1. 为什么要Normalization？其作用?

- 如果每次送入训练的数据分布都不同，显然会给网络的训练带来困难

  所以可以通过类似Batch norm的操作，将分布统一

- 我们知道sigmoid激活函数和tanh激活函数存在梯度饱和的区域，其原因是激活函数的输入值过大或者过小，其得到的激活函数的梯度值会非常接近于0，使得网络的收敛速度减慢。传统的方法是使用不存在梯度饱和区域的激活函数，例如ReLU等。

  归一化也可以缓解梯度饱和的问题加速收敛过程，它的策略是在调用激活函数之前将值归一化到梯度值比较大的区域。

2.Batch Normalization

> BN是对batch中所有样本的第N个词向量做归一化

BN 强行将数据拉回到均值为0，方差为1的正太分布上，**这样不仅让每个batch的数据分布一致，而且避免发生梯度消失**。

BN算法过程：

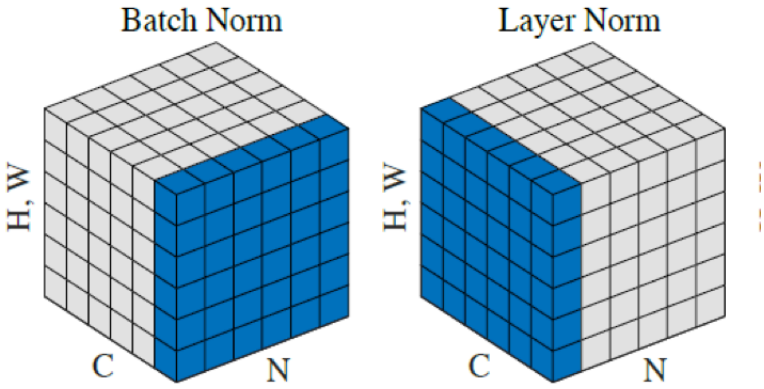
- 沿着通道计算每个batch的均值u
- 沿着通道计算每个batch的方差σ^2
- 对x做归一化，x’=(x-u)/开根号(σ^2+ε)
- 加入缩放和平移变量γ和β ,归一化后的值，y=γx’+β
  - 如果直接做归一化不做其他处理，将参数拉到非饱和区域（线性区域），神经网络是学不到任何东西的，但是加入这两个参数后，事情就不一样了，先考虑特殊情况下，如果γ和β分别等于此batch的方差和均值，那么参数不就还原到归一化前的x了吗，也即是缩放平移到了归一化前的分布，相当于batchnorm没有起作用，β 和γ分别称之为 平移参数和缩放参数 。这样就保证了每一次数据经过归一化后还保留的有学习来的特征，同时又能完成归一化这个操作，加速训练。
  - 这两个参数是用来学习的参数。

缺点：

1. 不适合batch_size较小场景： batch_size较小的话，方差、均值具有局限性
2. 对于深度相同的CNN网络很方便，但是不适合sequence长度不同的RNN，可能存在一个特殊sequence比其他sequence长很多。

2. Layer Normalization

> LN是对batch中某一个样本所有词向量做归一化，因此可以用于batchsize为任意大小的网络和RNN。


Batch Norm
Layer Norm
H, W
C
N
H, W
C
N