YOLO系列算法中的参数说明

# YOLO基础参数

## Batch\_size（批尺寸）

该参数主要用于批梯度下降 [算法](http://lib.csdn.net/base/datastructure) (Batch Gradient Descent)中，批梯度下降算法是每次迭代都遍历批中的所有样本， 由批中的样本共同决定最优的方向，Batch\_Size 正是批中的样本数量。

若数据集比较小，可以采用全数据集(Full Batch Learning)的形式，由全数据确定的方向能更好地代表样本总体，从而更准确地朝向极值所在的方向；但该种方式并不适用于[大数据](http://lib.csdn.net/base/hadoop" \o "Hadoop知识库)集；   
另一个极端是每次只训练一个样本，即Batch\_Size=1，每次修正方向以各自样本的梯度方向修正，横冲直撞各自为政，难以达到收敛。   
在合理范围内增大Batch\_Size,可以   
（1）提高内存利用率，进而提高大矩阵乘法的并行效率；   
（2）跑完一次epoch（全数据集）所需的迭代次数减少，对于相同数据量的处理速度进一步加快；   
（3）在一定范围内，一般来说Batch\_Size越大，其确定的下降方向越准，引起的训练震荡越小。   
盲目增大Batch\_Size的坏处：   
（1）超出内存容量；   
（2）跑完一次epoch(全数据集)所需的迭代次数减小，要想达到相同的精度，所需要的epoch数量越多，对参数的修正更加缓慢；   
（3）Batch\_Size 增大到一定程度，其确定的下降方向已经基本不再变化；   
Batch\_Size参数调试：   
大的Batch\_Size在显存能允许的情况下收敛速度是比较快的但有时会陷入局部最小的情况；小Batch\_Size引入的随机性会更大些，有可能会有更好的效果，但是收敛速度会慢一些；当Batch\_Size太小，而类别数又比较多的时候,会导致loss函数震荡而不收敛。具体调试过程中，一般可根据GPU显存，设置为最大，而且一般要求是8的倍数，选择一部分数据，跑几个Batch看看loss是否在变小，再选择合适的Batch\_Size。   
每Batch\_Size个样本更新一次参数。

darknet代码中，是将batch/subdivisions命名为batch；

# YOLOV5私有参数