

합성곱 신경망 (Convolutional Neural Network, CNN)

2021. 9. 8. 21:05

#CNN #합성곱신경망 #컨볼루션 #풀링레이어 #커널 #필터 #스트라이드 #패딩



목차

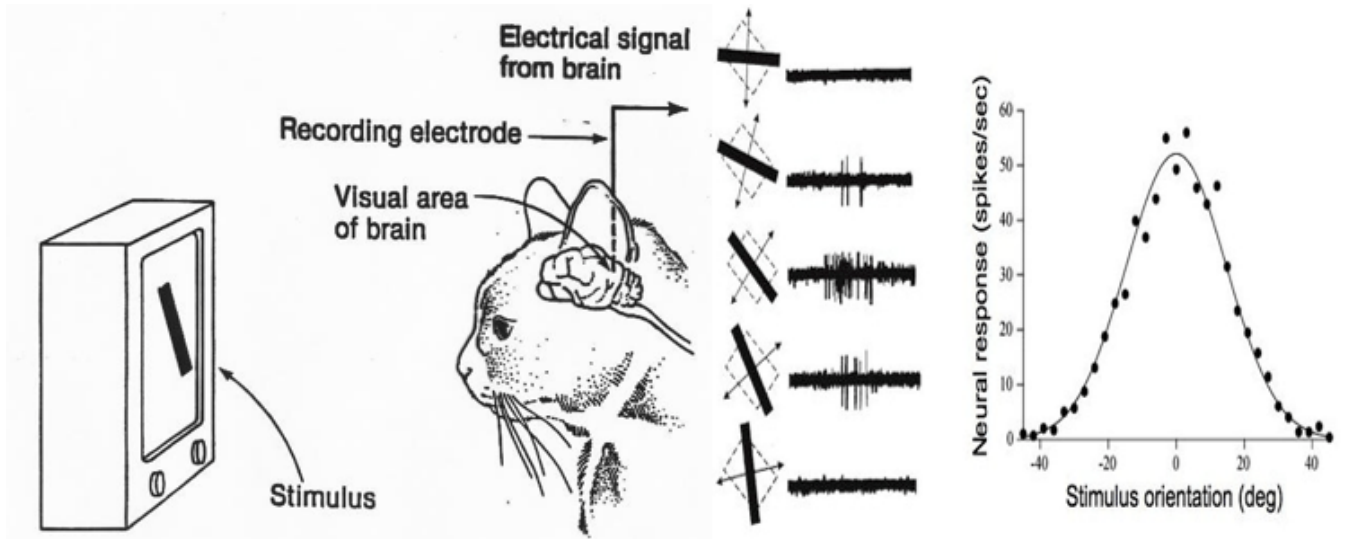
- 합성곱 신경망(Convolutional Neural Network, CNN)
- 완전 연결 계층의 문제점
- 이미지 처리와 필터링 기법
- 합성곱 신경망 아키텍처
- 컨볼루션 레이어(Convolution Layer)
- 풀링 레이어(Pooling Layer)

합성곱 신경망(Convolutional Neural Network, CNN)

합성곱 신경망은 이미지 데이터를 학습하고 인식하는데 특화된 알고리즘에 해당한다. Convolution의 의미는 **신호처리** 분야에서 사용되는 용어로 이미지 프로세싱에서 **일정한 패턴**으로 변환하기 위해 수행하는 **행렬연산**이라

는 의미를 가진다. Convolution은 특정한 수가 조합된 행렬인 필터(=커널)를 사용하는데, 이러한 필터의 값이 어떻게 구성되어 있느냐에 따라 패턴이 달라진다[1].

CNN은 1989년에 나온 모델로 현재의, 이미지를 인식하고 처리하는, 컴퓨터 비전 분야에서 가장 많이 사용된다. CNN 모델의 아이디어는 아래의 실험으로부터 차용하였다.



