Physcalの大魔導書

某HFUT的蒟蒻，ICT/VIPL的直博狗，SeetaTech的码农，还是当大魔导师好了(=￣ω￣=)。

深度神经网络结构以及Pre-Training的理解

Logistic回归、传统多层神经网络

1.1 线性回归、线性神经网络、Logistic/Softmax回归

线性回归是用于数据拟合的常规手段，其任务是优化目标函数：$h(\theta) = \theta + \theta_1 x_1 + \theta_2 x_2 + \ldots \theta_n x_n$

线性回归的求解法通常为两种：

①解优化多元一次方程（矩阵）的传统方法，在数值分析里通常被称作”最小二乘法"，公式$\theta = (X^T X)^{-1} X^T Y$

②迭代法：有一阶导数（梯度下降）优化法、二阶导数（牛顿法）。

方程解法局限性较大，通常只用来线性数据拟合。而迭代法直接催生了用于模式识别的神经网络诞生。

最先提出Rosenblatt的感知器，借用了生物神经元的输入-激活-传递输出-接受反馈-矫正神经元的模式，将数学迭代法抽象化。

并且在线性回归输出的基础上，添加了输出校正，通常为阶跃函数，将回归的数值按正负划分。

为了计算简单，此时梯度下降优化法被广泛采用，梯度优化具有计算廉价，但是收敛慢的特点（一次收敛，而牛顿法是二次收敛）。

为了应对精确的分类问题，基于判别概率模型$P(Y|X)$被提出，阶跃输出被替换成了广义的概率生成函数Logistic/Softmax函数，从而能平滑生成判别概率。

这三个模型，源于一家，本质都是对输入数据进行线性拟合/判别，当然最重要的是，它们的目标函数是多元一次函数，是**凸函数**。

1.2 双层经典BP神经网络

由Hinton协提出多层感知器结构、以及Back-Propagation训练算法，在80年代~90年代鼎盛一时。

经过近20年，即便是今天，也被我国各领域的CS本科生、研究生，其他领域（如机械自动化）学者拿来吓唬人。

至于为什么是2层，因为3层效果提升不大，4层还不如2层，5层就太差了。 [Erhan09]

这是让人大跌眼镜的结果，神经网络，多么高大上的词，居然就两层，这和生物神经网络不是差远了？

所以在90年代后，基于BP算法的MLP结构被机器学习界遗弃。一些新宠，如决策树/Boosing系、SVM、RNN、LSTM成为研究重点。

1.3 多层神经网络致命问题：非凸优化

这个问题得从线性回归一族的初始化Weight说起。线性家族中，W的初始化通常被置为0。

如果你曾经写过MLP的话，应该犯过这么一个错误，将隐层的初始化设为0。

然后，这个网络连基本的异或门函数 [参考] 都难以模拟。先来看看，线性回归和多层神经网络的目标函数曲面差别。

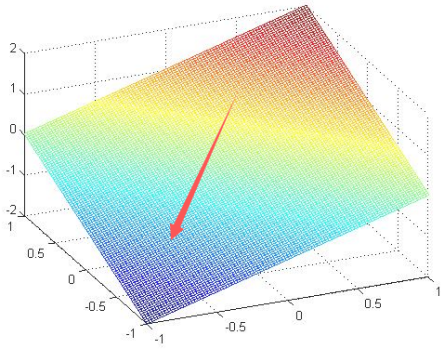
线性回归，本质是一个多元一次函数的优化问题，设$f(x,y) = x + y$

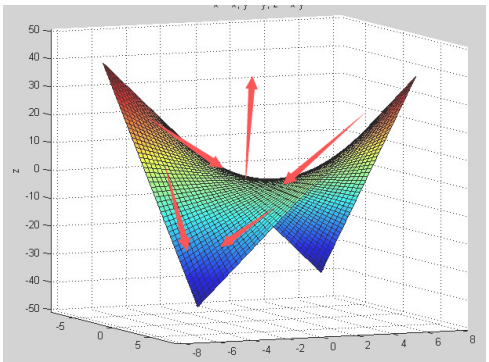
多层神经网络（层数K=2），本质是一个多元K次函数优化问题，设$f(x,y) = xy$

