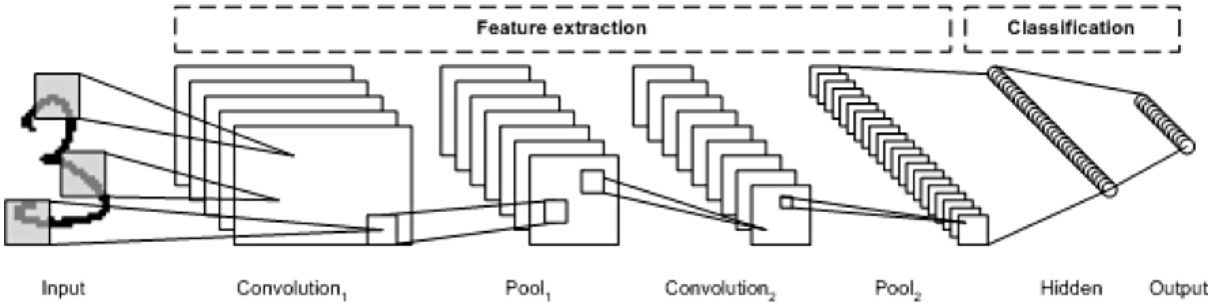
CNN



Fully Connected Layer로 구성된 인공 신경망의 입력데이터는 1차원 배열 형태로 한정됨.

But 한 장의 컬러 사진은 3차원 데이터

배치 모드에 사용되는 여러장의 사진은 4차원 데이터

따라서, 사진 데이터로 Fully Connected 신경망을 학습시켜야 할 때, 3차원 사진 데이터를 1차원으로 평면화시켜야 함 -> 이 과정에서 공간 정보의 손실이 있음

이미지의 공간 정보를 유지한 상태로 학습이 가능한 모델이 CNN

CNN이 Fully Connected Nueral Network와 비교하여 갖는 차별성

1. 각 레이어의 입출력 데이터의 형상 유지

2. 이미지의 공간 정보를 유지하면서 인접 이미지와의 특징을 효과적으로 인식

3. 복수의 필터로 이미지의 특징을 추출하고 학습

4. 추출한 이미지의 특징을 모으고 강화하는 Pooling 레이어

5. 필터를 공유 파라미터로 사용하기 때문에, 일반 인공 신경망과 비교하여 학습 파라미터가 매우 적음

1) 이미지의 특징을 추출하는 부분 => Feature Extraction

: Convoluation Layer와 Pooling Layer를 여러 겹 싸는 형태

- Convolution Layer : 입력데이터에 필터를 적용 후 활성화 함수를 반영하는 필수 요소로, Filter를 사용하여 공유 파라미터 수를 최소화하면서 이미지의 특성을 찾음

- Pooling Layer : 선택적 레이어로 특징을 강화하고 모은다

2) 클래스를 분류하는 부분 => Classification

: CNN 마지막에 이미지 분류를 위한 Fully Connected 레이어 추가됨

- 이미지의 특징을 추출하는 부분과 이미지를 분류하는 부분 사이에 이미지 형태의 데이터를 배열 형태로 만드는 Flatten 레이어 존재

Convolution Layer

- Filter 크기, Stride, Padding 적용 여부, Max Pooling 크기에 따라 출력 데이터의 Shape이 변경됨

용어 정리

- Convolution 합성곱

http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

- Channel 채널

  컬러 사진은 천역색을 표현하기 위해 각 픽셀을 RGB 3개의 실수로 표현한 3차원 데이터.

  흑백 사진은 2차원 데이터로 1개의 채널로 구성.

  ex) 높이가 39픽셀, 폭이 31픽셀인 컬러 사진의 데이터 shape = (39, 31, 3)

  ex) 높이가 39픽셀, 폭이 31픽셀인 흑백 사진의 데이터 shape = (39, 31, 1)

  => Convolution Layer에 유입되는 입력 데이터에는 한 개 이상의 필터가 적용된다.

  => 1개 필터는 Feature Map의 채널이 된다. n개의 필터가 적용된다면, 출력데이터는 n개의 채널은 갖게 된다.

