Assignment6-cuda

컴퓨터전공

2013011491

안찬영

1. Implementation

-Max Pooling

int idx = row\*input\_size\*filter\_size+col\*filter\_size;

float max=0;

if(input\_size-(col\*filter\_size)>=filter\_size){

if(idx<input\_size\*input\_size){

for(int i=0; i<filter\_size;i++){

for(int j=0; j<filter\_size; j++){

if(max<input[idx+j+i\*input\_size])

max=input[idx+j+i\*input\_size];

}

}

}

output[col+row\*(input\_size/filter\_size)]=max;

}

하나의 thread는 filter\_size\* filter\_size matrix에서 max값을 뽑아내도록 구현하였습니다. 첫번째 조건문에서 input\_size가 filter\_size의 multiple이 아닐 때 left\_over 부분을 무시하기 위해 사용되었습니다. 그 다음은 filter\_size로 나눠진 matrix에서 loop를 돌면서 최댓값을 output에 저장합니다.

-simple GEMM

if(col==(input\_size-1)&&((col+1)%TILE\_WIDTH!=0)){

for(int i=1; (col+i)%TILE\_WIDTH!=0; ++i){

if(row\*input\_size+p\*TILE\_WIDTH+tx+i>=(i\*input\_size)-1) continue;

s\_a[ty][tx+i]=a[row\*input\_size+p\*TILE\_WIDTH+tx+i];

}

}

if((p\*TILE\_WIDTH+tx)<(input\_size)&&row<input\_size){

s\_a[ty][tx]=a[row\*input\_size+p\*TILE\_WIDTH+tx];

}else{

s\_a[ty][tx]=0.0;

}

일부분만 설명하면 첫 조건문에서 input\_size가 TILE\_WIDTH의 multiple이 아닐 때, 행렬 마지막 column index를 찾아냅니다. 이 경우에 block에서 tx가 filter\_size보다 작기 때문에 s\_a[ty][tx]의 배열에 없는 tx의 대해서 값을 불러오지 않기 때문에, 이 때에만 연산이 추가로 행해지게 됩니다. 나머지 부분에 대해서는 tile 사이즈 만큼 공유메모리에 값을 불러와 block 단위로 계산하게 됩니다.

1. performance