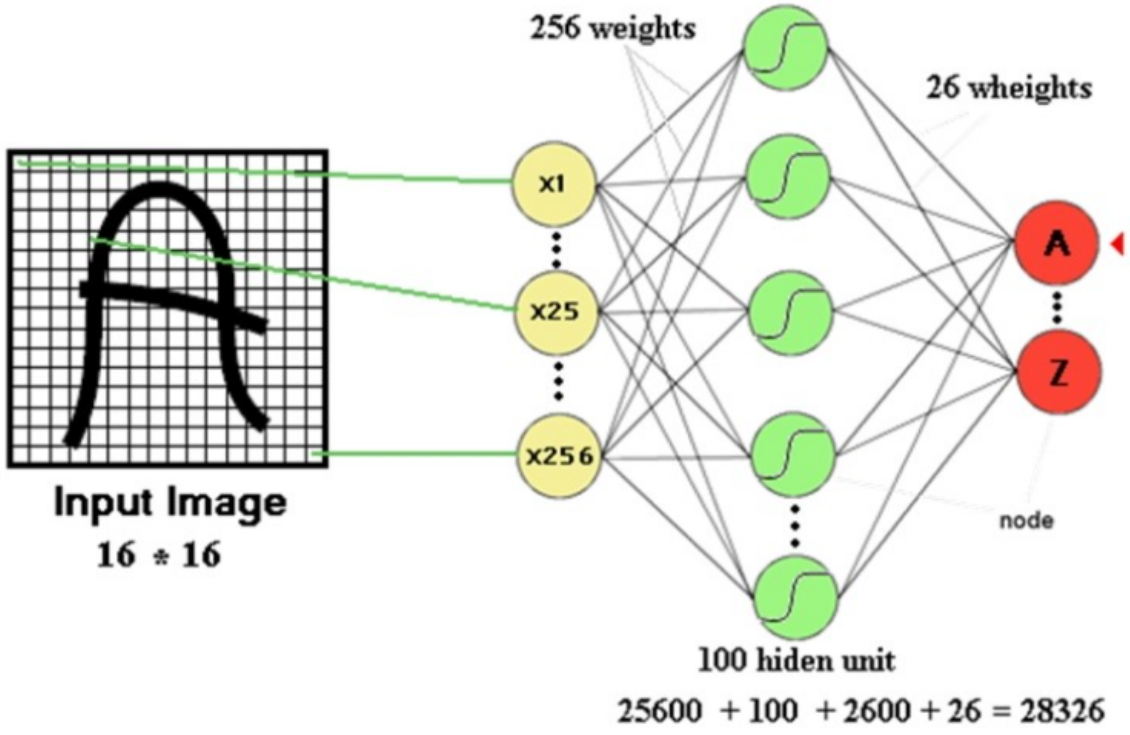
4. CNN student

도입

- 글자를 인식하는 방법을 학습해보자.
    - 기계라면? 픽셀 단위로 쪼개서 학습할듯
    - 사람이라면? 전체적인 모양을 학습할듯

기존 MLNN (Multi-Layer Neural Network)의 문제점

- MLNN 이미지 인식
- 아래와 같은네트워크 형태를 Fully-connected Network 라고 한다.
    - 모든 입력이 위상과는 상관없이 동일한 중요도를 갖고있다고 보기때문에 모든 레이어를 연결한것



- 16*16 의 필기체를 인식하기 위해서 hidden layer 가 10개인 네트워크를 고려한다면 가중치와 바이어스는 총 28,326개가 필요하다.
- 글자가 이동하거나 회전하는 경우, 글자의 크기가 달라지는 경우 입력값의 위치가 달라짐에 따라 서로 다른데이터가 됨에 따라 다른 결과를 나타내게 된다.
    - 글자의 위상적 형태는 고려하지 않고 Raw Data 에대해 직접적으로 처리하기 때문에 위와같이 가중치와 바이어스의 갯수가 늘어나게 된다.

