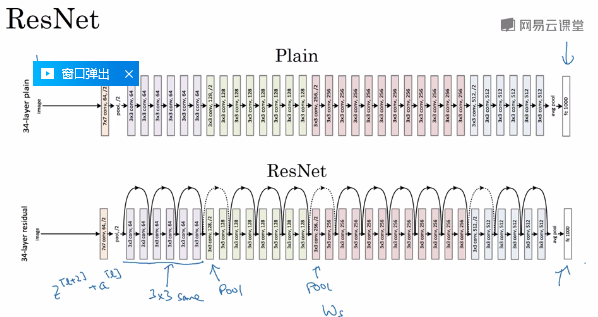
这些残差块学习identify function很容易，当构建更深的神经网络使用了残差块时，,如果使用了权重衰减，那么W参数和b参数假设将近0，，所以即使增加了深度，但是跟普通神经网络相比，同样可以达到好的效果，同时还便于学习，确定网络的性能不会受到影响，有时候甚至可以提高或者至少不减少效率。



另外一点，假如的维度相同，这个残差网络使用了很多的same convolution，所以的维度等于这个输出层的维度，那么就很容易得到输出。假如两者的维度不一样，那么在前面乘以一个矩阵,不需要对这个参数做什么，这是通过学习自己得到的，可以把后面的用0填充到与一样的维度。

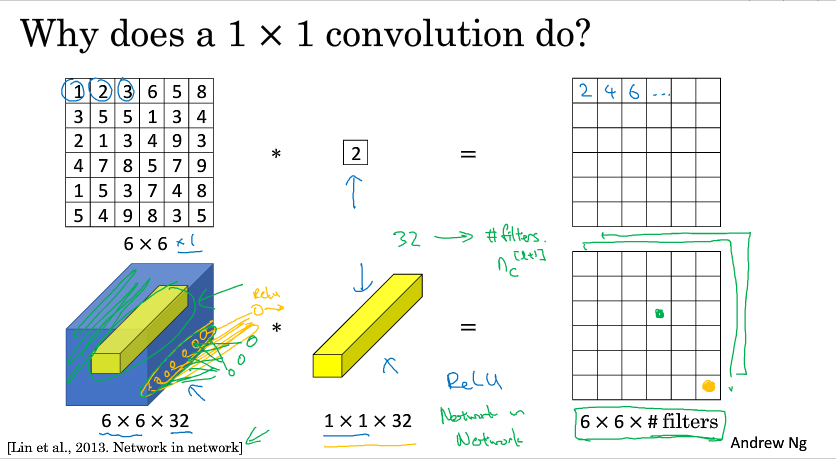
看看ResNets在图像识别上，

这个网络有很多3\*3的卷积，而且大多都是相同的，这就是添加等维特征向量的原因，所以这些卷积层（不是全连接层），因为它们是相等的卷积，维度得以保留。

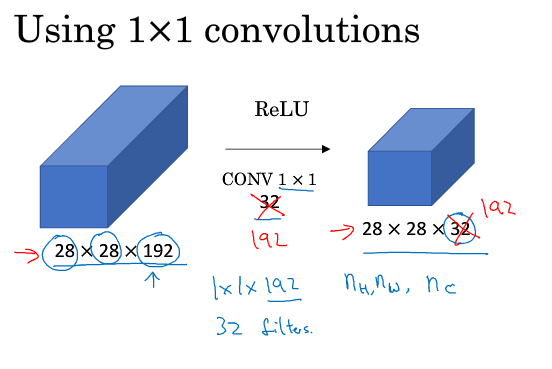


2.5 网络中的网络以及1\*1卷积

Network in network也就是使用1\*1卷积，对于一个灰度图片矩阵，只是平面的，使用1\*1卷积就是相当于给输入矩阵都乘上一个数，看起来作用不大，但对于一个有通道或者说立体的原始输入来说，对其中的某个元素乘以某个1\*1的数，穿过所有通道就相当于给这个管道上的数都乘了权重，如果有很多的过滤器，那么久相当于多了一个全连接层，对应成了很多组权重，产生很多个隐藏单元。所以最后的输出就是原来的高\*原来的宽\*通道数。

