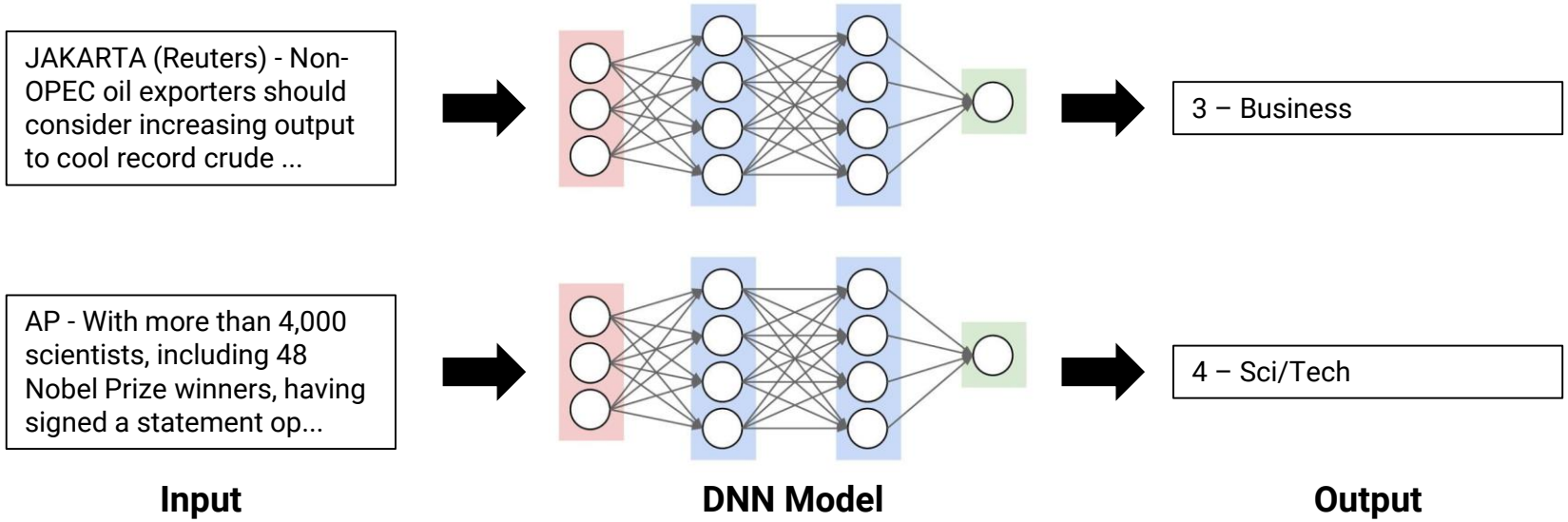
Final Project – Text Classifier

Project due: 2023.12.18. 11:59 PM

Last modified: 2023.12.05. 14:00

프로젝트 목표

- 주어진 DNN 추론 프로그램을 병렬화 및 최적화
  - CNN-based text classification model
  - 순차 코드 (CPU, single-thread) 로 구현되어 있음
- 4개 계산 노드의 모든 자원을 사용해 최대한 높은 성능을 달성할 것



뼈대 코드

- `/shpc23/skeleton/final-project`
  - 수정 가능한 파일: `classifier.cu Makefile run.sh`
  - 수정 불가능한 파일: `main.cpp util.cpp util.h classifier.h`
  - 수정 불가능한 파일들은 채점 시 뼈대 코드의 것으로 덮어씌워짐

- `make` 로 컴파일, `./run.sh` 스크립트로 실행
  - `-n`: 입력 텍스트 개수 지정 (`1 ~ 8192`)
  - `-v`: 결과 검증 여부 지정

뼈대 코드 (cont'd)