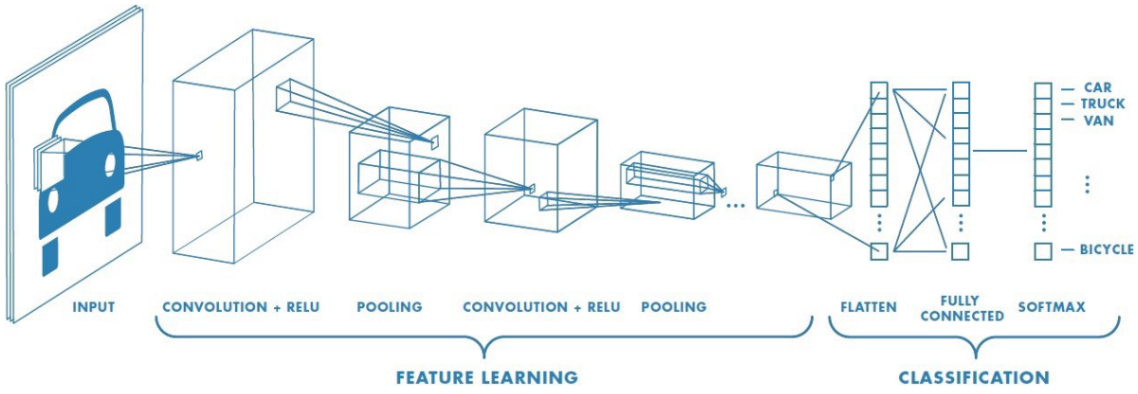
대뇌 시각피질 실험

- Hubel & Wiesel 의 실험
  - 고양이는 사선형 Edge를 Detect 할 수 있다.
  - 이를 가능하게 하는 뉴론은 다음과 같이 구성되어 있다.
    - 기울어진 크고 작은 edge 요소들의 합성 과정을 통해 전체 이미지를 구성한다.
    - 활성화 되는 뉴런에 따라 인식하는 모양이 달라짐.
    - 인간은 여기에 색깔을 인식할수 있는 시각세포도 존재.

우리가 생각해 봐야 하는것들

- 이미지를 인식하기 위해선 어떻게 이미지를 처리해야 할까?
  - 이미지의 윤곽을 찾아내면 된다.
  - Gaussian blur 기법을 이용해 윤곽을 찾아낸다.
- 이미지의 윤곽을 인식했다면 어떤 사물인지 어떻게 알수 있을까?

CNN의 전체구조



각 계층별 요소

컨볼루션층 은 다음과 같은 구성요소로 이루어 졌다.

- `Conv`
  - 입력데이터의 특징을 추출하는 역할
- `Relu`
  - 입력 정보를 0보다 크면 그대로 값을 내보내고, 0보다 작으면 0을 내보내는층
- `Pooling`
  - 입력으로 주어지는 정보를 최대/최소/평균값으로 압축하여 데이터 연산량을 줄여주는 역할 수행 (즉 대푯값을 추출)
  - max pooling
  - min pooling
  - averager pooling

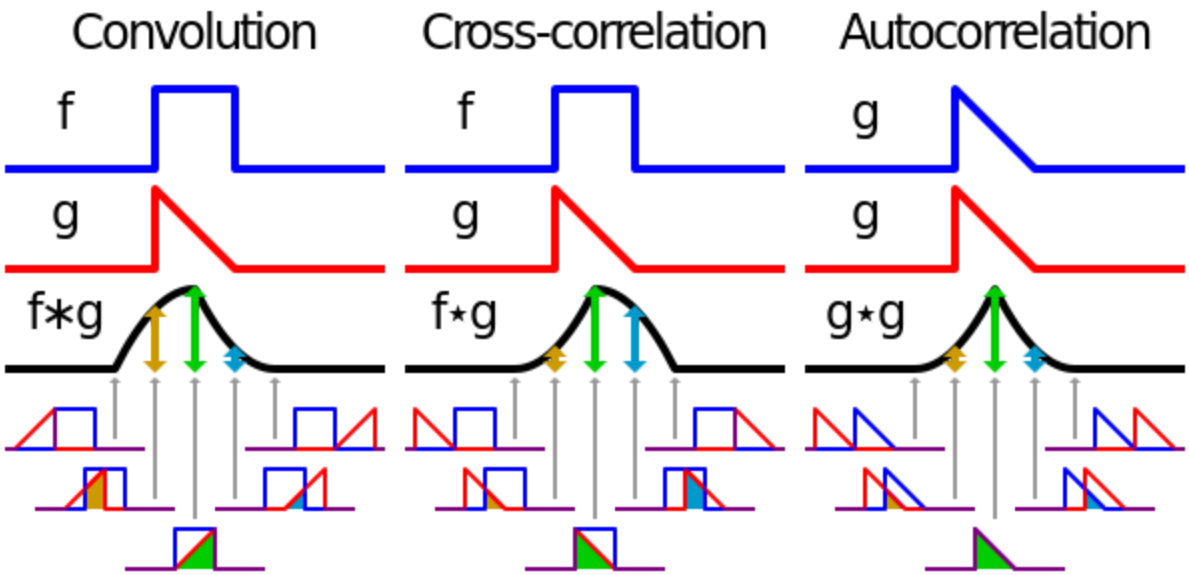
Conv 계층

- Convolution 은 일정 영역의 값들에 대해 가중치를 적용하여 하나의 값을 만드는 연산이다.

