#include "network.h"

#include "utils.h"

#include "parser.h"

#include "option\_list.h"

#include "blas.h"

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

{

srand(time(0));

float avg\_loss = -1;

char \*base = basecfg(cfgfile);

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

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);

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

int classes = 10;

int N = 50000;

char \*\*labels = get\_labels("data/cifar/labels.txt");

int epoch = (\*net.seen)/N;

data train = load\_all\_cifar10();

while(get\_current\_batch(net) < net.max\_batches || net.max\_batches == 0){

clock\_t time=clock();

float loss = train\_network\_sgd(net, train, 1);

if(avg\_loss == -1) avg\_loss = loss;

avg\_loss = avg\_loss\*.95 + loss\*.05;

printf("%d, %.3f: %f, %f avg, %f rate, %lf seconds, %ld images\n", get\_current\_batch(net), (float)(\*net.seen)/N, loss, avg\_loss, get\_current\_rate(net), sec(clock()-time), \*net.seen);

if(\*net.seen/N > epoch){

epoch = \*net.seen/N;

char buff[256];

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

save\_weights(net, buff);

}

if(get\_current\_batch(net)%100 == 0){

char buff[256];

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

save\_weights(net, buff);

}

}

char buff[256];

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

save\_weights(net, buff);

free\_network(net);

free\_ptrs((void\*\*)labels, classes);

free(base);

free\_data(train);

}

void train\_cifar\_distill(char \*cfgfile, char \*weightfile)

{

srand(time(0));

float avg\_loss = -1;

char \*base = basecfg(cfgfile);

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

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);

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

int classes = 10;

int N = 50000;

char \*\*labels = get\_labels("data/cifar/labels.txt");

int epoch = (\*net.seen)/N;

data train = load\_all\_cifar10();

matrix soft = csv\_to\_matrix("results/ensemble.csv");

float weight = .9;

scale\_matrix(soft, weight);

scale\_matrix(train.y, 1. - weight);

matrix\_add\_matrix(soft, train.y);

while(get\_current\_batch(net) < net.max\_batches || net.max\_batches == 0){

clock\_t time=clock();

float loss = train\_network\_sgd(net, train, 1);

if(avg\_loss == -1) avg\_loss = loss;

avg\_loss = avg\_loss\*.95 + loss\*.05;

printf("%d, %.3f: %f, %f avg, %f rate, %lf seconds, %ld images\n", get\_current\_batch(net), (float)(\*net.seen)/N, loss, avg\_loss, get\_current\_rate(net), sec(clock()-time), \*net.seen);

if(\*net.seen/N > epoch){

epoch = \*net.seen/N;

char buff[256];

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

save\_weights(net, buff);

}

if(get\_current\_batch(net)%100 == 0){

char buff[256];

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

save\_weights(net, buff);

}

}

char buff[256];

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

save\_weights(net, buff);

free\_network(net);

free\_ptrs((void\*\*)labels, classes);

free(base);

free\_data(train);

}

void test\_cifar\_multi(char \*filename, char \*weightfile)

{

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

if(weightfile){

load\_weights(&net, weightfile);

}

set\_batch\_network(&net, 1);

srand(time(0));

float avg\_acc = 0;

data test = load\_cifar10\_data("data/cifar/cifar-10-batches-bin/test\_batch.bin");

int i;

for(i = 0; i < test.X.rows; ++i){

image im = float\_to\_image(32, 32, 3, test.X.vals[i]);

float pred[10] = {0};

float \*p = network\_predict(net, im.data);

axpy\_cpu(10, 1, p, 1, pred, 1);

flip\_image(im);

p = network\_predict(net, im.data);

axpy\_cpu(10, 1, p, 1, pred, 1);

int index = max\_index(pred, 10);

int class\_id = max\_index(test.y.vals[i], 10);

if(index == class\_id) avg\_acc += 1;

free\_image(im);

printf("%4d: %.2f%%\n", i, 100.\*avg\_acc/(i+1));

}

}

void test\_cifar(char \*filename, char \*weightfile)

{

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

if(weightfile){

load\_weights(&net, weightfile);

