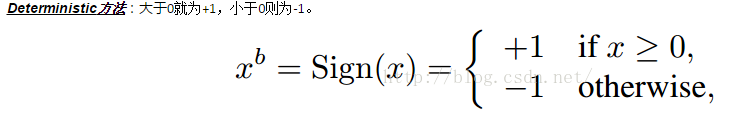
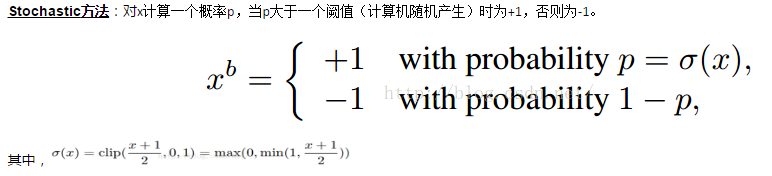
## BNN

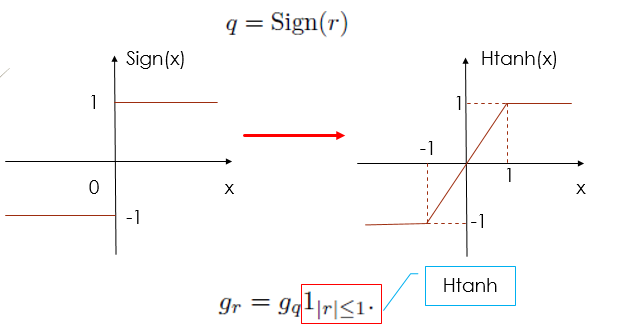
1. 二值化





分析：由于当计算机生成随机数的时候非常耗时，出于初衷加速考虑，所以一般以第一种方法进行实施。

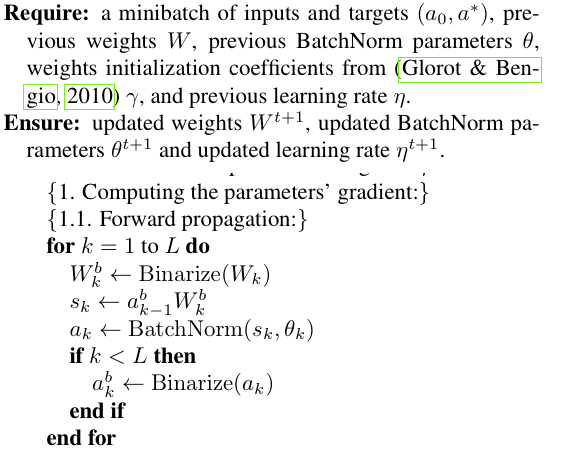
但第一种方法函数的倒数处处为0，并不能进行梯度反向传播。另外梯度具有累加效果，即梯度都带有一定的噪音，而噪音一般认为是服从正态分布的，所以，多次累加梯度才能把噪音平均消耗掉（有梯度的优势所以要有梯度）。

对第一种方法的函数进行改进，

接上，在前向传播阶段，对weights和activation的二值化相当于对网络的参数引入噪声，可以提高网络抗过拟合的能力，这可以看做是dropout的一种变形。Dropout是将激活值的一般变成0，从而造成一定的稀疏性，而二值化则是将另一半变成1，从而可以看做是进一步的dropout。

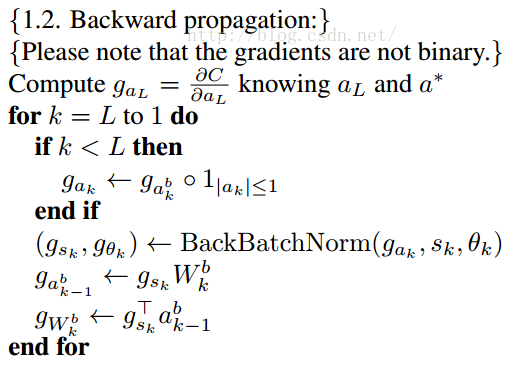
1. 模型训练

2.1前向传播



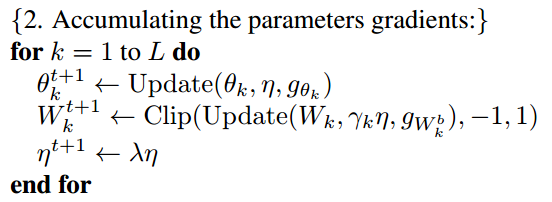
从第一层一直到最后一层，需要说明的是除了最后一个隐含层到输出层的连接权值和激活是实参外，其他的权值为二值化参数。Binarize为二值化函数。在一个神经元处需要做的操作有：二值化连接权值—>权值与输入相乘-->BatchNorm(BatchNormalization)得到这一层的激活值ak—>将 ak二值化。即：在隐含层计算阶段所有的值都为二值化后的结果。

* 1. 反向传播阶段



由于二值网络中，除了输出层，其他隐藏层都经过二值化。所以在求Batch Normalization的参数时，必须先求二值操作层（我们把二值化也当做一层来看待）的梯度，即。其中，笔者猜想是因为在-1---1之间自变量的值等于因变量，而在区间之外（梯度很大取消），取最大最小梯度分别为-1和1。注意：二值网络在对权值求梯度的时候，是对二值化后的权值求梯度，而不是对二值化前的实数型权值求梯度。这是因为二值化前的权值并没有真正的参与网络的前向传播过程。

* 1. 权值更新



求权值（W）梯度的时候是对二值化后的权值求梯度，但是在权值更新的时候，是利用上面求得的权值梯度对实数型的权值进行更新。