Yoshua Bengio等大神传授：26条深度学习经验

发表于 2015-09-17 08:21 | 21682次阅读 | 来源 Marek Rei | 0 条评论 | 作者 Marek Rei

深度学习 Python 大数据 并行处理 性能优化

摘要：究竟需要有哪些手段可以帮助我们来深度学习，Marek Rei参加了蒙特利尔深度学习暑期学校的课程，在他的个人网站为我们带来了26条深度学习的经验。

【编者按】8月初的蒙特利尔深度学习暑期班，由Yoshua Bengio、 Leon Bottou等大神组成的讲师团奉献了10天精彩的讲座，剑桥大学自然语言处理与信息检索研究组副研究员Marek Rei参加了本次课程，在本文中，他精炼地总结了学到的26个有代表性的知识点，包括分布式表示，tricks的技巧，对抗样本的训练，Neural Machine Translation，以及Theano、Nvidia Digits等，非常具有参考价值。

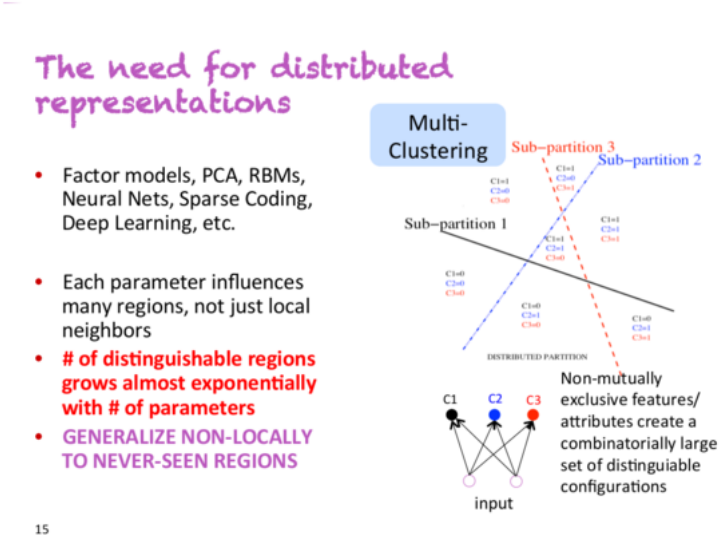
八月初，我有幸有机会参加了蒙特利尔深度学习暑期学校的课程，由最知名的神经网络研究人员组成的为期10天的讲座。在此期间，我学到了很多，用一篇博客也说不完。我不会用60个小时的时间来讲解神经网络知识的价值，而会以段落的方式来总结我学到的一些有趣的知识点。