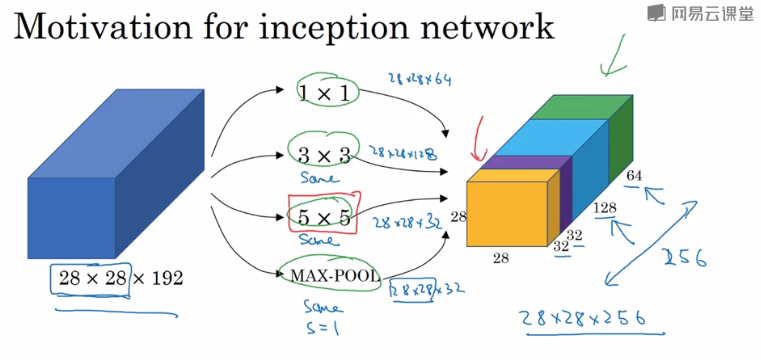


尽管没有被很多的应用，但是它的思维方式却影响了很多网络的构建，就比如后面的inception。1\*1的卷积作用：压缩通道并且减少计算或者为神经网络添加了一个非线性函数，从而或减少或保持通道数不变，当然也可以增加。

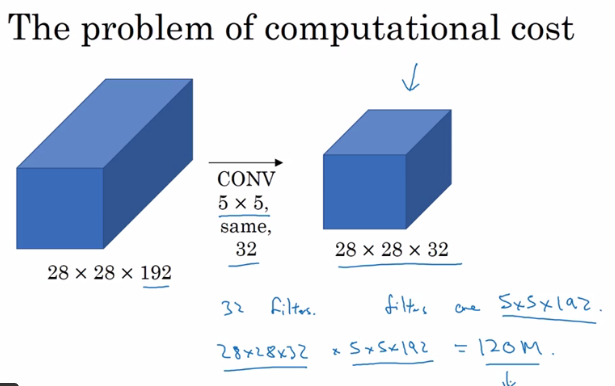


2.6 谷歌inception网络介绍

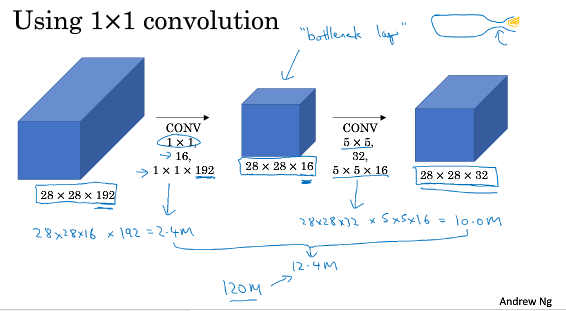
基本思想是inception网络不需要人为决定使用哪种过滤器，或是否需要池化，而是由网络自行确定这些参数。可以给网络添加这些参数的所有可能值，然后把这些输出连接起来，让网络自己学习它需要什么样的参数，采用哪种过滤器组合。



Inception层有一个问题，就是计算成本。我们来试着计算一下使用5\*5卷积层需要的计算成本，计算成本=输出的所有个数\*每个输出值所需的乘法计算次数。这个例子计算出来计算代价大概是12亿。

