

문)

1. CNN의 동작 원리를 설명하시오.

2. 이때 사용되는 활성화 함수는 무엇이 있는지 나열 및 간단하게 설명하시오.

답)

1. CNN의 동작 원리를 설명하시오.

가. Convolution 의 동작 원리

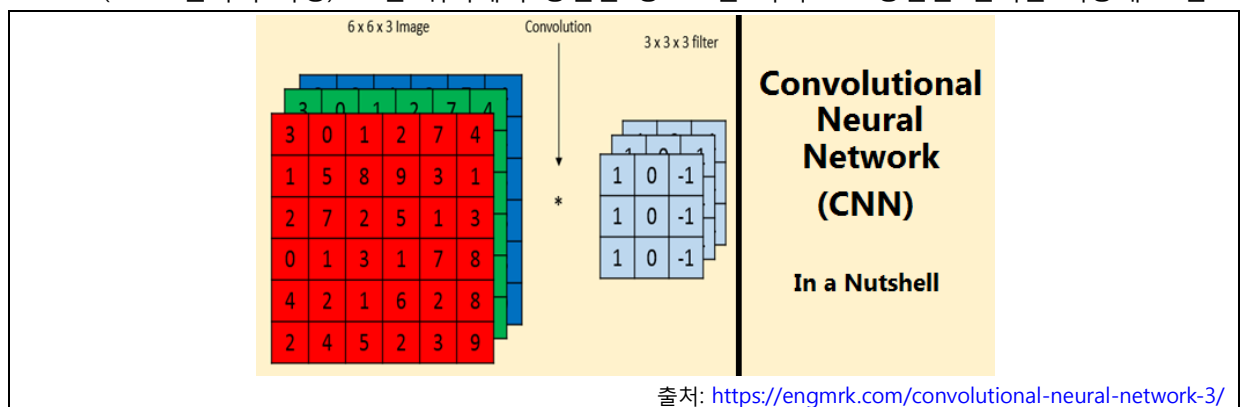
- Convolution 단순한 연산(내적: 덧셈 곱셈만으로 가능)을 이용한 Filtering으로 Filter와 같은 방향성(Edge)을 가진 정보에 더 큰 값을 할당
- 여러 종류의 필터를 이용해 방향성 정보를 도출하고 결과값을 만들어냄

분류	의미	설명	
Mathematically	Inner product	$(a,b) \cdot (c,d) = ac + bd$ $= a b \cos \theta$	
Physically	Filtering	<ul style="list-style-type: none"> <li>- <math>\cos \theta</math>는 <math>\theta=0</math> 일 때 최대</li> <li>= 같은 모양의 필터를 가질 때 값이 최대</li> </ul>	

- 전통적 Neural network는 너무 많은 가중치를 필요로 함으로 이를 공유(동일한 필터)해 보자는 개념 등장 => CNN(Convolution Neural Network)

나. CNN의 동작 원리 설명

- (CNN): 입력된 이미지에서 다시 한번 특성을 추출하기 위해 마스크(필터, 윈도우, 커널 등)를 도입하는 기법
- (CNN 원리의 가정): 모든 위치에서 동일한 중요도를 가지므로 동일한 필터를 적용해도 됨