在撰写本文时，暑期学校网站仍可以访问，并附有全部的演示文稿。所有的资料和插图都是来自原作者。暑期学校的讲座已经录制成了视频，它们也可能会被上传到网站上。

好了，我们开始吧。

1、分布式表示（distributed representations）的需要

在Yoshua Bengio开始的讲座上，他说“这是我重点讲述的幻灯片”。下图就是这张幻灯片：

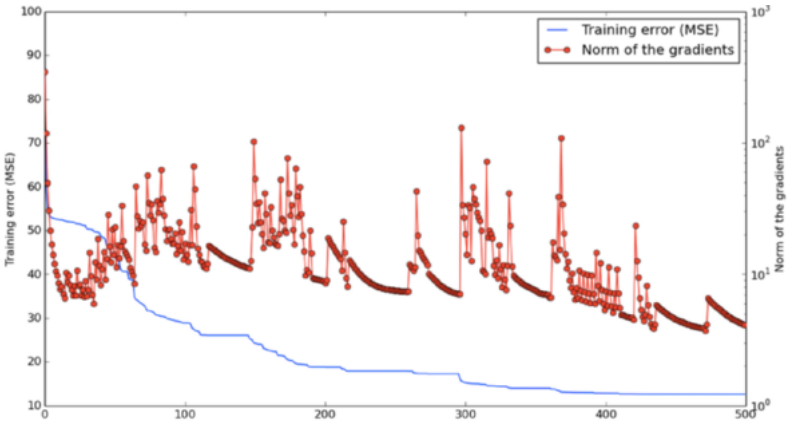


假设你有一个分类器，需要分类人们是男性还是女性，佩戴眼镜还是不佩戴眼镜，高还是矮。如果采用非分布式表示，你就在处理2*2*2=8类人。为训练精准度高的分类器，你需要为这8类收集足够的训练数据。但是，如果采用分布式表示，每一个属性都会在其他不同维度中有所展现。这意味着即使分类器没有碰到佩戴眼镜的高个子，它也能成功地识别他们，因为它学会了从其他样本中单独学习识别性别，佩戴眼镜与否和身高。

2、局部最小在高维度不是问题

Yoshua Bengio的团队通过实验发现，优化高维度神经网络参数时，就没有局部最小。相反，在某些维度上存在鞍点，它们是局部最小的，但不是全局最小。这意味着，在这些点训练会减慢许多，直到网络知道如何离开这些点，但是我们愿意等足够长的时间的话，网络总会找到方法的。

下图展示了在网络训练过程中，两种状态的震动情况：靠近鞍点和离开鞍点。



给定一个指定的维度，小概率p表示点是局部最小的可能性，但不是此维度上全局最小。在1000维度空间里的点不是局部最小的概率**和**就会是，这是一个非常小的值。但是，在**某些**维度里，这个点是局部最小的概率实际上比较高。而且当我们同时得到多维度下的最小值时，训练可能会停住直到找到正确的方向。

另外，当损失函数接近全局最小时，概率p会增加。这意味着，如果我们找到了真正的局部最小，那么它将非常接近全局最小，这种差异是无关紧要的。

3、导函数，导函数，导函数

Leon Bottou列出了一些有用的表格，关于激活函数，损失函数，和它们相应的导函数。我将它们先放在这里以便后续使用。

	Propagation	Back-propagation
Sigmoid	$y_s = \frac{1}{1+e^{-x_s}}$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \frac{1}{(1+e^{x_s})(1+e^{-x_s})}$
Tanh	$y_s = \tanh(x_s)$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \frac{1}{\cosh^2 x_s}$
ReLu	$y_s = \max(0, x_s)$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \mathbb{I}\{x_s > 0\}$
Ramp	$y_s = \min(-1, \max(1, x_s))$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \mathbb{I}\{-1 < x_s < 1\}$

		Propagation	Back-propagation
Square		$y = \frac{1}{2}(x-d)^2$	$\frac{\partial E}{\partial x} = (x-d)^T \frac{\partial E}{\partial y}$
Log	$c = \pm 1$	$y = \log(1 + e^{-cx})$	$\frac{\partial E}{\partial x} = \frac{-c}{1+e^{cx}} \frac{\partial E}{\partial y}$
Hinge	$c = \pm 1$	$y = \max(0, m - cx)$	$\frac{\partial E}{\partial x} = -c \ \mathbb{I}\{cx < m\} \frac{\partial E}{\partial y}$
LogSoftMax	$c = 1 \dots k$	$y = \log(\sum_k e^{x_k}) - x_c$	$\left[\frac{\partial E}{\partial x}\right]_s = (e^{x_s}/\sum_k e^{x_k} - \delta_{sc}) \frac{\partial E}{\partial y}$
MaxMargin	$c = 1 \dots k$	$y = \left[\max_{k \neq c}\{x_k + m\} - x_c\right]_+$	$\left[\frac{\partial E}{\partial x}\right]_s = (\delta_{sk^*} - \delta_{sc}) \mathbb{I}\{E > 0\} \frac{\partial E}{\partial y}$