值得一提的是，SVM是个K=2的神经网络，但是Vapnik转换了目标函数，将二次优化变成了二次规划。

相对于盲目搜索的优化问题，规划问题属于凸优化，容易搜到精确解。但是缺陷是，没人能将三次优化变成三次规划。

也就是说，K>=3的神经网络，如果训练到位，是可以轻松超越SVM的。





在线性回归当中，从任意一个点出发搜索，最终必然是下降到全局最小值附近的。所以置0也无妨。

公告

这是一个属于轻松描写魔导书平凡日常的故事，请不要过度期待。还有，请保持屋内明亮离开电视3米以上再观看。(=￣ω￣=)

昵称：Physcal
园龄：3年
粉丝：302
关注：19
+加关注

最新随笔

1. [深度学习大讲堂]从NN...
2. [深度学习大讲堂]文化、..
3. 前馈网络求导概论(一)·S..
4. 从零开始山寨Caffe·拾贰.
5. 从零开始山寨Caffe·拾：.
6. 从零开始山寨Caffe·玖：.
7. 从零开始山寨Caffe·捌：.
8. 从零开始山寨Caffe·柒：.
9. 从零开始山寨Caffe·陆：.
10. 从零开始山寨Caffe·伍...

随笔分类(164)

ACM(113)
Haskell(3)
Qt(1)
并行计算(3)
机器学习理论(28)
机器学习系统设计(12)
模式识别(4)

随笔档案(163)

2016年12月 (1)
2016年6月 (2)
2016年3月 (8)
2016年2月 (5)
2015年11月 (1)
2015年10月 (1)
2015年9月 (2)
2015年8月 (7)
2015年7月 (1)
2015年6月 (8)
2015年5月 (19)
2015年4月 (3)

而在多层神经网络中，从不同点出发，可能最终困在（*stuck*）这个点所在的最近的吸引盆（*basin of attraction*）。 [Erhan09, Sec 4.2]

吸引盆一词非常蹩脚，根据百度的解释：它像一个汇水盆地一样,把处于山坡上的雨水都集中起来,使之流向盆底。

其实就是右图凹陷的地方，使用梯度下降法，会不自觉的被周围最近的吸引盆拉近去，达到局部最小值。此时一阶导数为0。从此训练停滞。

局部最小值是神经网络结构带来的挥之不去的阴影，随着隐层层数的增加，非凸的目标函数越来越复杂，局部最小值点成倍增长。 [Erhan09, Sec 4.1]

因而，如何避免一开始就吸到一个倒霉的超浅的盆中呢，答案是权值初始化。为了统一初始化方案，通常将输入缩放到$[-1,1]$

经验规则给出，$W \sim Uniform(-\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}})$，Uniform为均匀分布。

Bengio组的Xavier在2010年推出了一个更合适的范围，能够使得隐层Sigmoid系函数获得最好的激活范围。 [Glorot10]

对于Log-Sigmoid： $[-4 * \frac{\sqrt{6}}{\sqrt{LayerInput+LayerOut}}, 4 * \frac{\sqrt{6}}{\sqrt{LayerInput+LayerOut}}]$

对于Tanh-Sigmoid： $[\frac{\sqrt{6}}{\sqrt{LayerInput+LayerOut}}, \frac{\sqrt{6}}{\sqrt{LayerInput+LayerOut}}]$

这也是为什么多层神经网络的初始化隐层不能简单置0的原因，因为0很容易陷进一个非常浅的吸引盆，意味着局部最小值非常大。

糟糕的是，随机均匀分布尽管获得了一个稍微好的搜索起点，但是却又更高概率陷入到一个稍小的局部最小值中。 [Erhan10, Sec 3]

所以，从本质上来看，深度结构带来的非凸优化仍然不能解决，这限制着深度结构的发展。

1.4 多层神经网络致命问题：Gradient Vanish

这个问题实际上是由激活函数不当引起的，多层使用Sigmoid系函数，会使得误差从输出层开始呈指数衰减。见 [ReLu激活函数]

