

Homework 08

Image Classification



과제 개요

- Multi Layer Perceptron을 사용해 Image Classification을 하는 과제



Bed Room



Living Room



Dining Room

Prediction

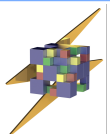
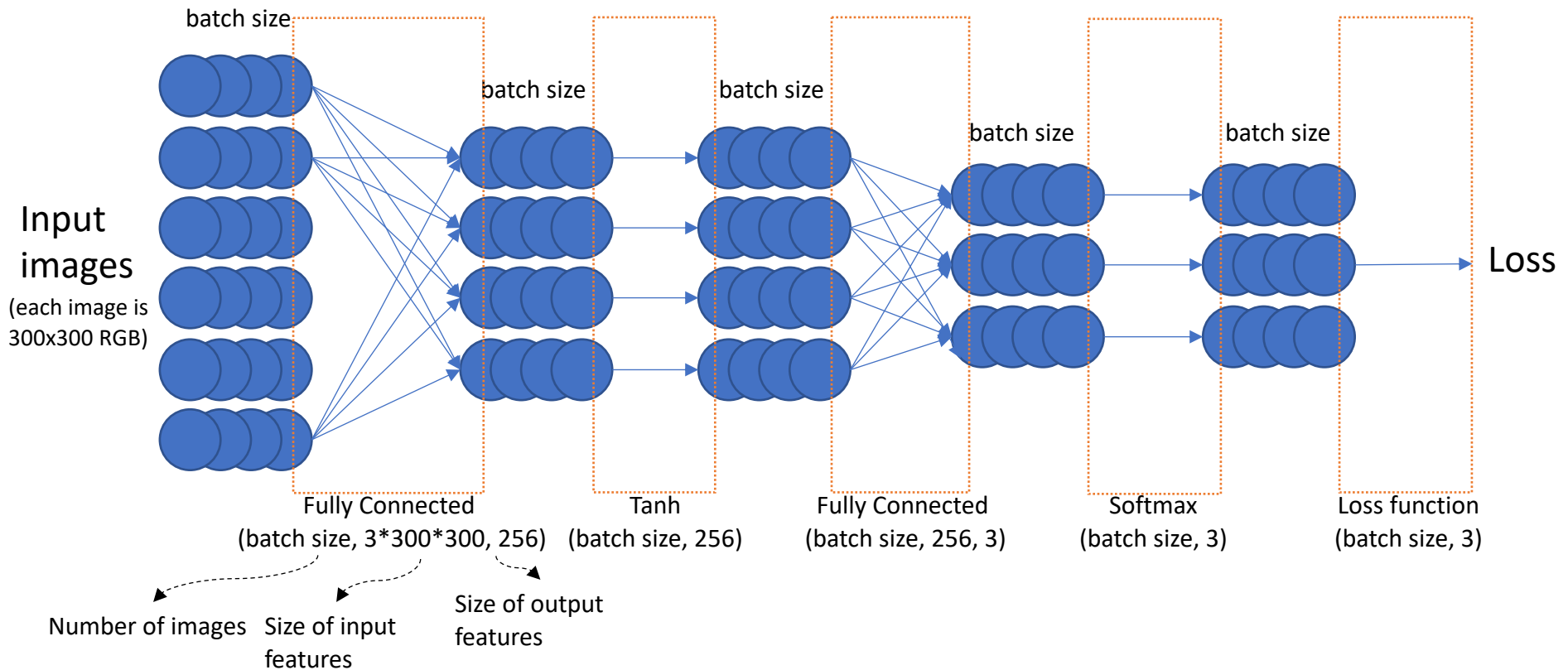