更新：根据评论指出，斜率公式中的最小最大函数应该调换。

4、权重初始化策略

目前在神经网络中建议使用的权重初始化策略是将值归一化到范围[-b,b]，b为：

$$b = \sqrt{\frac{6}{H_k + H_{k+1}}}$$

H_k和H_{k+1}分别是权值向量之前和之后的隐藏层大小。

由Hugo Larochelle推荐，Glorot和Bengio发布（2010）。

5、神经网络训练技巧

Hugo Larochelle给出的一些实用建议：

- 归一化实值数据。减去平均值，再除以标准差。
- 降低训练过程中的学习率。
- 更新使用小批量数据，梯度会更稳定。
- 使用动量，通过停滞期。

6、梯度检测

如果你手动实现了反向传播算法但是它不起作用，那么有99%的可能是梯度计算中存在Bug。那么就用梯度检测来定位问题。主要思想是运用梯度的定义：如果我们稍微增加某个权重值，模型的误差将会改变多少。

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon}$$

这里有更详细的解释：Gradient checking and advanced optimization。

7、动作跟踪

人体动作跟踪可以达到非常高的精准度。下图是来自Graham Taylor等人（2010）发表的论文Dynamical Binary Latent Variable Models for 3D Human Pose Tracking中的例子。该方法使用的是条件受限的玻尔兹曼机。

8、使用语法还是不使用语法？（即“需要考虑语法吗？”）

Chris Manning和Richard Socher已经投入了大量的精力来开发组合模型，它将神经嵌入与更多传统的分析方法组合起来。这在Recursive Neural Tensor Network这篇论文中达到了极致，它使用加法和乘法的互动将词义与语法解析树组合。

然后，该模型被Paragraph向量（Le和Mikolov，2014）打败了（以相当大的差距），Paragraph向量对语句结构和语法完全不了解。Chris Manning将这个结果称作“创造‘好的’组合向量的一次失败”。

然而，最近越来越多的使用语法解析树的工作成果改变了那一结果。Irsoy和Cardie（NIPS，2014）在多维度上使用更深层的网络成功地打败了Paragraph向量。最后，Tai等人（ACL，2015）将LSTM网络与语法解析树结合，进一步改进了结果。

这些模型在斯坦福5类情感数据集上结果的精准度如下：

Method	Accuracy
RNTN (Socher et al. 2013)	45.7
Paragraph Vector (Le & Mikolov 2014)	48.7
DRNN (Irsoy & Cardie 2014)	49.8
Tree LSTM (Tai et al. 2015)	50.9

从目前来看，使用语法解析树的模型比简单方法更胜一筹。我很好奇下一个不基于语法的方法何时出现，它又将会如何推动这场比赛。毕竟，许多神经模型的目标不是丢弃底层的语法，而是隐式的将它捕获在同一个网络中。

9、分布式与分配式

Chris Manning本人澄清了这两个词之间的区别。

分布式：在若干个元素中的连续激活水平。比如密集词汇嵌入，而不是1-hot向量。

分配式：表示的是使用上下文。word2vec是分配式的，当我们使用词汇的上下文来建模语义时，基于计数的词汇向量也是分配式的。

10、依赖状态分析

Penn Treebank中的依赖分析器比较：

Parser	Unlabelled Accuracy	Labelled Acccuracy	Speed (sent/s)
MaltParser	89.8	87.2	469
MSTParser	91.4	88.1	10
TurboParser	92.3	89.6	8
Stanford Neural Dependency Parser	92.0	89.7	654
Google	94.3	92.4	?