1950년대 허블과 비셀은 고양이의 시각 피질 실험에서 고양이 시야의 한 쪽에 자극을 주었더니 전체 뉴런이 아닌 특정 뉴런만이 활성화되는 것을 발견했다. 또한 물체의 형태와 방향에 따라서도 활성화되는 뉴런이 다르며 어떤 뉴런의 경우 저수준 뉴런의 출력을 조합한 복잡한 신호에만 반응한다는 것을 관찰했다. 이 실험을 통해 동물의 시각 피질 안의 뉴런들은 일정 범위 안의 자극에만 활성화되는 '근접 수용 영역(local receptive field)'을 가지며 이 수용 영역들이 서로 겹쳐져 전체 시야를 이룬다는 것을 발견했다.

이러한 아이디어에 영향을 받은 얀 르쿤 교수는 인접한 두 층의 노드들이 전부 연결 되어있는 기존의 인경신경망이 아닌 특정 국소 영역에 속하는 노드들의 연결로 이루어진 획기적인 인공신경망을 고안해냈고 이것이 바로 합성곱 신경망이다.

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

Feature extraction using convolution 위의 그림을 통해 특정한 필터를 통해 이미지 데이터에 Convolution(행렬 연산)이 적용되어 변환하는 과정을 확인할 수 있다. 이미지 데이터는 픽셀들의 합으로 이루어져 있는데 한 픽셀은 RGB의 세가지 색으로 구성된다. 따라서 100x100의 아주 작은 이미지라 하여도 $(100 \times 100 \times 3) = 30,000$ 개라는 큰 크기의 데이터로 구성되고 이에 따라 수 많은 이미지를 일반적인 신경망에 그대로 입력시키게 되면 학습에 있어 많은 시간이 필요하게 된다.

위와 같이 3x3 사이즈의 필터를 통해 Convolution을 수행하면 오른쪽 그림과 같이 데이터의 크기가 축소되는 효과를 얻을 수 있고, 이러한 Convolution을 여러번 적용하여 데이터의 크기를 줄이면서 이미지의 특징(패턴)을 추출할 수 있다는 것이 합성곱 신경망의 특징에 해당한다[2].

완전 연결 계층의 문제점



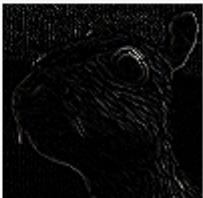

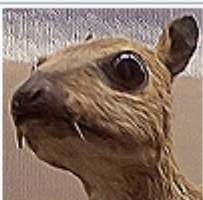
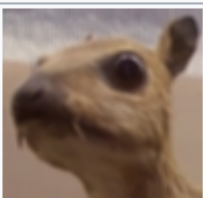
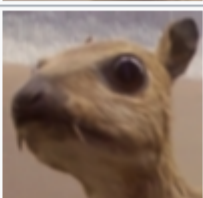
완전 연결 계층의 경우 이미지 전체를 하나의 데이터로 생각하여 입력으로 받아들이기에, 이미지의 특성을 찾지 못하고 이미지의 위치가 조금만 달라지거나 왜곡된 경우 올바른 성능을 내지 못한다. 즉, 완전연 연결 계층을 이용하여 이미지 분류를 하면 3차원(높이, 폭, 채널)인 이미지 데이터를 입력층에 넣어주기 위해 3차원에서 1차원으로 데이터 변환하는 과정을 거치는데, 이 때 **데이터의 형상이 무시**된다는 점이다. 이미지 데이터의 경우 3차원(높이, 폭, 채널)의 형상을 가지며, 이 형상은 **공간적 구조(Spatial Structure)** 또는 **공간 정보**라 불리는 정보를 가진다.

이미지에 있어 공간 정보라는 것은 말 그대로 이미지가 가지는 공간적인 정보를 의미하는데, 가까운 픽셀들은 서로 픽셀값이 비슷하여 사진에서 음영, 선, 질감 등으로 보여진다거나, RGB의 각 채널은 서로 밀접히 관련되어 있거나, 거리가 먼 픽셀간의 관련이 없는 등의 정보들을 의미하며 이러한 정보들은 완전 연결 계층에 입력을 위해 3차원에서 1차원으로 Shift되는 순간 정보가 사라지는 문제가 발생한다[3]. 그러나 합성곱 신경망은 완전 연결 신경망과 달리 3차원의 이미지를 그대로 Input Layer에 입력하고 출력 또한 3차원 데이터로 출력하여 다음 Layer로

전달하기 때문에 데이터의 원래 구조를 유지할 수 있다. 이렇게 할 경우 이미지가 왜곡된다하여도 입력 데이터의 공간정보를 잃지 않기 때문에 CNN에서는 형상을 가지는 데이터의 특성을 추출할 수 있어 제대로 학습할 수 있게 된다.

