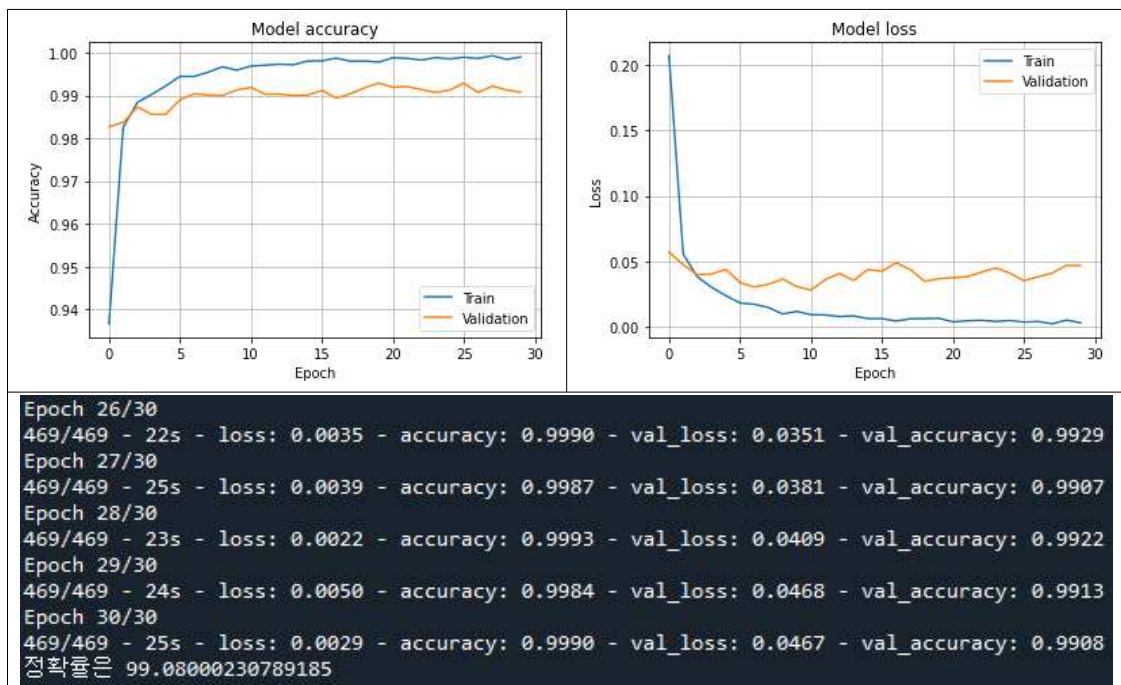


13주차 과제

2021254008 최준혁

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

1-1) 프로그램 6-1 결과 정리



30에포크 동안 정확도는 점점 상승하나, 일정 에포크 이후로 비슷한 정확도를 보이고 실행결과 99.08%의 정확률이 나왔습니다.

1-2) 프로그램 6-1 프로그램 동작 설명

```
# MNIST 데이터셋을 읽고 신경망에 입력할 형태로 변환
(x_train,y_train),(x_test,y_test)= mnist.load_data()
x_train=x_train.reshape(60000,28,28,1)
x_test=x_test.reshape(10000,28,28,1)
x_train=x_train.astype(np.float32)/255.0
x_test=x_test.astype(np.float32)/255.0
y_train=tf.keras.utils.to_categorical(y_train,10)
y_test=tf.keras.utils.to_categorical(y_test,10)
```

