3-5 images can be anything (randomly drawn from training set is possible), but preferably contains all of your classes (e.g. car and dogs, then 3-5 images should better have both of them instead of only one). Not being able to overfit such a small training set means the learning rate are too big; or there is bug in the code.

I recommend disabling noise augmentation during this overfit step by setting argument allobj = None in <https://github.com/thtrieu/darkflow/blob/master/net/yolo/data.py#L69>, setting learning rate smaller (say 1e-5) and try overfitting again.

When you first start training on custom data, you should overfit the network on a subset of images (3 - 5). keep training until you get perfect bounding boxes on those images with 0.9 confidence or greater. After that, begin training on the entire dataset.

So, let’s assume you type the following command to begin training a new model:

flow --model cfg/yolo\_custom.cfg --load bin/yolo.weights --train --annotation labelImg/annotations\_overfit --dataset labelImg/images\_overfit --epoch 1000 --gpu 1.0

Note that you are loading the pretrained weights in the —load flag.

You want to have only a subset of your training data (3-5 images and annotations) in the dataset and annotation directories. Train on just that subset until your bounding boxes cover the object as expected and you have a confidence score on all objects in the subset of at least 0.9. So far, every overfitting I’ve done has taken only 1000 epochs, but it could be different for you. After you’ve overfit the subset, begin training on the remaining training data (I.e., change —load to -1, and —dataset and —annotation to the directories with the rest of the training data). You should be all set! Did that clear it up for you?

I did following to get down to 3.8:

1. Initialize fresh model and train it with learning rate 0.01 and batch 5
2. After it's around 10 loss (quite fast), increase batch to 10 and learning rate to 0.0001
3. When it's around 4.5 increase batch to 40 and learning rate down too 0.00008

I guess it's all about increasing batch size and decreasing learning rate when moving average loss stops moving (e.g. if 2 minutes relative difference is less than 0.9). It certainly works for test images but's not perfect:

<https://github.com/thtrieu/darkflow/issues/411>

Overfit training produces no detection

<https://github.com/thtrieu/darkflow/issues/9>

# what is the lowest loss value can reach?

|  |
| --- |
| def setDefaults(self): |
|  | self.define('imgdir', './sample\_img/', 'path to testing directory with images') |
|  | self.define('binary', './bin/', 'path to .weights directory') |
|  | self.define('config', './cfg/', 'path to .cfg directory') |
|  | self.define('dataset', '../pascal/VOCdevkit/IMG/', 'path to dataset directory') |
|  | self.define('labels', 'labels.txt', 'path to labels file') |
|  | self.define('backup', './ckpt/', 'path to backup folder') |
|  | self.define('summary', '', 'path to TensorBoard summaries directory') |
|  | self.define('annotation', '../pascal/VOCdevkit/ANN/', 'path to annotation directory') |
|  | self.define('threshold', -0.1, 'detection threshold') |
|  | self.define('model', '', 'configuration of choice') |
|  | self.define('trainer', 'rmsprop', 'training algorithm') |
|  | self.define('momentum', 0.0, 'applicable for rmsprop and momentum optimizers') |
|  | self.define('verbalise', True, 'say out loud while building graph') |
|  | self.define('train', False, 'train the whole net') |
|  | self.define('load', '', 'how to initialize the net? Either from .weights or a checkpoint, or even from scratch') |
|  | self.define('savepb', False, 'save net and weight to a .pb file') |
|  | self.define('gpu', 0.0, 'how much gpu (from 0.0 to 1.0)') |
|  | self.define('gpuName', '/gpu:0', 'GPU device name') |
|  | self.define('lr', 1e-5, 'learning rate') |
|  | self.define('keep',20,'Number of most recent training results to save') |
|  | self.define('batch', 16, 'batch size') |
|  | self.define('epoch', 1000, 'number of epoch') |
|  | self.define('save', 2000, 'save checkpoint every ? training examples') |
|  | self.define('demo', '', 'demo on webcam') |
|  | self.define('queue', 1, 'process demo in batch') |
|  | self.define('json', False, 'Outputs bounding box information in json format.') |
|  | self.define('saveVideo', False, 'Records video from input video or camera') |
|  | self.define('pbLoad', '', 'path to .pb protobuf file (metaLoad must also be specified)') |
|  | self.define('metaLoad', '', 'path to .meta file generated during --savepb that corresponds to .pb file') |