- `main.cpp`
  - 파일에서 입출력 및 모델 파라미터를 읽어오고 classifier 함수들을 호출
  - 입출력 배열 및 모델 파라미터 정보는 루트 프로세스 (rank=0) 에만 있음에 주의
- `classifier.cu`
  - `initialize_classifier()` - 모델 파라미터 등 작업에 필요한 메모리를 할당하는 부분
  - `finalize_classifier()` - 동적으로 할당한 자원을 해제하는 부분
  - `classifier()` - 실제 연산을 수행하는 부분
- `run.sh`
  - 자신의 병렬화 방식에 맞게 실행 스크립트를 수정할 것
  - 몰래 4노드보다 더 많은 자원을 사용하면 0점 처리

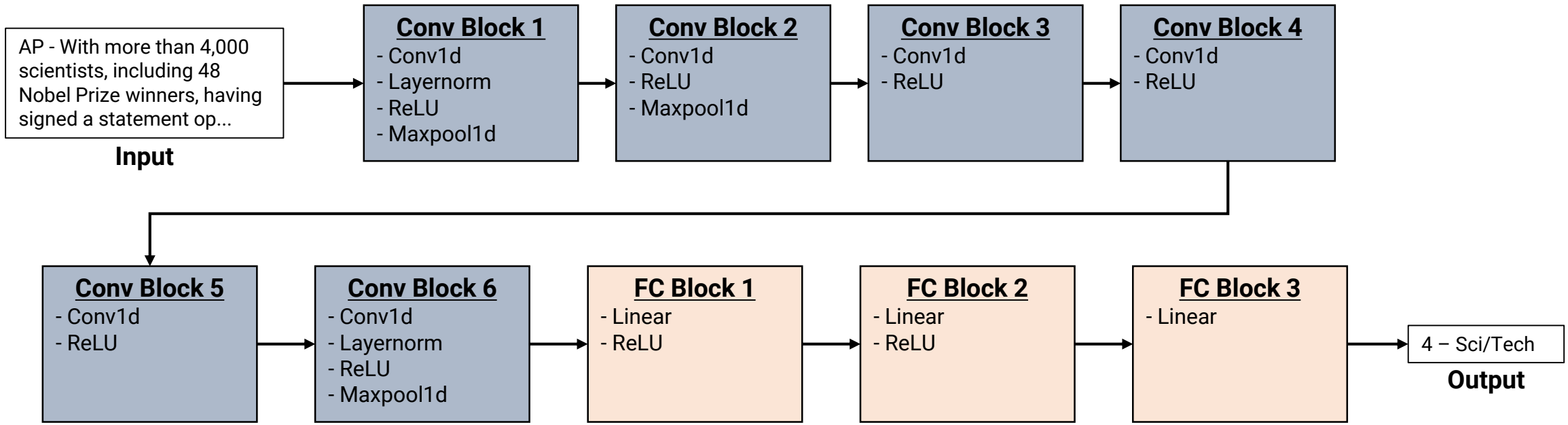
실행 예시

```
shpcta@login0:~/final-project$ ./run.sh -n 10 -v

Model : Classifier
=======================================
Number of inputs : 10
Validation : ON
---------------------------------------
Loading inputs ... DONE
Initializing classifier ... DONE
Classifying 10 articles ... DONE
=======================================
Elapsed time : 4.876319 s
Throughput   : 2.050727 input(s)/sec
Validation   : Pass
```

최적화 대상 딥 러닝 모델 구조

- “Character-level Convolutional Networks for Text Classification” 논문을 기반으로, 조교가 수정한 모델



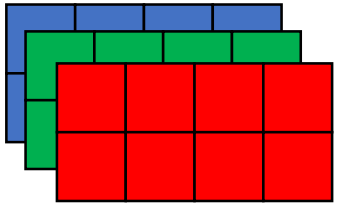
텐서 (Tensor)

- 딥 러닝에서 데이터를 다루는 기본 단위
  - 다차원 텐서의 원소들이 메모리에 저장되어 있는 순서를 잘 이해할 것
  - 행렬 (2D Tensor) 에서 시작해 다차원 텐서로 확장해서 생각해보기

1D Tensor
(e.g., Vector)

