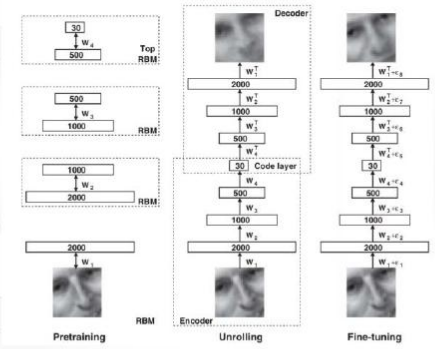
参数初始化

目录

参数初始化

参数初始化

◆ 深度学习兴起的开端：逐层无监督预训练方法，解决深度模型初始化、训练不稳定的问题

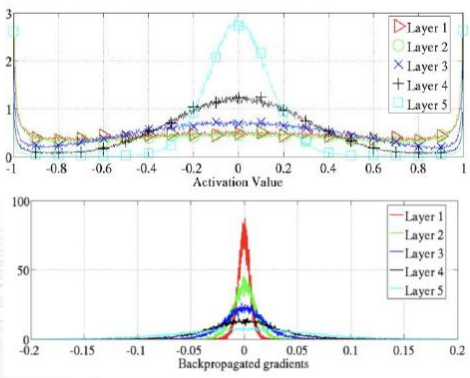


- 逐层无监督预训练玻尔兹曼机，首先把数据向量x和第一层隐藏层作为一个RBM，训练出该RBM的参数。然后固定训练好的参数，将h1视作可见向量，把h2视作隐藏向量，训练第二个RBM，依次类推。
- 利用无监督的RBM网络来预训练一个深度信念网络，然后将其作为一个多层前馈神经网络的初始化权重，再使用反向传播算法微调，“预训练+微调”有效解决深层模型难以训练的问题。

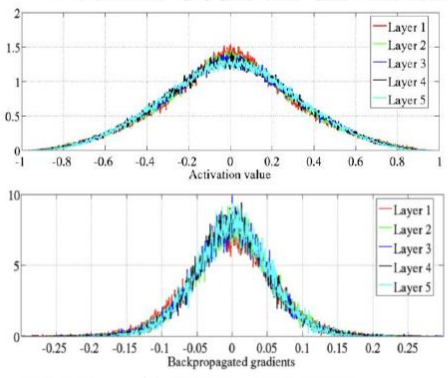
2006年，Geoffrey Hinton等人论文“Reducing the dimensionality of data with neural networks”

什么是好的参数初始化

- 各层激活值不为0也不出现饱和；每层的权重方差、梯度与层数无关，更有利于优化



标准初始化，各层激活值、梯度差异很大



Xavier初始化，各层激活值、梯度差异很小

Glorot X, Bengio Y “Understanding the difficulty of training deep feedforward neural networks”

常见初始化方法

简单初始化

◆ 全零初始化与随机初始化方法

全零初始化：在第一次更新的时候，除了输出层之外，所有的中间层的节点的值都为零。一般神经网络拥有对称的结构，导致进行第一次误差反向传播时，更新后的网络参数将会相同，在下一次更新时，相同的网络参数提取不到有用的特征。即使是对于非对称的网络结构，这样的随机参数也不利于接下来的优化。

随机初始化：np.random.randn(n)，用随机值进行初始化。参数的初始值不能取得太小，因为较小的参数在反向传播时会导致过小的梯度，会产生梯度弥散问题，降低参数的收敛速度。而过大不仅会造成震荡，对于Sigmoid等激活函数也会进入梯度饱和区。

标准初始化

◆ 固定方差的初始化方法：输出方差与输入方差有稳定的关系

均匀分布：　~　　概率密度函数　　—，（　），期望等于—，方差等于——
，其他

（）　———　—→　　（　）　　（）－　（）

训练时需要配合归一化方法使用

Xavier初始化

◆ 方差缩放(Variance Scaling)的初始化方法

Glorot条件：各层的激活值和梯度的方差在传播过程中保持一致，激活函数对称，每层的输入均值都是0。

前向传播：（ ） （ ） （ ）

同时考虑前向与反向过程：

$$\forall i, \quad n_i Var[W^i] = 1$$
$$\forall i, \quad n_{i+1} Var[W^i] = 1$$

→ （ ） ——— ——— → $\left[\frac{\sqrt{\ }}{\sqrt{\ }} \quad \frac{\sqrt{\ }}{\sqrt{\ }} \right]$

根据神经元数量自动调整均匀初始化方差分布

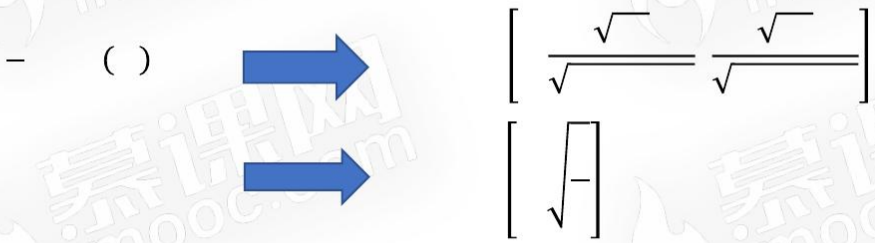
Xavier初始化

◆ 不同激活函数与分布下的激活形式

激活函数	均匀分布[-a,a]	高斯分布
Logistic函数	$\frac{\sqrt{\ }}{\sqrt{\ }}$	$\frac{\sqrt{\ }}{\sqrt{\ }}$
Tanh函数	$\frac{\sqrt{\ }}{\sqrt{\ }}$	$\frac{\sqrt{\ }}{\sqrt{\ }}$

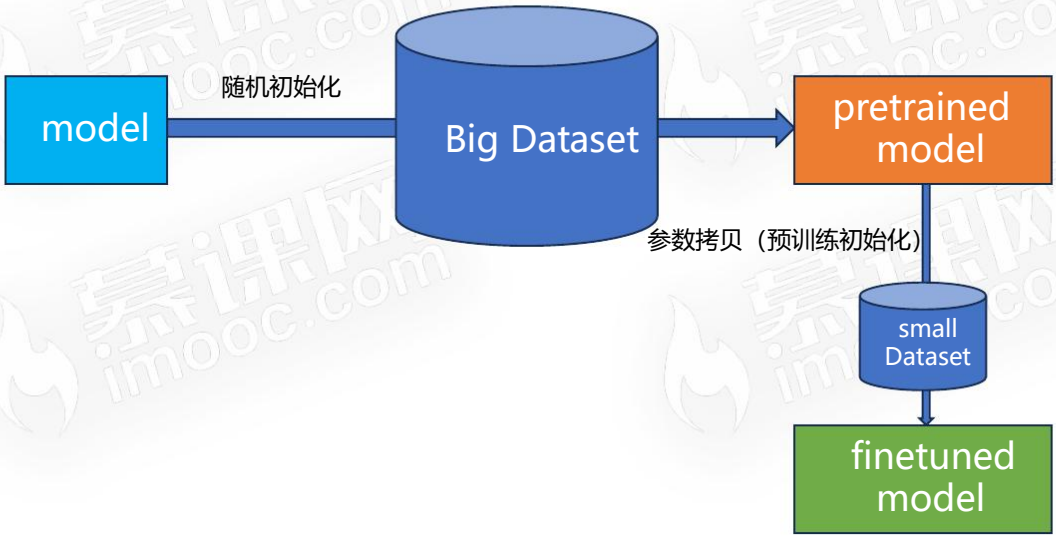
MSRA初始化

◆ 更适合ReLU的初始化



思考

◆ 如何获得更好的初始化?



下次预告：标准化