1. 准备数据集

数据集分为三类：包括训练集，校验集和测试集，每个数据集都会有一个txt的标签分类文件（标签分类文件是对每个数据集中的图片进行分类，以进行训练和校验测试的对比）

训练集：train.txt

校验集：val.txt

测试集：test.txt

备注：需要保证标签分类文件中的图片的名称以及相应的标签和数据集中的图片一一对应，否则在数据预处理和测试过程中会出错

错误一：标签分类文件中的图片名称灭有和数据集中的图片对应，执行生成lmdb文件（数据集预处理）时会提示找不到文件

错误二：标签分类文件中图片标签错误，对数据集进行验证和测试时会导致精确度很低（甚至为0）

1. 数据集预处理

数据集预处理：根据数据集及其相应的标签分类文件生成lmdb目录，三类数据集可以单独生成，只要修改相应的脚本即可，执行生成脚本就会在指定的目录下生成lmdb，用于后续的训练、校验和测试

下面以imagenet（任意一年）数据集预处理脚本为例：

|  |
| --- |
| create\_imagenet.sh |
| #!/usr/bin/env sh  # Create the imagenet lmdb inputs  # N.B. set the path to the imagenet train or val or test data dirs  set -e  #lmdb保存位置  EXAMPLE=/home/cnn/caffe/examples/imagenet  #标签分类文件所在目录  DATA=/home/cnn/caffe/data/ilsvrc12  #图片预处理工具convert\_imageset所在目录  TOOLS=/home/cnn/caffe/build/tools  #数据集所在目录  VAL\_DATA\_ROOT=/home/cnn/caffe/data/ILSVRC2012\_img\_test/  # 设置图片缩放大小  # Set RESIZE=true to resize the images to 256x256. Leave as false if images have  # already been resized using another tool.  RESIZE=true  if $RESIZE; then  RESIZE\_HEIGHT=256  RESIZE\_WIDTH=256  else  RESIZE\_HEIGHT=0  RESIZE\_WIDTH=0  fi  #判断图片数据集所在目录是否正确  if [ ! -d "$VAL\_DATA\_ROOT" ]; then  echo "Error: VAL\_DATA\_ROOT is not a path to a directory: $VAL\_DATA\_ROOT"  echo "Set the VAL\_DATA\_ROOT variable in create\_imagenet.sh to the path" \  "where the ImageNet validation data is stored."  exit 1  fi  #对数据集中每张图片进行缩放和处理，保存在lmdb中  #lmdb是一个目录，里面包含处理后的图片数据  GLOG\_logtostderr=1 $TOOLS/convert\_imageset \  --resize\_height=$RESIZE\_HEIGHT \  --resize\_width=$RESIZE\_WIDTH \  --shuffle \  #数据集所在目录  $VAL\_DATA\_ROOT \  #数据集对应的标签分类文件  $DATA/test.txt \  #保存的lmdb目录名  $EXAMPLE/ilsvrc12\_test\_lmdb  echo "Done." |