이미지 처리와 필터링 기법

이미지 처리에는 다음과 같이 여러 필터링 기법이 존재한다.

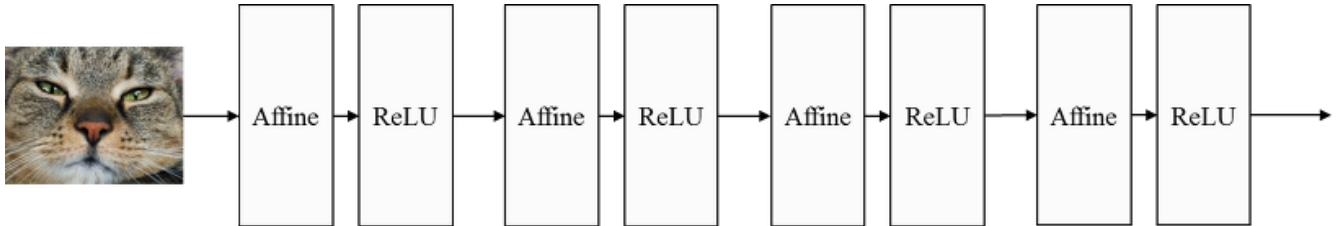
Operation	Kernel ω	Image result $g(x,y)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3×3 (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

[https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))위와 같이 기존에는 이미지 처리를 위해서 고정된 필터를 사용했고, 이미지를 처리하고 분류하는 알고리즘을 개발할 때 필터링 기법을 사용하여 분류 정확도를 향상시켰다.

하지만 이러한 필터링 기법을 사용할때 있어 필터를 사람의 직관이나 반복적인 실험을 통해 최적의 필터를 찾아야만 했다. 따라서 합성곱 신경망은 이러한 필터를 일일이 수동으로 고정된 필터를 찾는 것이 아니라 자동으로 데이터 처리에 적합한 필터를 학습하는 것을 목표로 나오게 되었다.

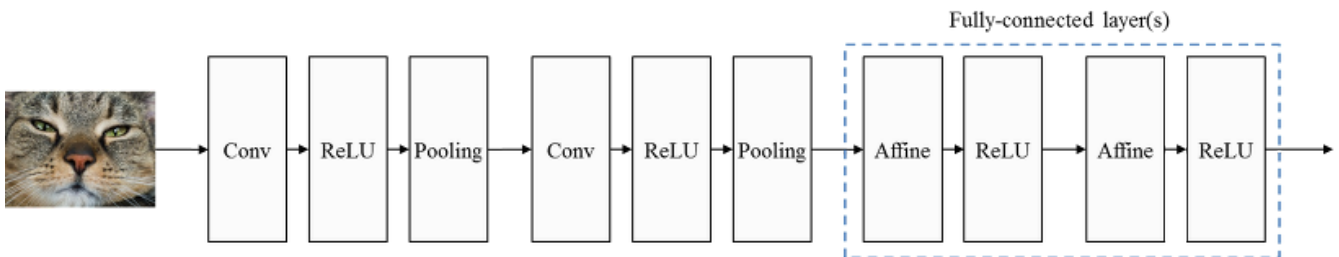
합성곱 신경망 아키텍처

일반적인 신경망의 경우 아래 그림과 같이 Affine으로 명시된 Fully-Connected 연산과 ReLU와 같은 활성화 함수의 합성으로 이루어진 계층을 여러 개 쌓은 구조이다.



