

13주차 과제

과목명	딥러닝 실제
담당교수	전명근 교수님
학과	산업인공지능학과
학번	2021254009
이름	정원용



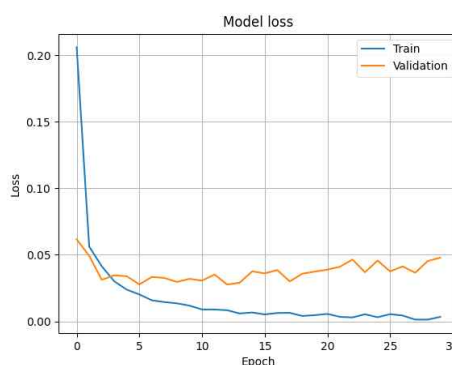
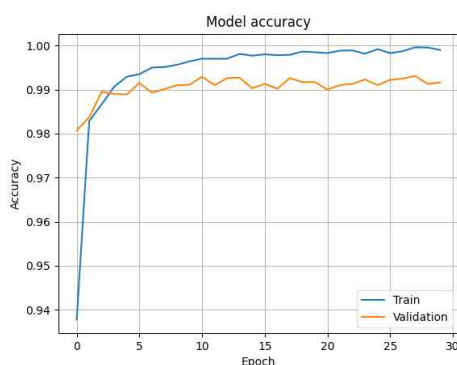
1. 프로그램 6-1과 6-2를 수행하여 결과를 정리하고, 프로그램의 동작을 설명하시오.

1.1 수행 결과 (6-1)

```
# LeNet-5 신경망 모델 설계
cnn=Sequential()
cnn.add(Conv2D(6,(5,5),padding='same',activation='relu',input_shape=(28,28,1)))
cnn.add(MaxPooling2D(pool_size=(2,2)))
cnn.add(Conv2D(16,(5,5),padding='same',activation='relu'))
cnn.add(MaxPooling2D(pool_size=(2,2)))
cnn.add(Conv2D(120,(5,5),padding='same',activation='relu'))
cnn.add(Flatten())
cnn.add(Dense(84,activation='relu'))
cnn.add(Dense(10,activation='softmax'))
```

#코드 해석

- 입력 이미지가 단채널 28X28이고, 5X5인 필터가 6개 있는 컨볼루션 레이어를 정의한다.
- 입력 영상 크기를 반으로 줄인 영상을 만들기 위해 MaxPooling 레이어를 사용한다.
- 5X5인 필터가 16개 있는 컨볼루션 레이어를 정의한다.
- 입력 영상 크기를 반으로 줄인 영상을 만들기 위해 MaxPooling 레이어를 사용한다.
- 5X5인 필터가 120개 있는 컨볼루션 레이어를 정의한다.
- 영상을 일차원으로 바꿔주기 위해 Flatten 레이어를 사용한다.
- 84개의 출력 뉴런을 갖고 활성화 함수로 'relu'를 사용한다.
- 10개의 출력 뉴런을 갖고 활성화 함수로 'softmax'를 사용한다.



Epoch 1/30 469/469 - 22s - loss: 0.2059 - accuracy: 0.9378 - val_loss: 0.0617 - val_accuracy: 0.9807 Epoch 2/30 469/469 - 28s - loss: 0.0562 - accuracy: 0.9828 - val_loss: 0.0493 - val_accuracy: 0.9837 Epoch 3/30 469/469 - 24s - loss: 0.0414 - accuracy: 0.9867 - val_loss: 0.0313 - val_accuracy: 0.9895 Epoch 4/30 469/469 - 25s - loss: 0.0301 - accuracy: 0.9907 - val_loss: 0.0346 - val_accuracy: 0.9890 Epoch 5/30 469/469 - 24s - loss: 0.0238 - accuracy: 0.9929 - val_loss: 0.0339 - val_accuracy: 0.9889 Epoch 6/30 469/469 - 24s - loss: 0.0202 - accuracy: 0.9935 - val_loss: 0.0278 - val_accuracy: 0.9915 Epoch 26/30 469/469 - 24s - loss: 0.0054 - accuracy: 0.9983 - val_loss: 0.0376 - val_accuracy: 0.9922 Epoch 27/30 469/469 - 23s - loss: 0.0044 - accuracy: 0.9987 - val_loss: 0.0413 - val_accuracy: 0.9925 Epoch 28/30 469/469 - 25s - loss: 0.0013 - accuracy: 0.9996 - val_loss: 0.0365 - val_accuracy: 0.9931 Epoch 29/30 469/469 - 23s - loss: 0.0013 - accuracy: 0.9995 - val_loss: 0.0453 - val_accuracy: 0.9913 Epoch 30/30 469/469 - 25s - loss: 0.0034 - accuracy: 0.9990 - val_loss: 0.0479 - val_accuracy: 0.9916 정확률은 99.1599977016449
---	---

- 데이터 학습 정확도는 99.16%이며, Validation 데이터와 Train 데이터가 5 epoch 큰 차이를 보여주지 않는다.
- Train 데이터의 손실율은 낮지만, epoch를 거듭할수록 Validation 데이터는 조금씩 증가한다.