因而，最滑稽的一个问题就是，靠近输出层的隐层训练的比较好，而靠近输入层的隐层几乎不能训练。

以5层结构为例，大概仅有第5层输出层，第4层，第3层被训练的比较好。误差传到第1、2层的时候，几乎为0。

这时候5层相当于3层，前两层完全在打酱油。当然，如果是这样，还是比较乐观的。

但是，神经网络的正向传播是从1、2层开始的，这意味着，必须得经过还是一片混乱的1、2层。（随机初始化，乱七八糟）

这样，无论你后面3层怎么训练，都会被前面两层给搞乱，导致整个网络完全退化，真是连鸡肋都不如。

幸运的是，这个问题已经被Hinton在2006年提出的逐层贪心预训练权值矩阵变向减轻，最近提出的ReLu则从根本上提出了解决方案。

2012年，Hinton组的Alex Krizhevsky率先将受到Gradient Vanish影响较小的CNN中大规模使用新提出的ReLu函数。

2014年，Google研究员贾扬清则利用ReLu这个神器，成功将CNN扩展到了22层巨型深度网络，见 知乎 。

对于深受Gradient Vanish困扰的RNN，其变种LSTM也克服了这个问题。

1.5 多层神经网络致命问题：过拟合

Bengio在 Learning Deep Architectures for AI 一书中举了一个有趣的例子。

他说：最近有人表示，他们用传统的深度神经网络把训练error降到了0，也没有用你的那个什么破Pre-Training嘛！

然后Bengio自己试了一下，发现确实可以，但是是建立在把接近输出层的顶隐层神经元个数设的很大的情况下。

于是他把顶隐层神经元个数限到了20，然后这个模型立马露出马脚了。

无论是训练误差、还是测试误差，都比相同配置下的Pre-Training方法差许多。

也就是说，顶层神经元在对输入数据直接点对点记忆，而不是提取出有效特征后再记忆。

这就是神经网络的最后一个致命问题：过拟合，庞大的结构和参数使得，尽管训练error降的很低，但是test error却高的离谱。

过拟合还可以和Gradient Vanish、局部最小值混合三打，具体玩法是这样的：

由于Gradient Vanish，导致深度结构的较低层几乎无法训练，而较高层却非常容易训练。

较低层由于无法训练，很容易把原始输入信息，没有经过任何非线性变换，或者错误变换推到高层去，使得高层解离特征压力太大。

如果特征无法解离，强制性的误差监督训练就会使得模型对输入数据直接做拟合。

其结果就是，A Good Optimation But a Poor Generalization，这也是SVM、决策树等浅层结构的毛病。

Bengio指出，这些利用局部数据做优化的浅层结构基于先验知识（Prior）：*Smoothness*

即，给定样本(x_i, y_i)，尽可能从数值上做优化，使得训练出来的模型，对于近似的x，输出近似的y。

然而一旦输入值做了泛型迁移，比如两种不同的鸟，鸟的颜色有别，且在图像中的比例不一，那么SVM、决策树几乎毫无用处。

因为，对输入数据简单地做数值化学习，而不是解离出特征，对于高维数据（如图像、声音、文本），是毫无意义的。

然后就是最后的事了，由于低层学不动，高层在乱学，所以很快就掉进了吸引盆中，完成神经网络三杀。

特征学习与Pre-Training

2.1 Local Represention VS Distrubuted Represention

经典的使用Local Represention算法有：

①在文本中常用的：One-Hot Represention，这种特征表达方式只是记录的特征存在过，

2015年3月 (7)
2015年2月 (10)
2014年11月 (17)
2014年10月 (71)

