请注意BN和LRN的计算过程和计算量分析。

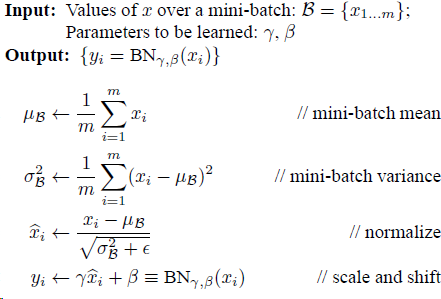
BN计算公式如下：其中x指一幅feature map中每一个点的值，m为feature map长x宽。

计算思想如下：

（1）针对每一幅feature map计算各自的平均值。

（2）针对每一幅feature map计算各自的均方差。

（3）针对每一幅feature map中的每一个pixel做normalize和scale、shift。



对于BN，training和inference过程中的运算方式略有区别。inference时是没有batch的概念的（输入图像只有1个），运算量也和training是有区别。请参考：http://blog.csdn.net/malefactor/article/details/51476961

之前给大家解释batchnorm有一点没有解释清楚，training阶段，batchnorm中均值和方差的计算是以batch为单位。

计算量方面，training阶段其计算量是我们之前统计的计算量再乘以batch num。

之前batchnorm的理解在inference时有一些误区，在inference时，batchnorm中的均值和方差不用计算，在training已经得到。在inference时，可以离线预先根据training时计算的均值和方差，对其做一定的调整。

因此，inference时，在caffe中batchnorm只做norm和scale、shift计算。根据变换，batchnorm可以转换为一个y=ax+b得线性变换，a和b可以离线计算得到，x为卷积计算后的值。

LRN计算公式如下：

通常情况k为1。

计算思想如下：

一、channel内

（1）针对每一个feature map，根据预设的filter size，计算该窗口内对应位置pixel value的平方、并累加量化。

（2）计算以β为指数的幂运算。

二、channel间

（1）针对不同的channel，根据filter size大小，在不同channel的同一位置选择对应的pixel value，计算平方、并累加量化。

（2）计算以β为指数的幂运算。

LRN和BN按照标准公式，不包含任何查表、近似、展开等操作。计算量如下：

设features map size为W,H，channel数为N，LRN某一个系数为Y，filter size为Fw，Fh，

LRN：

乘法： 2\*W\*H\*N。

加法： Fw\*Fh\*W\*H\*N。

幂运算：W\*H\*N次指数为Y的幂运算。

BN：

乘法： (3\*W\*H+1)\*N，或者经过优化(2\*W\*H+1)\*N

加法： (5\*W\*H-2)\*N

幂运算：N次指数为0.5的幂运算。