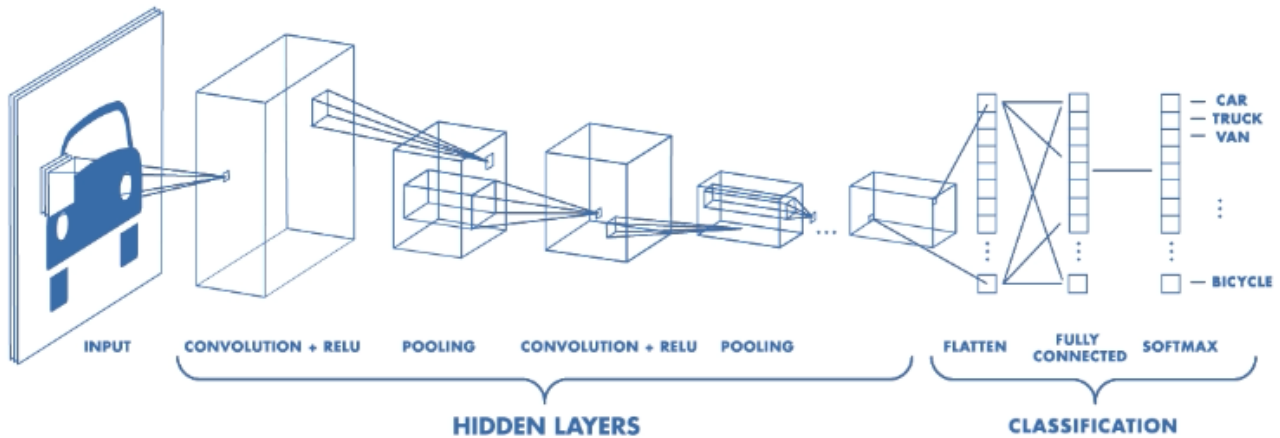
ANN의 구조

CNN은 아래의 그림과 같이 합성곱 계층(Convolutional Layer)와 풀링 계층(Pooling Layer)라고 하는 새로운 층을 Fully-Connected 계층 이전에 추가함으로써 원본 이미지에 필터링 기법을 적용한 뒤 필터링된 이미지에 대해 분류 연산이 수행되도록 구성된다.



CNN의 구조

위의 CNN의 구조를 조금 더 도식화 하면 아래와 같다.



CNN 구조 도식화합성곱 신경망은 크게 “필터링”과 “분류”의 과정을 거쳐 결과를 출력하게 되는데 달리 말하여 CNN은 이미지의 특징을 추출하는 부분과 클래스를 분류하는 부분으로 나눌 수 있다. 특징 추출 영역의 경우 위 그림과 같이 Hidden Layer에 속하는 Convolution Layer와 Pooling Layer를 여러 겹 쌓는 형태로 구성된다.

Convolution Layer는 입력 데이터에 대해 필터를 적용한 후 활성화 함수를 반영하는 필수 요소에 해당하며, 다음에 위치하는 Pooling Layer는 선택적인 Layer에 해당한다. CNN의 마지막 부분에는 이미지 분류를 위한 Fully Connected Layer가 추가된다. 이미지의 특징을 추출하는 부분과 이미지를 분류하는 부분 사이에 이미지 형태의 데이터를 배열 형태로 만드는 Flatten Layer가 위치한다.

CNN은 이미지 특징 추출을 위해 입력된 이미지 데이터를 필터가 순회하며 합성곱을 계산하고, 그 계산 결과를 이용하여 Feature Map을 생성한다. Convolution Layer는 필터의 크기, Stride, Padding의 적용 여부, Max Pooling의 크기 등에 따라 출력 데이터의 Shape이 변경되며, 앞서 언급한 이러한 요소의 경우 CNN에서 사용하는 학습 파라미터 또는 하이퍼 파라미터에 해당한다.

채널(Channel)

이미지를 구성하는 픽셀은 실수이다. 컬러 이미지의 경우 색을 표현하기 위해 각 픽셀을 Red, Green, Blue인 3개의 실수로 표현한 3차원 데이터이다. 컬러 이미지는 3개의 채널로 구성되며, 흑백 이미지는 1개의 채널을 가지는 2차원 데이터로 구성된다. 높이가 64 픽셀, 폭이 32 픽셀인 컬러 사진의 데이터 shape은 (64, 32, 3)으로 표현되며, 높이가 64 픽셀, 폭이 32 픽셀인 흑백 사진의 데이터의 shape은 (64, 32, 1)이다.

RED Channel



Green Channel



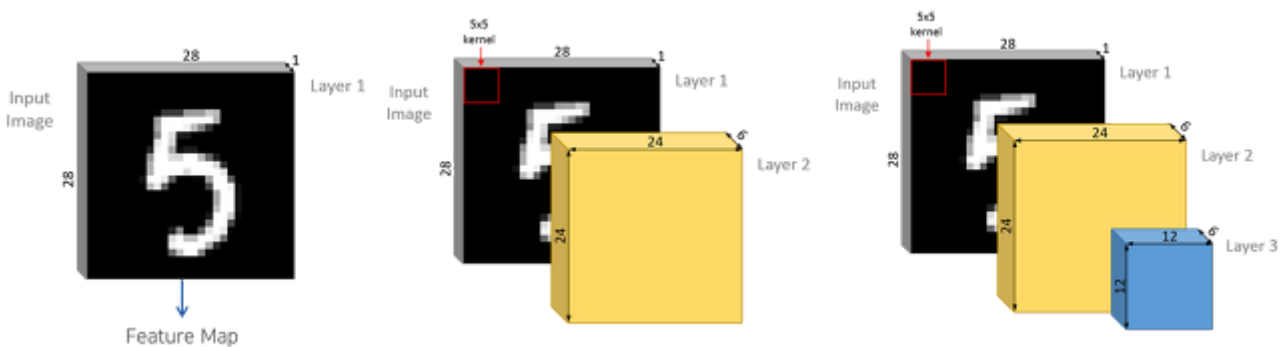
Blue Channel



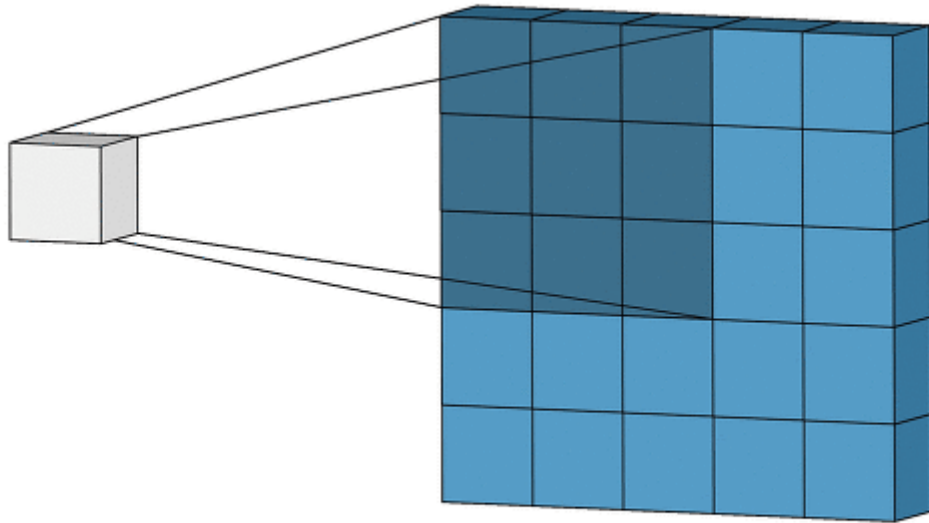
3개의 채널로 만들어진 컬러 이미지

합성곱(Convolution)

합성곱 층은 CNN에서의 가장 중요한 구성요소로, 완전연결 계층과 달리 입력 데이터의 형상을 유지한다. 아래와 같이 3차원 이미지를 그대로 입력층에 받으며, 출력 또한 3차원 데이터로 출력하여 다음 계층으로 전달한다.



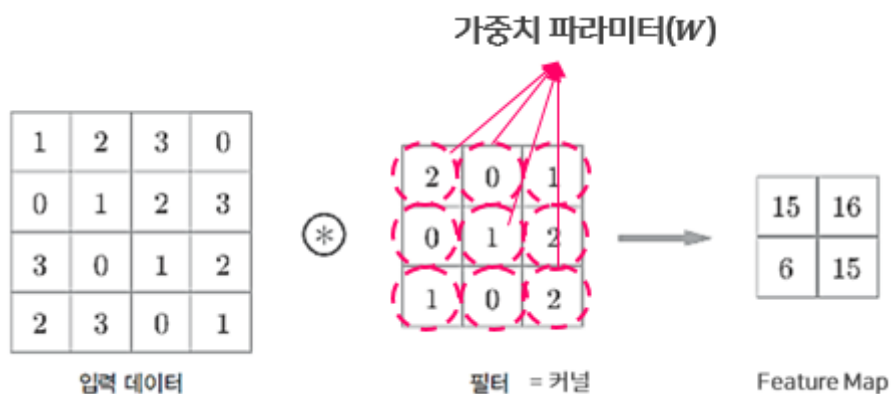
합성곱 층의 뉴런의 경우 아래와 같이 입력 이미지의 모든 픽셀에 연결되는 것이 아닌, 합성곱 층의 뉴런의 수용 영역안에 있는 픽셀에만 연결되기 때문에, 앞의 합성곱층에서는 저수준 특성에 집중하고, 그 다음 합성곱 층에서는 고수준 특성으로 조합해 나가도록 한다[4].



