**System Programing**

Assignment #2 . Thread-based Parallel CNN

**컴퓨터 공학과**

**201811242 김재우**

**Content**

1. Design ( 설계 ) & Implementation ( 구현 )

2. Function description ( 함수 설명 )

**1. Design ( 설계 ) & Implementation ( 구현 )**

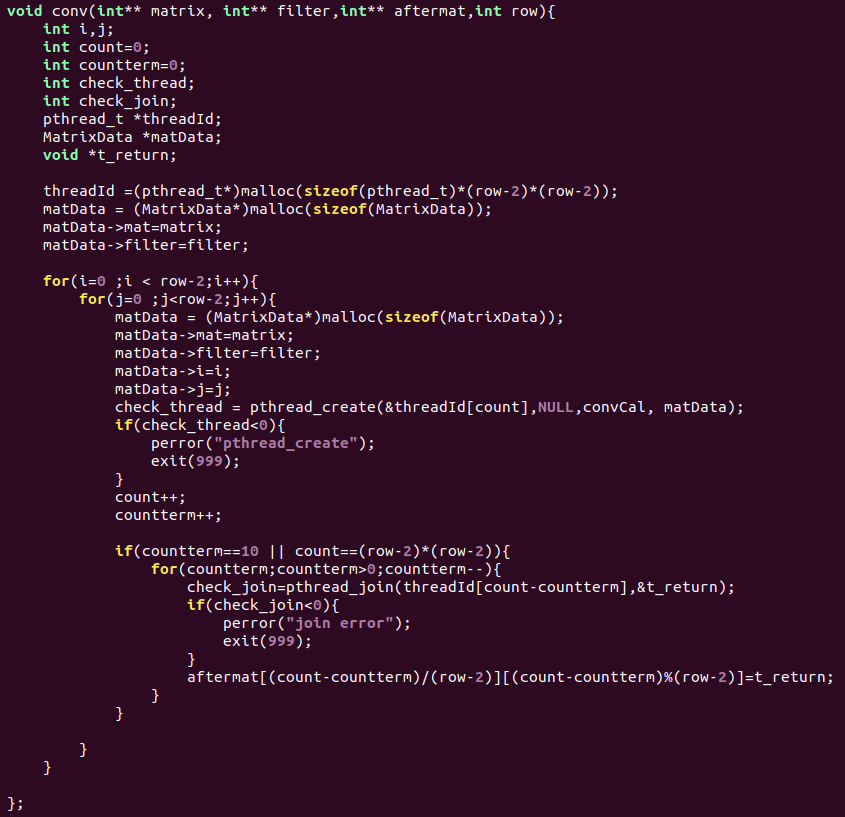
지난번 과제였던 Parallel CNN을 Message Queue가 아닌 Thread를 이용한 Parallel CNN을 구현하는 과제였습니다. 이번 과제에 대한 설명은 Thread를 이용하여 데이터를 계산하고, 이때 계산할 데이터를 File I/O를 이용하여 input file을 받아와 계산된 결과값을 output file에 저장시키게 합니다. 그러므로 지난번에 구현하였던 Convolution과 Maxpooling은 유사하게 사용하되, Thread를 사용하기 위해 데이터를 다른 방식으로 처리하였습니다.

Message Queue달리 Thread는 같은 메모리 주소 공간을 공유하기 때문에 다른 Thread가 사용 중인 변수를 건드리는 문제가 발생할 수도 있습니다. 따라서 이 문제를 처리하기 위해 구조체를 사용하였습니다. Convolution과 Maxpooling할 때 필요로 하는 layer, filter(Convolition), 인덱스인 i, j를 한 구조체로 구현해주었습니다. 이 구조체를 구조체pointer타입으로 선언해주어 현재 사용하는 위 값들의 주소 값을 넘겨주어서 thread는 이 값을 이용하여 계산을 한 뒤 저장된 주소에 값을 전달해서 데이터가 올바르게 전달되도록 하였습니다. 지난번 과제에서 구현했던 것처럼 이중 for문을 이용하여 각 인덱스 i, j의 경우마다 thread를 생성하여 병렬로 계산을 시킵니다. 이때 input의 size(=row)가 너무 클 경우 Thread의 제한 숫자를 넘어 버리는 경우가 발생하기 때문에 일정 개수의 thread를 생성할 경우 쓰레드를 종료하였습니다.