- Filter 필터 : 이미지의 특징을 찾아내기 위한 공용 파라미터로 정사각 행렬이다

  CNN에서 학습의 대상이 필터 파라미터

  입력 데이터를 지정된 간격으로 순회하며 채널별로 합성곱을 하고 모든 채널의 합성곱의 합을 Feature Map으로 만든다.

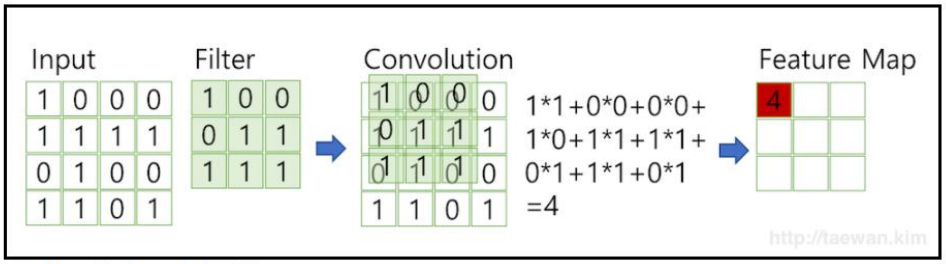


그림 3: 합성곱 계산 절차

=> Feature Map의 크기는 Input보다 가로/세로가 1씩 작아짐

- Kernel 커널 = Filter

- Stride 스트라이드 : 지정된 간격으로 필터를 순회하는 간격

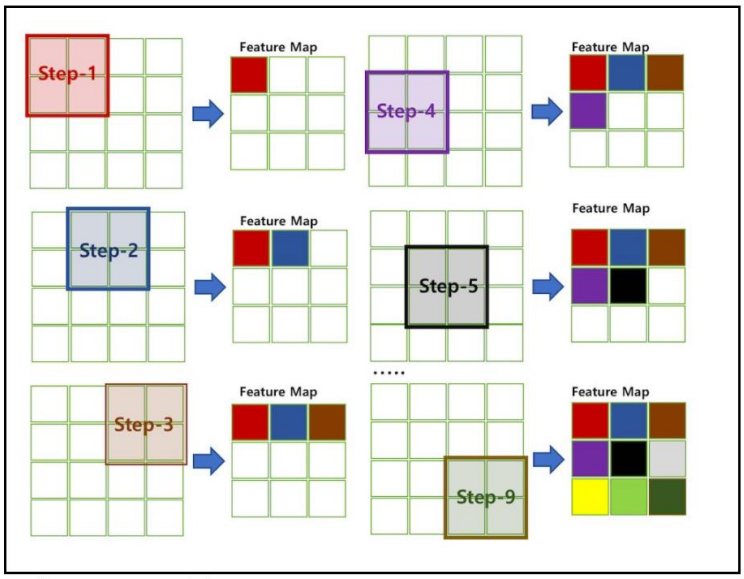


그림 4: Feature Map 과정

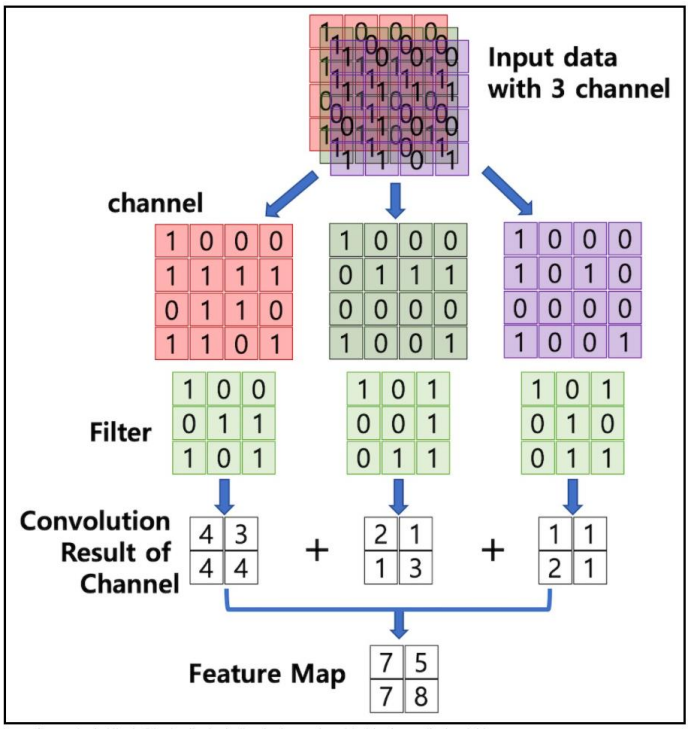


그림5: 멀티 채널 입력 데이터에 필터를 적용한 합성곱 계산 절차

=> 입력데이터가 여러 채널을 갖을 경우 필터는 각 채널을 순회하며 합성곱을 계산한 후, 채널별 Feature Map을 만든다. 그 후, 각 채널의 Feature Map을 합산하여 최종 Feature Map으로 반환한다. 입력 데이터는 채널 수와 상관없이 필터별로 하나의 Feature Map이 만들어 진다.