1.2 동작 설명 (6-1)

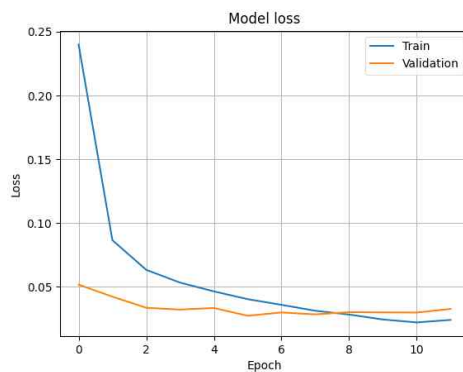
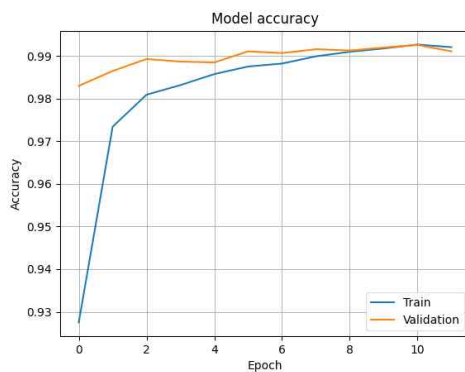
- 60,000 개의 Train 데이터와 10,000 개의 Validation 데이터 사용
- 전체 신경망 구조는 C-P-C-P-C-FC-FC 사용
- 신경망 구조는 LeNet-5와 비슷한 형태로 설계함.

2.1 수행 결과 (6-2)

```
# 신경망 모델 설계
cnn=Sequential()
cnn.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
cnn.add(Conv2D(64,(3,3),activation='relu'))
cnn.add(MaxPooling2D(pool_size=(2,2)))
cnn.add(Dropout(0.25))
cnn.add(Flatten())
cnn.add(Dense(128,activation='relu'))
cnn.add(Dropout(0.5))
cnn.add(Dense(10,activation='softmax'))
```

#코드 해석

- 입력 이미지가 단채널 28X28이고, 3X3인 필터가 32개 있는 컨볼루션 레이어를 정의한다.
- 3X3인 필터가 64개 있는 컨볼루션 레이어를 정의한다.
- 입력 영상 크기를 반으로 줄인 영상을 만들기 위해 MaxPooling 레이어를 사용한다.
- 25%의 weight의 Dropout 레이어를 추가한다.
- 영상을 일차원으로 바꾸주기 위해 Flatten 레이어를 사용한다.
- 128개의 출력 뉴런을 갖고 활성화 함수로 'relu'를 사용한다.
- 50%의 weight의 Dropout 레이어를 추가한다.
- 10개의 출력 뉴런을 갖고 활성화 함수로 'softmax'를 사용한다.



Epoch 1/12 469/469 - 53s - loss: 0.2400 - accuracy: 0.9275 - val_loss: 0.0518 - val_accuracy: 0.9830 Epoch 2/12 469/469 - 53s - loss: 0.0866 - accuracy: 0.9734 - val_loss: 0.0423 - val_accuracy: 0.9865 Epoch 3/12 469/469 - 54s - loss: 0.0634 - accuracy: 0.9809 - val_loss: 0.0336 - val_accuracy: 0.9893 Epoch 4/12 469/469 - 54s - loss: 0.0534 - accuracy: 0.9832 - val_loss: 0.0322 - val_accuracy: 0.9887 Epoch 5/12 469/469 - 56s - loss: 0.0465 - accuracy: 0.9858 - val_loss: 0.0335 - val_accuracy: 0.9885 Epoch 6/12 469/469 - 56s - loss: 0.0404 - accuracy: 0.9876 - val_loss: 0.0274 - val_accuracy: 0.9911	Epoch 7/12 469/469 - 56s - loss: 0.0359 - accuracy: 0.9882 - val_loss: 0.0300 - val_accuracy: 0.9907 Epoch 8/12 469/469 - 56s - loss: 0.0313 - accuracy: 0.9899 - val_loss: 0.0285 - val_accuracy: 0.9916 Epoch 9/12 469/469 - 55s - loss: 0.0283 - accuracy: 0.9910 - val_loss: 0.0302 - val_accuracy: 0.9913 Epoch 10/12 469/469 - 55s - loss: 0.0245 - accuracy: 0.9918 - val_loss: 0.0300 - val_accuracy: 0.9920 Epoch 11/12 469/469 - 55s - loss: 0.0221 - accuracy: 0.9927 - val_loss: 0.0299 - val_accuracy: 0.9926 Epoch 12/12 469/469 - 55s - loss: 0.0241 - accuracy: 0.9921 - val_loss: 0.0327 - val_accuracy: 0.9911 정확률은 99.1100013256073
- 30 epoch을 실행했던 6-1 프로그램보다 빠른 학습과 Train 데이터와 Validation 데이터와의 차이가 매우 적음을 확인할 수 있다. - Dropout 레이어를 통해 오버피팅을 줄일 수 있었다.	

2.2 동작 설명 (6-2)

- 60,000 개의 Train 데이터와 10,000 개의 Validation 데이터 사용
- 전체 신경망 구조는 C-C-P-dropout-FC-dropout-FC 사용
- dropout은 활성화 함수처럼 층의 출력 연산을 적용하는 역할에 불과하므로, 별개의 층으로 볼 수 없음.
- FC층을 제외하면 C-C-P 3개의 층만 있음.