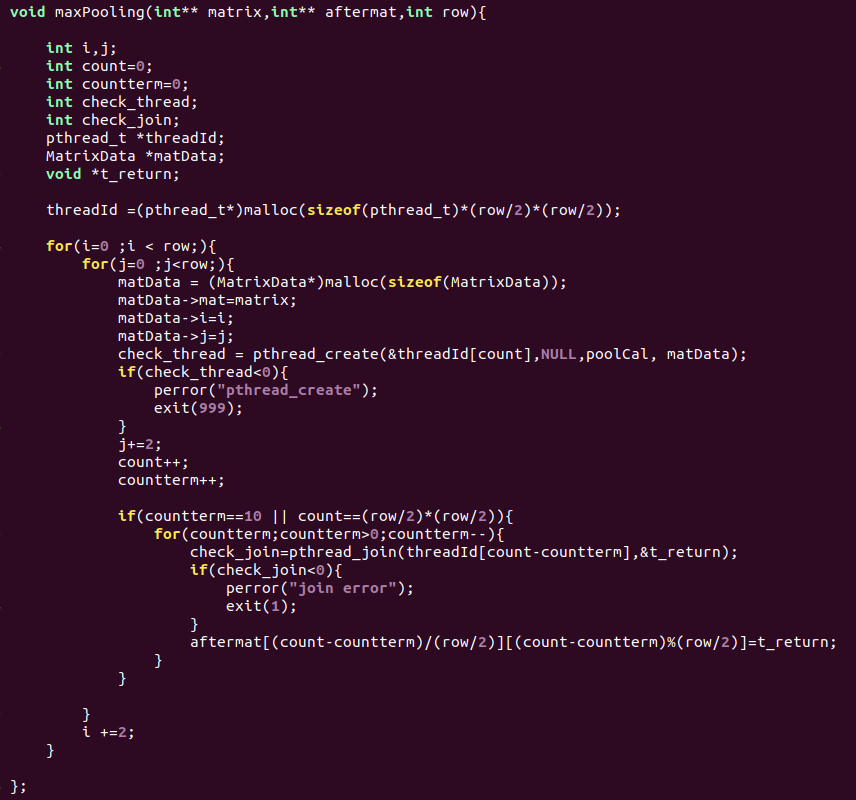
또한 이번 과제는 File I/O를 이용하여 input file에서 데이터를 가져와 계산 후 계산된 결과 값을 output file에 저장해야 했습니다. 이때 프로그램 실행 시 두가지 입력 값을 같이 받습니다. (ex) ./ku\_tconv input\_file output\_file ) 이 받은 input\_file이름으로 된 파일에서 주어진 형식은 첫 줄은 layer의 크기 두번째 줄부터 일정한 형식으로 쓰여진 데이터들이 존재합니다. 따라서 input을 읽어올 경우 첫째줄이 몇 글자인지 그때그때 다르므로 개행 문자를 만날 경우까지 읽어준 뒤 그 값을 row에 저장시킵니다. 그 후 input을 읽어와 layer에 하나씩 저장시켜줍니다. 이때 input의 경우 문자열로 읽어보기 때문에 layer에 저장해줄 때 atoi를 이용하여 정수형으로 전환한 뒤 저장해 줍니다. Output도 동일하게 저장해야 할 형식이 존재하므로 convolution과 maxpooling을 한 크기인 (row-2)/2으로 output을 저장해줍니다. Output은 input과 반대로 int형이기 때문에 문자열로 변환해준 뒤 저장해줍니다.

Thread를 이용한 Convolution과 Mxpooling의 자세한 코드는 아래 사진으로 넣어 놨습니다.

1.1 Convolutional Layer



2.1 Max-pooling



**2. Function description ( 함수 설명 )**

2.1 Convolution 과 Maxpooling을 시작하는 함수

|  |  |  |
| --- | --- | --- |
| Fuuction Name | Arguments | Description |
| conv | Int\*\* matrix; | Convolution을 계산하기 위해 데이터가 들어 있는 행렬 |
| Int\*\* filter; | Convolution을 하기 위해서 필요한 filter 행렬 |
| Int\*\* aftermat; | Convolution을 한 뒤 계산 값을 저장해서 보내주기 위한 행렬 |
| Int row; | 계산을 할 행렬의 가로(=세로) 값 |
| return void |  |

|  |  |  |
| --- | --- | --- |
| Fuuction Name | Arguments | Description |
| maxPooling | Int\*\* matrix; | Max-pooling을 계산하기 위해 데이터가 들어 있는 행렬 |
| Int\*\* aftermat; | Max-pooling을 한 뒤 계산 값을 저장해서 보내주기 위한 행렬 |
| Int row; | 계산을 할 행렬의 가로(=세로) 값 |
| return void |  |

2.2 Convolution과 Maxpooling중 index에서 계산 해주는 함수

|  |  |  |
| --- | --- | --- |
| Fuuction Name | Arguments | Description |
| convCal | void \*temp; | Convolution을 계산하기 위해 데이터가 들어 있는 구조체 포인터 타입을 받기 위한 void\*타입의 변수 |
| return void\* | matrix[i][j]를 기준으로 filter만큼 Convolution을 한 계산 값 (단, 쓰레드를 이용하기 위해 void\*타입을 이용하였습니다.) |

|  |  |  |
| --- | --- | --- |
| Fuuction Name | Arguments | Description |
| poolCal | void \*temp; | Max-pooling을 계산하기 위해 데이터가 들어 있는 구조체 포인터 타입을 받기 위한 void\*타입의 변수 |
| return void\* | matrix[i][j]를 기준으로 정해진 Max-pooling의 region만큼 Max-pooling을 한 계산 값  (단, 쓰레드를 이용하기 위해 void\*타입을 이용하였습니다.) |

2.3 동적할당 및 동적할당 정리 해주는 함수

|  |  |  |
| --- | --- | --- |
| Fuuction Name | Arguments | Description |
| makeAllocation | Int\*\* matrix ; | 동적 할당을 시켜줄 int\*\*타입의 matrix |
| Int row ; | 동적할당을 해줄 가로(=세로)의 값 |
| return int\*\* | 동적할당이 된 int\*\*타입의 값 |

|  |  |  |
| --- | --- | --- |
| Fuuction Name | Arguments | Description |
| freeAllocation | Int\*\* matrix ; | 동적 할당할 정리할 int\*\*타입의 matrix |
| Int row ; | 동적할당을 정리 할 가로(=세로)의 값 |
| return void |  |