=> 하나의 Convolution Layer에 크기가 같은 여러 개의 필터 적용 가능. 이 경우 Feature Map에는 필터 개수 만큼의 채널이 만들어짐. 즉, 입력데이터에 적용한 필터의 개수는 출력 데이터인 Feature Map의 채널이 됨.

- Padding 패딩 : Convolution Layer의 출력 데이터(Feature Map)의 데이터가 줄어드는 것을 방지하기 위한 방법

=> 입력 데이터의 외각에 지정된 픽셀만큼 특정 값으로 채워 넣는 것으로 보통 0으로 채워 넣는다,

=> 출력 데이터의 사이즈 조절 기능 외 인공 신경망이 이미지의 외각을 인식하는 학습 효과도 있다.

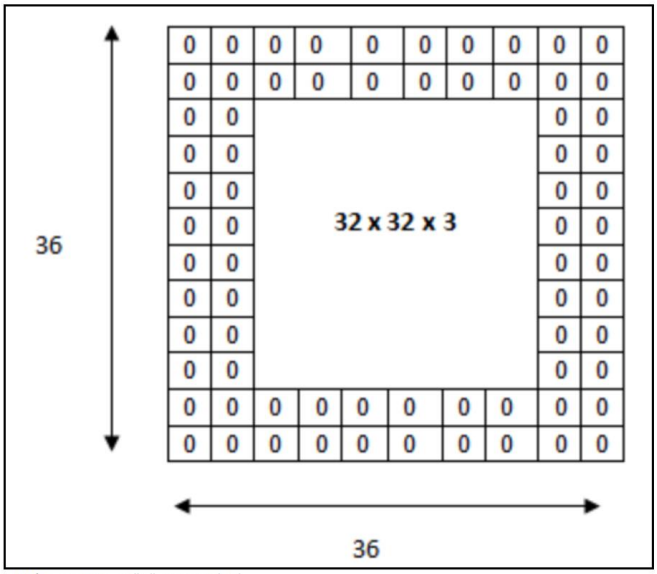


그림 6: padding 예제: 2pixel 추가

- Feature Map 피처 맵 : Convolution Layer의 입력 데이터를 필터가 순회하며 합성곱을 통해 만든 출력으로 합성곱 계산으로 만들어진 행렬

- Activation Map 액티베이션 맵 : Feature Map 행렬에 활성 함수를 적용한 결과로 Convolution Layer의 최종 출력 결과

- Pooling Layer 풀링레이어 : Convolution Layer의 출력 데이터를 입력으로 받아 출력 데이터(Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용

