DNN-train 使用教程

1. train.py
   1. 输入命令 python train.py <model> <datacfg> <weightfile> 进行训练，第一个参数<model>是在model.py中，有两种model模式，全通道和半通道。<datacfg>是指你的训练集txt文件，文件内容主要是你的图片文件路径以及相应的图片标签数据。<weightfile>是权重文件，训练文件夹中有给出初始预训练权重文件mobilenetv19.weights，你可以使用这个文件进行训练，在训练过程中你可以在backup文件夹找到你训练过程中所得到的权重文件，并可以使用其中权重文件进行训练；
   2. 如果要使用GPU进行训练，首先要在train.py中第47-48行进行更改，将第47行注释掉，将第48行注释解除，然后将177行到193行中带有cuda的注释代码行的注释去掉，然后将相应的无cuda的代码注释即可，其他需要修改的文件部分相同。
2. region\_loss.py
   1. 该文件主要是计算loss的实现，实现了对x, y, w, h以及conf五个不同loss值的计算，即在训练过程所看到的五个loss数值。
   2. 在FPGA上实现时是不需要backward的过程的，也就是说loss计算这部分是不用需要硬件实现的。
3. models.py
   1. 文件中分为两种不同的model，全通道和半通道，即卷积函数中第二个参数的区别。该文件主要是模型的参数配置，包括具体的网络结构，以及输入图片的参数，因为本次竞赛所使用的数据集图片大小不一，所以可能会需要在读取图片之后对图片进行resize操作，统一到同样大小。
   2. 你可以在该文件中进行模型的更改，当然，更改的前提是相应参数是适配的，否则会发生隐形的bug。
   3. 模型的好坏一定程度上决定了下个阶段比赛内容难易程度，如果你的模型太过庞大，可能你在第一阶段的成绩很好，但是第二阶段你的比赛任务完全无法完成；相应的如果你的模型过于简单，那么第一阶段的成绩将会比较差，所以请选择合适的模型，并且在第二阶段对模型进行一定的压缩优化。
4. dataset.py
   1. 其中listDataset是和train list和test list 直接相关的一个类，涉及到相关文件的读取以及标签信息载入转换操作等。
5. image.py
   1. 该文件主要是对图片的一些随机操作，从而达到数据增强的目的；
   2. 旋转图片，随机旋转，裁剪等，以及相应的label信息的更新。
6. utils.py
   1. 该文件中是一些重要函数的实现；
   2. Sigmoid函数，softmax函数，IoU计算函数，NMS函数，区域框函数，画框函数，读label信息函数，检测函数……