贾扬清微信讲座文字版

一、讲座正文：

大家好！我是贾扬清，目前在Google Brain，今天有幸受雷鸣师兄邀请来和大家聊聊Caffe。

没有太多准备，所以讲的不好的地方还请大家谅解。

我用的ppt基本上和我们在CVPR上要做的tutorial是类似的，所以大家如果需要更多的内容的话，可以去tutorial.caffe.berkeleyvision.org，也欢迎来参加我们的tutorial：）

网页上应该还有一些python的样例帮助大家上手，所以欢迎参观。

ppt比较长，所以我想我主要就介绍一下背景以及high level的内容，然后更多关注大家有一些什么具体的问题，希望大家觉得OK。

OK，所以大家最近一段时间应该已经听到很多关于deep learning的八卦了。

deep learning比较流行的一个原因，主要是因为它能够自主地从数据上学到有用的feature。特别是对于一些不知道如何设计feature的场合，比如说图像和speech。deep learning可以学习到比以往比如说sift或者MFCC这样手工设计的feature更好的方法，而且像slide 4显示的一样，这些feature有很强的semantic的含义。

所以很多时候在用到其他的一些task的时候会很有效，这也是为什么我们可以用一个feature来实现很多比如说识别，检测，物体分割这样的不同task的缘故。anyway，deep learning其实说回来是个挺久的话题了。Yann Lecun在89年的时候就提出了convolutional Neural Net的idea。然后在手写数字上获得了很大的成功。最近deep learning重新受到关注，最大的原因是两个：一个是大规模的数据集使得我们可以学习到远比digit更加复杂的概念；另外一个是大规模并行计算让我们可以做很快的优化，使得以前我们没法想象的计算量都变成小case了：）所以这些都很美好。。。但是问题是，写code还挺麻烦的。所以大家肯定希望有个比较好用的框架来很快上手和试试这些deep learning的算法。所以这就是Caffe了：）Caffe是我在Berkeley写thesis的时候想学习C++和cuda写的，然后写完了觉得我自己用太亏了，所以想贡献给community让大家来用，所以如果你看见一些写得很烂的code，不要骂我：P

caffe的好处是，我们基本上可以用一个比较简单的语言（google protobuffer）来定义许多网络结构，然后我们可以在CPU或者GPU上面执行这些代码，而且cpu和gpu在数学结果上是兼容的，然后，所有的模型和recipe我们都会公布出来，使得我们可以很容易地reproduce互相发布的结果，这也是我感到很幸运的一个地方，大家都很喜欢caffe，也很喜欢分享自己paper里的成果（比如说MIT的place net和VGG的模型）。

anyway，这就是Caffe的简单介绍了，最开始是一个hobby project，但是最近Berkeley和其他公司比如说NVidia，Yahoo在很认真地maintain它，希望能够把整个架构做的更好用。

然后我大概讲一下caffe的design吧。基本上，caffe follow了神经网络的一个简单假设 - 所有的计算都是以layer的形式表示的，layer做的事情就是take一些数据，然后输出一些计算以后的结果，比如说卷积，就是输入一个图像，然后和这一层的参数（filter）做卷积，然后输出卷积的结果。每一个layer需要做两个计算：forward是从输入计算输出，然后backward是从上面给的gradient来计算相对于输入的gradient，只要这两个函数实现了以后，我们就可以把很多层连接成一个网络，这个网络做的事情就是输入我们的数据（图像或者语音或者whatever），然后来计算我们需要的输出（比如说识别的label），在training的时候，我们可以根据已有的label来计算loss和gradient，然后用gradient来update网络的参数，这个就是Caffe的一个基本流程。如果大家需要自己实现一个layer的话，可以参考slide28的格式。

我简单解释一下，比如说输入是x，我们可以想象一个layer的forward function就是y=f(x)，然后，我们会有一个loss function，记成L(.)，在做backward的时候，网络得到的是上层给出的gradient，dL/dy。然后网络需要做的计算是dL/dx = dL/dy \* dy/dx，dy/dx也就是f'(x)，于是，这样我们就可以一层一层往后计算gradient，我找一下具体的slide在哪里：）slide 31简单介绍了一下这个forward和backward的结构，anyway，Caffe里面实现的solver主要也是为了神经网络设计的，在做training的时候，我们一般都会做SGD，就是每次输入一个小batch，做计算，update参数，然后再输入下一个batch，Caffe也实现了许多实际应用上比简单SGD要更有效的算法，比如说momentum 和Adagrad，（顺便插一句，Ilya Sutskever有paper解释说，momemtum其实已经可以很好地实现quasi second order的优化，所以建议大家可以从momentum sgd开始尝试做training）。基本上，最简单地用caffe上手的方法就和slide 35说的一样，先把数据写成caffe的格式，然后设计一个网络，然后用caffe提供的solver来做优化看效果如何，如果你的数据是图像的话，可以从现有的网络，比如说alexnet或者googlenet开始，然后做fine tuning，如果你的数据稍有不同，比如说是直接的float vector，你可能需要做一些custom的configuration，caffe的logistic regression example（slide 36）兴许会很有帮助：）