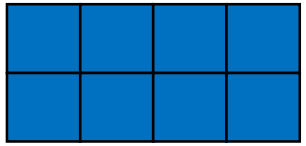
Shape = {4}

3D Tensor
(e.g., RGB
image)



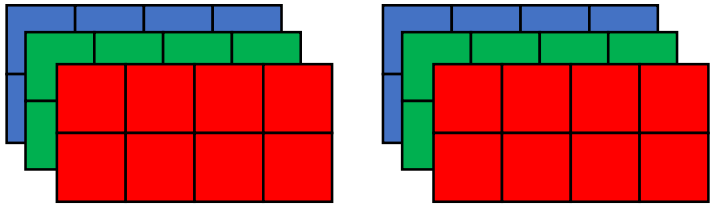
Shape = {3, 2, 4}

2D Tensor
(e.g., Matrix)



Shape = {2, 4}

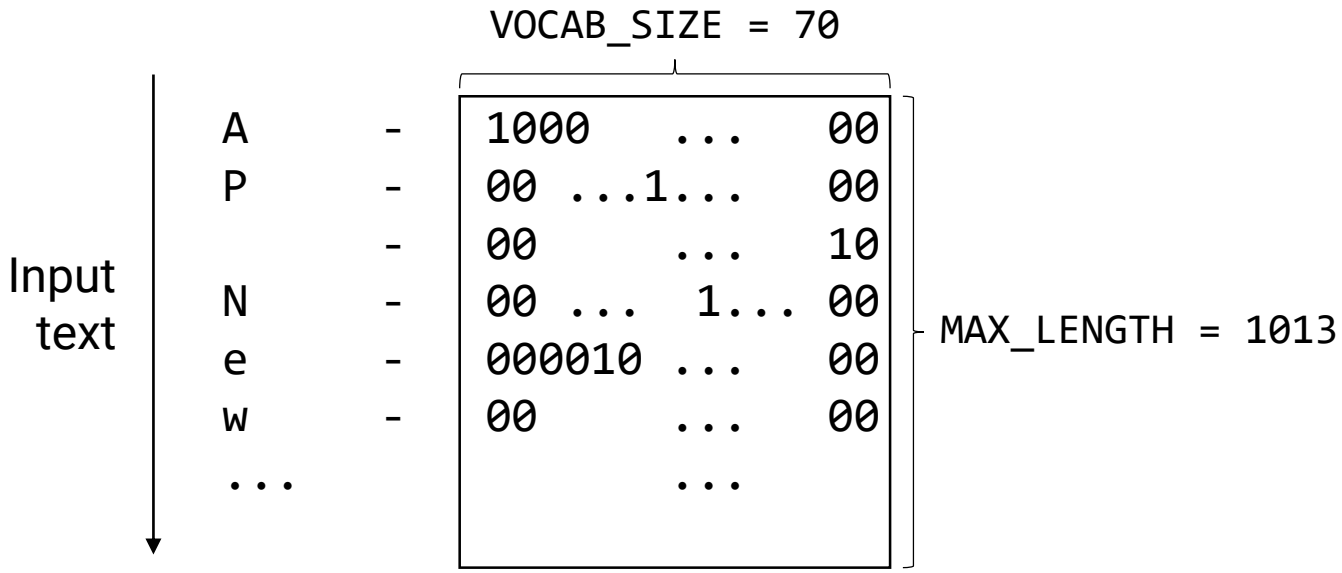
4D Tensor
(e.g., CNN filter)



Shape = {2, 3, 2, 4}

입/출력 데이터 형태

- 입력 텍스트는 one-hot 벡터 형태로 주어짐
  - [N x VOCAB_SIZE x MAX_LENGTH] 크기를 가지는 실수 배열



- 출력은 해당 텍스트의 분류를 나타내는 정수 1개
  - [N] 크기를 가지는 실수 배열

딥 러닝 레이어 목록

1. Conv1d - https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html
2. ReLU - https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html
3. Maxpool1d - https://pytorch.org/docs/stable/generated/torch.nn.MaxPool1d.html
4. Linear - https://pytorch.org/docs/stable/generated/torch.nn.Linear.html
5. Layernorm - https://pytorch.org/docs/stable/generated/torch.nn.LayerNorm.html

