第六讲 训练神经网络一

2019年4月30日 10:54

1.激活函数

1.1 大纲

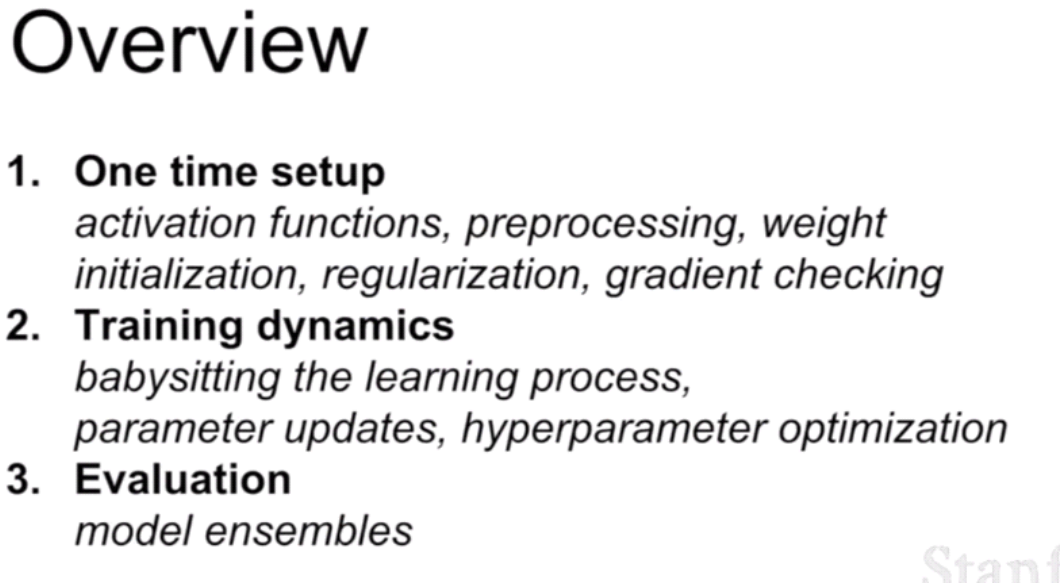


图6.1.1 训练神经网络小节大纲

1.搭建

激活函数、数据预处理、权重初始化、正则项、梯度检查

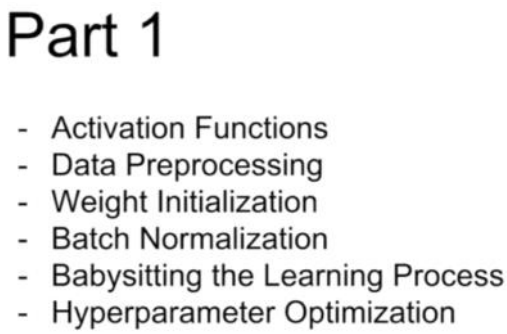
2.训练中的动态变化

如何监测整个学习过程、参数(权重)更新、超参数优化

3.模型评估

模型集成

1.2 part 1



1.2.1 激活函数

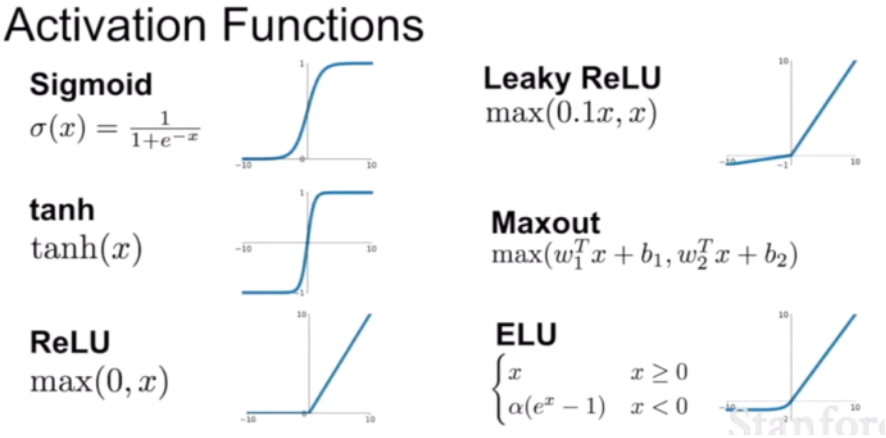


图6.1.2 常用激活函数

① **sigmoid： historically popular!**

缺点：1.会使**饱和神经元梯度消失**

2.输出不是以0为中心的分布

3.函数中包含exp运算，代价太大

当输入远大于10的时候sigmoid函数的梯度接近0

② **tanh： 优点：0中心分布**

缺点：存在梯度消失问题

③**ReLU：优点：不会出现饱和**

计算代价小

缺点：非0中心分布

dead ReLU

④**Leaky ReLU：max(0.1x,x)** **变种：PReLU = max(px,x)**

优点：不会出现饱和

计算代价小

收敛速度快

不会dead

讲师经验之谈!