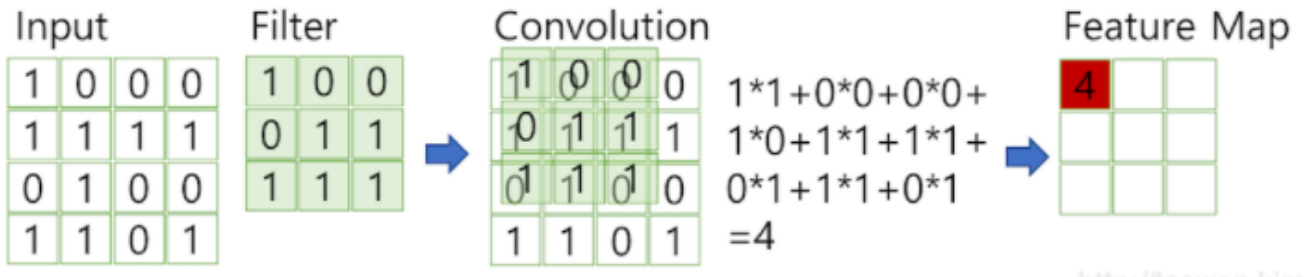
합성곱 처리 절차

필터(Filter)

위의 합성곱에서 언급한 수용영역을 Convolution Layer에서는 필터 또는 커널이라 부른다. 아래의 그림과 같이 필터는 합성곱 층에서의 가중치 파라미터(w)에 해당하며, 학습단계에서 적절한 필터를 찾도록 학습된다. 이후 합성곱 층에서 입력 데이터에 필터를 적용하여 필터와 유사한 이미지의 영역을 강조하는 특성 맵(Feature Map)을 출력하여 다음 층으로 전달한다.



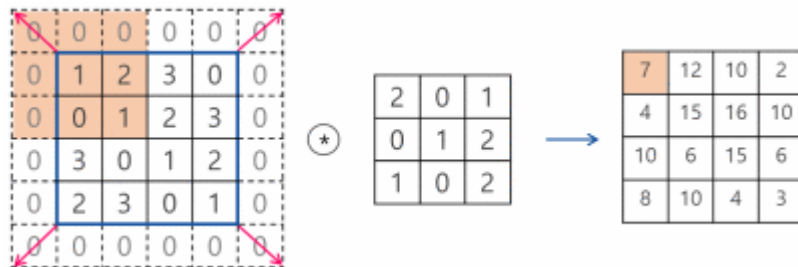
필터는 이미지의 특징을 찾아내기 위한 공용 파라미터인데, 필터는 일반적으로 (4, 4)나 (3, 3)과 같은 정사각 행렬로 정의된다. 위의 합성곱 처리 절차와 같이 입력 데이터를 지정된 간격으로 순회하며 채널별로 합성곱을 하고 모든 채널(컬러의 경우 3개)의 합성곱의 합을 Feature Map으로 생성한다.