1. 配置文件

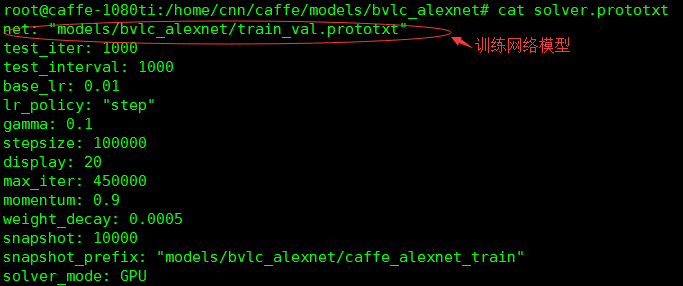
训练、校验和测试所需的配置文件包含以下3个：

1. Solver.prototxt:定义参数，比如深度网络模型，测试迭代次数等等

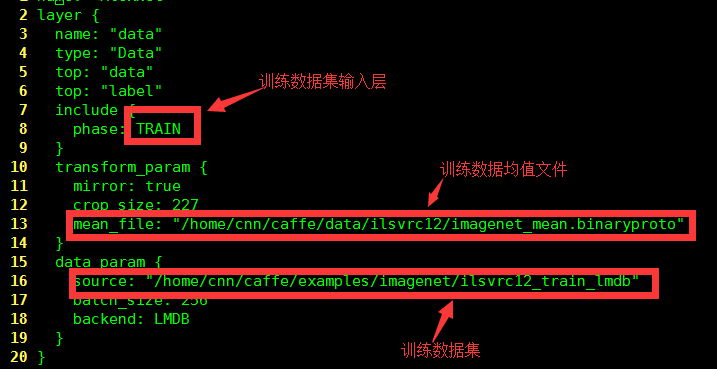
详细参数解析见网址：

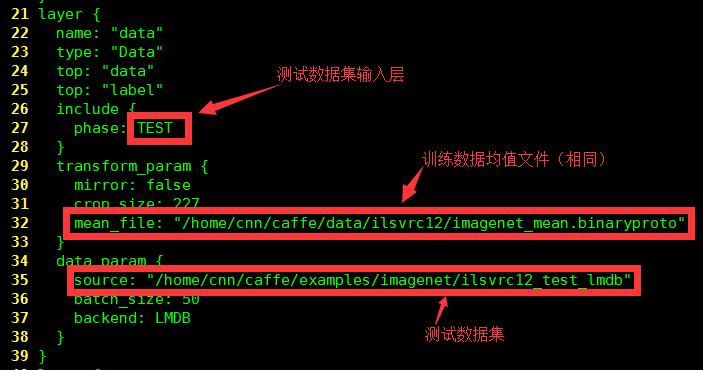
http://blog.csdn.net/u012746763/article/details/51549184

以alexnet的solver文件为例



1. Train\_val.prototxt：深度网络模型，定义训练模型的网络拓扑图，包括数据输入层，卷积层等等。被sovler配置文件引用。一般情况下，只要修改输入数据层的数据集路径和训练数据的均值文件路径即可





1. 训练数据集均值文件

均值文件一般从网上下载。如果模型是自己训练的，有下载相应的数据集，可以使用make\_imagenet\_mean.sh脚本生成均值文件

|  |
| --- |
| #!/usr/bin/env sh  # Compute the mean image from the imagenet training lmdb  # N.B. this is available in data/ilsvrc12  #训练数据集的lmdb，由数据预处理步骤生成  EXAMPLE=examples/imagenet  #均值文件保存位置  DATA=data/ilsvrc12  #均值文件生成工具compute\_image\_mean所在路径（最好是绝对路径）  TOOLS=build/tools  $TOOLS/compute\_image\_mean $EXAMPLE/ilsvrc12\_train\_lmdb \  $DATA/imagenet\_mean.binaryproto  echo "Done." |

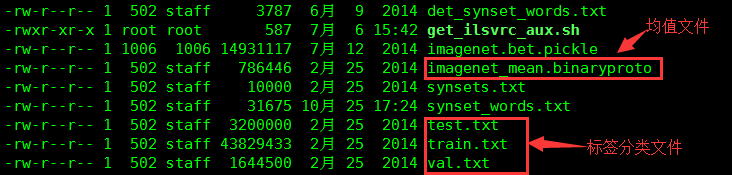
1. bvlc\_alexnet.caffemodel：训练模型参数文件，经过使用训练数据集对alexnet模型进行训练后会输出的参数文件（还没训练过），在数据测试时使用
2. 测试步骤：使用caffe自带alexnet训练模型，对imagenet12的测试数据集进行测试
3. 下载测试数据集，到caffe/data/目录下，并进行解压

解压文件到指定目录：tar –xvzf xxx.tar.gz –C /a/b

百度云下载链接：<https://pan.baidu.com/s/1pLt83J9>（该测试数据集较大，最好提前下载并上传到服务器上）

1. 执行/home/cnn/caffe/data/ilsvrc12/目录下成都的get\_ilsvrc\_aux.sh脚本，获取数据集标签分类文件和均值文件

|  |
| --- |
| root@caffe-1080ti:~# cd /home/cnn/caffe/data//ilsvrc12 |
| root@caffe-1080ti:/home/cnn/caffe/data/ilsvrc12# ./get\_ilsvrc\_aux.sh |



1. 测试数据预处理（生成lmdb）：进入/home/cnn/caffe/examples/imagenet/目录下，其中create\_imagenet.sh为数据集预处理脚本，修改脚本中的路径参数等（该脚本具体编写和修改方法见上面的数据预处理部分）

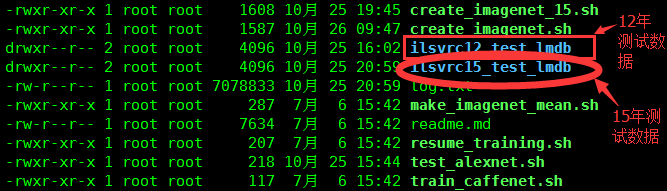
在数据预处理过程中，需要图片进行缩放，缩放尺寸的设定要求和均值文件中的大小相同（目前均值文件默认为256\*256），而且必须大于等于训练网络中的crop\_size（目前默认是227）