我在和人聊的时候发现大家都比较喜欢fine tune的方法，所以我也简单介绍一下。基本上，finetuning的想法就是说，我在imagenet那么大的数据集上train好一个很牛的网络了，那别的task上肯定也不错，所以我可以把pretrain的网络拿过来，然后只重新train最后几层，重新train的意思是说，比如我以前需要classify imagenet的一千类，现在我只想识别是狗还是猫，或者是不是车牌，于是我就可以把最后一层softmax从一个4096\*1000的分类器变成一个4096\*2的分类器，这个strategy在应用中非常好使，所以我们经常会先在imagenet上pretrain一个网络，因为我们知道imagenet上training的大概过程会怎么样，所以我觉得算法上主要就是这些了，大概再讲一下最近一些比较有意思的方向吧：）

首先是multi-GPU的训练，caffe有一个Flickr的branch可以用来做multi-GPU，不过目前好像把它merge进master得过程有点慢，不过，如果你有兴趣的话，其实multi-GPU不是很难：）比如说，用MPI实现一个GPU之间的synchronization，然后把data transfer和computation 并行起来，基本上就可以实现一个比较直接的single machine multi-gpu training了，当然希望flickr的branch尽早merge。

另外，sequence model （RNN, LSTM）也是一个比较热门的方向，一个比较简单地实现RNN的方法是unrolling，就是说，我不来实现一个for loop，而是确定地说我的sequence就是一个固定的长度，这样，整个网络就依然是一个feedforward的网络，除了需要一些weight sharing以外，依然是可以用原先的架构来实现的，另外就是NVidia的cuDNN，NVidia在cuda上做了很多的优化，所以无论大家是用caffe还是实现自己的code，都可以关注一下它，cudnn最近会准备出v3，效果应该比v2还会更快一些。另外一个比较值得关注的数学计算库是Eigen，在CPU上的优化还是挺显著的。Caffe没有太多地用到Eigen，但是我觉得值得提一下：）anyway，我觉得咱们要不还是多留一些时间来讨论大家关注的问题，所以我就先打住了，我们Caffe的主要的contributer都在slide 89上，大家都很nice，如果你在CVPR上碰见我们的话欢迎来聊天：）

二、问答环节：

问：在finetuning的时候，新问题的图像大小不同于pretraining的图像大小，只能缩放到同样的大小吗？”

答：对的：）

问：目前dl在时序序列分析中的进展如何？研究思路如何，能简单描述一下么

答：这个有点长，可以看看google最近的一系列machine translation和image description的工作。

问：2个问题：1.目前Caffe主要面对CV或图像的任务，是否会考虑其它任务，比如NLP？2.如果想学习Caffe代码的话，能给一些建议吗？

答：Caffe的确主要是做vision的，但是也可以做nlp，caffe的代码学习我觉得主要还是follow tutorial，另外知乎上我记得有一位兄台做过一些解析，但是不是很记得link了...

问："请问下师兄，在移动端用深度学习可以实现实时人脸检测么？谢谢"

答：人脸检测可能目前用传统方法还是很competitive的，但是做一些识别等等，我觉得目前的移动设备应该是可以支持的。

问“1、fine tuning过程是用已有的模型来初始化现有的模型，那在fine tuning的过程中，怎么在fine tuning的时候，不更新某些层的参数呢？”

答：这个在caffe里面可以设置一些layer的learning rate为零来实现：）

问：“我一直想问的问题 就是slide1上的黄嘌呤是什么意思 ，现在的卷积能实现化合物feature的识别吗？”

答：那个其实是咖啡因（caffeine）的分子式 :P

问：“请问 训练过程中batch的大小对结果影响大吗？受限于我的gpu内存，我的batchsize不能选太大，我怀疑这个会导致结果的不收敛”

答：理论上batch小是不会影响收敛的。小batch主要的问题是在FC层的计算可能会不是很efficient，但是数学上没有问题。

问：“ 现在在caffe里实现的imagnet那个 caffenet，是不是 2-GPU的吗？”

答：是单GPU的，其实AlexNet可以直接用单GPU来实现，大家觉得AlexNet是2GPU的缘故是，Alex当年train网络的时候GPU内存太小，他只好用两个GPU来实现：）后来大家一般都是用一个GPU的。

问："师兄您好，想用caffe做下反卷积，发现里面有自带deconv层代码，但是不大会使用，官网也没有相关资料"

答：这个的确有点tricky。。。我个人没用过deconv层，所以不是很好解释，你可以在caffe-users@googlegroups.com上问问：）

问：“用caffe训练自己的数据时，网络层数、卷积核大小、滑动步长，学习速率这些参数的设置有没有一个规律可循呢？ ”

答：这个相对比较tricky，我觉得更多的还是通过已有的架构然后来做一些微调，个人也没有太好的insights可以分享：微软的paper，vgg，googlenet可能有帮助。

问：“目前deep learning用在小数据集上有什么好的方法吗？在小数据集的问题上是不是可以通过减少网络的层数来减少过拟合？”

答：小数据集基本上需要通过小的模型来防止overfit，当然如果数据集是图像等等，也可以通过finetuning。另外一个可能是直接手标更多数据，有时候糙快猛但是还挺好使的。

“我在自己的数据集上训练，训练的loss函数一直不降低，调小过偏置大小，学习率也改过很多，但是每次都很快的迭代到一个大的值，不再变化，而且测试准确率就等于瞎猜的准确率” 这个可能是learning rate太大或者初始值的问题？可以缩小初始值的scale事实

“请问在s层，如何确定该用mean pooling还是max pooling？”基本上靠试 :P

"目前dl近几年在siamese nets distances结构上的进展如何？研究思路如何？" Yann Lecun有paper讲这个，值得看看

“师兄您好，我想问下不使用matlab或python接口，直接在C++的caffe代码里对图像进行分类有什么好的方式吗，速度会不会比matlab和python更快” 我觉得速度应该差不多，因为matlab和python的overhead不会太大

“dl能实现FFT吗” facebook其实有fft的code，参见fbfft:)

"2、caffe内部的Convolution计算是图像拉伸成向量进行的计算，这种方式会比普通的方法和fft的方法计算更快吗？

放大点说，caffe做了哪些算法上的优化 使得计算速度比较快呢？" 那个其实是我的weekend hack，所以推荐大家用其他的优化，比如说cudnn等等。说实话写caffe的时候我没太关注速度....

“师兄，您好！用caffe纯粹做分类的话（前向），需要softmax层吗？看代码有个pro层和softmax一样吗？” 不是很清楚pro层是哪个，不过也可以用logistic，任何传统的分类函数应该都是可以的

“3、对于cxxnet，您是怎么看待的呢？ ” 我还挺喜欢cxxnet的一些设计的，基本上就是大家选自己喜欢的codebase来用吧：）

关于时序的问题统一回答一下 - 大家可以参考最近的machine translation，im2txt等等的一系列文章

“请问，想cxxnet，这些新的框架，也集成了bn，prelu等新的模块，caffe是否会内置这些模块呢>” 我觉得会的，这个在code层面上其实没有太大的问题。我最近主要在做一些refactor，然后还有一些公司的事情，所以没有关注在push新的模块上：）

“caffe能否在多个层都连接loss函数，同时进行反向传播” 可以的，关键是要处理好gradient merge的问题，其他都是OK的：）

“caffe里面的激活函数可以自行修改成其他自己设计的激活函数吗” 可以的，你可以参考ReLU层的code，然后改一下relu的函数就可以了

“CNN可以应用到对图像进行深度图提取吗？效果会怎样呢？”最近nyu应该有一篇stereo的文章，应该比较类似？

“caffe会内置rbm的模块吗。nin相关的会不会也会考虑添加。” rbm可能不会，因为最近用得好像比较少。nin其实已经支持了 - nin的本质是1x1的convolution，可以参考googlenet

“我现在是在做机器学习，还没有深入deep learning，是不是要先打好机器学习的基础再学dp会好一点，谢谢贾老师了” 这个我其实也不是很清楚，很多想法其实都是相通的（比如说优化的问题），所以可以都看一些，然后按照自己的需求深入：）

“用hdf5layer实现多label的过程不是很清楚，举个例子说，比如，输入低分辨图像，label是高分辨图像，，这种有没有详细一点的教程，或者师兄能不能简单提一下” 这个主要就是要设计一个input层能够输出不同的top blob，其实caffe在这一点上做的不是很好（因为太关注classification了），可能看一下这些典型的输入层的实现会有帮助。

“caffe能支持lstm、rnn的训练吗？另外，对于百度的dlmc您有什么看法？” Jeff Donahue有一个branch可以来做lstm，我自己在refactor的一些code应该也是可以的，但是因为公司review政策的缘故没法保证什么时候能release :) dmlc我觉得是个挺好的effort，在开源界看到更多中国学生的身影很兴奋!

“师兄您好。想问一个问题，如何将已知的世界知识，比如说语法规则等有效融入到深度学习中？” 这个是个好问题，目前大家都有点倾向于learning from scratch，所以我也说不好怎么做融合，但是应该是一个值得考虑的研究方向

“请问调参方面有什么比较细致的资料或文献集” “solver里的 lr\_policy: 选择有什么规律么 我看到有fixed inv” 这两个问题，基本上我觉得还是靠经验。marc'aurelio ranzato曾经有一个presentation讲一些有用的trick，容我找找，anyway，不太好找，但是marc'aurelio的网站在这，应该是其中的某一个slides：<http://www.cs.toronto.edu/~ranzato/>

“用自己的数据（并不属于imagenet的1000个类）在imagenet训练的网络上做finetune时，发现怎么调整参数最后几乎都无法用来分类，这是什么原因呢？” 这个可能需要看一下图片是否类似，比如说imagenet的模型用来做医学图像识别效果就很可能会不是很好，还是需要看这两个task的数据之间是否有相似性.

“接着上一轮的提问，caffe实现多层loss反向传播，我能不能直接在prototxt里每一层后加一层loss，最后的结果会是怎样？” 唔，这个得看loss是什么了，比如说googlenet用到了几个branch来inject softmax，所以基本上还是要寻找和问题相关的loss term

“可否评论一下nature 新出的DL文章？reinforcement learning之类的会是下一个主要结合的点吗？” 哈，Hinton本人的说法是“you won't learn much from that paper”。那个更多的是一个overview，如果希望了解一下DL的来龙去脉的话值得读一下。RL其实还是挺热门的，deepmind做的就有点像RL，berkeley Pieter Abbeel组也做了很多RL的工作

“,lstm97年就出来了，为何最近又火起来” 我觉得是因为LSTM的确可以很好地model sequence data，为啥会有冷热的问题，这个很神秘：）谁也说不清楚，你看CNN也是冷了好几年然后忽然热了。

“dl能实现FFT吗” facebook其实有fft的code，参见fbfft:)” fb是利用了FFT去快速计算，不是我问的意思。用傅立叶变换其实是提取了频域特征，根据应用的不同，最优的变换不一定是FT，可能是时频变换、分数阶FT等等变换。那么问题就来了：利用深度学习算法，能否学习到最优的时频特征表出？如果可以，是不是可以把信号处理里面的固定分析方法都扔掉？” 这个我就的确不是专家了，我觉得这个有点类似于model design的问题，深度学习相当于也是设计了一大类的model，然后在这一类model当中寻找最优的，所以如果有一些oracle knowledge（比如说已有的固定分析的经验）可以知道如何rectify数据，我觉得应该还是有帮助的

“caffe有没有对分布式的支持？”目前在parallel branch里面

“3.caffe的训练过程如何使用gpu对计算性能进行优化” 这个更多的是在code层面上调速度了，如果有兴趣的话，nvidia的nvprof应该会很有帮助

“记得有一篇说论文说 在imagenet上，把30%的标签打乱，反而使得最后的结果更好和更鲁棒。那么是不是意味着我们不需要强定义的数据（不需要那么仔细的标注数据） 就可以训练得到一个不错的模型呢？” 我觉得基本上就是数据越干净，数据越多，效果一般就越好（实际应用上我们有时候会让human rater去再次确认一些不确定的标注）。鲁棒性的问题，我觉得可能是因为增加了regularization？imagenet上基本上还是标准的protocol来training效果最好。