队友の魔導書

esxgx
MaticsL
Pentium
战亿熊猫

而不能体现特征之间的关联（ 比如从语义、语法、关联性上 ）。且特征表示过于稀疏，带来维数灾难。

②高斯核函数：看起来要高明一些，它将输入悬浮在核中心，按照距离远近来决定哪些是重要的，哪些是不重要的。

将特征转化成了连续的数值，避免了表达特征需要的维数过高。但是正如KNN一样，片面只考虑重要的，忽视不重要的，

会导致较差的归纳能力，而对于高度特征稠密的数据（如图像、声音、文本），则可能都无法学习。

③簇聚类算法：将输入样本划分空间，片面提取了局部空间特征，导致较差的归纳能力。

④决策树系：同样的将输入样本划分空间问题。

以上，基本概括了数据挖掘十大算法中核心角色，这说明，数据挖掘算法基本不具备挖掘深度信息的能力。

相比之下，处理经过人脑加工过的统计数据，则更加得心应手。

因而，模式识别与数据挖掘的偏重各有不同，尽管都属于机器学习的子类。

为了提取出输入样本模式中的泛型关联特征，一些非监督学习算法在模式识别中被广泛使用，如PCA、ICA。

PCA的本质是：$[Input] => (Decompose)[Output] * [LinearBase]$。

即线性分解出特征向量，使得输入->输出之间做了一层线性变换，有用的关联特征信息被保留，相当于做了一个特征提取器。

分解出来的特征称之为Distrubuted Represention，NLP中词向量模型同样属于这类特征。

而RBM、AutoEncoder的本质是：$[Input] => (Decompose)[Output] * [Non - LinearBase]$。

显而易见，非线性变换要比线性变换要强大。

2.2 判别模型与生成模型

这是个经典问题，非监督学习的生成模型$P(X)$和监督学习的判别模型$P(Y|X)$之间的关系，到底是亲兄弟，还是世仇呢？

这个问题目前没有人能给出数学上解释，但是从生物学上来讲，肯定是关系很大的。

尽管CNN目前取得了很大的成功，但是也带来的很大忧虑，[知乎专栏] 的评论区，看到有人这么评论：

Ng说，你教一个小孩子认一个苹果，是不会拿几百万张苹果的图给他学的。

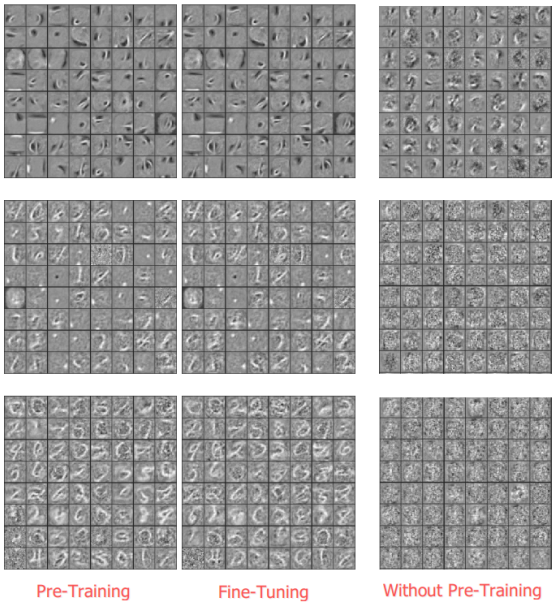
如果两者之间没有关系，那么$P(X)$初始化得出的参数，会被之后$P(Y|X)$改的一团糟，反之，则只是被$P(Y|X)$进行小修小改。

Hinton在DBN(深信度网络)中，则是利用此假设，提出了逐层贪心初始化的方法，进行实验：

①Stage 1：先逐层用RBM使得参数学习到有效从输入中提取信息。进行生成模型$P(X)$。(Pre-Training)。

②Stage 2: 利用生成模型得到的参数作为搜索起点，进行判别模型$P(Y|X)$。(Fine-Tuning)。

[Erhan10 Sec6.4] 将Stage1、Stage2、纯监督学习三种模型训练到完美之后的参数可视化之后，是这个样子：



可以看到，由于Gradient Vanish影响，较高层比较低层有更大的变动。

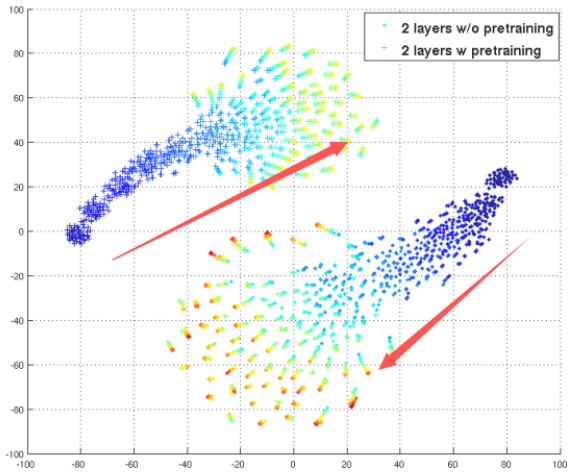
但是从整体上，Fine-Tuning没有太大改变Pre-Training的基础，也就是说$P(Y|X)$的搜索空间是可以在$P(X)$上继承的。