TLDR: In practice:

- Use ReLU. Be careful with your learning rates
- Try out Leaky ReLU / Maxout / ELU
- Try out tanh but don't expect much
- Don't use sigmoid

1.2.2 数据预处理

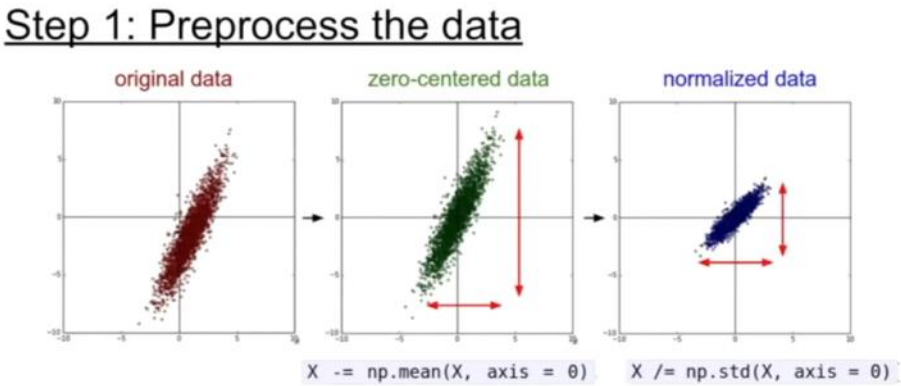


图6.1.3 数据预处理

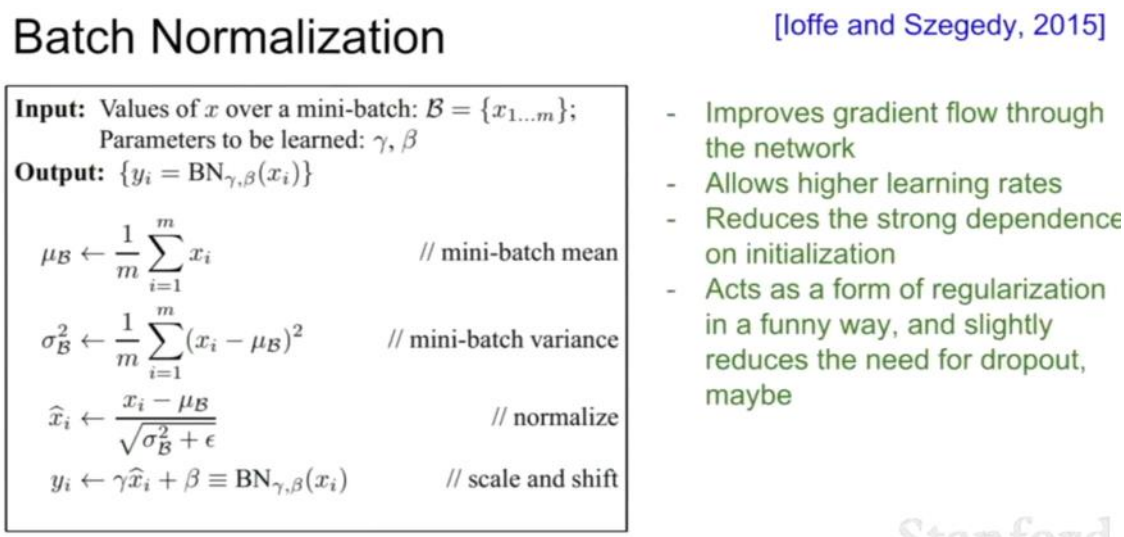
1. 数据分布中心化处理
2. 归一化处理
3. PCA 主成成分分析

   降维，分析数据相关性
4. 白化

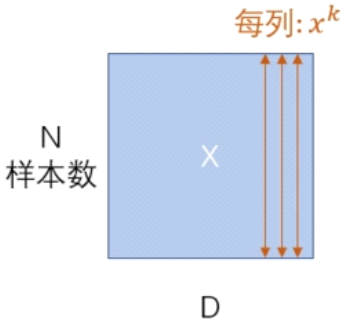
2.批量归一化

《Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift》

背景：在网络训练的过程中，参数的更新会导致网络的各层输入数据的分布不断变化，那么各层在训练的过程中就需要不断的改变以适应这种新的数据分布，从而造成网络训练困难，收敛变慢（而且网络越深越难），在论文中这个问题被称为“Internal Covariate Shift”。为了解决这个问题出现了批量归一化的算法，他对每一层的输入进行归一化，保证每层的输入数据分布是稳定的，从而加速训练。



$$\hat{x}^{(k)} = \frac{x^k - E[x^k]}{\sqrt{Var[x^k]}}$$



计算数据每个维度（每列）的均值和方差:

第k维的平均值$E[x^k] = \frac{1}{N}\sum_{j=1}^{N} x_j$

第k维的标准差$\sqrt{Var[x^k]} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_j - E[x^k])^2}$

再用上面的公式对每一维进行归一化

BN:	https://blog.csdn.net/hjimce/article/details/50866313