工作汇报

Shilong Zhang

9.24至 10.7

1.深度学习理论相关
1.将基本所有GAN的基础知识补上了。
2.读了几个经典GAN网络的代码。
3.CS231。
4.一些机器学习理论相关

1 deep learning理论

并非浅层网络表达能力差 并非深层网络相比浅层网络有更好的表示能力，能拟合更复杂的函数，实际上，浅层网络只要有足够的参数，亦可以拟合任意复杂的函数(前提是具有一定的平滑性，是L-Lipschitz的 ,给定任意小的误差ϵ,利用的足够多的Relu激活函数就可以组合出来)，其实，深层网络真正的作用的在拟合相同复杂的函数时候，需要的参数更少。

深层网络的好处 之前证明利用浅层网络可以用Relu来拟合任意L-Lipschitz时候，有一个重要的指标就是网络可以组合出来的分段的数目，我们可以证明相同神经元数目下，两层之间分段可以按指数级别增加，而每个层仅仅可以让分段以线性增加。即有关理论研究给出了网络可以产生的Linear Pieces给出的lower Bound，K是每层的宽度，H是深度的话，可以至少产生K^H个piece,有理论表明，如果想用一个浅层网络逼近深层网络，往往浅层网络需要非常宽(即参数需要非常多)。

lowlayer 的参数是最重要的 后面网络仿佛是在对折lowlayer产生的pieces，nips上有验证实验，1 不同层的参数加噪声，越往前的层数对噪声更敏感。

2.单层训练，即除去某一层外都随机产生，那么仅仅训练第一层比仅仅训练其他层有高的多的正确率。

通过NN的到底是如何拟合fuction的？ 可以找到x^2的一种通过ReLU进行拟合的方法，通过x^2组合出mutiply功能，从而可以拟合任意多项式，那么就可以拟合任意的有多项式展开的函数。

optimization deep learning 即如何在网络能表达的函数集中选出最接近target function的,nn的loss function 并不是convex的，即寻求最有解是的困难的，但是很幸运，虽然不是convex的，但是近期研究表明，loss fuction似乎具有很好的性质。有实验表明，deep learning 的local minima与 globle minima几乎是差不多的，loss function 的梯度为零的点在 training error 较大的点有较大的可能是saddle point ，在training error较小的点有较大的可能是local minima,但是在large network 里面local minima是很少见的。

影响最后收敛的结果的因素 发现采用不同的initialization与不同的优化算法，经过训练之后，虽然最后的loss function的值差不多，但是最后的实际上网络参数差别非常多，并且哪怕是在训练的最后阶段更换优化算法，最后的收敛的loss都会不同，也可以说，似乎每种优化算法都会有他自己的特性，在不知道哪种优化算法更是和问题时候，可以采用不同的算法进行优化。

Generation Gap 原本机器学习理论在训练集上loss差不多的时候，我们应该选择capacity更小一些的model，这样trainin loss 与eneralization loss在概率上差别更小。大门时deep learning 中有神奇的现象，在training loss已经收敛时候(比如降到0)，这时候继续增大paramter的数目，在 test set上的误差还会下降。有人认为deep network是自带regularization的，deep network 会倾向于用一种比较简单的方式去拟合数据，也就是说虽然deep network的capacity非常强，代表的的function space非常大，但是它并没有我们想的那么容易 over fitting。但是，当数据比较差的时候，还是会出现deep network 强记数据的情况，即over fitting，发现L2 norm 在network 过拟合的时候会比较大，可以用来防止过拟合。

泛化能力与sensitivity 可以用jocobian矩阵来表示出网络对数据的sensitivity，实验表明sensitivity与网络对数据的泛化能力是正相关的，network对分布在training data附近的数据sensitivoity比较低，也就是对training 附近的数据预测比较稳定，因此做数据扩充是有必要的。那么对一批test data,就可以先计算数据的sensitivity,让网络仅仅预测sensitivity比较低的数据(答对的概率较大)，而sensertivity比较大的数据就单独交给人来标记预测，还是有很大实用性的。