2.3 殊途同归的搜索空间

神经网络的目标函数到底有多复杂，很难去描述，大家只知道它是超级非凸的，超级难优化。

但是，带来的一个好处就是，搜索到终点时，可能有几百万个出发起点，几百万条搜索路径，路径上的权值有几百万种组合。

[Erhan09 Sec4.6] 给出了基于Pre-Training和非Pre-Training的各400组随机初始化W，搜索输出降维后的图示，他指出：

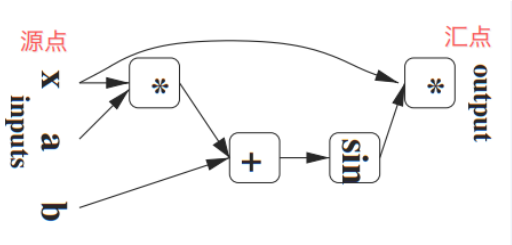


①Pre-Training和非Pre-Training的模型参数，在搜索空间中，从不同点出发，最后停在了不同的搜索空间位置。

②散开的原因，是由于陷入的吸引盆的局部最小值中，明显不做Pre-Training，散开的范围更广，说明非常危险。

可以看到，尽管两种模型取得了近似的train error和test error，但是搜索空间是完全不同的，参数形成也不同的。

从图论的网络流角度，多层神经网络，构成了一个复杂的有向无环流模型：



原本最大流模型下，每个结点的流量就有不同解。但是神经网络的要求的流是近似流，也就是说，近似+不同衍生出更多不同的解。

目前关于Pre-Training的最好的理解是，它可以让模型分配到一个很好的初始搜索空间，按照 [Erhan09, Sec 4.2] 中说法：

The advantage of pre-training could be that it puts us in a region of parameter space

where basins of attraction run **deeper** than when picking starting parameters

at random. The advantage would be due to a better optimization.

来自Bengio的采访稿的一段，~ Link~

通常来讲，我所知道的**模型都会受到不可计算性的影响**（至少从理论上看，训练过程非常困难）。

SVM之类的模型不会受到此类影响，但是如果你没有找到合适的特征空间，这些模型的普适性会受到影响。

（寻找是非常困难的，深度学习正是解决了寻找特征空间的问题）。

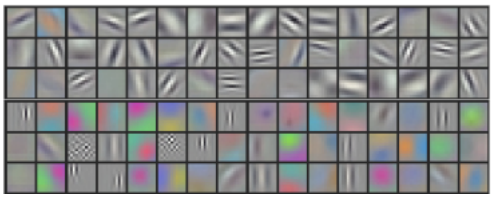
从Bengio的观点来看，Pre-Training带来的搜索空间，不仅有更好的计算（Optimation）性，还有更好的普适（Generalization）性。

2.4 特殊的特征学习模型——卷积神经网络

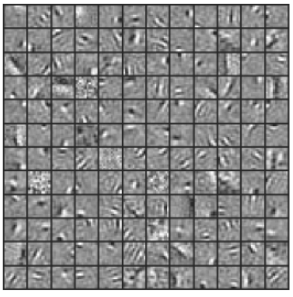
CNN经过20年发展，已经是家喻户晓了，即便你不懂它的原理，你一样可以用强大的Caffe框架做一些奇怪的事情。

Bengio指出，CNN是一种特殊的神经网络，参数少，容易层叠出深度结构。

最重要的是，它根据label，就能有效提取出稀疏特征，将其卷积核可视化之后，居然达到了近似生成模型的效果，确实可怕。



（CNN第一层卷积核可视化 by AlexNet）



（DBN第二层可视化）

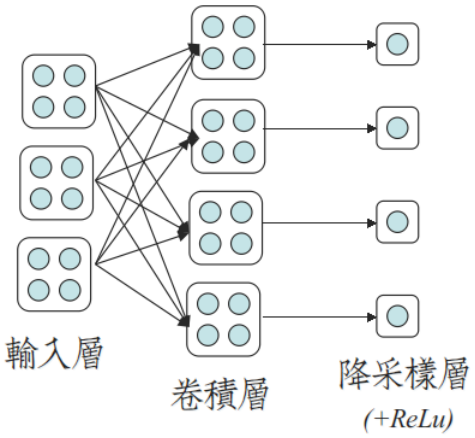
上面是对自然图片的学习结果，Hinton指出，自然图片的参数可视化后，应该近似Gabor特征。