最后一个结果是从谷歌“提取出所有stops”得到的，将海量数据源来训练斯坦福神经语法解析器。

11、Theano

我之前对Theano有所了解，但是我在暑期学校学习到了更多。而且它实在是太棒了。

由于Theano起源自蒙特利尔，直接请教Theano的开发者会很有用。

关于它大多数的信息都可以在网上找到，以交互式Python教程的形式。

12、Nvidia Digits

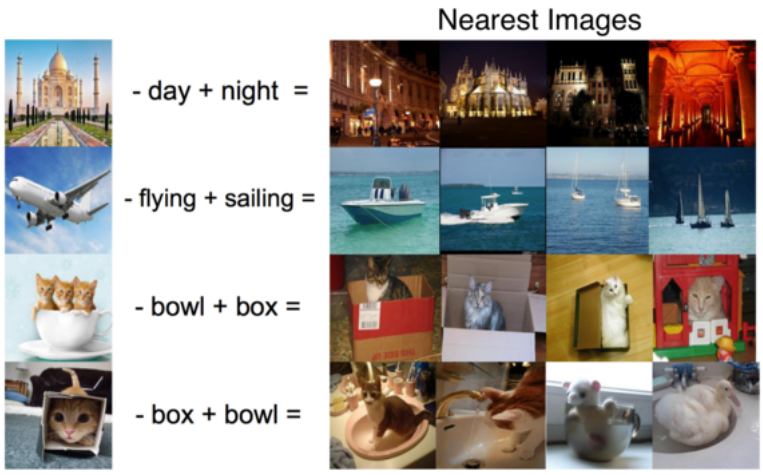
英伟达有一个叫做Digits的工具包，它可以训练并可视化复杂的神经网络模型而不需要写任何代码。并且他们正在出售DevBox，这是一款定制机器，可以运行Digits和其他深度学习软件（Theano，Caffe等）。它有4块Titan X GPU，目前售价15,000美元。

13、Fuel

Fuel是一款管理数据集迭代的工具，它可以将数据集切分成若干小部分，进行shuffle操作，执行多种预处理步骤等。对于一些建立好的数据集有预置的功能，比如MNIST，CIFAR-10和谷歌的10亿词汇语料库。它主要是与Blocks结合使用，Blocks是使用Theano简化网络结构的工具。

14、多模型语言学规律

记得“国王-男性+女性=女王”吗？事实上图片也能这么处理（Kiros等人，2015）。



(Kiros, Salakhutdinov, Zemel, TACL 2015)

15、泰勒级数逼近

当我们在点处，向移动时，那么我们可以通过计算导函数来估计函数在新位置的值，我们将使用泰勒级数逼近：

$$f(x) = f(x_0) + (x - x_0)f'(x) + \frac{1}{2}(x - x_0)^2 f''(x) + \cdots$$

同样地，当我们将参数更新到时，我们可以估计损失函数：

$$J(\theta) = J(\theta_0) + (\theta - \theta_0)^T g + \frac{1}{2}(\theta - \theta_0)^T H(\theta - \theta_0) + \cdots$$

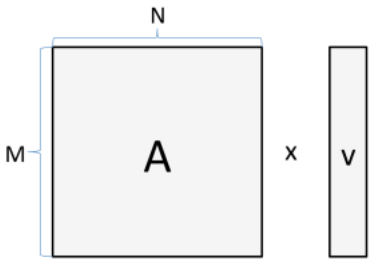
其中g是对θ的导数，H是对θ的二阶Hessian导数。

这是二阶泰勒逼近，但是我们可以通过采用更高阶导数来增加准确性

16、计算强度

Adam Coates 提出了一种分析GPU上矩阵操作速度的策略。这是一个简化的模型，可以显示花在读取内存或者进行计算的时间。假设你可以同时计算这两个值，那么我们就可以知道那一部分耗费时间更多。

假设我们将矩阵和一个向量相乘：



如果M=1024，N=512，那么我们需要读取和存储的字节数是：

> 4 bytes ×(1024×512+512+1024)=2.1e6 bytes

计算次数是：

> 2×1024×512=1e6 FLOPs

如果我们有块6TFLOP/s的GPU，带宽300GB/s的内存，那么运行总时间是：

> max{2.1e6 bytes /(300e9 bytes/s),1e6 FLOPs/(6e12 FLOP/s)}=max{7μs,0.16μs}

这意味着处理过程的瓶颈在于从内存中复制或向内存中写入消耗的7μs，而且使用更快的GPU也不会提升速度了。你可能会猜到，在进行矩阵-矩阵操作时，当矩阵/向量变大时，这一情况会有所好转。