- PyTorch 문서 및 뼈대 코드를 참고해 각 레이어의 작동방식을 이해할 것
  - 각 레이어들의 이론적 배경을 이해할 필요는 없음
  - 레이어별 연산 특징 및 연산간 dependence 를 이해하는 것이 중요

제약 조건

- x86 intrinsics, Pthread, OpenMP, MPI, CUDA 외 라이브러리 사용 불가능
  - 사용 불가능한 라이브러리 예시 : cuBLAS, cuDNN, MAGMA, BLIS, PyTorch, Tensorflow, ...
  - OpenCL 을 사용하고자 할 경우 조교에게 문의
- 프로그램 로직 혹은 모델 구조를 변경하는 수정 불허
  - 가능한 수정의 예
    - 메모리 레이아웃 변경, 루프 순서 변경, 패딩 데이터/연산 추가, fusion 등
  - 불가능한 수정의 예
    - `initialize_classifier()` 에서 모델 추론 관련 연산 수행
    - 동일한 출력을 만들어내는 다른 모델/알고리즘 사용
    - 일부 연산 생략
- 4개 노드보다 많은 자원을 사용하거나, 상식 선에서 허용되지 않는 수정 불허
- 애매한 것은 etl 게시판에 문의할 것

채점 방식

- 성능 (80%) - 프로그램 성능(Throughput)에 따른 log-scale 상대 평가
  - 1등에 비해 2배 느릴때마다 10% 감점
  - E.g., 1등 프로그램이 100 inputs/sec 라면 25 inputs/sec 프로그램은 성능 점수의 80% 부여
  - 원하는 2의 지수승 입력 개수 (n) 에 대해 성능 측정
  - 보고서와 run.sh 에 입력 개수를 적어 둘 것
  - 답이 틀릴 경우 0점

- 보고서 (20%)
  - report.pdf
  - 양식 및 분량은 자유
  - 자신의 병렬화, 최적화 방식을 위주로 간결하게 작성
  - 자신이 측정한 프로그램의 성능을 포함할 것. 조교가 측정한 성능과 크게 차이가 나는 경우 확인을 위함

프로젝트 제출

- Deadline : 12월 18일 오후 11:59:59
  - **Grace day 사용 불가**
  - **일찍 시작할 것**

- 제출 스트립트를 사용해 4개 파일을 제출
  - `shpc-submit submit final-project classifier.cu`
  - `shpc-submit submit final-project Makefile`
  - `shpc-submit submit final-project report.pdf`
  - `shpc-submit submit final-project run.sh`

Comments

- 뼈대 코드를 잘 이해한 뒤 시작하는 것을 추천

- 항상 근거를 가지고 최적화를 해야 함
  - 최적화에 앞서 어느 부분이 문제인지 찾아내는 것이 필수

- 생각 해볼만한 것들 (조교의 추측일 뿐, 정답은 없음)
  - 여러개의 입력을 묶음으로 (batch) 처리
  - 그동안 해 왔던 행렬 곱 최적화 방식을 다른 연산에도 적용
  - 레이어 내 최적화와 레이어 밖의 최적화를 모두 고려할 것
  - 프로그램이 의도한 대로 잘 작동하는지 확인하기
  - 주어진 HW 자원을 잘 활용하고 있는지 확인하기

Comments (cont'd)

- 일찍 시작할 것
  - 마감 1~2일 전에는 작업이 몰려 제대로 된 실험이 불가능
  - 마감 기한 연장 불가
  - Grace day 사용 불가

- 추가 내용은 본 슬라이드 뒤에 업데이트 됨

Updates

[2023.12.05 14:00] 입력 텍스트 개수 최댓값을 8192 로 변경