CNN的强大，归结起来有四大创新点：

①块状神经元局部连接：CNN的神经元比较特殊，它是一个2D的特征图，这意味着每个像素点之间是没有连接的。

全连接的只是特征图，特征图是很少的。由于这种特殊的连接方式，使得每个神经元连接着少量上一层经过激活函数的神经元。

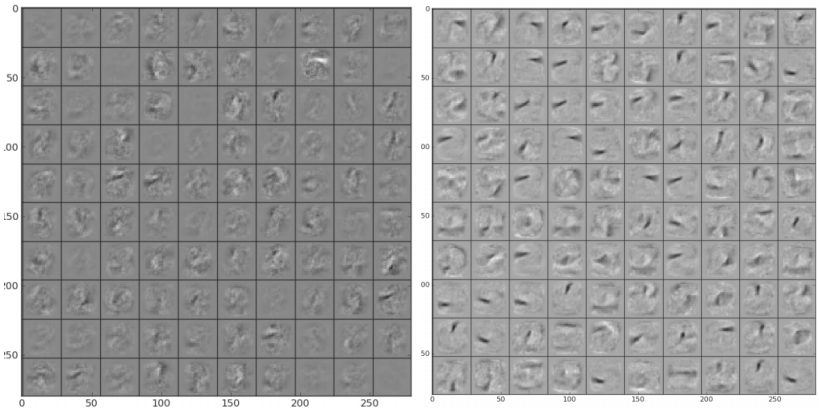
减轻了Gradient Vanish问题。使得早期CNN在非ReLU激活情况下，就能构建不退化的深度结构。



其中，降采样层、为非全连接的"虚层"，也就是说，真正构成压力的只有卷积层。

Hinton的Dropout观点来看，块状神经元使得一个卷积核在一张feature map中固定学习一部分输入，而不依赖全部输入。

这是为什么卷积核的可视化效果较好的原因。因为它模仿出了输入的局部特征。



左图是FC网络非Dropout，右图是Dropout。 [Hinton12]

②参数权值共享: 直观上理解是一个小型卷积核（如5x5)在30x30的图上扫描，30x30像素用的都是5x5参数。

实际原因是1D连接变成了2D连接，原来的点对点参数现在变成了块对块参数，且卷积核块较小。更加容易提取出鲁棒性特征。

当然，局部最小值问题也被减轻，因为参数量的减少，使得目标函数较为简单。

③卷积计算：对比原来的直接点对点乘，卷积方法能快速响应输入中的关键部分。

④降采样计算：添加了部分平移缩放不变性。

Alex Krizhevsky在 [Krizhevsky12] 对传统CNN提出的几点改进，使得CNN结构变得更加强大：

①将Sigmoid系激活函数全部换成ReLu，这意味着多了稀疏性，以及超深度结构成为可能（如GoogleNet)

②添加局部响应归一化层，在计算神经学上被称作神经元的侧抑制，根据贾扬清说法，暂时似乎没发现有什么大作用。 [知乎]

Caffe中之所以保留着，是为了尊敬长辈遗留的宝贵成果。

③弱化FC层神经元数：ReLu使得特征更加稀疏，稀疏特征具有更好的线性可分性。 [Glorot11]

这意味着FC层的多余？GoogleNet移除了FC层，根据贾扬清大牛的说法： [知乎]