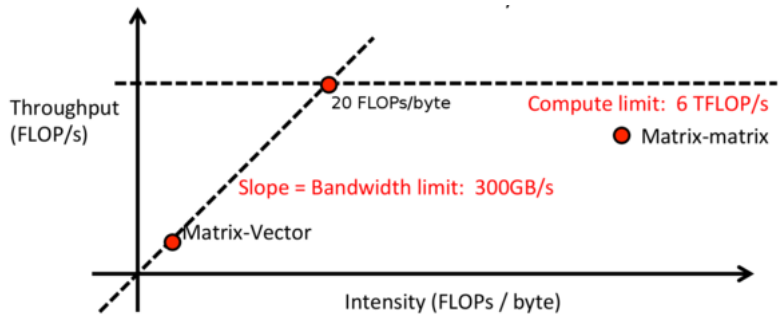
Adam同样给出了计算操作强度的算法：

> 强度=（#算术操作）/（#字节加载或存储数）

在之前的场景中，强度是这样的：

> 强度=（1e6 FLOPs）/（2.1e6 bytes）= 0.5FLOPs/bytes

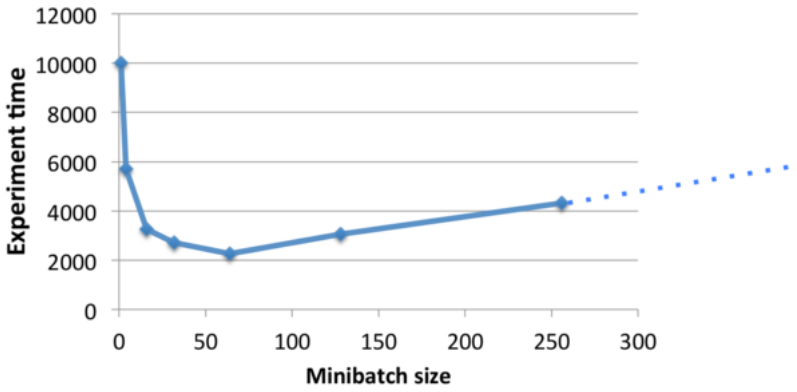
低强度意味着系统受内存大小的牵制，高强度意味着受GPU速度的牵制。这可以被可视化，由此来决定应该改进哪个方面来提升整体系统速度，并且可以观察最佳点的位置。



