

# CNN(Convolution Neural Network)

## 학습 내용

- 최대 풀링을 사용하는 이유

In [5]:

```
# 이미지 처리 분야에서 가장 유명한 신경망 모델인 CNN 을 이용
from keras import layers
from keras import models

from keras.datasets import mnist
from keras.utils import to_categorical

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

## 왜 최대 풀링 연산(Max Pooling)을 이용하여 특성 맵을 다운샘플링할까?

### 최대 풀링층을 빼고 만든 모델

In [6]:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)))
model.add(layers.Conv2D(64, (3,3), activation='relu'))
model.add(layers.Conv2D(64, (3,3), activation='relu'))
```

In [7]:

```
model.summary()
```

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 26, 26, 32)	320
conv2d_7 (Conv2D)	(None, 24, 24, 64)	18496
conv2d_8 (Conv2D)	(None, 22, 22, 64)	36928
Total params: 55,744		
Trainable params: 55,744		
Non-trainable params: 0		

## 생각해 보기

**01 최종 특성 맵은  $64 \times 3 \times 3 \times 64(ch) + 64 = 36928$ 개의 가중치를 가진다 아주 많**

다.

- 작은 모델치고는 너무 많은 가중치이고, 심각한 과대적합이 발생.
- 처리할 특성맵의 가중치 개수를 줄이기 위해

## 02 특성의 공간적 계층 구조를 학습하는데 도움이 되지 않는다.

- 층이 깊어지면 질수록 뒤쪽의 특징맵의 값은 초기 입력의 정보가 아주 적어져 숫자 분류를 학습하기에 충분하지 않을 것이다.
- 세번째 층은  $3 \times 3$ 은 원도의 초기 입력의  $7 \times 7$  원도의 영역의 정보만 담고 있다.

In [ ]:

