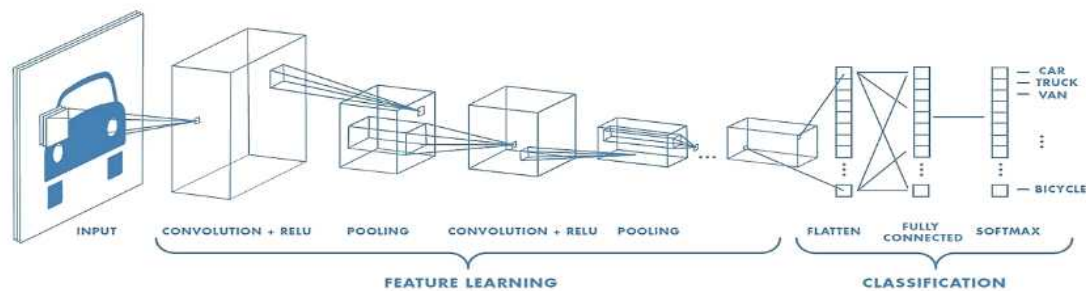


# CNN

## 1. CNN구조

- : Fully connected layer 만으로 구성된 인공 신경망의 입력 데이터는 1차원 형태로 한정.
- : 데이터의 공간 정보를 유지한 상태로 학습이 가능한 모델이 바로 Convolutional Neural Network



- : 이미지의 특징을 추출하는 부분과, 클래스를 분류하는 부분으로 나눌 수 있음
- > 특징 추출 영역은 Convolutional layer(필터+활성화함수) + Pooling Layer(선택적) 여러 겹
- > 분류 영역은 Fully Connected Layer
- > 특징과 분류 영역 사이에 Flatten 레이어 위치

## 2. Convolution Layer (= Locally Connected)

- : 높이x너비x채널 이 4x4x1 + 필터 3x3 + stride 1 => 2x2x1 텐서 형태의 이미지 생성
- : 추가로 bias 라는 스칼라값을 동일하게 더하도록 구현하기도 함

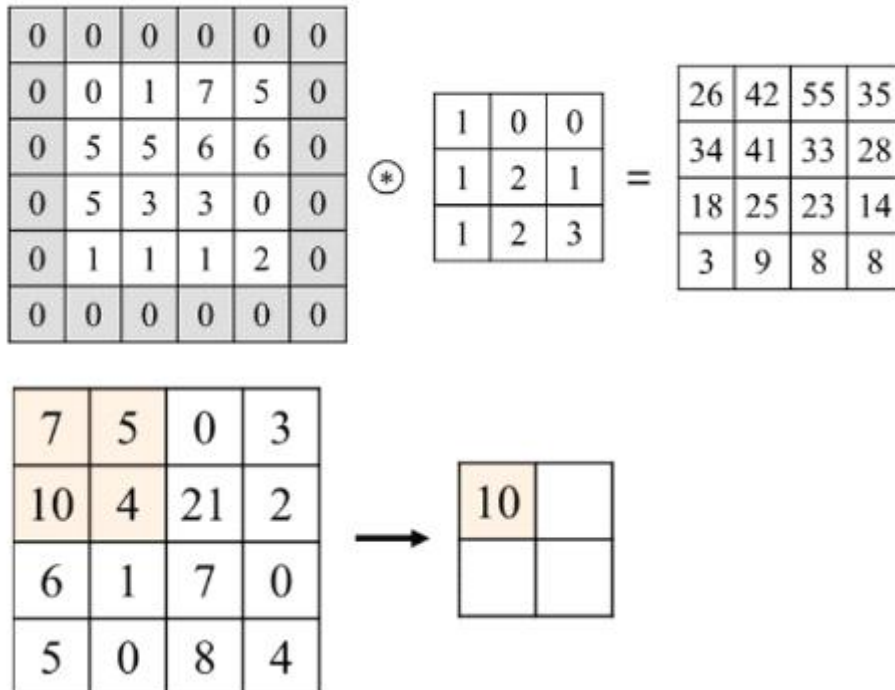
$$\begin{bmatrix} 0 & 1 & 7 & 5 \\ 5 & 5 & 6 & 6 \\ 5 & 3 & 3 & 0 \\ 1 & 1 & 1 & 2 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 0 \\ 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 40 & \\ & \end{bmatrix}$$

- > stride : 필터의 이동량을 의미, 주로 1로 설정하긴 함
- > padding : 입력 이미지와 출력 이미지의 크기를 같거나 비슷하게 만드는 역할을 함
- ex) 이미지의 가장자리에 0의 값을 갖는 픽셀을 추가하는 것을 zero-padding 이라고 함

### 3. Pooling Layer

: 적당히 크기도 줄이고, 특정 feature를 강조하는 역할

: 처리 방법은 Max, Average, Min pooling이 있다. 주로 Max 사용



### 4. Fully Connected Layer

: Flatten Layer -> 데이터 타입을 Fully Connected 네트워크 형태로 변형, shape 변경만 수행

-> 즉, pooling output을 벡터화

ex) 4x4x20 -> 세로줄을 일렬로 세워 320-dimension인 벡터로 변환

: The first fully Connected layer -> 입력을 취해 정확한 분류를 위해 가중치를 적용

: Fully connected output layer -> 각 label에 대한 최종 확률을 제공

즉,

- (1) n차원 데이터를 1차원 배열로 평탄화
- (2) 활성화 함수(Relu, Tanh 등)로 뉴런을 활성화
- (3) 분류기(softmax) 함수로 분류

※ 입력 채널이 1개라도, 출력 채널을 여러 개 지정했을 경우 이점

#### 1. 더 다양한 특징 추출

: 여러 개의 필터를 통해 각 필터가 입력 데이터의 다른 특징 학습 가능

: 입력 데이터의 복합적이고 추상적인 특징을 더 잘 표현할 수 있음

#### 2. 표현력 향상

: 네트워크가 입력 데이터에서 더 많은 정보를 추출하고 유지 가능  
; 더 많은 출력 채널을 통해 네트워크는 더 추상적이고 고차원적 특징 학습 가능

### 3. 성능 향상

: 오버피팅을 줄이고 더 나은 성능 발휘에 도움

### 4. 하위 모듈 재사용

: 다른 네트워크 모듈에 재사용 가능성을 높임