#include "network.h"

#include "detection\_layer.h"

#include "cost\_layer.h"

#include "utils.h"

#include "parser.h"

#include "box.h"

void train\_swag(char \*cfgfile, char \*weightfile)

{

char \*train\_images = "data/voc.0712.trainval";

char\* backup\_directory = "backup/";

srand(time(0));

char \*base = basecfg(cfgfile);

printf("%s\n", base);

float avg\_loss = -1;

network net = parse\_network\_cfg(cfgfile);

if(weightfile){

load\_weights(&net, weightfile);

}

printf("Learning Rate: %g, Momentum: %g, Decay: %g\n", net.learning\_rate, net.momentum, net.decay);

int imgs = net.batch\*net.subdivisions;

int i = \*net.seen/imgs;

data train, buffer;

layer l = net.layers[net.n - 1];

int side = l.side;

int classes = l.classes;

float jitter = l.jitter;

list \*plist = get\_paths(train\_images);

//int N = plist->size;

char \*\*paths = (char \*\*)list\_to\_array(plist);

load\_args args = {0};

args.w = net.w;

args.h = net.h;

args.paths = paths;

args.n = imgs;

args.m = plist->size;

args.classes = classes;

args.jitter = jitter;

args.num\_boxes = side;

args.d = &buffer;

args.type = REGION\_DATA;

pthread\_t load\_thread = load\_data\_in\_thread(args);

clock\_t time;

//while(i\*imgs < N\*120){

while(get\_current\_batch(net) < net.max\_batches){

i += 1;

time=clock();

pthread\_join(load\_thread, 0);

train = buffer;

load\_thread = load\_data\_in\_thread(args);

printf("Loaded: %lf seconds\n", sec(clock()-time));

time=clock();

float loss = train\_network(net, train);

if (avg\_loss < 0) avg\_loss = loss;

avg\_loss = avg\_loss\*.9 + loss\*.1;

printf("%d: %f, %f avg, %f rate, %lf seconds, %d images\n", i, loss, avg\_loss, get\_current\_rate(net), sec(clock()-time), i\*imgs);

if(i%1000==0 || i == 600){

char buff[256];

sprintf(buff, "%s/%s\_%d.weights", backup\_directory, base, i);

save\_weights(net, buff);

}

free\_data(train);

}

char buff[256];

sprintf(buff, "%s/%s\_final.weights", backup\_directory, base);

save\_weights(net, buff);

}

void run\_swag(int argc, char \*\*argv)

{

if(argc < 4){

fprintf(stderr, "usage: %s %s [train/test/valid] [cfg] [weights (optional)]\n", argv[0], argv[1]);

return;

}

char \*cfg = argv[3];

char \*weights = (argc > 4) ? argv[4] : 0;

if(0==strcmp(argv[2], "train")) train\_swag(cfg, weights);

}