1-4. 딥러닝 네트워크 학습시키기

우리가 만든 네트워크의 입력은 `(데이터갯수, 이미지 크기 x, 이미지 크기 y, 채널수)` 와 같은 형태를 가집니다. 이전 스텝에서 첫번째 레이어에 `input_shape=(28,28,1)` 로 지정했던 것을 기억하시나요?
그런데 `print(x_train.shape)` 을 해보면, `(60000, 28, 28)` 로 채널수에 대한 정보가 없습니다. 따라서 `(60000, 28, 28, 1)` 로 만들어 주어야 합니다 (여기서 채널수 1은 흑백 이미지를 의미합니다. 컬러 이미지라면 R, G, B 세 가지 값이 있기 때문에 3이겠죠?).

[Input]

```
print("Before Reshape - x_train_norm shape: {}".format(x_train_norm.shape))
print("Before Reshape - x_test_norm shape: {}".format(x_test_norm.shape))

x_train_reshaped=x_train_norm.reshape( -1, 28, 28, 1)  # 데이터갯수에 -1을 쓰면 reshape시 자동계산됩니다.
x_test_reshaped=x_test_norm.reshape( -1, 28, 28, 1)

print("After Reshape - x_train_reshaped shape: {}".format(x_train_reshaped.shape))
print("After Reshape - x_test_reshaped shape: {}".format(x_test_reshaped.shape))
```

실행

[Output]

그러면 이제 `x_train` 학습 데이터로 딥러닝 네트워크를 학습시켜 봅시다. 여기서 epochs=10 은 전체 60,000개의 데이터를 10번 반복 사용해서 학습을 시키라는 뜻입니다. 물론 model의 입력 정의에 형태를 맞춘 `x_train_reshaped` 가 사용되어야겠죠. 자 그러면 코드를 실행해 봅시다.

[Input]

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
model.fit(x_train_reshaped, y_train, epochs=10)
```

실행

[Output]

각 학습이 진행됨에 따라 epoch 별로 어느 정도 인식 정확도(**accuracy**)가 올라가는지 확인할 수 있습니다. 인식 정확도가 0.9413에서 0.9957까지 매우 높게 올라가는군요. 9 epoch정도부터는 인식률의 상승이 미미합니다. 10 epoch정도 학습을 시키면 충분할 것 같네요.