Input images



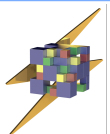
타겟 모델

- Fully Connected Layer 와 Activation Function 으로 구성된 형태



모델에 사용된 레이어

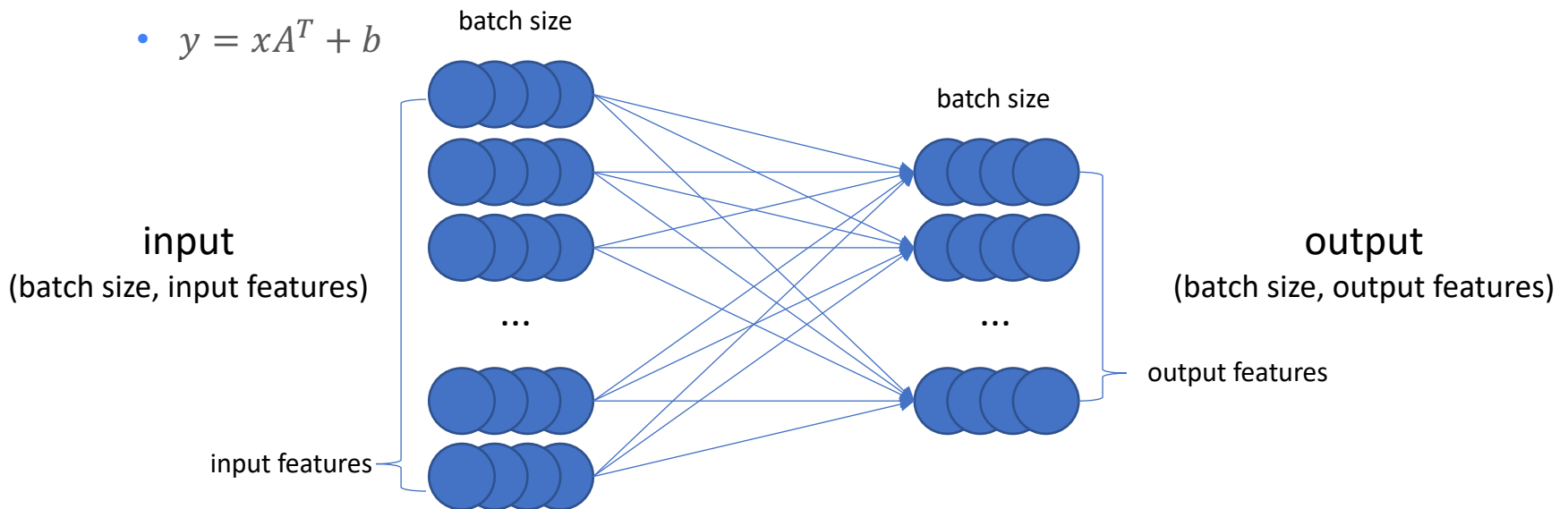
- Fully Connected Layer
- Activation Function
 - Tanh, Softmax
- Categorical Cross-Entropy Loss



Fully Connected Layer

- Applies a linear transformation to the incoming data

- $$y = xA^T + b$$

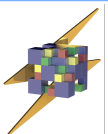


- Configurations

- 이번 과제에서 사용하는 FC layer는 weight 값만 있고 bias는 없음
- 총 2종류의 FC Layer가 사용됨
 - FC0(270000, 256), FC1(256,3)