방법1) Max Pooling : CNN에서 주로 사용

방법2) Average Pooling

방법3) Min Pooling

=> Pooling의 크기와 Stride를 같은 크기로 설정하여 모든 원소가 한 번씩 처리되도록 설정

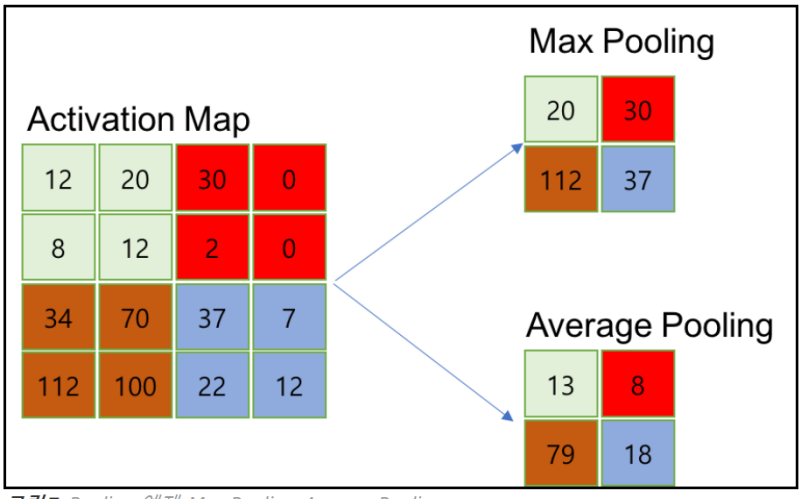


그림 7: Pooling 예제: Max Pooling, Average Pooling

- 학습대상 파라미터가 없음

- Pooling Layer를 통과하면 행렬의 크기 감소

- Pooling Layer를 통해서 채널 수 변경 없음

Convolution Layer 출력 데이터 크기 산정

- H : 입력 데이터 높이

- W : 입력 데이터 폭

- FH : 필터 높이

- FW : 필터 폭

- S : Stride 크기

- P : 패딩 사이즈

▪ *식 1. 출력 데이터 크기 계산*

$$OutputHeight = OH = \frac{(H + 2P - FH)}{S} + 1$$
$$OutputWeight = OW = \frac{(W + 2P - FW)}{S} + 1$$

=> 식의 결과값이 자연수를 만족하도록 Filter, Stride, Pooling의 크기 및 패딩 크기 조절

=> Pooling Layer가 온다면, Feature Map의 행/열 크기는 Pooling 크기의 배수여야 한다.

Pooling Layer 출력 데이터 크기 산정

: Stride와 같은 크기로 만들어 모든 요소가 한번씩 Pooling 되도록 한다.

=> 입력 데이터의 행/열 크기는 Pooling 사이즈의 배수여야 한다.

=> Pooling Layer의 출력 데이터 크기 = 행/열 크기 / Pooling 사이즈

ex) Pooling의 크기가 (2, 2) 라면 출력 데이터 크기는 입력 데이터의 행/열 크기를 2로 나눈 몫

- *식 2. 출력 데이터 크기 계산*

$$OutputRowSize = \frac{InputRowSize}{PoolingSize}$$
$$OutputColumnSize = \frac{InputColumnSize}{PoolingSize}$$

CNN의 구조

: Convolution Layer와 Max Pooling Layer를 반복적으로 stack을 쌓는 Feature Extraction + 마지막 출력층의 Softmax

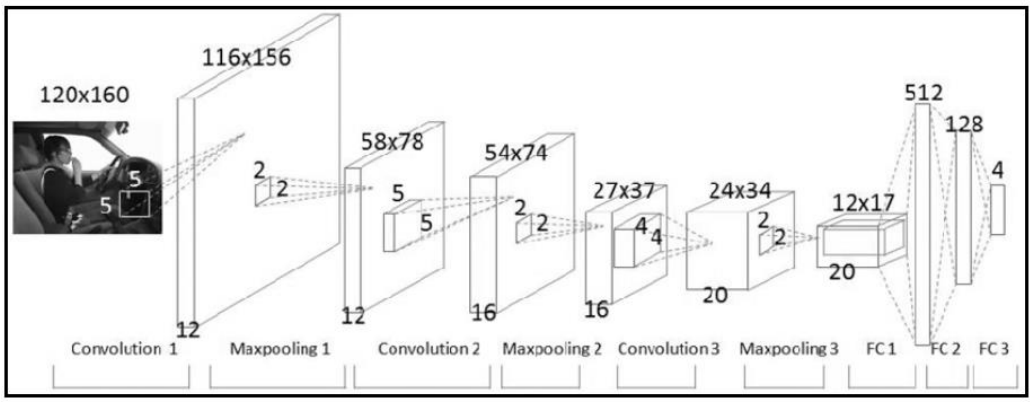


그림 8: 전형적인 CNN, 출처: https://www.researchgate.net/figure/Architecture-of-our-unsupervised-CNN-Network-contains-three-stages-each-of-which_283433254

=> Feature Extraction 부분은 Filter, Stride, Padding을 조절하여 입/출력 크기를 맞추는 작업이 중요하다.

