**Normalization详解**

为什么需要Normalization?

答：由于ICS的问题存在，“理论正确”的方法就是对每一层的数据都进行白化操作。然而标准的白化操作代价高昂，特别是我们还希望白化操作是可微的，保证白化操作可以通过反向传播来更新梯度。因此，以 BN 为代表的 Normalization 方法退而求其次，进行了简化的白化操作。

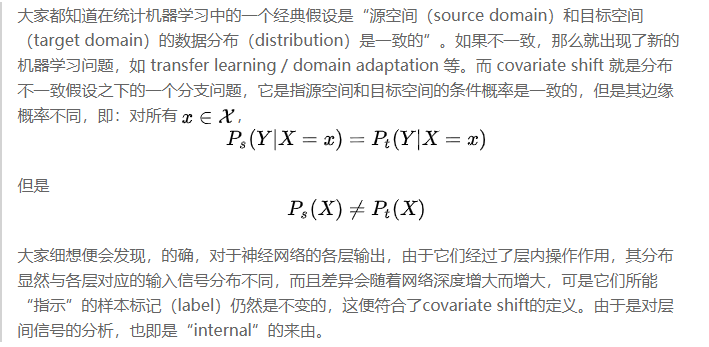
1. 独立同分布

机器学习模型中，独立同分布并非所有模型的必然条件，但独立同分布的数据可以简化常规机器学习模型的训练、提升机器学习模型的预测能力，已经是一个共识。

因此，“白化”是机器学习的一个重要的数据预处理步骤，原因如下：

1. 去除特征之间的相关性 —> 独立；
2. 使得所有特征具有相同的均值和方差 —> 同分布
3. Internal Covariate Shift

在深度学习中，深度神经网络涉及到很多层的叠加，而每一层的参数更新会导致上层的输入数据分布发生变化，通过层层叠加，高层的输入分布变化会非常剧烈，这就使得高层需要不断去重新适应底层的参数更新。为了训好模型，我们需要非常谨慎地去设定学习率、初始化权重、以及尽可能细致的参数更新策略。Google 将这一现象总结为 Internal Covariate Shift，简称 ICS。

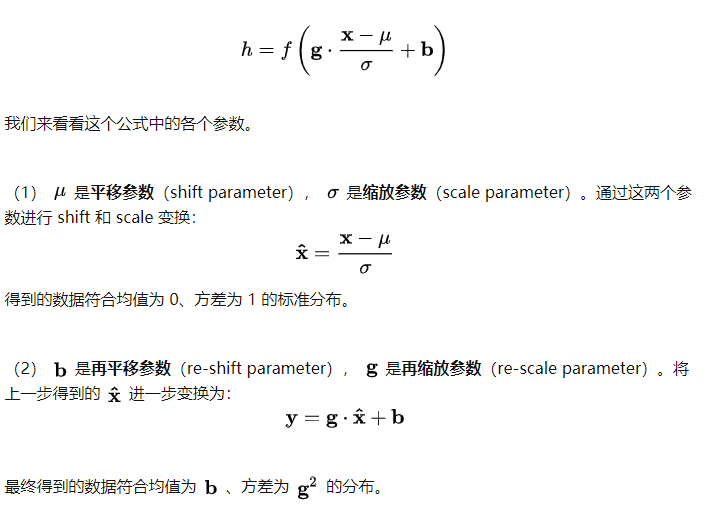


ICS 会导致什么问题：

1. 上层参数需要不断适应新的输入数据分布，降低学习速度。
2. 下层输入的变化可能趋向于变大或者变小(必然的)，导致上层落入饱和区，使得学习过早停止。
3. 每层的更新都会影响到其它层，因此每层的参数更新策略需要尽可能的谨慎。

简言之，每个神经元的输入数据不再是“独立同分布”

Normalization的基本思想



答：主要是讲x缩放到N(0,1),然后通过g,b缩放到N(b,g\*g)；既然已经缩放到标准分布，为何又变回去，答案是为了保证模型的表达能力不因为规范化而下降。Normalization并非意义上的同分布，只是映射到一个确定的区间范围而已，独立方面还有待研究。

BN：针对batch进行归一化，是纵向、元素级别的规范化；BN要求每一个batch整体数据都应该是同分布。对于每一个batch的数据差异不大进行BN，相当于适当引入噪声，可以增加模型的鲁棒性（图片就很适合，每一张图片的元素都有用，范围空间都是0-256，很适合）；但是对于每一个batch数据之间差异很大的话，此时对不同的batch数据进行BN，则会进行不一样的数据变换，增加模型训练难度。适用场景：batch比较大、训练之前做好充分shuffle.

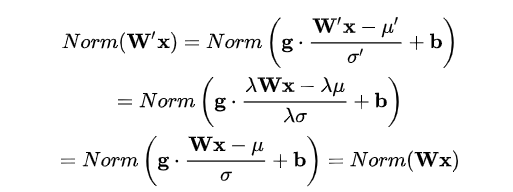
LN：是一种横向的规范化，只针对单个训练样本进行，不依赖于其他数据，避免batch小的问题，LN 不需要保存 mini-batch 的均值和方差，节省了额外的存储空间。与BN的区别：BN的转换是针对单个神经元可训练的，不同神经元的输入经过再平移和再缩放后分布在不同的区间；而 LN 对于一整层的神经元训练得到同一个转换，所有的输入都在同一个区间范围内。如果不同输入特征不属于相似的类别（比如颜色和大小），那么 LN 的处理可能会降低模型的表达能力。

Normalization为什么有效

答：

1. Normalization的权重伸缩不变性：

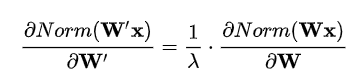
① 有效提高反向传播效率



参数W乘以λ之后进行Norm得到的值不变（权重伸缩不变性）

权重的伸缩变化不会影响反向梯度的 Jacobian 矩阵，因此也就对反向传播没有影响，避免了反向传播时因为权重过大或过小导致的梯度消失或梯度爆炸问题，从而加速了神经网络的训练。

② 有参数正则化的效果，避免参数的大幅震荡，提高网络的泛化性能，可以使用更高的学习率



1. Normalization的数据伸缩不变性

当x\_=λ\*x时，Norm(W\*x)=Norm(W\*x\_) （数据x伸缩不变性）

1. 数据伸缩不变性可以有效减少梯度弥散，简化对学习率的选择

每一层神经元的输出依赖于底下各层的计算结果。如果没有正则化，当下层输入发生伸缩变化时，经过层层传递，可能会导致数据发生剧烈的膨胀或者弥散，从而也导致了反向计算时的梯度爆炸或梯度弥散。