因为全连接层（Fully Connected）几乎占据了CNN大概90%的参数，但是同时又可能带来过拟合（overfitting）的效果。

这意味着，CNN配SVM完全成为鸡肋的存在，因为FC层+Softmax≈SVM

④为卷积层添加Padding，使得做了完全卷积，又保证维度不会变大。

⑤使用重叠降采样层，并且在重叠降采样层，用Avg Pooling替换Max Pooling（第一仍然是Max Pooling）获得了5%+的精度支持。

⑥使用了Hinton提出的DropOut方法训练，减轻了深度结构带来的过拟合问题。

2.5 Pre-Training

Pre-Training的理论基石是生成模型$P(X)$。

Hinton提出了对比重构的方法，使得参数W可以通过重构，实现$arg\max_{W} \prod_{v \epsilon V} P(v)$

生成模型的好处在于，可以自适应从输入中获取信息，尽管可能大部分都是我们不想要的。

这项专长，可以用来弥补某些模型很难提取特征的不足，比如满城尽是FC的传统全连接神经网络。

FC神经网络最大的缺陷在于很难提取到有用的特征，这点上最直观的反应就是 A Good Optimation But a Poor Generalization。

最近刷微博看到有人贴1993年的老古董 [论文] ，大概内容就是：

证明神经网络只需1个隐藏层和n个隐藏节点，即能把任意光滑函数拟合到1/n的精度。

然而并没有什么卵用，这是人工智能，不是数学函数模拟。你把训练函数拟合的再好，归纳能力如此之差，仍然会被小朋友鄙视的。

为了证明FC网络和CNN的归纳能力之差，我用了Cifar10数据集做了测试，两个模型都没有加L1/L2。

FC网络：来自 [Glorot11] 提出了ReLU改进FC网络，预训练：正向ReLU，反向Softplus，数据缩放至[0,1]

网络[1000,1000,1000]，lr三层都是0.01，pre-lr：0.005,0.005,0.0005。

CNN：来自Caffe提供的Cifar10快速训练模型， [参考]

[FC]

负似然函数	1.69	1.55	1.49	1.44	1.32	1.25	1.16	1.07	1.05	1.00
验证集错误率	55%	53%	52%	51%	49%	48%	49%	49%	49%	49%

[CNN]

负似然函数	1.87	1.45	1.25	1.15	1.05	0.98	0.94	0.89	0.7	0.63
验证集错误率	55%	50%	44%	43%	38%	37%	35%	34%	32%	31%

可以看到，尽管已经做了Pre-Training，在相同似然函数值情况下，FC网络的Generalization能力真是比CNN差太多。

这也是为什么要使用多层网络，而不是1层网络的原因，为了得到更好的归纳。根据人视觉皮层的机理，多层次组合特征。

可以获得更好的归纳效果，而不是就为了那点训练error，那点函数优化拟合。

来自Bengio的采访稿的一段，~ Link~

全局逼近器并不会告诉你需要多少个隐含层。对于不确定的函数，增加深度并不会改进效果。

然而，如果函数能够拆分成变量组合的形式，深度能够起到很大作用。

无论从统计意义（参数少所需训练数据就少）来讲，还是从计算意义（参数少，计算量小）来讲。

模式识别三大领域：Speech，CV，NLP都已经被证明了，其高级特征可以由低级特征组合得到。

所以，在这些领域当中，使用深度结构多层叠加低级特征，以获取高级特征，会得到更好的Generalization。

训练error代表着样本的损耗度，一旦error逼近0，那么说明这个数据集已经玩完了。

如果此时仍然Generalization很差，真是神仙也救不了你。

深度学习，真的是很深度嘛？

3.1 DBN & Stack AutoEncoder

先让我们来看看百度词条是怎么解释 [深度学习] 的：

深度学习的概念由Hinton等人于2006年提出。基于深信度网(DBN)提出非监督贪心逐层训练算法，为解决深层结构相关的优化难题带来希望。

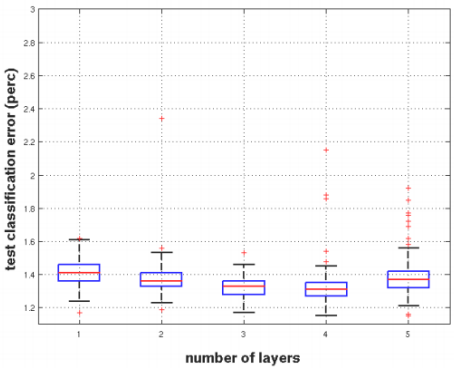
随后提出多层自动编码器深层结构。此外Lecun等人提出的卷积神经网络是第一个真正多层结构学习算法，它利用空间相对关系减少参数数目以提高训练性能。