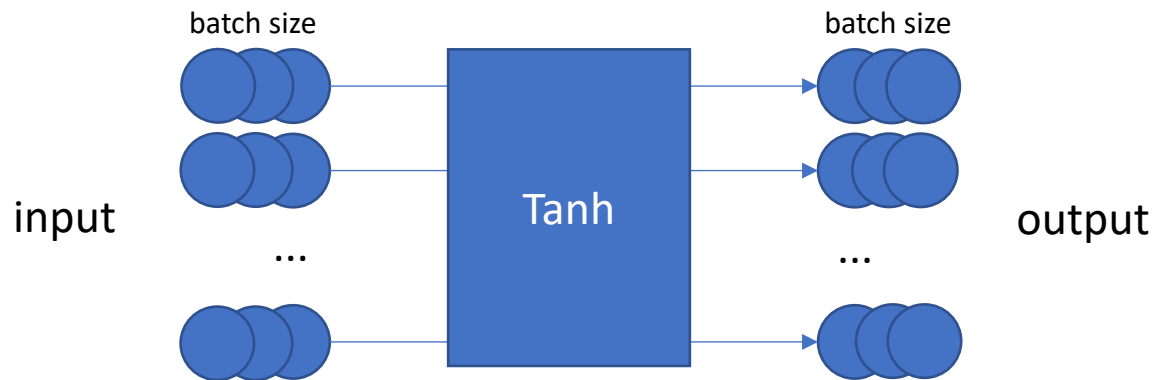
Size of input

Size of output

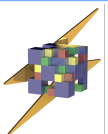


Tanh

- Returns a hyperbolic tangent of the elements



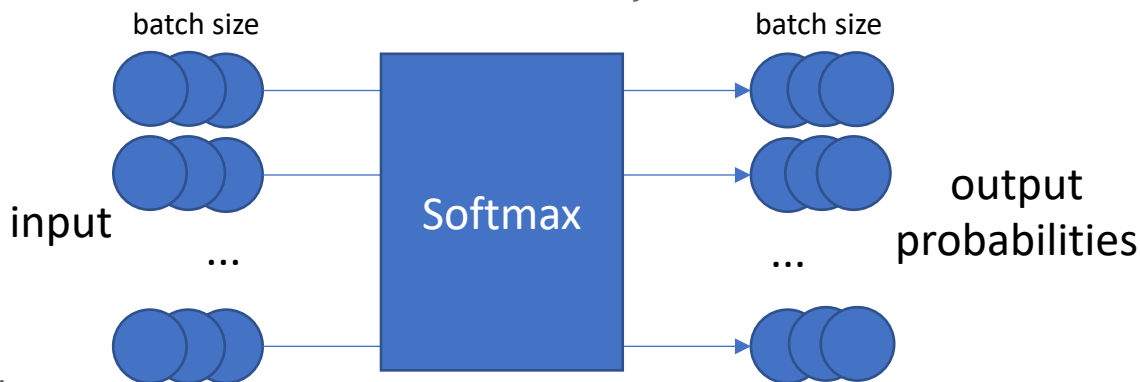
- Configuration
 - input : (batch size, 256)
 - output : (batch size, 256)



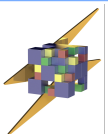
Softmax

- Applies the Softmax function to an n-dimensional input rescaling them so that the elements of the n-dimensional output lie in the range $[0,1]$ and sum to 1

- Softmax function
$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$



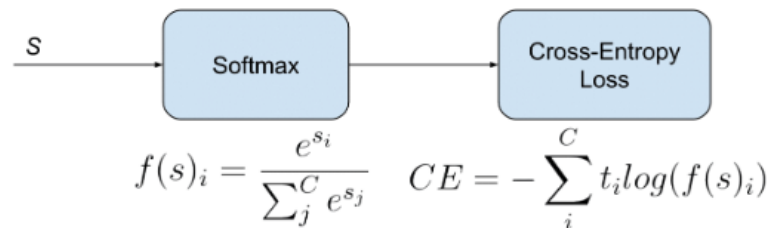
- Configuration
 - input : (batch size, 3)
 - output : (batch size, 3)



Categorical Cross-Entropy Loss

- Computes the cross entropy loss between input and target
- Softmax의 output 값을 토대로 실제 target값과의 loss를 계산
 - 3개의 class (bedroom, dining room, living room) 를 one hot encoding

- bedroom = [1, 0, 0]
- dining room = [0, 1, 0]
- living room = [0, 0, 1]

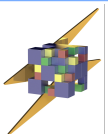
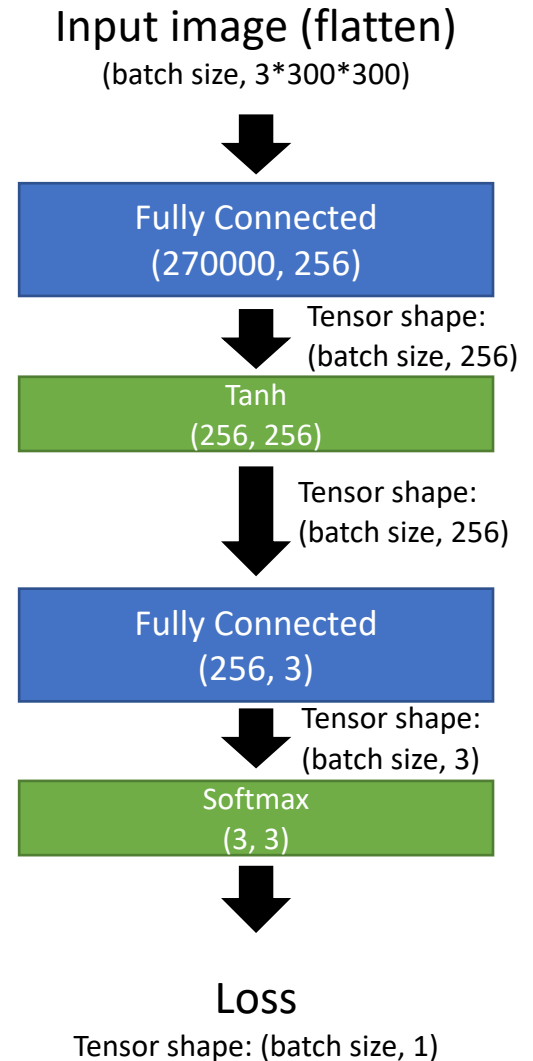


- For example,
 - Softmax의 output이 [0.1, 0.85, 0.05] 이고,
 - 실제 target값이 living room인 경우,
 - CrossEntropy Loss = $-1 * (0 * \ln 0.1 + 0 * \ln 0.85 + 1 * \ln 0.05) = 2.9957$



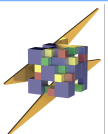
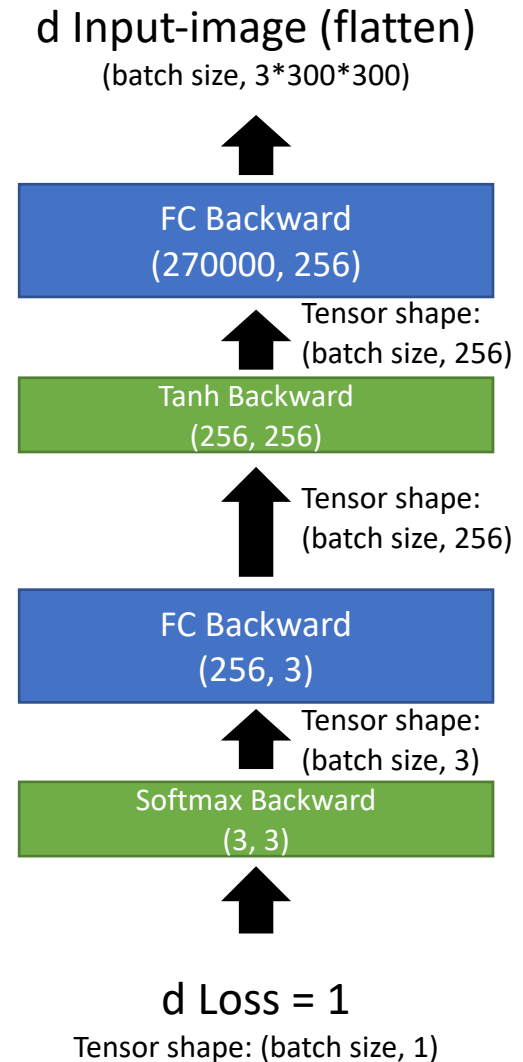
Forward Pass

- Image 1개는
 - 300x300 pixels
 - RGB channels (Red, Green, Blue)
 - 따라서 image 1개를 표현하기 위한 tensor의 shape은 $3 \times 300 \times 300$
- Input
 - Batch size개의 images
 - (batch size, $3 \times 300 \times 300$)
- Output
 - Batch size개의 images 각각에 대한 loss



Backward Pass

- Input gradient
 - Backward pass에 loss function의 input으로 들어가는 gradient (dLoss)는 1
- Output
 - Input image에 대한 gradient 값