“caffe用的GPU大概成本需要多少” 取决于GPU，我觉得从200到1000美元不等？当然土豪用5000块钱的K80这种事情也是可以的。

“师兄您好！用SGD的时候，收敛充分的前提下，不同的学习率衰减策略是不是结果都差不多？” 恩，一般会差不多

“dl 在ctr预测上有什么好的论文或者资料么？” 我不是很清楚，不过余凯师兄以前讲过百度用DL做CTR效果很好，所以还是很promising的

“不好意思，我的问题可能没表达清楚，您之前说多层loss反向传播，需要处理好gradient的merge，我想问，如果只是在prototxt里，每一层后加上需要的loss函数，那么caffe最终的反向传播会是怎样进行的” 哦，应该是这样的，每一层后面需要一个split层，把这一层的输入变成两个blob，一个继续往下传播，一个输入到loss层里面。在backprop的时候，split层会把这两条路径的gradient加起来

"其实我对师兄解释的dl在时序方面的应用还是不太清楚，能多分析一下吗？" DL在时序方面的应用主要是RNN/LSTM这方面，主要是用来理解sequence的信息，两个用法：（1）提取sequence的feature，然后来做classification或者embedding，（2）从sequence到sequence，比如说输入语音，输出识别的句子

“1.caffe的训练过程能否保持对象的旋转不变性 怎样做到这点” 目前不是很好explicit地输入这样的constraint，主要还是靠data augmentation（输入各种旋转以后的图）来实现

“2.caffe对不同尺度的同一对象的分类和识别有哪些特殊的处理方法” 这个倒也不单是caffe的问题，在图像识别上如果需要处理不同尺度，一般就是做multi-scale的detection，可以参考一下selective search，R-CNN等等的工作

“用自己的数据集，且类型和和imagenet的类型不太一样（比如细胞类型），想用caff训练的话，最少得需要多少数据量，才比较好？” 这个说不太好，所以最好还是先用一部分数据测试一下，然后你可以用从少到多的数据来训练，然后外推一下可能会需要多少数据

“现在caffe上有一些已经训练好的，准确率比较高的模型吗？我在caffe主页下载的几个分类的精度都不高，cifar10和imagenet的都是 百分之八十几，有精度更高的吗？” 基本上imagenet的模型算是准确度最高的了，包括googlenet和vggnet

“softmax\_layer和softmax\_loss\_layer有什么区别。” softmax\_layer是做softmax变换（就是把输入的score变成sum to 1的概率值）， softmax\_loss是计算prediction和true label之间的cross entropy loss function

“Caffe现在怎么处理变长的图片，因为Conv对变长不明感，而且可以用Dynamic Pooling？”变长的图片可以用SPPNet这样的思路，最后做一个固定输出大小的pooling

“请问多任务学习的DL有什么经验可以分享吗？比如数据分布的均匀性的影响” 数据分布均匀性一般都还是挺tricky的，实际操作上一般我觉得cap一些frequency（如果某一类太多了，就downsample一下）会使得training更好一些

“想问一下：在神经网络的训练过程中，如何能够并行或者说更快地计算？” 主要是靠两点吧，一个是写更快的code（比如说用cudnn优化convolution），一个是写并行计算的框架（这方面我推荐用MPI入手，因为MPI虽然没有fault tolerance等等的好处，但是并行非常简单，可以作为最开始的测试）

“autoencoder 模型中，单个隐含层和多隐层 模型，效果差别很多啊吗？” 这个可能和具体实现有关，隐层多了以后，representation power增加，很可能会提升效果，但是也可能会overfit，所以需要更仔细的training

“请问除了从分类结果看特征表出的优劣，有没有一种通行的方式去看特征表出的优劣？还有一个问题：lstm简直就是一个编码模型…以后机器学习的结构都要往电子工程上靠了吗？我觉得结构越来越复杂正背离dl的初衷了…” 其实大家经常批评DL的问题就是说，我们从设计feature变成了设计model（我记得原话是jitendra malik讲的...啊我太八卦了）。所以这个的确也是一个难解的问题，兴许我们可以做一个算法来自动生成很多model然后evolve这些model？MIT曾经有一篇paper来自动学习网络的结构，但是目前state of the art的模型还经常靠手调

“DL中，能否预知到底学到了一个怎样的物理模型,来实现分类的？” 参见上面的回答：）目前比较困难，在图片上，大家做过一些有意思的实验来检测模型到底学了什么，可以参考karen simonyan的文章（用CNN来生成一个"最像"某一个类别的图像）