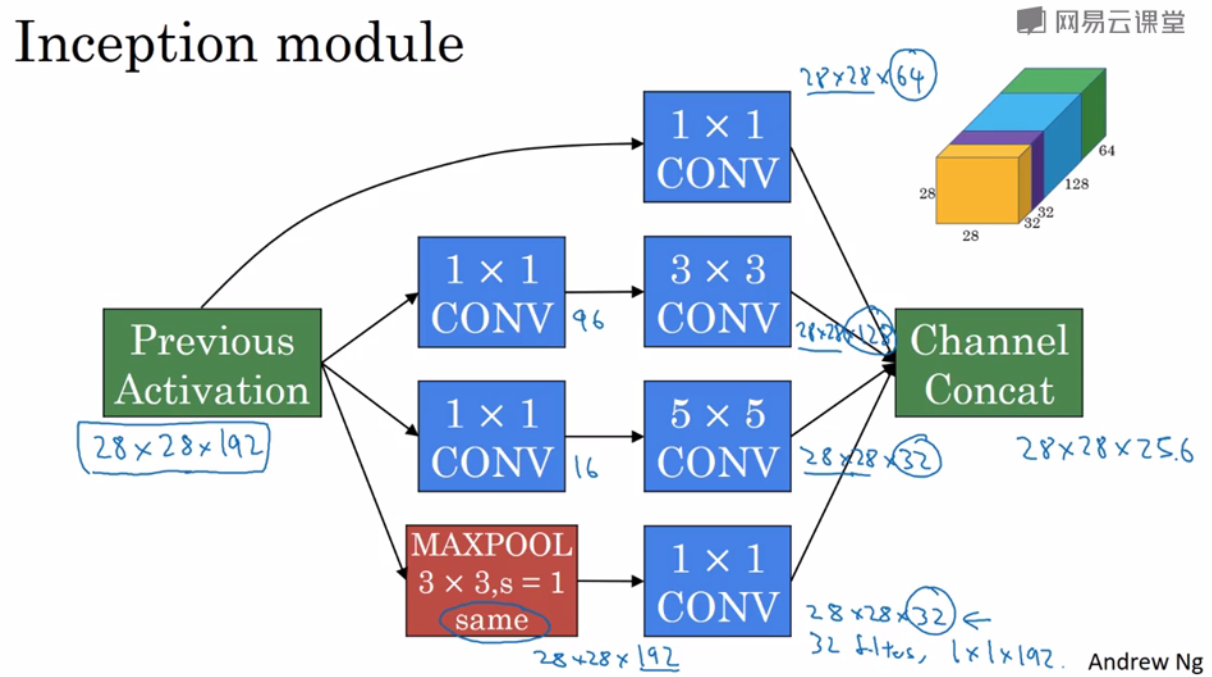


再来看看另一种方案，前后的维度不变，但是多了一个中间步骤，先将原始输入使用1\*1卷积缩小为一个比较小的中间模型，再使用卷积运算扩大。中间就被称为瓶颈层。采用这种方式最后的计算成本是1240万，和前面相比，12亿就降成了10分之1。需要的加法和乘法差不多，所以都只统计了乘法运算。

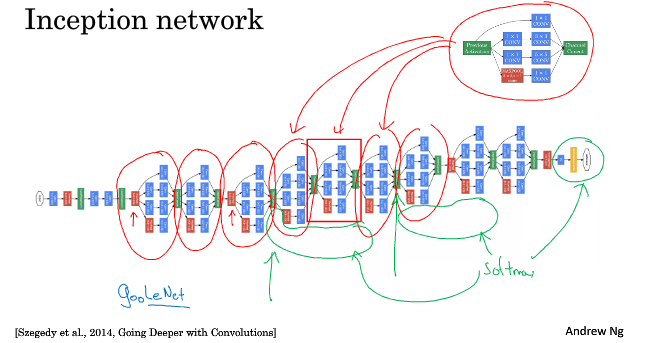


总的来说，无法决定使用什么层以及什么大小的过滤器，inception模块就是最好的选择。可能会问，仅仅大幅缩小表示层规模会不会影响神经网络的性能，事实证明，只要合理构建瓶颈层，既可以缩小表示层规模，又不会降低网络性能，从而节省了大量计算。