Step 1

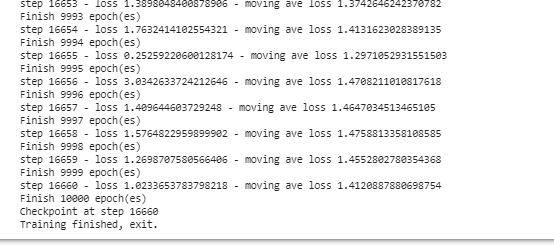
Overfit the model with 3 images. First, we train the model with a learning rate equals to 1e-5,withput noise augmentation, and with the next 3 images.

!python flow --model cfg/yolov2-tiny-7c.cfg –load bin/yolov2-tiny-7c.wegiths --train --gpu 1.0 --annotation Dataset/train/labels --dataset Dataset/train/images --epoch 10000 --batch 32 --lr 1e-5

|  |  |
| --- | --- |
| Shelve (1).jpg | Shelve (2).jpg |
| Shelve (4).jpg |  |

Step2 When the loss function reaches a value close to 1, we decrease the learning rate to 1e-6

!python flow --model cfg/yolov2-tiny-7c.cfg --load 6660 --train --gpu 1.0 --annotation Dataset/train/labels --dataset Dataset/train/images --epoch 10000 --batch 32 --lr 1e-6



1 hora

|  |  |
| --- | --- |
|  |  |
|  |  |

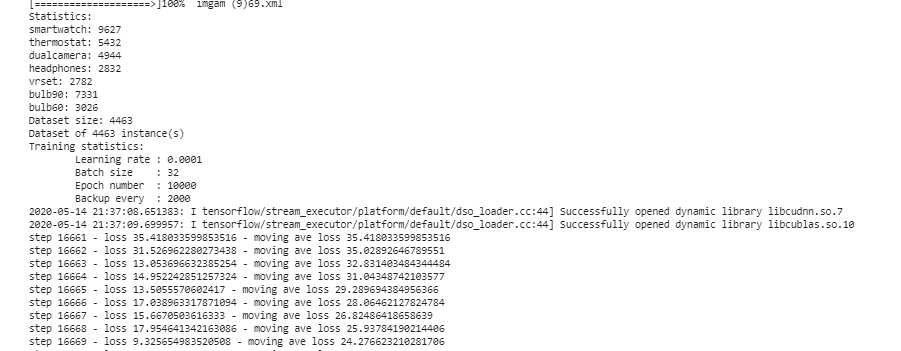
Test images

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

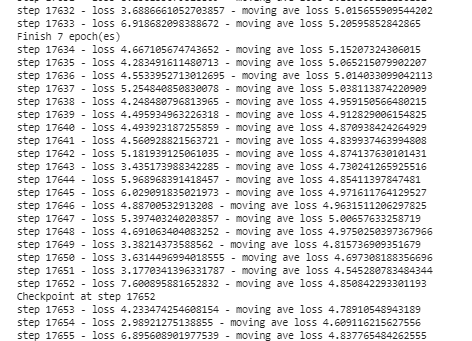
Step 3 a

Train the model from the last step saved and with the rest of images, including nosie augmentation learning rate equals to 1e-4

!python flow --model cfg/yolov2-tiny-7c.cfg --load 16660 --train --gpu 1.0 --annotation Dataset/train/labels --dataset Dataset/train/images --epoch 10000 --batch 32 --lr 1e-4



After 7 epochs and 17652 steps the loss is 4.7 to 4.8. (50 min)



Original train images

|  |  |
| --- | --- |
|  |  |
|  |  |

Test images

|  |  |
| --- | --- |
|  |  |
|  |  |

We decide to decrease learning rate to 1e-6

Without noise augmentation

!python flow --model cfg/yolov2-tiny-7c.cfg --load 6250 --train --gpu 1.0 --annotation Dataset/train\_min/labels --dataset Dataset/train\_min/images --epoch 100000 --batch 32 --lr 1e-5  --save 100

(python36) C:\Users\oscar.frausto.perez\Accenture\Reyna, Daniel - smart-shelves\_development\YOLO\darkflow-master>python flow --imgdir sample\_img/ --model cfg/yolov2-tiny-7c.cfg --load 9290