

## [CNN(Convolution Neural Network)]

### ※ CNN에서 사용되는 레이어들

1. Convolution Layer(모듈명: **Conv2D**): 특정한 패턴의 특징이 어디서 나타나는지를 확인 하는 도구를 컨볼루션(convolution)이라 하며, 필터로 특징을 뽑아준다.
2. Maxpool Layer(모듈명: **MaxPool2D**): 사소한 변화는 무시하도록 설정한다.
3. Flatten Layer(모듈명: **Flatten**): 영상을 1차원으로 변경한다.

### ※ CNN을 이용한 mnist(손글씨데이터) 학습과정 코딩의 일부

(1) 필요한 패키지 import	<pre>import tensorflow as tf from tensorflow.keras.datasets import mnist # mnist 데이터 셋 import tensorflow.keras.utils as utils # 원 핫 인코딩 from tensorflow.keras.models import Sequential # 모델 생성 from tensorflow.keras.layers import Dense, Activation from tensorflow.keras.layers import <b>Conv2D</b>, <b>MaxPool2D</b>, <b>Flatten</b> import numpy as np</pre>
(2) 데이터 셋 준비 & 훈련 셋, 검증 셋 분리 & 원 핫 인코딩(라벨링 전환)	<pre>(x_train, y_train), (x_test, y_test) = mnist.load_data( ) # 데이터 셋 준비 x_val = x_train[50000: ] # 데이터 총 6만행 중 처음부터 50000행까지 검증 셋 설정 y_val = y_train[50000: ] x_train = x_train[: 50000] # 데이터 총 6만행 중 50001행부터 끝까지 훈련 셋 설정 y_train = y_train[: 50000]  y_train = <b>utils.to_categorical</b>(y_train) # 원 핫 인코딩 y_val = <b>utils.to_categorical</b>(y_val) y_test = <b>utils.to_categorical</b>(y_test)</pre>
(3) 모델 구성하기	<pre>model = Sequential( ) model.add(<b>Conv2D</b>(32, (3,3), activation='relu', <b>input_shape</b>=(8, 8, 1) # 채널 수: 32, 크기: 3*3, input_shape(너비, 높이, 채널 수) model.add(<b>MaxPool2D</b>(pool_size=(2,2))) # pool_size=( ): 수직, 수평의 축소 비율 지정 model.add(<b>Conv2D</b>(32, (3,3), activation='relu') model.add(<b>MaxPool2D</b>(pool_size=(2,2))) model.add(<b>Flatten</b>()) # 크기가 3*3 이었던 영상을 1차원으로 변경한다. model.add(Dense(256, input_dim=w*h, activation='relu') model.add(Dense(10, activation='softmax')</pre>