

WEEK13 Report

컨볼루션 신경망 프로그래밍 - 필기 숫자 인식

1. LeNet-5로 MNIST 인식
2. 컨볼루션 신경망으로 MNIST 인식

2021254019 김지현

1. (6-1)LeNet-5로 MNIST 인식

```
# 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) #6만개를 학습에 사용.
이미지 사이즈 28x28, 8bit 영상
x_test=x_test.reshape(10000,28,28,1) #만개를 테스트로 사용.
x_train=x_train.astype(np.float32)/255.0
x_test=x_test.astype(np.float32)/255.0
#0~255의 값을 255로 나누어 0~1사이의 값으로 표현
y_train=tf.keras.utils.to_categorical(y_train,10)
y_test=tf.keras.utils.to_categorical(y_test,10)
```

```
# LeNet-5 신경망 모델 설계
# 구조 : Sequential 모델로 시작 -> Conv -> Pooling -> Conv -> Pooling -
> Conv -> Fc -> FC 구조.
# FC란? Fully Connected Layer. Conv->Pooling 프로세스의 결과를
가지고 이미지를 정의된 라벨로 분류하는 것. 이 레이어는 이미지를
분류/설명하는데 가장 적합하게 예측한다.
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 모델 학습
cnn.compile(loss= ' categorical_crossentropy ' ,optimizer=Adam(),met
rics=[ ' accuracy ' ])
hist=cnn.fit(x_train,y_train,batch_size=128,epochs=30,validation_data=
(x_test,y_test),verbose=2)
```

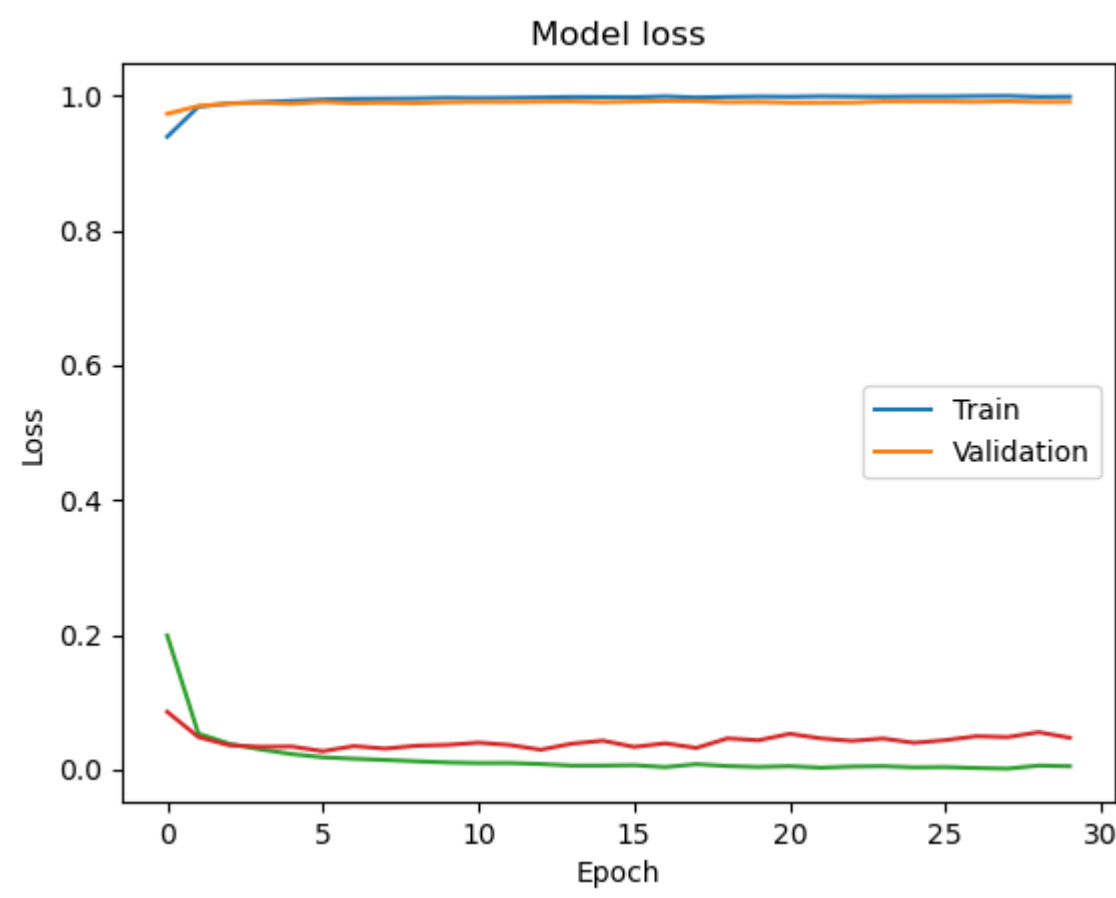
```
# 신경망 모델 정확률 평가
res=cnn.evaluate(x_test,y_test,verbose=0)
print( " 정확률은 " ,res[1]*100)
```

```
import matplotlib.pyplot as plt #그래프 그리기
```

1. (6-1)LeNet-5로 MNIST 인식

정확률은 99.08999800682068

결과 그래프 ↓



2. (6-2) 컨벌루션 신경망으로 MNIST 인식

1번 코드와 다른 부분만 설명.

신경망 모델 설계
구조 : Sequential 모델로 시작 -> Conv -> Conv -> Pooling -> dropout
-> FC -> dropout -> FC
dropout : 활성 함수 처럼 층의 출력에 연산을 적용하는 역할
따라서, dropout과 FC 층을 제외하는 Conv -> Conv -> Pooling
층으로 구성된다.

```
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'))
```

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) # 1번 코드와 다르게 Epoch을 12로 설정

정확률은 99.19999837875366
(1번 코드의 정확도와 0.11001 정도 차이가 보인다.)

