目录

1 WeightInitialization

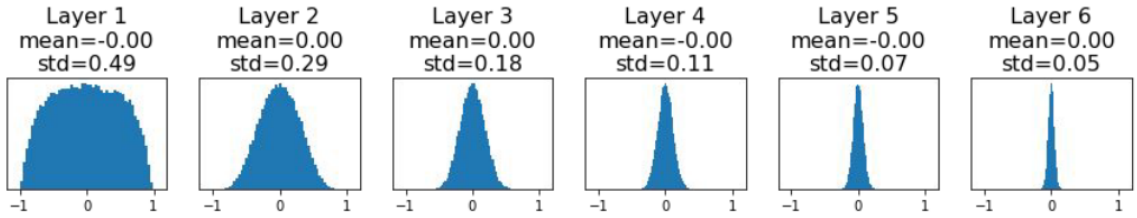
神经网络的权重初始化仍然是一个未解决的问题，但是我们可以鉴别出哪些权重初始化的方法不好，哪些方法会给神经网络带来比较好的效果。

1.1 Bad idea

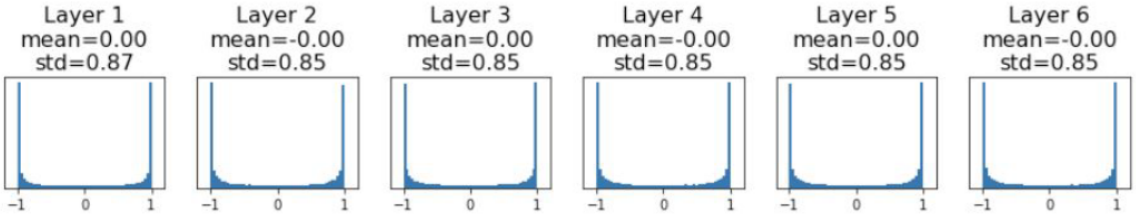
神经网络的权重不能是初始化为相同的值，如：将MLP的权重全部初始化为0或者其他某一个常数。如果初始时将所有的权重值设置为一样的常数，由于每一个神经元都接受相同的输入，因此权重相同会导致神经元输出相同的值，从而在反向传播计算梯度的时候，所有的神经元的梯度值一样，更新后每一个神经元的权重一样，这实际上可以看成是一个神经元构成一个层，不仅没有起到增加神经网络表达能力的目的，反而浪费了计算资源，因此我们不能把神经网络的权重设置成相同的值，因此，权重的随机初始化是必要的。

2 激活值的分布

假设我们采用高斯分布来初始化网络权重，使用sigmoid作为激活函数，初始化的时候只是简单地指定整个网络所使用的高斯分布的均值和方差，即：整个网络的所有层的权重的分布均一致。该初始化方法似乎并不存在什么问题，因为随机初始化打破了常数初始化方法的权重对称问题，能够有效地增强模型的表达能力，但是，对于深层的神经网络而言，会带来“梯度消失”问题。

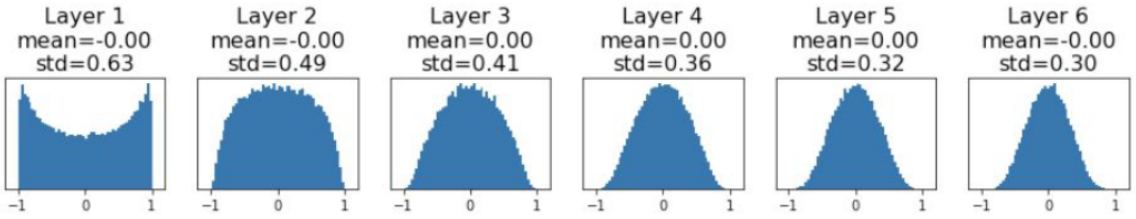


如图是一个简单的6层神经网络（MLP）的每一层的激活函数值的分布，可以看出，越深的层，其分布越趋近于零，若以sigmoid为激活函数，其激活函数值分布如图所示，可以看出，其分布趋近于0和1，由于sigmoid函数的饱和特性，权重的梯度将趋近于零，而downstream gradient又依赖于 upstream gradient，因此网络中的权重梯度将几乎全部趋近于零，这便是“梯度消失”现象。



2.1 Xavier Initialization

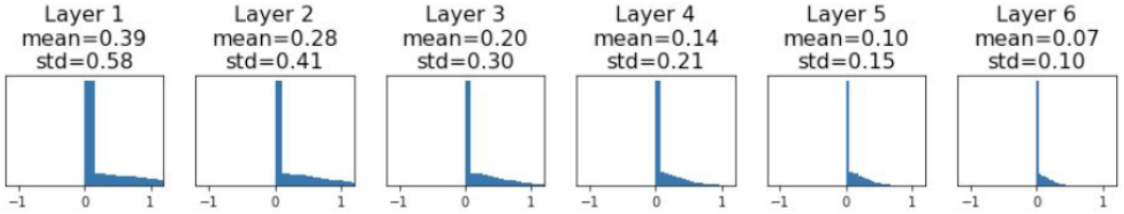
Xavier Initialization 能够避免上面说到的问题，其使用的高斯分布的均值仍然为0，但标准差是根据输入的维度确定的，即：$\sigma = \frac{1}{\sqrt{D_{in}}}$.同样的模型，使用 Xavier Initialization 后网络的每一层的激活函数值分布如图所示。



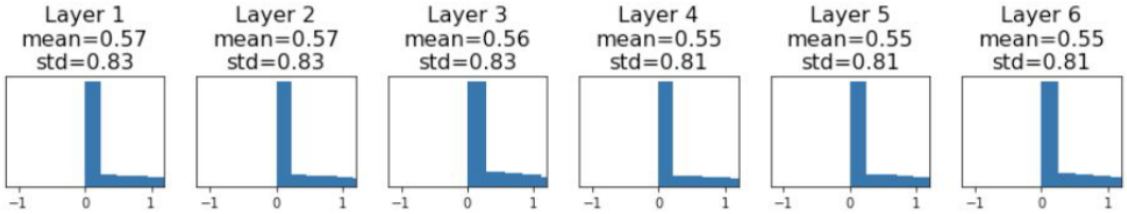
可以看出，Xavier Initialization 能够使激活函数值的分布范围更广泛，这一结论虽然是从全连接层得到的，但是也可以用于卷积神经网络，此时 $D_{in} = kernel\_size^2 * in\_channels$， $\sigma = \frac{1}{\sqrt{D_{in}}}$

2.2 Kaiming Initialization

需要注意的是，Xavier Initialization 在 sigmoid 上表现良好，但是在 ReLU 上表现就并不是很让人满意，其各层的激活函数值分布如下：



这时候我们可以看到，越深层的网络的其分布越靠近0，由上面的讨论我们知道，这显然不利于神经网络的权重更新，Kaiming Initialization 采用均值为0，方差 $\sigma = \sqrt{\frac{2}{D_{in}}}$, 使用 Kaiming Initialization 后，各层的分布如下：



可以看出，其分布显然更加均匀，能够避免梯度消失问题，是很多神经网络的首选的初始化方法。