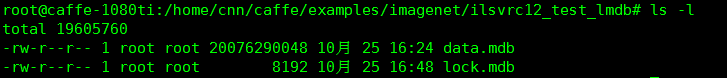
|  |
| --- |
| root@caffe-1080ti:~# cd /home/cnn/caffe/examples/imagenet/ |
| root@caffe-1080ti: /home/cnn/caffe/examples/imagenet/# vim create\_imagenet.sh |
| root@caffe-1080ti: /home/cnn/caffe/examples/imagenet/# ./ create\_imagenet.sh |

生成mdb文件的过程中，可能出现找不到相应测试图片的错误，可能存在原因有两个：a. create\_imagenet.sh脚本的测试数据集路径错误（最好采用绝对路径）;b. test.txt（标签分类文件）中的图片列表和测试图片集没有一一对应

注：对数据集进行预处理时，需要先删除之前生成的同名lmdb文件夹

对测试数据集预处理完成后会在/home/cnn/caffe/examples/imagenet/目录下生成lmdb文件夹，该文件加下是两个mdb文件





1. 对测试数据集进行测试：
2. 修改/home/cnn/caffe/models//bvlc\_alexnet目录下的网络拓扑配置文件train\_val.prototxt

|  |
| --- |
| root@caffe-1080ti:~# cd /home/cnn/caffe/models//bvlc\_alexnet |
| root@caffe-1080ti:/home/cnn/caffe/models/bvlc\_alexnet# vim train\_val.prototxt |

1. 修改/home/cnn/caffe/models//bvlc\_alexnet目录下的测试脚本test\_alexnet.sh

|  |
| --- |
| root@caffe-1080ti:~# cd /home/cnn/caffe/models//bvlc\_alexnet |
| root@caffe-1080ti:/home/cnn/caffe/models/bvlc\_alexnet# vim test\_alexnet.sh |

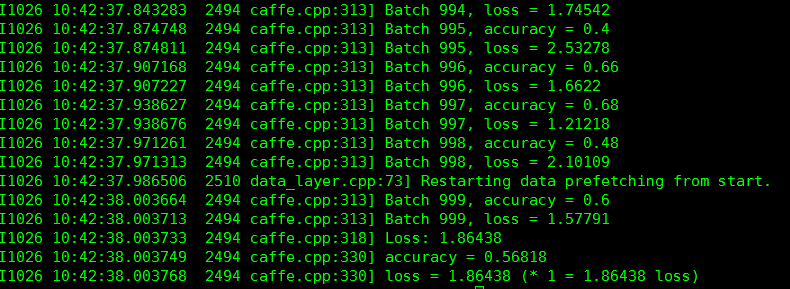
测试脚本具体内容个如下：

|  |
| --- |
| test\_alexnet.sh |
| #!/usr/bin/env sh  set –e  #可执行程序caffe，总共有四个参数  #test:表示将进行测试操作  #model：使用的深度网络拓扑  #weights：使用的训练参数模型  #gpu：使用gpu进行计算  #详细命令参数可以使用—help进行查询  /home/cnn/caffe/build/tools/caffe test \  -model=/home/cnn/caffe/models/bvlc\_alexnet/train\_val.prototxt \  -weights=/home/cnn/caffe/models/bvlc\_alexnet/bvlc\_alexnet.caffemodel -gpu=0 $@ |