17、小批量

继续说说计算强度，增加网络强度的一种方式（受计算而不是内存限制）是，将数据分成小批量。这可以避免一些内存操作，GPU也擅长并行处理大矩阵计算。

然而，增加批次的大小的话可能会对训练算法有影响，并且合并需要更多时间。重要的是要找到一个很好的平衡点，以在最短的时间内获得最好的效果。



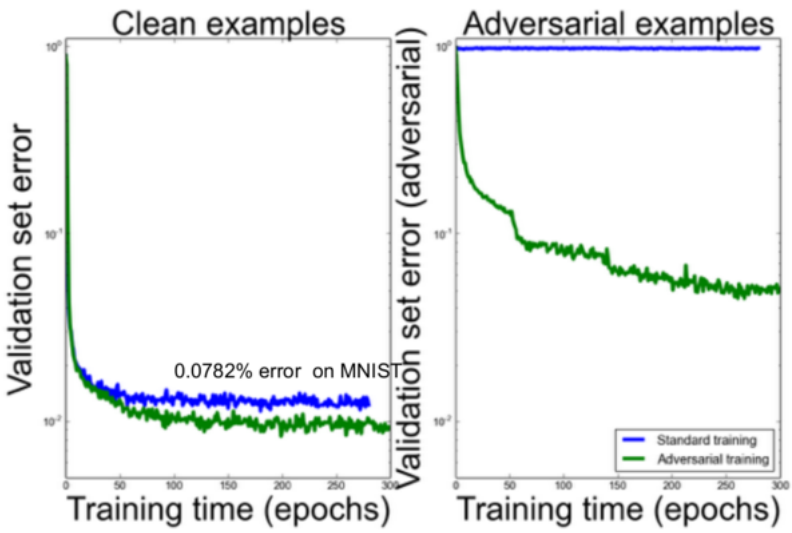
18、对抗样本的训练

据最近信息显示，神经网络很容易被对抗样本戏弄。在下面的案例中，左边的图片被正确分类成金鱼。但是，如果我们加入中间图片的噪音模式，得到了右边这张图片，分类器认为这是一张雏菊的图片。图片来自于Andrej Karpathy的博客 “Breaking Linear Classifiers on ImageNet”，你可以从那了解更多。

噪音模式并不是随机选择的，而是为了戏弄网络通过精心计算得到的。但是问题依然存在：右边的图像显然是一张金鱼而不是雏菊。

显然，像集成模型，多扫视后投票和无监督预训练的策略都不能解决这个漏洞。使用高度正则化会有所帮助，但会影响判断不含噪声图像的准确性。

Ian Goodfellow提出了训练这些对抗样本的理念。它们可以自动的生成并添加到训练集中。下面的结果表明，除了对对抗样本有所帮助之外，这也提高了原始样本上的准确性。



最后，我们可以通过惩罚原始预测分布与对抗样本上的预测分布之间的KL发散来进一步改善结果。这将优化网络使之更具鲁棒性，并能够对相似（对抗的）图像预测相似类分布。

19、万事万物皆为语言建模

Phil Blunsom 提出，几乎所有的NLP都可以构建成语言模型。我们可以通过这种方式实现，将输出与输入连接，并尝试预测整个序列的概率。

翻译：

> P(Les chiens aiment les os || Dogs love bones)

问答：

> P(What do dogs love? || bones .)

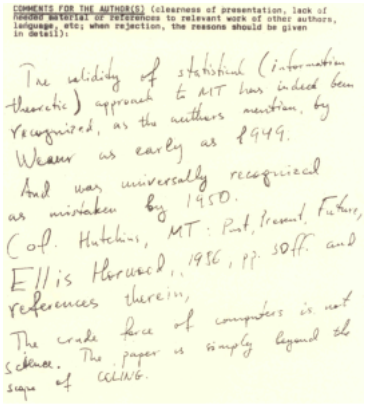
对话：

> P(How are you? || Fine thanks. And you?)

后两个必须建立在对世界已知事物了解的基础上。第二部分甚至可以不是词语，也可以是一些标签或者结构化输出，比如依赖关系。

20、SMT开头难

当Frederick Jelinek 和他在IBM的团队在1988年提交了关于统计机器翻译第一批之一的论文时，他们的到了如下的匿名评审：

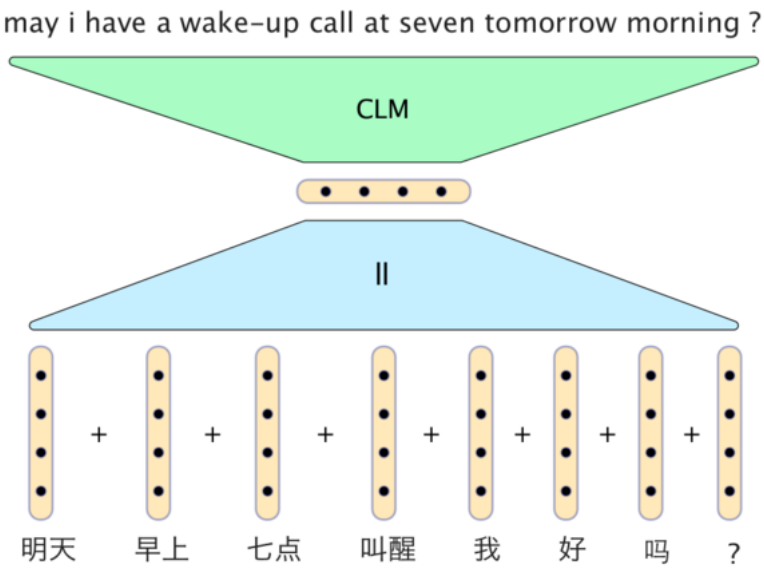

COMMENTS FOR THE AUTHOR(S) (clearness of presentation, lack of
recent material or references to relevant work of other authors,
language, etc; when rejection, the reasons should be given
in detail):