2.7 inception网络



这就是一个inception模块，最后将得到的不同的小模块拼接在一起，与输入矩阵的高和宽是保持不变的，通道数会增加很多。Inception网络所做的就是把这些模块组合在一起。



一个inception网络的例子，图中有两个分支，这些分支所做的就是通过隐藏层再加上一些全连接层来做出预测，最后都是softmax预测，输出结果的标签。保证了即使是隐藏单元和中间层，也参与了特征计算，也能预测图片的分类。在inception网络中起到调整的作用，并且能预防网络发生过拟合。

2.8 使用开源的实现方案

如果你正在开发深度学习应用，通常做法是先选择一个你喜欢的架构，接着寻找一个开源实现，从GitHub上下载下来，以此为基础开始创建，优点就是这些网络通常都需要很长的时间来训练，而或许有人已经使用多个gpu通过庞大的数据集预先训练了这些网络，一来就可以使用这些网络进行迁移学习。

现在github上得到下载地址，在cmd命令窗口里面输入git clone 【url】。

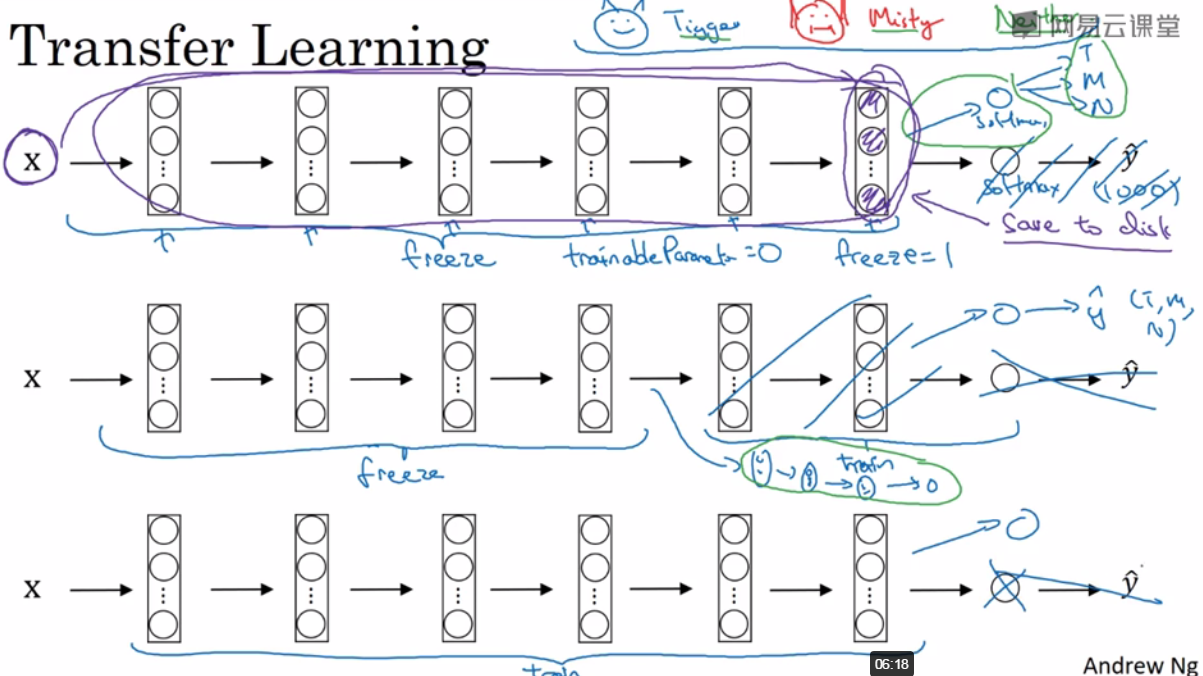
2.9 迁移学习

相比于自己从头构建网络，初始化参数等，将别人学习到的成果，训练好的参数直接应用到自己感兴趣的任务上，有效的多。可以把别人花了很多时间训练出来的开源的权重参数，把它当成一个很好的初始化，用在自己的神经网络上，用迁移学习的公共集上的问题迁移到自己的学习。