}

srand(time(0));

clock\_t time;

float avg\_acc = 0;

float avg\_top5 = 0;

data test = load\_cifar10\_data("data/cifar/cifar-10-batches-bin/test\_batch.bin");

time=clock();

float \*acc = network\_accuracies(net, test, 2);

avg\_acc += acc[0];

avg\_top5 += acc[1];

printf("top1: %f, %lf seconds, %d images\n", avg\_acc, sec(clock()-time), test.X.rows);

free\_data(test);

}

void extract\_cifar()

{

char \*labels[] = {"airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"};

int i;

data train = load\_all\_cifar10();

data test = load\_cifar10\_data("data/cifar/cifar-10-batches-bin/test\_batch.bin");

for(i = 0; i < train.X.rows; ++i){

image im = float\_to\_image(32, 32, 3, train.X.vals[i]);

int class\_id = max\_index(train.y.vals[i], 10);

char buff[256];

sprintf(buff, "data/cifar/train/%d\_%s",i,labels[class\_id]);

save\_image\_png(im, buff);

}

for(i = 0; i < test.X.rows; ++i){

image im = float\_to\_image(32, 32, 3, test.X.vals[i]);

int class\_id = max\_index(test.y.vals[i], 10);

char buff[256];

sprintf(buff, "data/cifar/test/%d\_%s",i,labels[class\_id]);

save\_image\_png(im, buff);

}

}

void test\_cifar\_csv(char \*filename, char \*weightfile)

{

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

if(weightfile){

load\_weights(&net, weightfile);

}

srand(time(0));

data test = load\_cifar10\_data("data/cifar/cifar-10-batches-bin/test\_batch.bin");

matrix pred = network\_predict\_data(net, test);

int i;

for(i = 0; i < test.X.rows; ++i){

image im = float\_to\_image(32, 32, 3, test.X.vals[i]);

flip\_image(im);

}

matrix pred2 = network\_predict\_data(net, test);

scale\_matrix(pred, .5);

scale\_matrix(pred2, .5);

matrix\_add\_matrix(pred2, pred);

matrix\_to\_csv(pred);

fprintf(stderr, "Accuracy: %f\n", matrix\_topk\_accuracy(test.y, pred, 1));

free\_data(test);

}

void test\_cifar\_csvtrain(char \*filename, char \*weightfile)

{

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

if(weightfile){

load\_weights(&net, weightfile);

}

srand(time(0));

data test = load\_all\_cifar10();

matrix pred = network\_predict\_data(net, test);

int i;

for(i = 0; i < test.X.rows; ++i){

image im = float\_to\_image(32, 32, 3, test.X.vals[i]);

flip\_image(im);

}

matrix pred2 = network\_predict\_data(net, test);

scale\_matrix(pred, .5);

scale\_matrix(pred2, .5);

matrix\_add\_matrix(pred2, pred);

matrix\_to\_csv(pred);

fprintf(stderr, "Accuracy: %f\n", matrix\_topk\_accuracy(test.y, pred, 1));

free\_data(test);

}

void eval\_cifar\_csv()

{

data test = load\_cifar10\_data("data/cifar/cifar-10-batches-bin/test\_batch.bin");

matrix pred = csv\_to\_matrix("results/combined.csv");

fprintf(stderr, "%d %d\n", pred.rows, pred.cols);

fprintf(stderr, "Accuracy: %f\n", matrix\_topk\_accuracy(test.y, pred, 1));

free\_data(test);

free\_matrix(pred);

}

void run\_cifar(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\_cifar(cfg, weights);

else if(0==strcmp(argv[2], "extract")) extract\_cifar();

else if(0==strcmp(argv[2], "distill")) train\_cifar\_distill(cfg, weights);

else if(0==strcmp(argv[2], "test")) test\_cifar(cfg, weights);

else if(0==strcmp(argv[2], "multi")) test\_cifar\_multi(cfg, weights);

else if(0==strcmp(argv[2], "csv")) test\_cifar\_csv(cfg, weights);

else if(0==strcmp(argv[2], "csvtrain")) test\_cifar\_csvtrain(cfg, weights);

else if(0==strcmp(argv[2], "eval")) eval\_cifar\_csv();

}