泛化能力与sharpness 现在认为loss函数的local minima周围比较sharp的泛化能力会比较差，而比较flat的泛化能力会比较好，这貌似可以解释deep learning的泛化能力会比较好，因为initialize时候很容易初始化在把比较平坦的地方，因为flat的相比sharp的还是在高维空间中占据更大的空间。而sharp还是flat一个重要影响因素是训练时候的batch size，小batch size的loss local minima 一般比较flat，泛化能力就比较好(是有实验做这样的事情)，但是最近有文章论证shape的local minima 也有很好的泛化能力。

Tips for training DNN 1.选择适合的loss function ，不同的loss function在梯度下降的时候，其表面surface 是不一样的，有的比较陡峭，易于梯度下降，有的比较平坦，难以训练。2. 选用mini-batch，在计算能力有限的时候，采用mini-batch参数更新更快。而且实验表明，采用mini-batch训练的网络往往具有更好的泛化能力。3. 采用合适的activation function 解决梯度消失问题。4. Learning rate 的选择，过大可能错过local minima，太小可能会收敛过慢，可以采用learning rate 随着epoch的增加进行减少在两者间进行权衡，其他的还有如同Adagrad，可以根据梯度大小来调整learning rate 。 5.Momentum 可以使得参数更新更平稳，避免震荡，更新更快，也有利于跳出不是很好的local minima。

避免overfitting 的方法 1.data augmentation,如图片，加噪声，旋转等等。2.Early stopping 3.weight decay,促使无用的参数消失。4.dropout层 5.Network Structure

2 机器学习理论

3 Paper Reading

3.1 An overview of gradient descent optimization algorithms

Batch gradient descent 最原始的梯度下降，每次梯度下降需要计算整个dataset的loss，更新慢，无法进习惯online training。

Stochastic gradient descent(SGD) 快速更新，每个example 更新一次，可以update online,但是每次更新是不稳定的，导致目标函数震荡严重。同时，batch gredient descent在initialization在哪个‘流域’那么最后就会下降到哪个local minima，但是SGD可能在梯度下降的时候会跳出当前的‘流域’，找到更好的一个local minima，

Mini-batch gradient descent 1.减少了每次更新loss的方差，使得更新过程更为稳定。2.数学上对于运算的优化使得对于mini-batch的梯度运算十分的高效。

Momentum 应该是增加每次SGD在relevant direction方向上的位移，就可以减少SGD的震荡，使得convege速度变快。

Nesterov accelerated gradient(NAG) 将求梯度的位置换为利用动量项目预测的位置，经过推导，实质是考虑了loss fuction的二阶导，推导链接`https://zhuanlan.zhihu.com/p/22810533`.

Adagrad 自适应选择learning rate，对更新较快的参数(一直梯度较大)的，减小步长，对梯度较小的参数增加步长，并且随着更新的次数增加，学习步长总体趋势都是下降的，可以理解为一开始距离local minima还比较远，可以快速下降，到了local minima附近应该放慢学习率，避免震荡。但是发现，因为learning rate 一直下降，在一些网络中出现训练停止的现象。

Adadelta 为了解决adagrad学习率下降过快，其对于过往梯度的累积通过一个时间窗实现，没有储存W个状态的梯度值，而是使用类似动量因子的平均方法，并利用一阶的优化算法去近似牛顿法(二阶法)。

RMSprop 解决RNN效果很好。

Adam 解决自适应学习速率下降过快的问题的同时，同时当前梯度也借用动量的思想进行指数移动平均。同时平滑时候初始化为零会使平滑结果偏向0，因此采用了修正来抵消偏差。adam在大多数情况下表现较好。

Adamax 对梯度的自适应分母采用无穷范数,无需像adam一样修正初始化偏差。

NAdam 更新中的一项平滑项改为当前已经更新的项。

which to use 1.数据稀疏使用自适应学习率的算法 2.Adam might be the best overall choise 3.如果网络复杂并且想快速收敛，建议使用自适应学习率算法。

Additional strategies for optimizing SGD 1.Shuffling and Curriculum Learning,shuffle the training data after every epoch,框架已经内置，curriculum learning,example的顺序也十分重要，有论文显示train lstm 时候先简单后复杂有更好的训练结果。3.batch normalization 4.early stopping 5.Gradient noise，起初在梯度上加上一个均值为零，方差退火的噪声，发现可以使得网络发现不同的local minima，对poor initialization也表现出较好的鲁棒性。