#include "avgpool\_layer.h"

#include "dark\_cuda.h"

#include "utils.h"

#include <stdio.h>

avgpool\_layer make\_avgpool\_layer(int batch, int w, int h, int c)

{

fprintf(stderr, "avg %4d x%4d x%4d -> %4d\n", w, h, c, c);

avgpool\_layer l = { (LAYER\_TYPE)0 };

l.type = AVGPOOL;

l.batch = batch;

l.h = h;

l.w = w;

l.c = c;

l.out\_w = 1;

l.out\_h = 1;

l.out\_c = c;

l.outputs = l.out\_c;

l.inputs = h\*w\*c;

int output\_size = l.outputs \* batch;

l.output = (float\*)xcalloc(output\_size, sizeof(float));

l.delta = (float\*)xcalloc(output\_size, sizeof(float));

l.forward = forward\_avgpool\_layer;

l.backward = backward\_avgpool\_layer;

#ifdef GPU

l.forward\_gpu = forward\_avgpool\_layer\_gpu;

l.backward\_gpu = backward\_avgpool\_layer\_gpu;

l.output\_gpu = cuda\_make\_array(l.output, output\_size);

l.delta\_gpu = cuda\_make\_array(l.delta, output\_size);

#endif

return l;

}

void resize\_avgpool\_layer(avgpool\_layer \*l, int w, int h)

{

l->w = w;

l->h = h;

l->inputs = h\*w\*l->c;

}

void forward\_avgpool\_layer(const avgpool\_layer l, network\_state state)

{

int b,i,k;

for(b = 0; b < l.batch; ++b){

for(k = 0; k < l.c; ++k){

int out\_index = k + b\*l.c;

l.output[out\_index] = 0;

for(i = 0; i < l.h\*l.w; ++i){

int in\_index = i + l.h\*l.w\*(k + b\*l.c);

l.output[out\_index] += state.input[in\_index];

}

l.output[out\_index] /= l.h\*l.w;

}

}

}

void backward\_avgpool\_layer(const avgpool\_layer l, network\_state state)

{

int b,i,k;

for(b = 0; b < l.batch; ++b){

for(k = 0; k < l.c; ++k){

int out\_index = k + b\*l.c;

for(i = 0; i < l.h\*l.w; ++i){

int in\_index = i + l.h\*l.w\*(k + b\*l.c);

state.delta[in\_index] += l.delta[out\_index] / (l.h\*l.w);

}

}

}

}