最初你没有很多的训练数据，建议从网上下载一些神经网络的开源实现，不仅把代码下载下来，也要把权重下载下来，有许多训练好的网络。举个例子，ImageNet有1000个不同的类别，因此这个网络会有一个Softmax单元，输出1000个可能类别之一，你可以去掉这个softmax层，创建自己的softmax层，建议吧前面所有的层看做是冻结的，冻结前面所有层的参数，只需要训练跟你的softmax层有关的参数。通过别人预训练的权重，很可能得到好的性能，即使只有一个小的数据集。大多数深度学习框架都支持这种操作，事实上取决于用的框架可能会有trainParameter=0这样的参数，对于前面的层，可以设置这样的参数，从而不训练这些权重，有时也会有freeze=1这样的参数，不同深度学习框架有不同的方式，允许你指定是否要训练特定层的权重。

另一个技巧，前面的层被冻结了，相当于一个固定函数，不需要改变也不需要训练，取特征输入X，把它映射到冻结部分的最后这层的激活函数，能加速这个训练的技巧就是先计算最后一层的特征值或激活值，把它们存在硬盘里，然后就相当于训练的很浅的softmax，用这个计算好的特征向量来做预测，所以技巧就是提前计算你训练集中所有样本的最后一层的激活值，在这基础上训练softmax分类器。优点就是不用每次都遍历数据集重新计算这个激活值。

如果你有大量的数据集的话，那么久冻结更少的层，然后训练更多的层。可以取这几层的权重作为初始化，然后使用梯度下降开始训练更新，或者可以直接去掉这几层，换成自己设计的隐藏层和输出层。

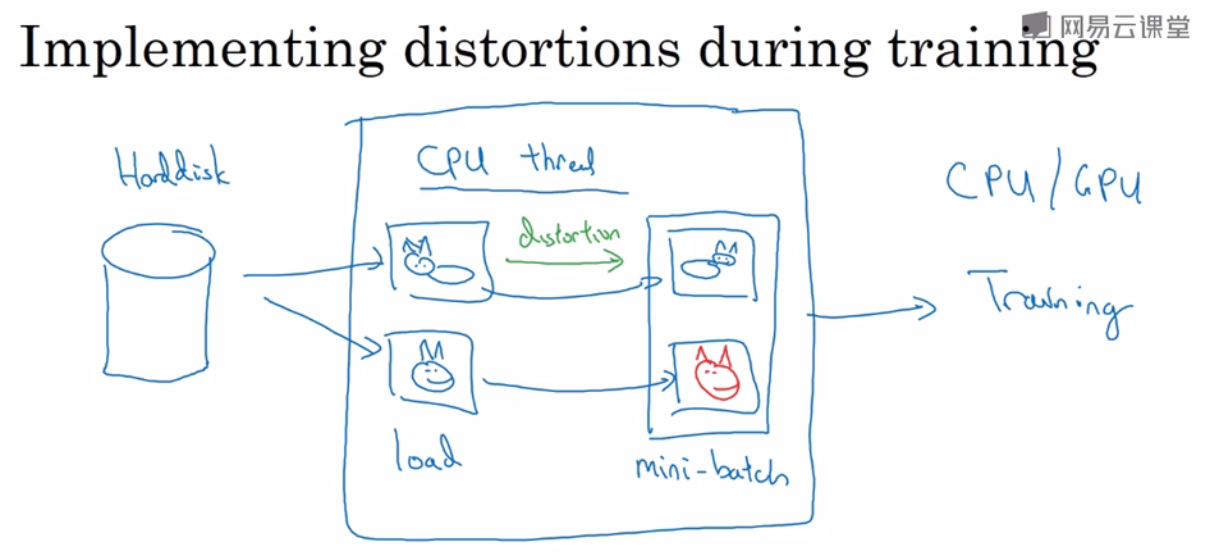


最后总结规律是，你拥有的数据越多，需要冻结的层数越少，能够训练的层数也越多。理念就是如果你有更大的数据集，也有足够的数据，不要单单训练一个softmax单元，而要考虑训练中等大小的网络，包含你最终要用的网络的后面几层。用开源的网络和权重，整个的当作初始化，然后训练整个网络。