<예제>

입력데이터 Shape : (39, 31, 1)

분류 클래스 : 100

layer	Input Channel	Filter	Output Channel	Stride	Max Pooling	activation function
Convolution Layer 1	1	(4, 4)	20	1	X	relu
Max Pooling Lyaer 1	20	X	20	2	(2, 2)	X
Convolution Layer 2	20	(3, 3)	40	1	X	relu
Max Pooling Lyaer 2	40	X	40	2	(2, 2)	X
Convolution Layer 3	40	(2, 2)	60	1	1	relu
Max Pooling Lyaer 3	60	X	60	2	(2, 2)	X
Convolution Layer 4	60	(2, 2)	80	1	1	relu
Flatten	X	X	X	X	X	X
fully connected Layer	X	X	X	X	X	softmax

ex) Convolution Layer1

입력 이미지에 Shape이 (4, 4)인 필터 20개를 적용 -> 학습시킬 대상임

- *식 3. Convolution Layer 1의 Activation Map 크기 계산*

$$RowSize = \frac{N - F}{Strid} + 1 = \frac{39 - 4}{1} + 1 = 36$$
$$ColumnSize = \frac{N - F}{Strid} + 1 = \frac{31 - 4}{1} + 1 = 28$$

- 입력 데이터 Shape : (39, 31, 1)

- 출력 데이터 Shape : (36, 28, 20)

- 학습 파라미터 = (입력 채널수) * (필터폭) * (필터높이) * (출력채널수)

= 1 * 4 * 4 * 20 = 320개

ex) Max Pooling Layer 1

- *식 4. Max Pooling Layer 1의 출력 데이터 크기 계산*

$$RowSize = \frac{36}{2} = 18$$
$$ColumnSize = \frac{28}{2} = 14$$

- 입력 데이터 Shape : (36, 28, 20)

- Max Pooling 크기 : (2, 2)

- 출력 데이터 Shape : (18, 14, 20)

- 학습 파라미터 = 0 (Max Pooling Layer는 학습 파라미터가 없다)

ex) Flatten Layer

: CNN의 데이터 타입을 Fully Connected Neural Network의 형태로 변경하는 Layer

- 파라미터가 존재하지 않고, 입력 데이터의 Shape 변경만 수행한다.

- 입력 데이터 Shape = (2, 1, 180)

- 출력 데이터 Shape = (160, 1)

ex) Softmax Layer

: 이 네트워크의 분류 클래스가 100개 이기 때문에 최종 데이터의 Shape은 (100, 1)이다.

- 입력 데이터 Shape = (160, 1)

- 출력 데이터 Shape = (100, 1)

- Weight Shape = (100, 160)

- Softmax Layer 파라미터 = 100 * 160 = 160000

layer	input channel	Filter	output channel	Stride	Pooling	활성함수	Input Shape	Output Shape	파라미터 수
Convolution Layer 1	1	(4, 4)	20	1	X	relu	(39, 31, 1)	(36, 28, 20)	320
Max Pooling Lyaer 1	20	X	20	2	(2, 2)	X	(36, 28, 20)	(18, 14, 20)	0
Convolution Layer 2	20	(3, 3)	40	1	X	relu	(18, 14, 20)	(16, 12, 40)	7,200
Max Pooling Lyaer 2	40	X	40	2	(2,2)	X	(16, 12, 40)	(8, 6, 40)	0
Convolution Layer 3	40	(2, 2)	60	1	1	relu	(8, 6, 40)	(6, 4, 60)	21,600
Max Pooling Lyaer 3	60	X	60	(2, 2)	60	X	(6, 4, 60)	(3, 2, 60)	0
Convolution Layer 4	60	(2, 2)	80	1	1	relu	(3, 2, 60)	(2, 1, 80)	19,200
Flatten	X	X	X	X	X	X	(2, 1, 80)	(160, 1)	0
fully connected Layer	X	X	X	X	X	softmax	(160, 1)	(100, 1)	160,000
합계	X	X	X	X	X	softmax	(160, 1)	(100, 1)	208,320

