

1. img-dnn

第一步：改写 tailbench 文件下 configs.sh 的文件路径 系统 jdk 的路径

Vim configs.sh 界面如下：

```
# Set this to point to the top level of the TailBench data directory
DATA_ROOT=/home/mgat/desk/tailbench/tailbench-v0.9

# Set this to point to the top level installation directory of the Java
# Development Kit. Only needed for Specjbb
JDK_PATH=/usr/java/jdk1.8.0_281

# This location is used by applications to store scratch data during execution.
SCRATCH_DIR=/home/mgat/desk/tailbench/tailbench-v0.9/scratch
```

主要是第一条，第二条和第三条暂时没有用到。

第二步，下载安装 libopencv-dev

在华为云的主机上用的是 Ubuntu20.安装 opencv4 参考连接

<https://blog.csdn.net/s15810751918/article/details/107705387>

安装好以后，修改 Makefile 文件

```
Makefile Makefile common.cpp common.h first-56潘世抱2.c img-dnn.cp
1 CXX = g++
2 CXXFLAGS += -g -O3 $(shell pkg-config --cflags opencv) -std=c++0x
3 LDFLAGS = $(shell pkg-config --libs opencv)
4
```

我下载的是 opencv4，所以将上图的 opencv 改成 opencv4，这两条 shell 得到的是 opencv4 的头文件的路径和链接库的位置。

进行编译，make，会有编译错误，显示参数没有定义，这可能跟 g++ 的版本有关。

```
112 |         reduce(act1.aHidden, pj, 1, CV_REDUCE_SUM);
    |                                     ^~~~~~
train.cpp: In function 'void softmaxRegressionCost(cv::Mat&, cv::Mat&, SMR&
ble)':
train.cpp:189:24: error: 'CV_REDUCE_SUM' was not declared in this scope
189 |         reduce(M, temp, 0, CV_REDUCE_SUM);
    |                                     ^~~~~~
train.cpp: In function 'void fineTuneNetworkCost(cv::Mat&, cv::Mat&, std::v
<SparseAutoencoder>&, SMR&, double)':
train.cpp:256:23: error: 'CV_REDUCE_MAX' was not declared in this scope
256 |         reduce(M, tmp, 0, CV_REDUCE_MAX);
    |                                     ^~~~~~
train.cpp:260:23: error: 'CV_REDUCE_SUM' was not declared in this scope
260 |         reduce(p, tmp, 0, CV_REDUCE_SUM);
    |                                     ^~~~~~
make: *** [Makefile:22: train.o] Error 1
root@ecs-c6s-xlarge-2-linux-20210331182611:~/tailbench/img-dnn# vim train.c
```

对于有问题的 cpp 文件将参数改成 cv::REDUCE_SUM(MAX)

编译 make

```

#!/bin/bash

DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
source ${DIR}/../configs.sh

THREADS=1
REQS=100000000 # Set this very high; the harness controls maxreqs

TBENCH_WARMUPREQS=5000 TBENCH_MAXREQS=10000 TBENCH_QPS=500 \
TBENCH_MINSLEEPNS=10000 TBENCH_MNIST_DIR=${DATA_ROOT}/img-dnn/mnist \
./img-dnn_integrated -r ${THREADS} \
-f ${DATA_ROOT}/img-dnn/models/model_60000pts_80kitters.xml -n ${REQS}
~
~
~

```

最后一行，文件的路径，要配置好，如果找不到，需要找到官方的 tailbench.input 的数据集里找到

2. masstree 进入官方的文档，执行

```

sudo apt install libgoogle-perftools-dev
bash build.sh
sudo bash run.sh

```

- 3 1、将 xapian（杨亚楠学长给的文件包）上传至云端，（这里可能没有 wiki，如果没有记得把 Wiki 一起传到云端，并放在 xapian 目录下。）

- 2、进入 xapian-core-1.4.4 文件夹（xapian-core-1.2 没用，安装不了）进行以下命令


```

mkdir install
./configure --prefix=$PWD/install
make
make install
cd ..

```

- 3、回到 xapian 目录下，make 如果编译失败，并提示.o 文件有问题，删除所有的.o 文件，在进行编译，如果还失败，进入 xapian/xapian-core-1.4.4/install/lib 执行

```

sudo cp libxapian.so libxapian.so.30 libxapian.so.30.3.1 /usr/lib
最后编译 make
Sudo bash run.sh

```

所有生成的参数数据保存在 lats.bin 二进制文件中，进入到相应的文件当中，可以用 Python3 parselats.py lats.bin 翻译生成 lats.txt。