# [Deep Learning 优化小结](http://blog.csdn.net/lifeitengup/article/details/10212867)

分类： [机器学习](http://blog.csdn.net/LiFeitengup/article/category/1293585) [Deep Learning](http://blog.csdn.net/LiFeitengup/article/category/1549309)2013-08-23 11:50 3523人阅读 [评论](http://blog.csdn.net/lifeitengup/article/details/10212867#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/lifeitengup/article/details/10212867#report)

目录[(?)[+]](http://blog.csdn.net/lifeitengup/article/details/10212867)

胡乱写一点，做个记录；不保证可读性。

Hinton 06 年的 A fast learning algorithm for deep belief nets首先提出了 pre-training + fine-tuning的优化 deep network的方法策略；

mutil-layer network多年前就提出，但优化一直是个问题；非凸且参数巨多， 单隐藏层NN的优化算法Backpropagation Algorith往往得不到较好的局部最优解；

pre-training 往往能够得到比较接近局部最优解的初始迭代点，然后使用fine-tuning来微调pre-training得到的参数。

pre-training是一种策略，不是具体的优化方法。pre-training使用无监督学习来训练一个单隐藏层NN，输入输出相同。从而学习到参数，然后保留第一层；递归进行多次。最后使用 有监督学习来trian一下 feature-label的参数，可以使用sigmoid/softmax regression等。

单隐藏层NN Autoencoder/Sparse Autoencoder等；另 RBM等。

RBM 用Hinton 02年的一篇文章提出的方法来trian 基本思想是 极大似然的时候不断抬高下界 做个两三层即可

单隐藏层NN 的训练方法有 梯度下降(GD)/随机梯度下降(SGD)/mini-Batch GD/L-BFGS/共轭梯度(CG)

其实单隐藏层 一般不需要这么多方法， 但是Deep Learning往往需要 大量的训练数据 才能刻画数据的分布，所以需要 优化技巧来提高训练速度。

这里有许多参数要调  比如Sparse Autoencoder如果使用KL-熵来做优化的目标函数 就需要选择 优化比例lambda等

GD/SGD中需要调节 learning rate 尤其是SGD；mini-Batch除步长外 还需调节 batchsize epoch等参数/L-BFGS 如果内存占用太多 还需调节选用个数的参数。

此外优化 还有添加 动量 or 平均随机梯度的方法等。

fine-tuning也是用 SGD/LBFGS/CG来微调各层参数，但是这种微调 其实对底层的参数影响比较小，多半调在了最上层的参数上；不同的task 会带来不同的微调效果 尽管他们使用相同的pre-training

近来，Hinton 还提出 使用 dropout的技术来trian，有点 Denoise Autoencoder的意思；Bengio顺带搞出来 maxout等技术。

Ng组 11年 有篇文章——On Optimization Methods for Deep Learning分析了SGD/L-BFGS/CG

华盛顿大学的 Pedro-Domingos等 提出的 Sum-Product Networks的优化 完全不是上面套路，reuse很多中间结果，据说训练很快 没细看

关于优化等问题 可以参考    部分章节

**[Neural Networks: Tricks of the Trade](http://link.springer.com/book/10.1007/978-3-642-35289-8/page/1" \t "_blank)**

Second Edition

Deep Learning在图像上 为了克服图像尺寸、平移、旋转等的问题会使用 patch/convolution-pooling等技术

早期Deep learning 基本用在了图像/语音的分类问题和提取特征上，近年有拓展应用领域的趋势：图像分割/图像检索等。

我也用 Stacked （Sparse） Autoencoder/Convolution NN等 在MNIST上做了一些实验，基本在 97.5%+的正确率上，更高的准确率，可以通过增加训练参数(旋转图像/平移图像等)、数据的预处理(白化等)、更细致的调参数、更换模型等来提高。