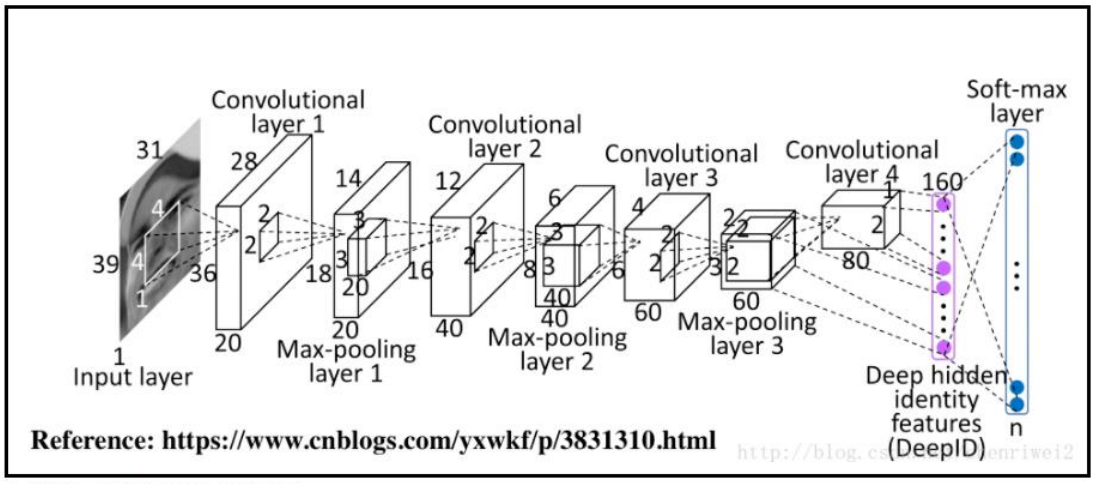


그림 9: 예제 CNN 이미지

<같은 예제를 Fully Connected Neural Network로 구성하였을 때>

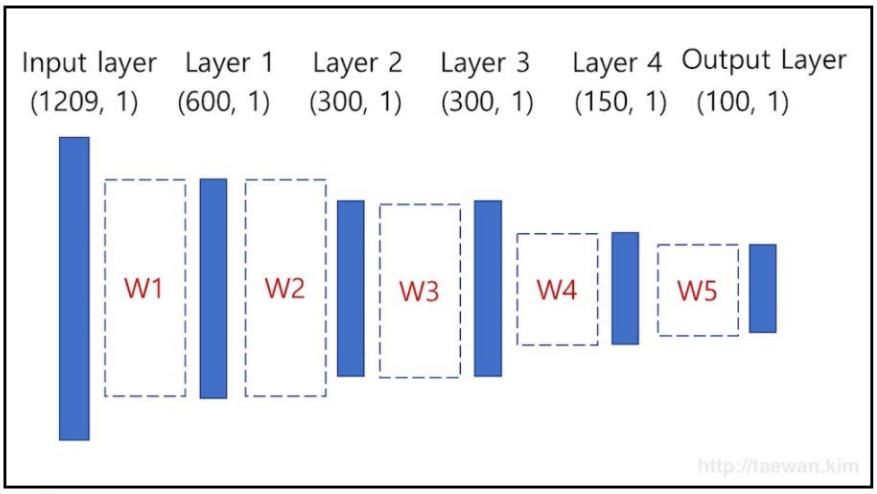


그림 10: FC 신경망 - 비교

레이어	입력 노드	출력 노드	Weight Shape	파라미터 수
Layer 1	1209	600	(1209,600)	725,400
Layer 2	600	300	(600,300)	180,000
Layer 3	300	300	(300,300)	90,000
Layer 4	300	150	(300,150)	45,000
Output	150	100	(150,100)	15,000
합계				1,055,400

- CNN 파라미터와 비교하면 10배 이상의 학습 파라미터를 갖는다. (CNN이 약 20% 규모)

- 따라서, CNN이 FC에 비해 학습이 쉽고 네트워크 처리 속도가 빠르다. 인식률도 CNN이 더 높다.