데이터 입력 부분으로 2차원 맵을 그대로 입력해야 하므로 reshape(60000, 28, 28, 1) 사용

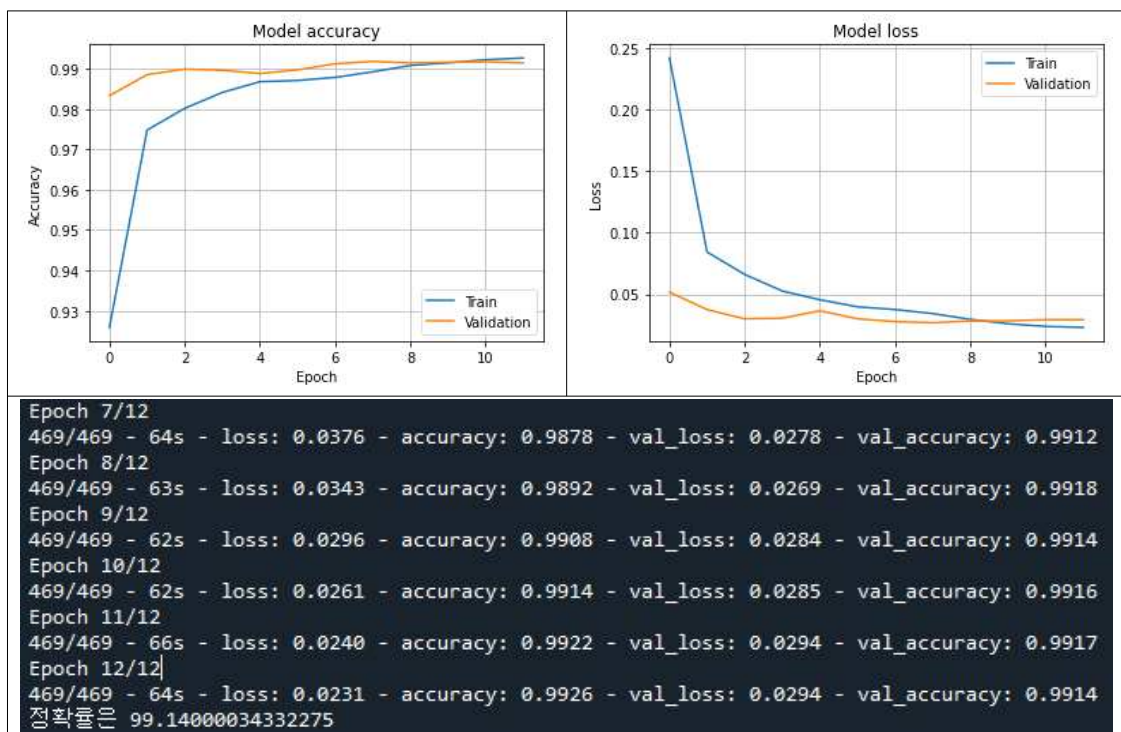
```
# LeNet-5 신경망 모델 설계
cnn=Sequential()
cnn.add(Conv2D(6,(5,5),padding='same',activation='relu',input_shape=(28,28,1)))
cnn.add(MaxPooling2D(pool_size=(2,2)))
cnn.add(Conv2D(16,(5,5),padding='same',activation='relu'))
cnn.add(MaxPooling2D(pool_size=(2,2)))
cnn.add(Conv2D(120,(5,5),padding='same',activation='relu'))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

신경망 설계부분으로 C-P-C-P-C-FC-FC 의 구조로 설계

```
# 신경망 모델 학습
cnn.compile(loss='categorical_crossentropy',optimizer=Adam(),metrics=['accuracy'])
hist=cnn.fit(x_train,y_train,batch_size=128,epochs=30,validation_data=(x_test,y_test),verbose=2)
```

batch 사이즈를 128, epochs을 30으로 하여 학습.

1-3) 프로그램 6-2 결과 정리



30에포크 동안 정확도는 점점 상승하고 실행결과 99.14%의 정확률이 나왔습니다. 6-2 프로그램은 12에포크만에 6-1 프로그램 보다 높은 정확도를 나타냈습니다.

1-4) 프로그램 6-2 프로그램 동작 설명

```
# MNIST 데이터셋을 읽고 신경망에 입력할 형태로 변환
(x_train,y_train),(x_test,y_test)=mnist.load_data()
x_train=x_train.reshape(60000,28,28,1)
x_test=x_test.reshape(10000,28,28,1)
x_train=x_train.astype(np.float32)/255.0
x_test=x_test.astype(np.float32)/255.0
y_train=tf.keras.utils.to_categorical(y_train,10)
y_test=tf.keras.utils.to_categorical(y_test,10)
```

데이터 입력 부분으로 2차원 맵을 그대로 입력해야 하므로 reshape(60000, 28, 28, 1) 사용

```
# 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'))
```

C-C-P-dropout-FC-dropout-FC로 구조로 설계

- 오버피팅을 막기 위해 학습 도중 일정 비율의 가중치를 무작위로 골라 불능으로 하는 dropout 적용함.

```
# 신경망 모델 학습
cnn.compile(loss='categorical_crossentropy',optimizer=Adam(),metrics=['accuracy'])
hist=cnn.fit(x_train,y_train,batch_size=128,epochs=12,validation_data=(x_test,y_test),verbose=2)
```

batch 사이즈를 128, epochs을 12으로 하여 학습.