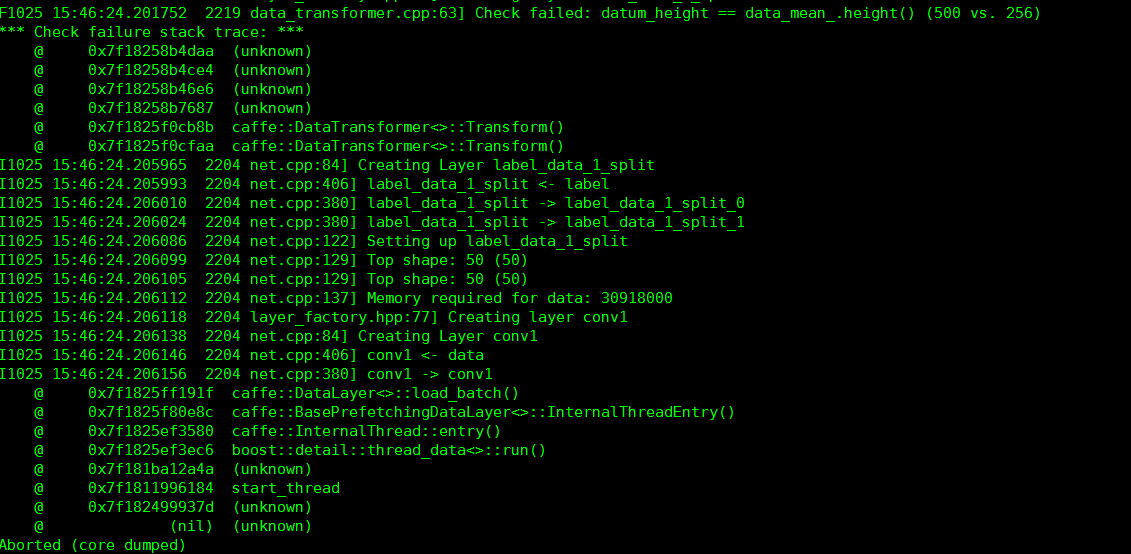
附件训练脚本

|  |
| --- |
|  |
| #!/usr/bin/env sh  set -e  #可执行程序caffe，总共有三个参数  #test:表示将进行测试操作  # solver：训练时使用solver配置文件  #gpu：使用gpu进行计算  #详细命令参数可以使用—help进行查询  ./build/tools/caffe train \  --solver=models/bvlc\_reference\_caffenet/solver.prototxt -gpu=0 $@ |

测试结果



1. 测试过程中可能出现的bug
2. 测试图片大小和均值文图片大小不匹配（或者测试图片尺寸小于网络拓扑文件中的图片大小crop\_size），错误提示如下：



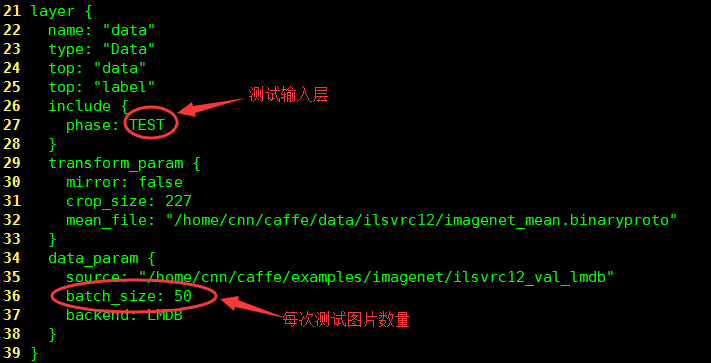
解决办法：在数据集预处理阶段，设置好图片的缩放尺寸

相关解决博客说明：<http://www.cnblogs.com/hust-yingjie/p/6526370.html>

1. 测试的迭代次数是在代码中设定的，测试的batch\_size则在深度网络拓扑文件中设置，两者需要根据测试数据集大小来调整，满足

Tset\_size = iterations \* batch\_size

batch\_size可以在网络拓扑文件（train\_val.prototxt）中调整

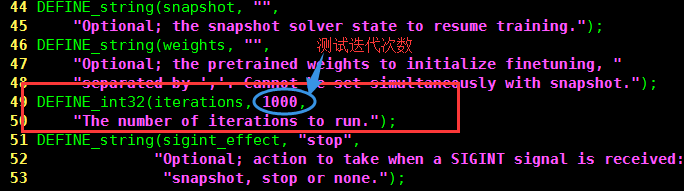


下面介绍测试迭代成都测试的调整方法

打开/home/cnn/caffe/tools的caffe.cpp文件，修改iterations参数的数值，如下图所示，修改完成后，进入caffe根目录，进行make编译。编译完成后即可进行正常测试

|  |
| --- |
| root@caffe-1080ti:~# cd /home/cnn/caffe/tools |
| root@caffe-1080ti:/home/cnn/caffe/tools# vim caffe.cpp |
| root@caffe-1080ti:/home/cnn/caffe/tools# cd ../ |
| root@caffe-1080ti:/home/cnn/caffe# make -j8  CXX tools/caffe.cpp  CXX/LD -o .build\_release/tools/caffe.bin |

caffe.cpp文件修改



相关解决博客说明：http://blog.csdn.net/zhikangfu/article/details/53308452