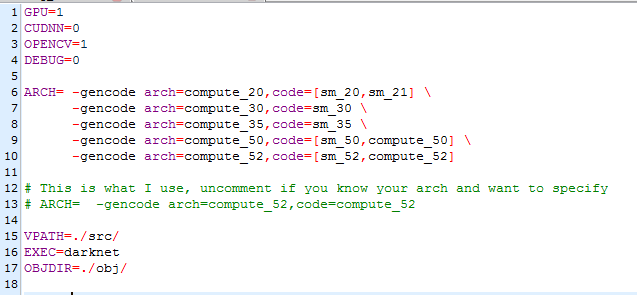
1. darknet 資料夾裡面有 Makefile檔案



第一行GPU，選擇數字1代表有使用GPU，選擇0代表沒使用到GPU。

第二行CUDNN，選擇數字1代表有使用CUDNN，選擇0代表沒使用到CUDNN。

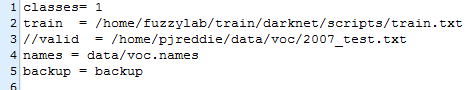
第三行OPENCV，選擇數字1代表有使用OPENCV，選擇0代表沒使用到OPENCV。

1. 編輯好Makefile檔案後，在終端機使用指令 make ，即可編譯。
2. 在終端機輸入指令

./darknet detector train cfg/yolov2.data cfg/yolov2.cfg yolov2.weights

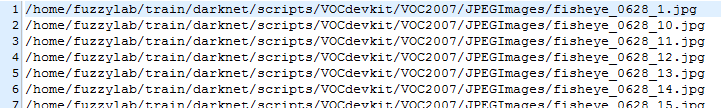
即可立即訓練網路。

1. 在cfg/voc.data 資料底下



第一行classes，訓練類別有幾類。

第二行train，為指出訓練圖片路徑的檔案。



第三行，valid，為指出驗證圖片路徑的檔案。

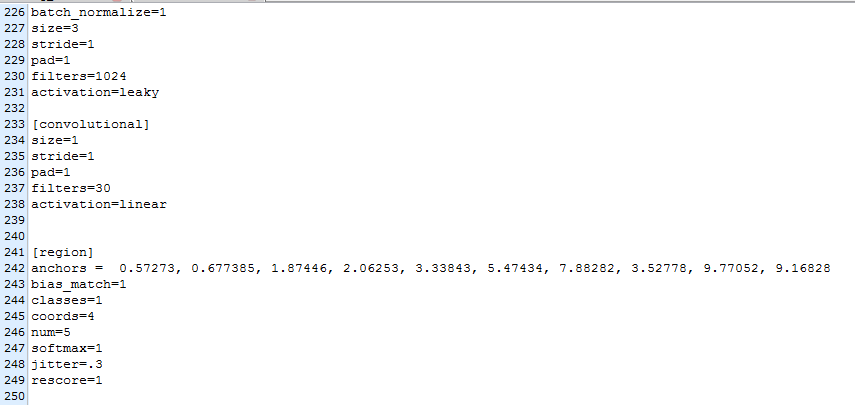
第四行names，為訓練類別的名稱。



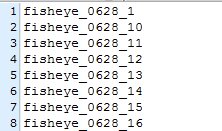
第五行backup，為儲存weight的路徑。

1. 在cfg/yolov2.cfg資料底下

在第237行，filters為 修改最後一層卷積層核參數個數，計算公式是依照自己數據的類别數filter=num ×（classes + coords + 1）=5×（1+4+1）=30



1. 資料準備，我在scripts/資料夾下放置VOCdevkit的資料夾，VOCdevkit的格式是依照VOC data的擺放方式。
   1. scripts/VOCdevkit/VOC2007/Annotations/ 底下放置.xml 檔 (bounding box資訊 )
   2. scripts/VOCdevkit/VOC2007/JPEGImages/ 底下放置.jpg檔
   3. scripts/VOCdevkit/VOC2007/ImageSets/Main/ 底下放置所有訓練圖片的名稱



1. 依照第6點處理後，執行scripts/voc\_label.py，在scripts/路徑下，指令下 python voc\_label.py，會跑出train.txt的檔案(第4點所述之train.txt)。在voc\_label.py裡面的第9行為類別名稱