DBN和SAE是原型结构都是基于FC网络的，使用的训练算法称为 "*Greedy Layer-Wise* Training"

然而，如果你仔细翻翻DBN和SAE的论文，发现其隐层结构不过就3~4层，和我们理解中的**“深度”**不知差了多远。

然而，最重要的是，通过Pre-Training之后，也就在FC网络基础上改进2~3%的错误率（Cifar10、MNIST），

Neuron	MNIST	CIFAR10	NISTP	NORB
With **unsupervised pre-training**				
Rectifier	**1.20%**	**49.96%**	**32.86%**	**16.46%**
Tanh	**1.16%**	**50.79%**	35.89%	17.66%
Softplus	**1.17%**	**49.52%**	**33.27%**	19.19%
Without **unsupervised pre-training**				
Rectifier	**1.43%**	**50.86%**	**32.64%**	**16.40%**
Tanh	1.57%	52.62%	36.46%	19.29%
Softplus	1.77%	53.20%	35.48%	17.68%



左图来自 [Glorot11] ，使用比DBN更优良的Stacked Denoising Autoencoder，对于Cifar10和MNIST依然是惨不忍睹。

右图来自 **[Erhan09, Sec 4.1]**，即便Pre-Training，在加上第五隐层之后，网络也开始退化。

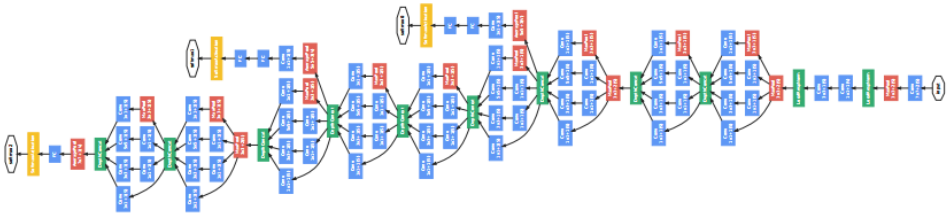
看来，FC结构真是祸害不浅，除了很难提取特征外，还给训练带来灾难，和CNN对抗？想得美！

所以说，Pre-Training给 解决深层结构相关的优化难题带来希望 而CNN才是 第一个真正多层结构。

至于基于FC结构的默认DBN和SAE，只能说不要想太多。

3.2 CNN倒是挺深度的

改进CNN在超深度方面确实惊人。 [Going deeper with convolutions By GoogleNet]



尽管外界质疑超深度CNN是一种过拟合 [知乎专栏] ,但我们起码离生物神经网络进了一步。

3.3 混合CNN与SDAE

DBN训练比SDAE慢(主要体现在GPU无法生成随机数，反复跑Gibbs Sampling很慢），而SDAE又比DBN效果好。

忘掉俗套的FC结构吧。我们有更先进的CNN结构。

最新进展

来自Hinton、Bengio、Lecun三位大师在Nature杂志AI开版的庆贺科普文 [Deep Learning] 。

其中提到了现在有不少人正在解决深度结构带来的局部最小值问题。如果数学上能够有突破，

能够有取代梯度下降，而且更加廉价、容易的训练算法，那么恐怕又是一个浪潮之巅。

这三位大师在DL的代表作品分别是，DBN&DBM，SAE&SDAE，CNN。

分别代表着深度学习的三大重镇：多伦多大学（Hinton组）、蒙特利尔大学（Bengio组）、纽约大学（Lecun组）。

还有一个比较容易忽视的大师是 Jürgen Schmidhuber，来自慕尼黑工大。

代表作是1997年提出的LSTM，一种解决了Gradient Vanish的深度RNN结构，在NLP、Speech领域非常火热。

他的学生Alex Graves最近加入了Hinton组，并且贡献出了预出版的RNN&LSTM的学习资料。 ~Link~

分类: 机器学习理论

Physcal
关注 - 19
粉丝 - 302
+加关注

6 0

« 上一篇： 词向量概况
» 下一篇： 基于Theano的DL的开源小框架：Dragon

posted @ 2015-06-14 19:06 Physcal 阅读(19724) 评论(1) 编辑 收藏

评论列表

#1楼 2017-07-25 14:44 Yuki_i

写的真好

支持(0) 反对(0)