2.10 数据扩充

数据增强，计算机视觉最主要的问题没有办法得到充足的数据。最常用的数据扩充就是垂直镜像对称，另一个随机裁剪，给定一个数据集开始随机裁剪，可以得到不同的图片放到数据集中，并不是很完美的一个数据，在实践中，作用还是很大的。旋转rotation，局部弯曲local warping，shearing剪切。

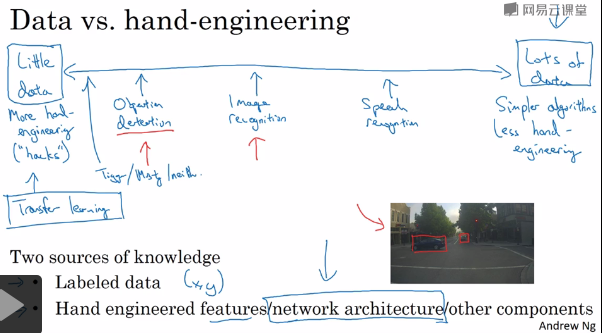
第二个方式就是色彩转换，也就是说给一张图片R G B三个通道都加上不同的失真值，例子中失真的比较夸张，实际上红蓝绿通道是根据某种概率分布来决定的，这样的改变也可能很小。对R G B有不同的采样方式，其中一种颜色扭曲的算法是PCA，主成分分析，PCA颜色增强的意思大概是如果你的图片呈现紫色，即主要含有红色和紫色，绿色很少，那对RG就会变化很多，绿色变化相对少点，使总体的颜色保持一致。



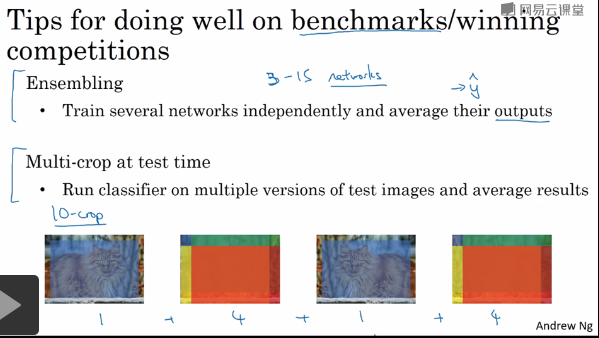
常用的做法使用一个专门的cpu进程从磁盘上读取图片并且实现扭曲等数据增强，再输入另一个进程进行训练，可以实现并行操作。另外关于这个数据增强也会有参数，最好的办法就是借鉴别人的研究成果，如果选择当然也可以自己选择学习参数。

2.11 计算机视觉形状

深度学习已经很好的运用于计算机视觉，自然语言处理，语音识别，在线广告，物流还有其他许多问题。



在基准测试上使用集成或者ten-crops（就是在一张图片和它的镜像上取一个中心和五个角落图片进行操作），可能会取得比较高的正确率，但是需要更多的计算能力和资源，所以一般在基准测试上和竞赛的时候效果比较好，在实际应用生产的时候不会使用。



当你有很小的数据集的时候，就需要我们使用更多的人工设计的架构，组件等，尽管现在关于图片的数据很多了，但是对于计算机视觉这个领域来说，这些数据还是算少的，所以我们还需要人工设计等，但是在数据少的情况下，我们仍然可以通过在别人的好的训练的基础上，完成我们自己的应用。