The validity of statistical (information theoretic) approach to MT has indeed been recognized, as the authors mention, by Weaver as early as 1949.
And was universally recognized as mistaken by 1950.
Cf. Hutchins, MT: Past, Present, Future, Ellis Horwood, 1986, pp. 30ff. and references therein.
The crude force of computers is not science. The paper is simply beyond the scope of COLING.

正如作者提到的，早在1949年Weaver就肯定了统计（信息论）方法进行机器翻译的有效性。而在1950年被普遍认为是错误的（参见Hutchins, MT – Past, Present, Future, Ellis Horwood, 1986, p. 30ff 和参考文献）。计算机的暴力解决并不是科学。该论文已经超出了COLING的范围。

21、神经机器翻译（Neural Machine Translation）现状

显然，一个非常简单的神经网络模型可以产生出奇好的结果。下图是Phil Blunsom的一张幻灯片，将中文翻译成英文的例子：



在这个模型中，汉字向量简单地相加在一起形成一个语句向量。解码器包含一个条件性语言模型，将语句向量和两个最近生成的英语单词中的向量结合，然后生成译文中下一个单词。

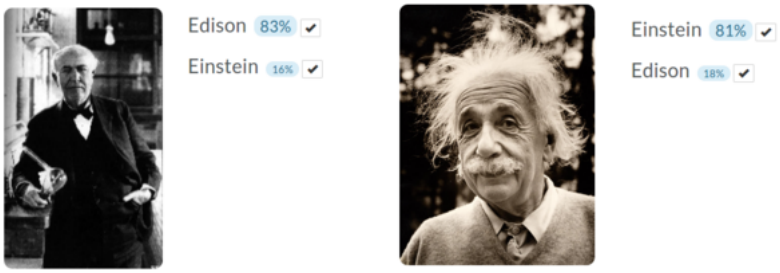
然而，神经模型仍然没有将传统机器翻译系统性能发挥到极致。但是它们已经相当接近了。Sutskever等人（2014）在"Sequence to Sequence Learning with Neural Networks"中的结果：

Model	BLEU score
Baseline	33.30
Best WMT'14 result	37.0
Scoring with 5 LSTMs	36.5
Oracle(upper bound)	~45

更新：@stanfordnlp指出，最近一些结果表明，神经模型确实会将传统机器翻译系统性能发挥到极致。查看这篇论文"Effective Approaches to Attention-based Neural Machine Translation"（Luong等人，2015）

22、伟大人物的分类例子

Richard Socher演示了伟大人物图像分类例子，你可以自己上传图像来训练。我训练了一个可以识别爱迪生和爱因斯坦（不能找到足够的特斯拉个人相片）的分类器。每个类有5张样本图片，对每个类测试输出图像。似乎效果不错。



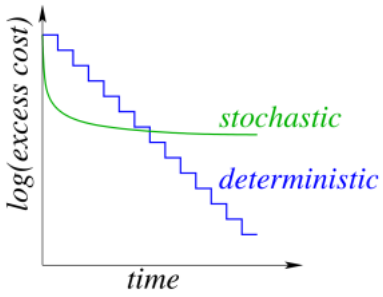
23、优化梯度更新

Mark Schmidt给出了两份关于在不同情况下数值优化的报告。

在**确定性**梯度方法中，我们在整个数据集上计算了梯度，然后更新它。迭代成本与数据集大小呈线性关系。

在**随机**梯度方法中，我们在一个数据点上计算了梯度，然后更新它。迭代成本与数据集大小无关。

随机梯度下降中的每次迭代要快许多，但是它通常需要更多的迭代来训练网络，如下图所示：



为了达到这两者最好效果，我们可以用批量处理。确切的说，我们可以对数据集先进行随机梯度下降，为快速达到右边的部分，然后开始增加批大小。梯度误差随着批大小的增加而减少，然而最终迭代成本大小还是会取决于数据集大小。

