

# Assignment\_Report\_pdf 버전

Assignment\_Report 원본 링크를 README에 올려놓았습니다 !

pdf는 폰트 사이즈 이슈로 내용을 몇가지 생략하여, 원본 링크를 봐주시길 부탁드립니다.

## 1. dataset.py

- github [dataset.py](#) 업로드

## 2. LeNet-5, MLP 모델 구현

**컨볼루션 레이어 파라미터 계산** : 파라미터 수 = (커널 높이×커널 너비×입력 채널 수 +1(bias))×출력 채널 수

**FC 레이어 파라미터 계산** : 파라미터 수 = (입력 특성의 수+1(bias))×출력 특성의 수 / 입력 특성의 수 : 이전 레이어의 출력 size

- 코드의 각 레이어의 주석으로 파라미터 계산 공식 적어놓음

**LeNet 파라미터 수 :**

Total parameters = 61,706 (156 + 2416 + 48120 + 10164 + 850)

Forward/backward parameters = 123,412

**CustomMLP 파라미터 수 :**

Total parameters = 61,720 (50240+4160+4160+2080+528+272+170+110)

Forward/backward parameters = 123,440

**Regularized LeNet-5 model 파라미터 수 :**

정규화에 따라 파라미터 수가 기존과 변경됨. 6번에서 설명

### 3. LeNet-5, custom MLP Training

#### LeNet-5 Model

Epoch: 10

Optimizer: SGD

Batch size: 32

#### epoch별 평균 성능

1 epochs

training loss: 0.0305

training accuracy: 0.5823

validation loss: 0.0253

validation accuracy: 0.8932

2 epochs

training loss: 0.0244

training accuracy: 0.9149

validation loss: 0.0241

validation accuracy: 0.9343

3 epochs

training loss: 0.02389

training accuracy: 0.9427

validation loss: 0.02379

validation accuracy: 0.9530

4 epochs

training loss: 0.0235

training accuracy: 0.9598

validation loss: 0.02356

validation accuracy: 0.9677

5 epochs

training loss: 0.0234

training accuracy: 0.9691

validation loss: 0.0234

validation accuracy: 0.9739

6 epochs

training loss: 0.0233

training accuracy: 0.9751

validation loss: 0.0233

validation accuracy: 0.9781

7 epochs

training loss: 0.0232

training accuracy: 0.9790

validation loss: 0.0233

validation accuracy: 0.9810

8 epochs

training loss: 0.0231

training accuracy: 0.9814

validation loss: 0.0232

validation accuracy: 0.9827

9 epochs

training loss: 0.0231

training accuracy: 0.9830

validation loss: 0.0232

validation accuracy: 0.9839

10 epochs

training loss: 0.0231

training accuracy: 0.9837

validation loss: 0.0232

validation accuracy: 0.9831

---

## Custom MLP model

Epoch: 10

Optimizer: SGD

Batch size: 32

### epoch별 평균 성능

1 epochs

training loss: 0.0360

training accuracy: 0.0979

validation loss: 0.0361

validation accuracy: 0.0958

2 epochs

training loss: 0.0360

training accuracy: 0.0981

validation loss: 0.0362

validation accuracy: 0.1010

3 epochs

training loss: 0.0360

training accuracy: 0.1059

validation loss: 0.0361

validation accuracy: 0.1135

4 epochs

training loss: 0.0360

training accuracy: 0.1121

validation loss: 0.0361

validation accuracy: 0.1135

5 epochs

training loss: 0.0360

training accuracy: 0.1124  
validation loss: 0.0361  
validation accuracy: 0.1135

6 epochs  
training loss: 0.0360  
training accuracy: 0.1124  
validation loss: 0.0361  
validation accuracy: 0.1135

7 epochs  
training loss: 0.0360  
training accuracy: 0.1124  
validation loss: 0.0361  
validation accuracy: 0.1135

8 epochs  
training loss: 0.0360  
training accuracy: 0.1124  
validation loss: 0.0361  
validation accuracy: 0.1135

9 epochs  
training loss: 0.0360  
training accuracy: 0.1124  
validation loss: 0.0361  
validation accuracy: 0.1135

10 epochs  
training loss: 0.0360  
training accuracy: 0.1144  
validation loss: 0.0360  
validation accuracy: 0.1135

