

Deep learning Report

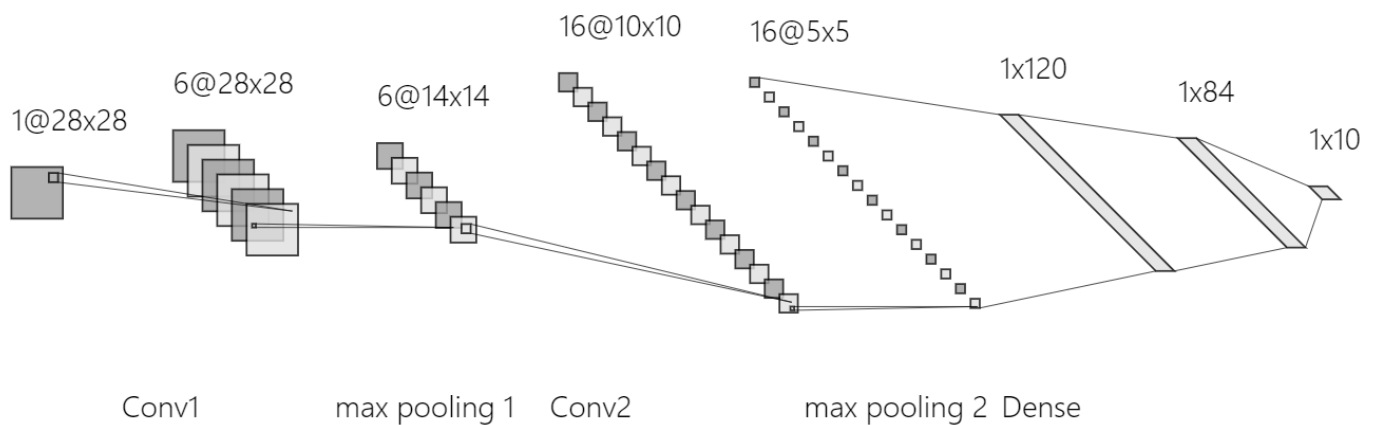
MNIST classification

데이터사이언스학과

20510076 김인조

1. Model Architecture and number of parameter

1-1. Lenet 5



1. Convolution : kernel size = 5, kernel num = 6, padding = same, activation = relu

- Parameters = $(5 \times 5 \times 1) \times 6 = 150$

2. Max pooling : kernel size, stride = 2

3. Convolution : kernel size = 5, kernel num = 16, padding = valid, activation = relu

- Parameters = $(5 \times 5 \times 6) \times 16 = 2,400$

4. Max pooling : kernel size, stride = 2

5. FCN

- Parameters = $(5 \times 5 \times 16) \times 120 = 48,000$
- Parameters = $120 \times 84 = 10,080$
- Parameters = $84 \times 10 = 840$

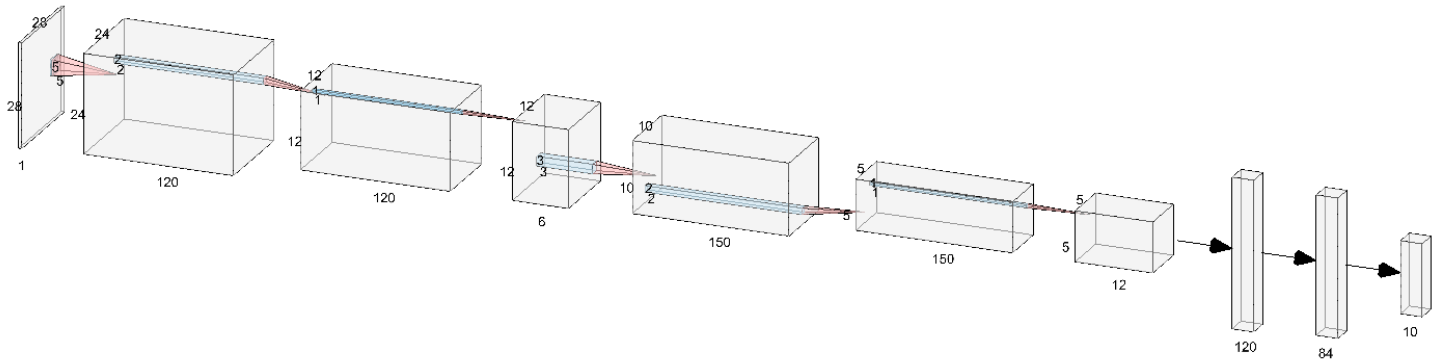
Total parameters = $61,470 \times 2(\text{backpropagation}) = 122,940$

1-2. Regularized Lenet 5

사용한 정규화 방법

1. Cost function에 weight term 추가 → 0.001
2. FCN layer에 dropout 추가 → rate : 0.3

1-3. Custom model



1. Convolution : kernel size = 5, kernel num = 120, activation = relu

- Parameters = $(5 \times 5 \times 1) \times 120 = 3,000$

2. max pooling : kernel size, stride = 2

3. Convolution : kernel size = 1, kernel num = 6, activation = relu

- Parameters = $(1 \times 1 \times 120) \times 6 = 720$

4. Convolution : kernel size = 3, kernel num = 150, activation = relu

- Parameters = $(3 \times 3 \times 6) \times 150 = 8,100$

5. max pooling : kernel size, stride = 2

6. Convolution : kernel size = 1, kernel num = 12, activation = relu

- Parameters = $(1 \times 1 \times 150 \times 12) = 1,800$

7. FCN

- Lenet의 FCN layer의 parameter와 동일 = $48,000 + 10,080 + 840$

Total parameters = $60,540 \times 2(\text{backpropagation}) = 121,080$

Convolution layer를 통과한 후에 1x1 kernel convolutional layer를 배치하여 학습 시간에서 유의미한 성능 향상 기대

