

딥러링 실제 : 13주차 실습과제

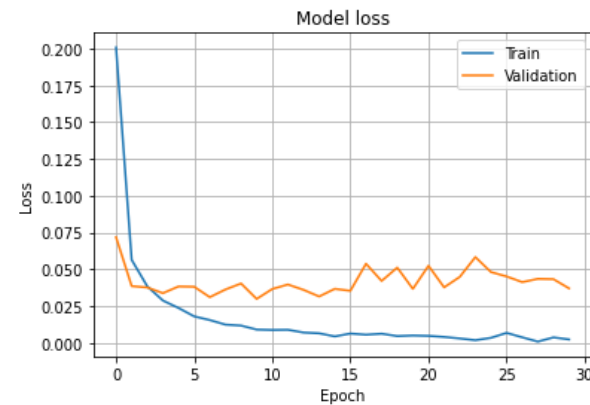
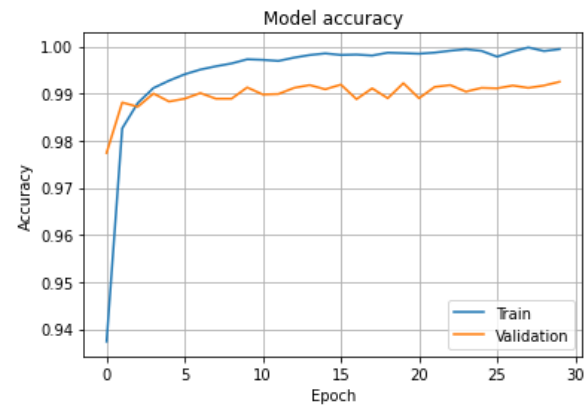
산업인공지능학과

2020254018

강윤구

1-1) 프로그램 6-1을 수행하여 결과를 정리

Epoch 28/30
469/469 - 2s - loss: 9.0555e-04 - accuracy: 0.9998 - val_loss: 0.0435 - val_accuracy: 0.9912
Epoch 29/30
469/469 - 2s - loss: 0.0037 - accuracy: 0.9990 - val_loss: 0.0433 - val_accuracy: 0.9917
Epoch 30/30
469/469 - 2s - loss: 0.0024 - accuracy: 0.9994 - val_loss: 0.0370 - val_accuracy: 0.9925
정확률은 99.25000071525574

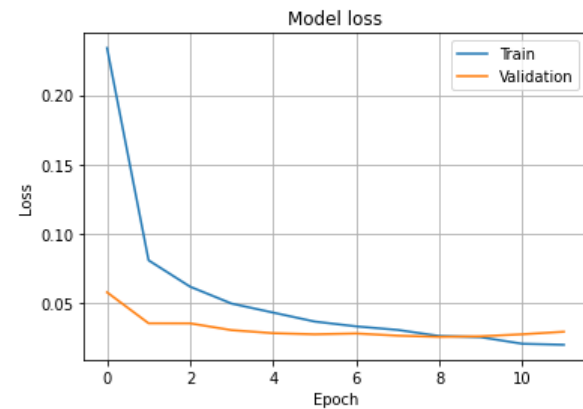
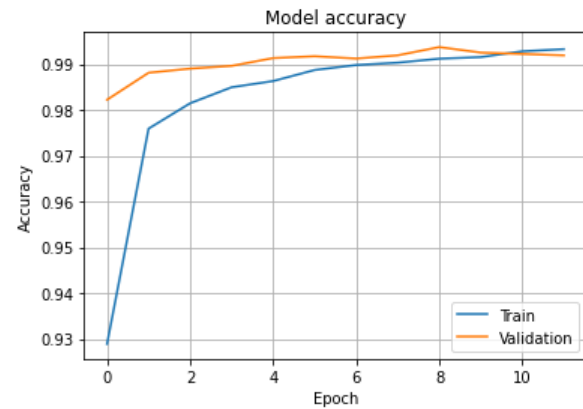


- 총30회 진행
- 정확률은 99.25%

<https://github.com/Yunkoo-GIT/Programming/blob/main/20210603.ipynb>

1-2) 프로그램 6-2를 수행하여 결과를 정리

Epoch 10/12
469/469 - 3s - loss: 0.0255 - accuracy: 0.9915 - val_loss: 0.0261 - val_accuracy: 0.9925
Epoch 11/12
469/469 - 3s - loss: 0.0208 - accuracy: 0.9928 - val_loss: 0.0276 - val_accuracy: 0.9922
Epoch 12/12
469/469 - 3s - loss: 0.0200 - accuracy: 0.9933 - val_loss: 0.0294 - val_accuracy: 0.9919
정확률은 99.19000267982483



- 총12회 진행
- 정확률은 99.19%

<https://github.com/Yunkoo-GIT/Programming/blob/main/20210603.ipynb>

2-1) 프로그램 6-1 의 동작을 설명

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.optimizers import Adam
```

```
# 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)
```

```
# 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 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)
```

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

```
import matplotlib.pyplot as plt
```

```
# 정확률 그래프
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='best')
plt.grid()
plt.show()
```

```
# 손실 함수 그래프
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='best')
plt.grid()
plt.show()
```

MNIST 데이터셋을 읽어 옴
학습데이터 60000개와 테스트 데이터 10000로 나눔

LeNet-5 신경망 모델 설계

신경망 모델을 학습

신경망 모델 정확률 평가

정확률, 손실함수 그래프 출력

2-1) 프로그램 6-2 의 동작을 설명

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
```

```
# 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)
```

```
# 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'))
```

```
# 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)
```

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

```
import matplotlib.pyplot as plt
```

```
# 정확률 그래프
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='best')
plt.grid()
plt.show()
```

```
# 손실 함수 그래프
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='best')
plt.grid()
plt.show()
```

MNIST 데이터셋을 읽어 옴
학습데이터 60000개와 테스트 데이터 10000로 나눔

컨볼루션 신경망 모델 설계

신경망 모델을 학습

신경망 모델 정확률 평가

정확률, 손실함수 그래프 출력