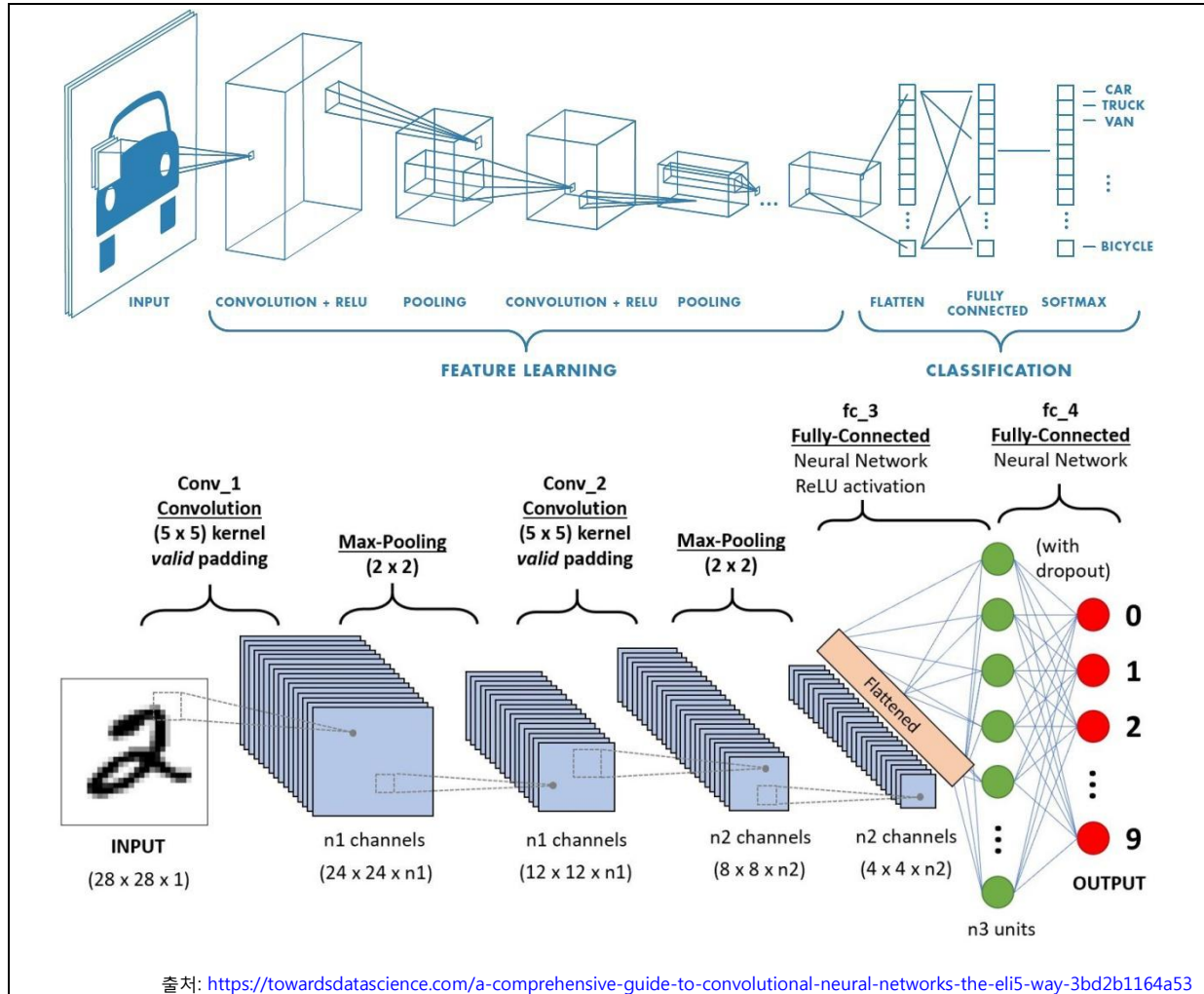


- 3X3 필터를(가중치 공유) 3개 사용하여 각 입력값의 방향성(Edge)을 학습
- 3X3X3 Filter 의 예

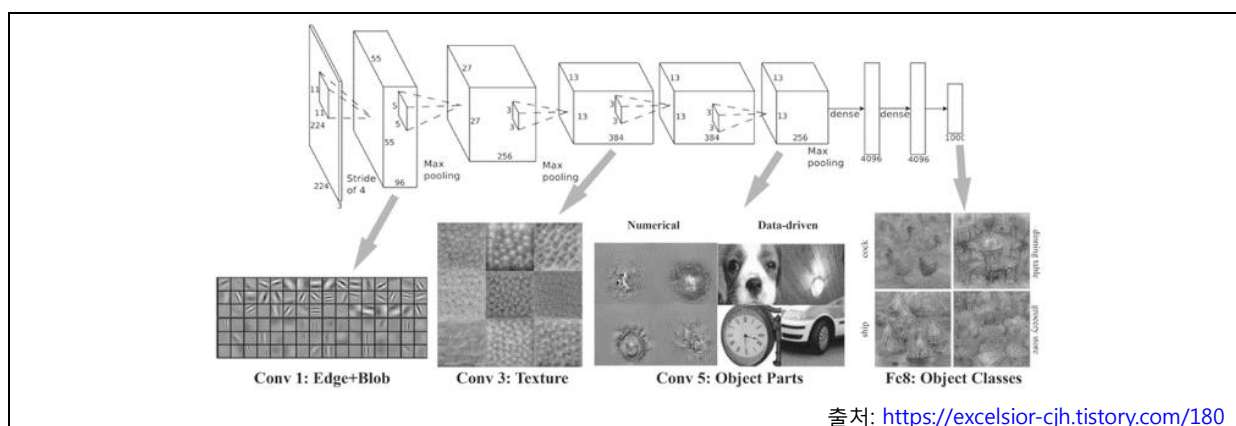
1	0	-1	0	-1	1	-1	1	0
1	0	-1	0	-1	1	-1	1	0
1	0	-1	0	-1	1	-1	1	0