2. Experiment & Result

2-1. 실험 조건

Epoch = 10

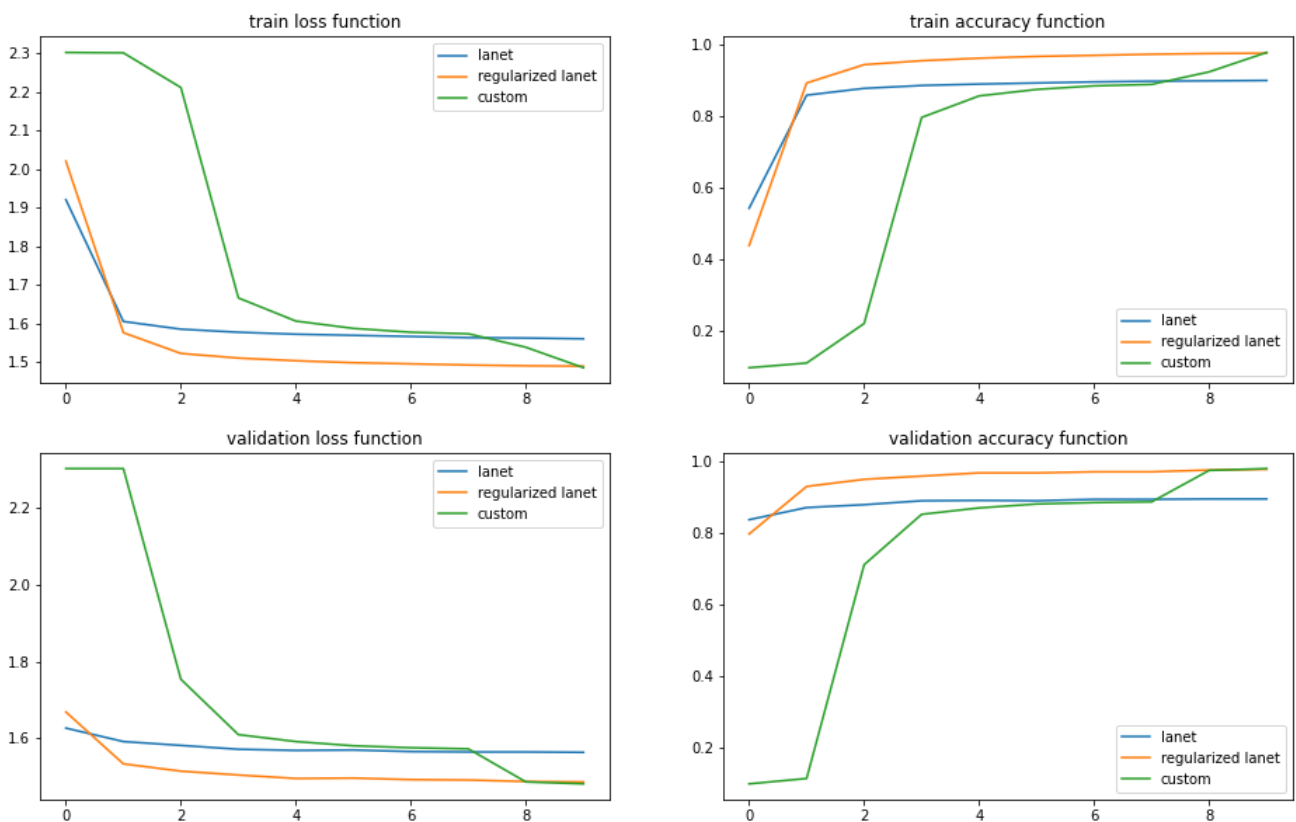
Cost function = cross entropy loss

Optimizer = SGD with learning rate 0.01, momentum 0.9

2-2. 실험 결과

실험 결과 epoch 15 내외에서 saturation됨

본 레포트에서는 각 모델 별 성능 비교를 위해 epoch를 10으로 한 결과를 첨부



실험 결과 regularized lenet 모델이 바닐라 모델에 비해 더 높은 정확도를 보임과 동시에 더 빠르게 saturation됨을 확인

Custom 모델은 학습 초반에는 기존 Lenet 모델보다 저조한 성능을 보이나 특정 epoch를 넘기는 시점에서 급격하게 성능이 향상함을 확인

이는 validation set에서도 완벽히 동일한 결과로 overfitting 없이 적절히 학습되었음을 알 수 있음

Lenet model과 custom model의 architecture가 달라 실행시간의 직접적인 비교는 어려움

100개 이상의 filter를 사용한 custom 모델이 lenet 모델보다 상대적으로 실행속도가 느림을 확인

대조군으로 custom model과 같은 수의 kernel을 사용하지만 1x1 kernel layer를 거치지 않는 모델과 비교 시 유의미한 속도의 향상이 있을 것으로 기대