과제 목표

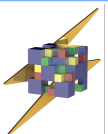
- 모델의 전체적인 뼈대와 이미지 입출력 등의 기능 등이 구현된 코드가 주어질 예정
 - Forward pass와 backward pass의 뼈대가 구현되어 있음
 - Train dataset, test dataset 이 포함되어 있음
- 모델에 사용된 각 레이어의 backward function을 구현하는 것이 이번 과제의 목표
 - Forward functions, Optimizer, 이미지 파일 I/O등의 기능들은 모두 구현되어 있음
 - 구현되어 있는 forward function 코드와 ETL에 업로드 되어 있는 강의자료를 참고하여 backward function 에 해당하는 코드를 작성하면 됨



과제 파일 구성

- Makefile
- config.hpp : MLP의 configuration이 적혀 있는 파일
- main.cpp : MLP를 train/inference하기 위한 내용이 적혀 있는 파일
- op.cpp : MLP에 사용되는 Layer가 구현된 파일
- tensor.cpp : MLP에 사용되는 Tensor가 구현된 파일
- util.cpp : File I/O 등 유틸리티 파일
- tool.cpp : Layer의 backward function 이 제대로 구현되었는지 테스트해주는 내용이 구현된 파일
- dataset : 이미지 파일 디렉토리

해당 파일에서 레이어 별 backward function만 구현하면 됨
(구현 해야할 부분이 코드 내에 TODO로 마킹되어 있음)



프로그램 실행

- 컴파일 방법
 - 주어진 Makefile사용
 - make run
 - Training이 끝나면 이어서 inference를 함
- 실행 예시
 - 우측 박스 참조
 - 제대로 구현했을 시 약 3~5분 정도의 실행시간이 소요됨
 - config.hpp 파일을 수정하여 모델의 하이퍼파라미터를 조정할 수 있음

```

Layer      Shape
=====
Linear_0    (30, 3*300*300, 256)
Tanh_0      (30, 256)
Linear_1    (30, 256, 3)
Softmax_0   (30, 3)
=====

Hyper parameters
=====
Epochs      3
Number of images for training  30
Number of images for test     9
Learning Rate  0.01
=====

Initialize Dataset from image files
=====
Loaded bed room images
Loaded dining room images
Loaded living room images

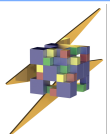
Start Training
=====
[Epoch 1/3] Loss : 1.09865
[Epoch 2/3] Loss : 1.09862
[Epoch 3/3] Loss : 1.0986

Inference : O(Correct), X(Wrong)
=====
Predict image[0] as Bed Room  (O)
Predict image[1] as Bed Room  (O)
...
=====
Accuracy : 0.333333
  
```



결과 검증

- Layer의 backward function 이 제대로 구현되었는지 테스트를 위한 툴 제공
 - make tool
 - FC, Tanh, Softmax, Categorical CrossEntropy Loss 함수의 결과 검증 가능
 - ./tool [test layer] [number of sample images]
 - 0 : FC layer
 - 1 : Tanh
 - 2 : Softmax
 - 3 : Categorical CrossEntropy Loss
 - Validation tool 사용 예시
 - ./tool 0 5 (fc layer, 5 images)
 - ./tool 2 6 (softmax, 6 images)



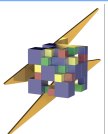
구현 시 주의 사항

- Training은 시간이 오래 걸리므로 제공되는 validation tool을 이용해 backward function이 제대로 구현되었는지 먼저 확인을 할 것
 - 본 슬라이드 13페이지 참고
- 별도의 library 사용 불가
 - C math library는 사용 가능
 - 애매하면 조교에게 문의



제출 안내

- 제출 내용
 - 보고서
 - Epoch, Learning Rate 등을 수정했을 때 loss 값, accuracy가 어떻게 변화하는지 관찰하고 간략하게 정리하여 적을 것
 - 실행 화면 1개 첨부
 - 소스 코드
 - Backward function 구현 내용만 채점할 것이므로 op.cpp 파일만 제출
- 제출 형태
 - cfdXXX_HW08.zip (또는 다른 압축파일 확장자)
 - report.pdf
 - op.cpp
- ETL로 제출
- 6월 17일 (금) 23:59까지 제출
- Grace day 사용 가능



평가

- 보고서 (20%)
- 소스 코드 (80%)
 - Backward function이 제대로 구현되어 있는지 평가