합성곱 계산 절차

스트라이드(Stride)

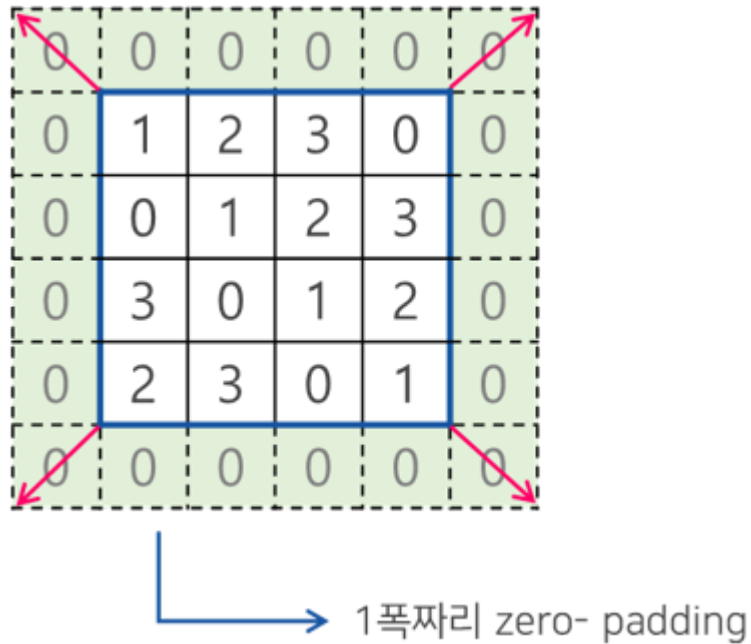
필터는 지정된 간격으로 이동하며 전체 입력데이터와 합성곱을하여 Feature Map을 생성하는데, 위의 그림은 채널이 1개인 입력 데이터를 (3, 3) 크기의 필터로 합성곱을 하는 과정을 설명한 것이다. 이 때 지정된 간격을 Stride 라고 부른다.



Stride는 출력 데이터의 크기를 조절하기 위해 사용하며, 일반적으로 Stride는 1과 같이 작은 값에 더 잘 동작하며, Stride가 1일 경우 입력 데이터의 Spatial 크기는 Pooling Layer에서만 조절할 수 있다. 위의 그림은 Stride 1과 이후 언급될 패딩(zero-padding)을 적용한 뒤 합성곱 연산을 수행후 Feature Map을 생성하는 예제이다.

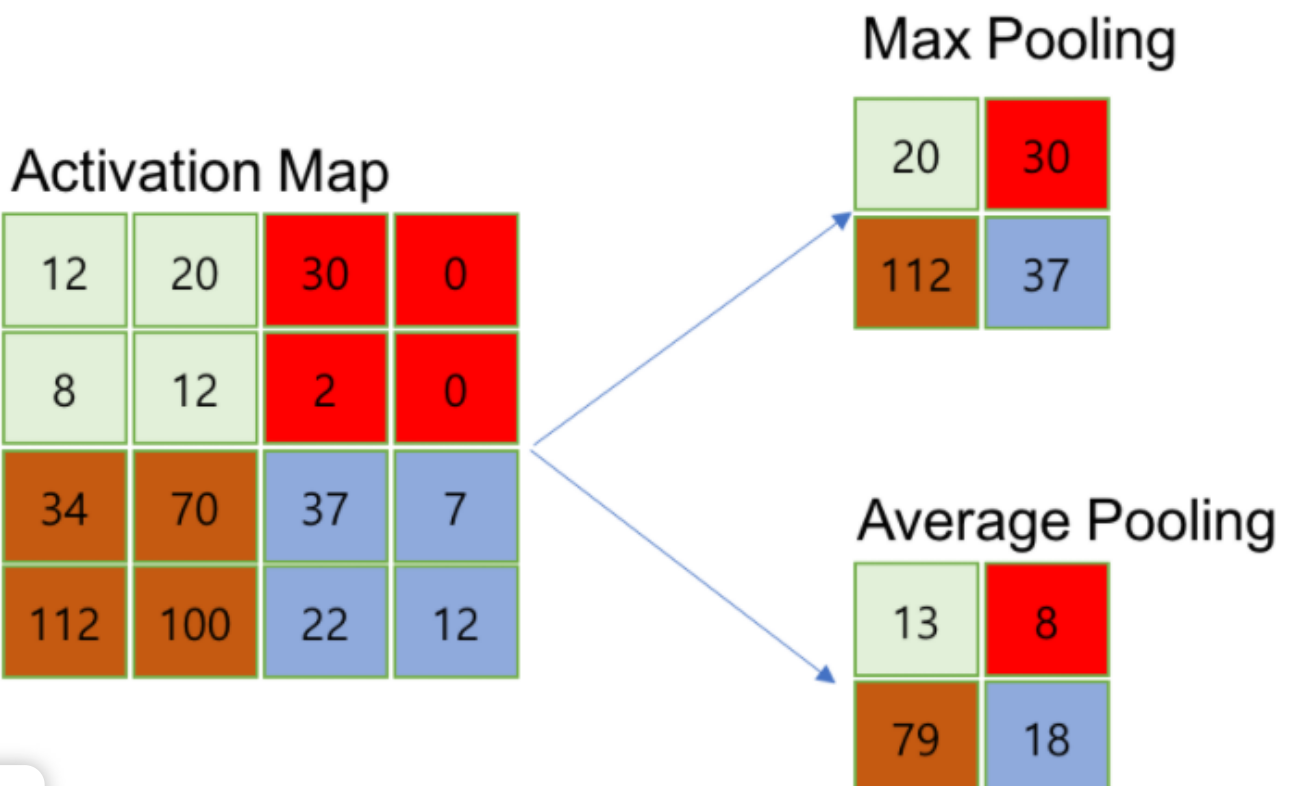
패딩(Padding)