다. CNN의 전체 구조의 동작

- 완전연결 계층은 데이터의 형상이 무시되는 문제가 존재, 합성곱 연산으로 유지
- CNN은 Conv연산을 반복하며 non-linearity를 증가시키고 활성화 함수를 이용하여 계층을 추상화하는 작업을 반복함



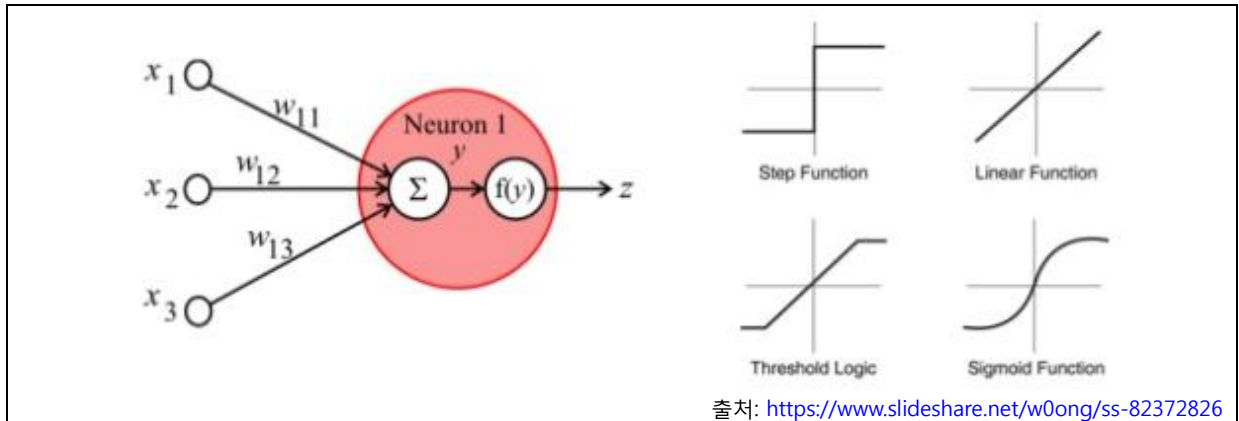
- 대개 Max-pooling을 포함한 Conv 연산과 추가적인 Conv 후 Fully-Connected 연산으로 결과를 도출함



- Edge + Blob -> Texture 에서 점점 더 복잡한 사물에 반응하며 점차 고급 정보로 변환

## 2. 이때 사용되는 활성화 함수는 무엇이 있는지 나열 및 간단하게 설명하시오.

### 가. 활성화 함수의 필요성



- 선형 함수의 경우 아무리 다층으로 구조를 만들어도 결과는 Linear
- 활성화 함수를 사용함으로써 non-linearity 증가

### 나. 활성화 함수의 유형과 설명

유형	설명	
ReLU	<ul style="list-style-type: none"> <li>- <math>ReLU(x) = \max(x, 0)</math></li> <li>- 최근 가장 많이 사용되는 활성화 함수</li> <li>- Conv-ReLU-(Pooling) 조합으로 많이 사용됨</li> </ul>	<p>The graph shows the ReLU function, which is 0 for negative inputs and increases linearly for positive inputs. The legend indicates three equivalent definitions: <math>\log(\exp(x) + 1)</math>, <math>\max(0, x)</math>, and <math>1/(1 + \exp(-x))</math>.</p>
Sigmoid	<ul style="list-style-type: none"> <li>- 첫 CNN 인 LeNet 에서 사용되었던 활성화 함수</li> </ul> <p>LeNet →</p>	<p>The diagram shows the architecture of LeNet-5, starting from an input image of a handwritten 'A'. It passes through two convolutional layers (C1 and C2) with subsampling, followed by two more convolutional layers (C3 and C4) with subsampling, and finally a fully connected layer (F5) leading to the output.</p>
Softmax	<ul style="list-style-type: none"> <li>- 마지막 출력층에서는 Affine-Softmax 조합을 그대로 사용하는 경우가 많음</li> </ul>	<p>The graph shows the Softmax function, which maps a vector of real numbers to a probability distribution where each element is between 0 and 1 and the sum of all elements is 1.</p>

- LeNet 에선 시그모이드 함수를 사용하는데 반해 현재는 주로 ReLU가 많이 사용 됨
- Affine-ReLU 에서 Conv-ReLU-(Pooling) 으로 바뀌는 추세

이미지 출처: <https://www.slideshare.net/ByoungHeeKim1/recurrent-neural-networks-73629152>  
<http://blog.naver.com/PostView.nhn?blogId=laonple&logNo=221218707503&categoryNo=0&parentCategoryNo=0&viewDate=&currentPage=1&postListTopCurrentPage=1&from=postView>  
<http://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/>