---

## Regularized LeNet-5 model

Epoch: 10

Optimizer: SGD

Batch size: 32

1 epochs

training loss: 0.0269

training accuracy: 0.8041

validation loss: 0.0238

validation accuracy: 0.9601

2 epochs

training loss: 0.0240

training accuracy: 0.9483

validation loss: 0.0235

validation accuracy: 0.9747

3 epochs

training loss: 0.0238

training accuracy: 0.9581

validation loss: 0.0235

validation accuracy: 0.9734

4 epochs

training loss: 0.0237

training accuracy: 0.9629

validation loss: 0.0234

validation accuracy: 0.9778

5 epochs

training loss: 0.0236

training accuracy: 0.9652

validation loss: 0.0234

validation accuracy: 0.9810

6 epochs

training loss: 0.0236

training accuracy: 0.9670

validation loss: 0.0234

validation accuracy: 0.9804

7 epochs

training loss: 0.0236

training accuracy: 0.9675

validation loss: 0.0234

validation accuracy: 0.9813

8 epochs

training loss: 0.0236

training accuracy: 0.9688

validation loss: 0.0235

validation accuracy: 0.9783

9 epochs

training loss: 0.0236

training accuracy: 0.9692

validation loss: 0.0234

validation accuracy: 0.9829

10 epochs

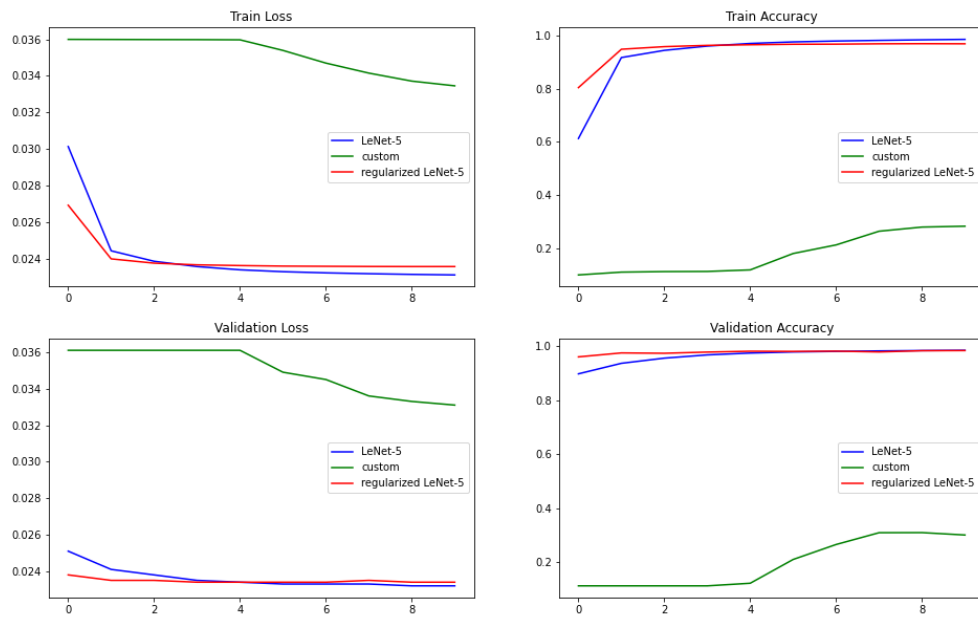
training loss: 0.0236

training accuracy: 0.9688

validation loss: 0.0234

validation accuracy: 0.9834

## 4. Plot above statistics



## 5. Compare the predictive performances of LeNet-5 and custom MLP

마지막 epoch을 기준으로 3개의 모델 중 평균 loss값은 custom model이 가장 높게 나왔다. regularized LeNet-5과 original LeNet-5모델은 큰 차이가 없었지만, original LeNet-5의 loss가 미세하게 더 낮은 것을 확인하였다. 실험에서는 epoch을 10까지 했지만, epoch을 늘릴 경우 regularized LeNet-5의 loss는 더 낮아질 것으로 예상된다.

평균 accuracy의 경우 LeNet-5 모델이 가장 높은 정확도를 보여주었으며, 뒤로 마찬가지로 accuracy에서도 미세하게 original LeNet-5 모델이 더 높은 것을 확인하였다. custom model의 accuracy는 나머지 두 모델에 비해서 낮은 값이 나왔다.

LeNet-5는 입력 이미지에서 공간적인 구조를 효과적으로 캡처할 수 있는 합성곱 레이어를 포함하고 있으므로 단순히 MLP만 쌓아서 구현한 Custom model보다 좋은 성능을 보이는 것으로 예상된다.



## 6. Improve LeNet-5 model

사용한 정규화 방법 : batchNorm, Dropout

정규화에 따라 파라미터 수가 기존과 변경됨

### BatchNorm2d(6)

- 첫 번째 컨볼루션 레이어의 in\_channels=6이므로, 각 채널당 2개의 파라미터 추가
- 따라서 총 12개의 파라미터 추가

### BatchNorm2d(16)

- 두 번째 컨볼루션 레이어 in\_channels=12이므로, 각 채널당 2개의 파라미터 추가
- 따라서 총 32개의 파라미터 추가

### BatchNorm1d(120)

- 첫 번째 FC layer 이후에는 in\_features=120이므로, 각 feature당 2개의 파라미터 추가
- 따라서 총 240개의 파라미터 추가

### BatchNorm1d(84)

- 두 번째 FC layer 이후에는 in\_features=84개이므로, 각 feature당 2개의 파라미터 추가
- 따라서 총 168개의 파라미터 추가

정규화로 인해  $12+32+240+168=452$ 개의 파라미터가 추가

**기존 파라미터 수 : 61706**

**변화된 파라미터 수 : 62158**