패딩은 Convolution Layer의 출력 데이터가 줄어드는 것을 방지하기 위해 사용하는 개념이며, 합성곱 연산을 수행하기 이전에 입력 데이터의 외각에 지정된 픽셀만큼 특정 값으로 채워 넣는 것을 의미하고, 패딩에 사용할 값은 하이퍼파라미터로 결정이 가능하지만 보통 zero-padding을 사용한다. 패딩은 또한 주로 합성곱 층의 출력이 입력 데이터의 공간적 크기와 동일하게 맞춰주기 위해 사용하기도 한다.



Pooling Layer

풀링 레이어의 경우 컨볼루션 레이어의 출력 데이터를 입력으로 받아서 출력 데이터에 해당하는 Feature Map(=Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용된다. 풀링 레이어를 사용하는 방법에는 크게 Max Pooling, Average Pooling, Min Pooling이 있다. 아래의 그림은 Max Pooling과 Average Pooling의 동작 방식을 설명하며, 일반적으로 **Pooling의 크기와 Stride를 같은 크기로 설정**하여 모든 원소가 한 번씩 처리되도록 설정한다.



풀링 레이어는 컨볼루션 레이어와 비교하여 다음과 같은 특징이 존재한다.

- 학습 대상 파라미터가 없음
- 풀링 레이어를 통과하면 행렬의 크기 감소
- 풀링 레이어를 통해서 채널 수 변경 없음

이때 필터링을 거친 원본 데이터를 피처맵(Feature Map)이라고 하며 필터를 어떻게 설정할지에 따라 피처맵이 달라진다.

Reference

[1] <https://ellun.tistory.com/104>

[2] cnn이란-합성곱-신경망의-구조-알아보기

[3] [딥러닝] 합성곱 신경망, CNN(Convolutional Neural Network) - 이론편

[4] 06. 합성곱 신경망 - Convolutional Neural Networks

[5] 합성곱 신경망 (Convolutional Neural Network, CNN)과 학습 알고리즘

[6] CNN, Convolutional Neural Network 요약

공감

구독하기

'AI Reasearch > 머신러닝-딥러닝' 카테고리의 다른 글

[머신러닝/딥러닝] Ubuntu 20.04에 딥러닝 환경 설치 (0)	2021.11.02
[인공지능 이론] Manifold Learning (1)	2021.09.08
<u>합성곱 신경망 (Convolutional Neural Network, CNN)</u> (0)	2021.09.08
[인공지능 이론] 인공지능의 역사 (0)	2021.09.08
[Pandas] 데이터 프레임에 컬럼 이름 추가하기 (0)	2020.05.12
[머신러닝 이론] 서포트 벡터 머신(Support Vector Machine) (0)	2020.04.12

NAME

PASSWORD

HOME PAGE



http://

SECRET ☐ WRITE

+ Recent posts





[독서노트#17] 타이탄의 도구...





[독서노트#16] 울트라 러닝 (…





[독서노트#15] 7막 7장 (홍정욱)





Powered by [Tistory](#), Designed by [wallel](#)

[Rss Feed](#) and [Twitter](#), [Facebook](#), [Youtube](#), [Google+](#)