随机平均梯度（SAG）可以避免这样的情况，每次迭代只有1个梯度，从而得到线性收敛速度。不幸的是，这对于大型神经网络是不可行的，因为它们需要记住每一个数据点的梯度更新，这就会耗费大量内存。随机方差降低梯度（SVRG）可以减少这种内存耗费的情况，并且每次迭代（加上偶然全部通过）只需要两次梯度计算。

Mark表示，他的一位学生实现了各种优化方法（AdaGrad，momentum，SAG等）。当问及在黑盒神经网络系统中他会使用什么方法时，这位学生给出了两种方法：Streaming SVRG（Frostig等人，2015），和一种他们还没发布的方法。

24、Theano分析

如果你将“profile=true”赋值给THEANO_FLAGS，它将会分析你的程序，然后显示花在每个操作上的时间。对寻找性能瓶颈很有帮助。

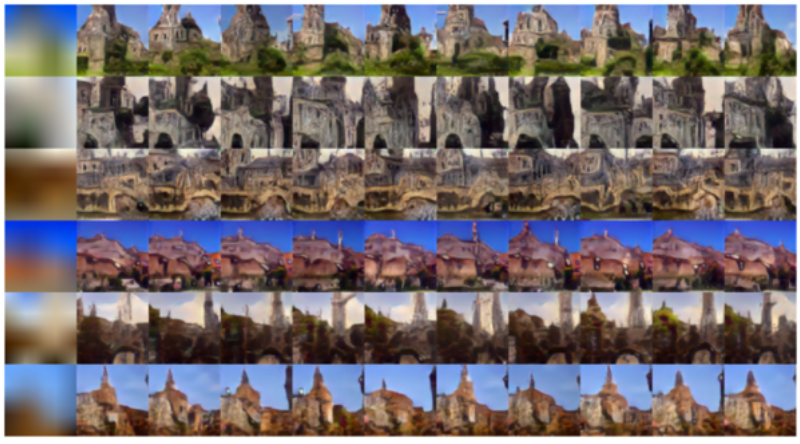
25、对抗性网络框架

继Ian Goodfellow关于对抗性样本的演讲之后，Yoshua Bengio 谈到了用两个系统相互竞争的案例。

系统D是一套判别性系统，它的目的是分类真实数据和人工生成的数据。

系统G是一套生成系统，它试图生成可以让系统D错误分类成真实数据的数据。

当我们训练一个系统时，另外一个系统也要相应的变的更好。在实验中这的确有效，不过步长必须保持十分小，以便于系统D可以更上G的速度。下面是“Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks”中的一些例子——这个模型的一个更高级版本，它试图生成教堂的图片。



26、arXiv.org编号

arXiv编号包含着论文提交的年份和月份，后面跟着序列号，比如论文1508.03854表示编号3854的论文在2015年8月份提交。很高兴知道这个。

原文链接：26 THINGS I LEARNED IN THE DEEP LEARNING SUMMER SCHOOL（译者/刘翔宇 审校/赵屹华、朱正贵、李子健 责编/仲浩）

关于译者： 刘翔宇，中通软开发工程师，关注机器学习、神经网络、模式识别。

推荐阅读相关主题： 自然语言处理 deep learning 机器学习 模式识别 海量数据 分布式

相关文章 最新报道

实时大数据分析：网络分析的一种新方法
LinkedIn技术高管Jay Kreps：Lambda架构剖析
如何利用Signiant SkyDrop进行数据大迁移？
专访贾磊：百度语音实现技术创新，打破汉语语音识…
深度学习： 推进人工智能的梦想
让机器搞懂100万种隐含语义，腾讯Peacock大规模…