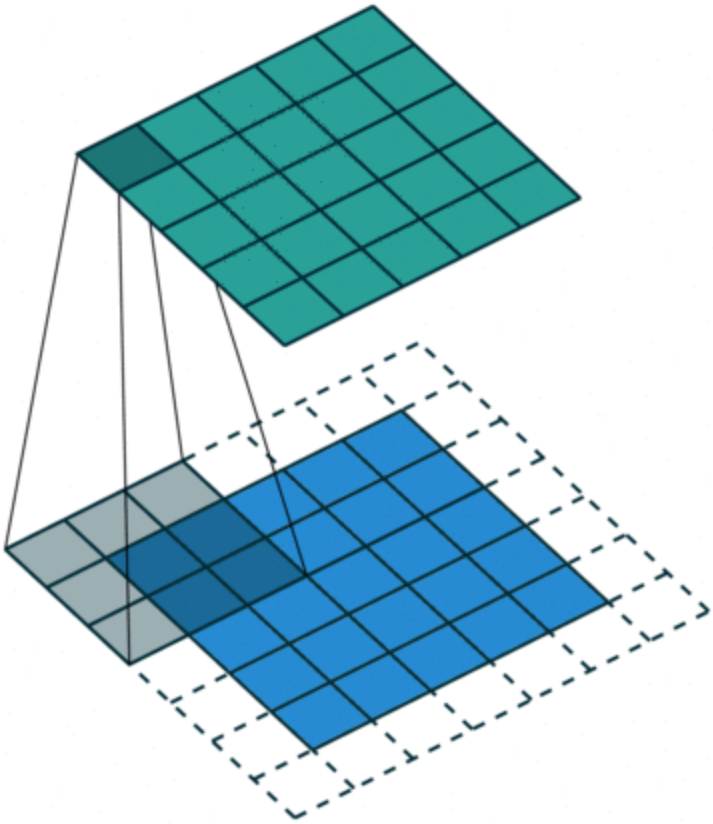
합성곱 연산

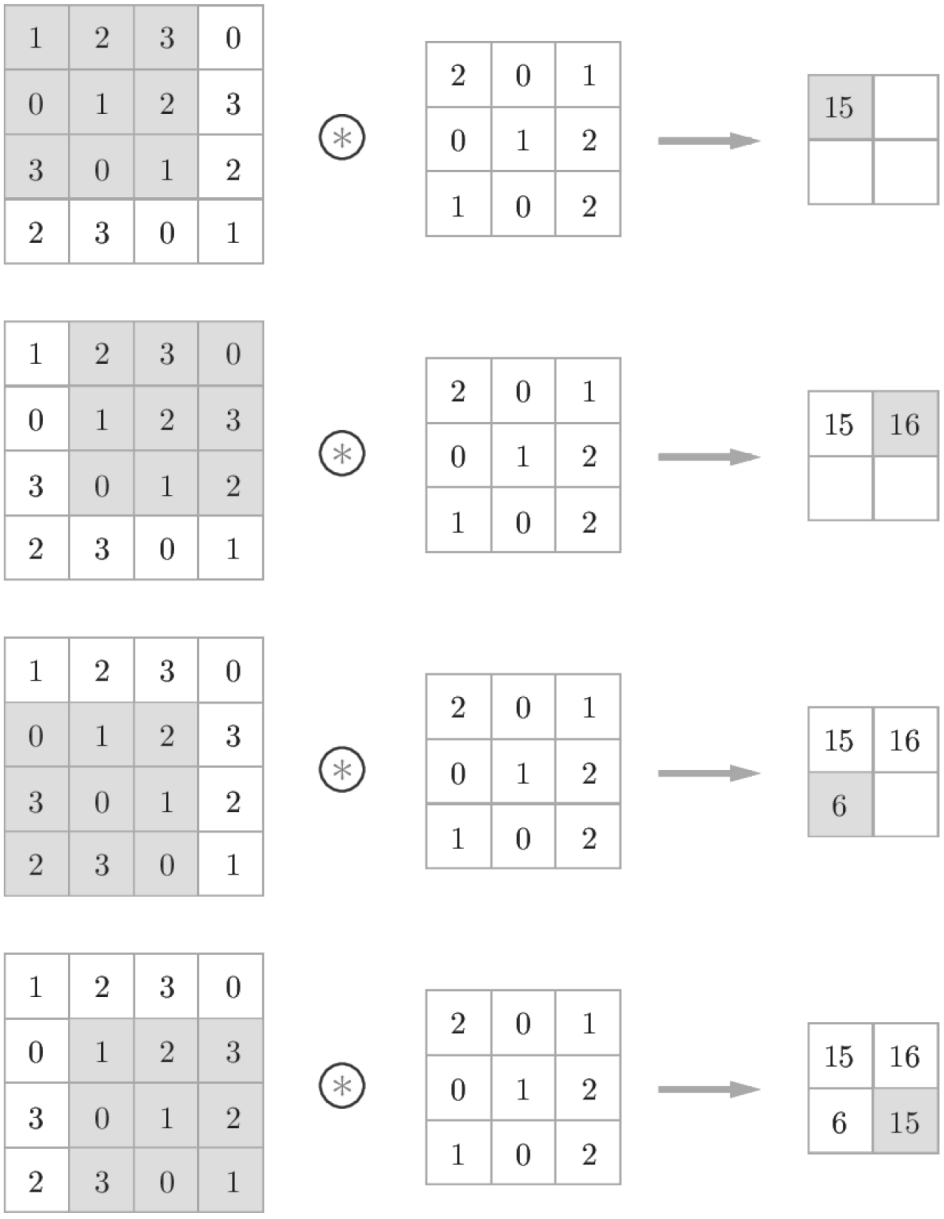
Convolution 이란?

두개의 신호를 합성해서 내보내는 연산을 의미한다. (곱한다음 적분한다.)

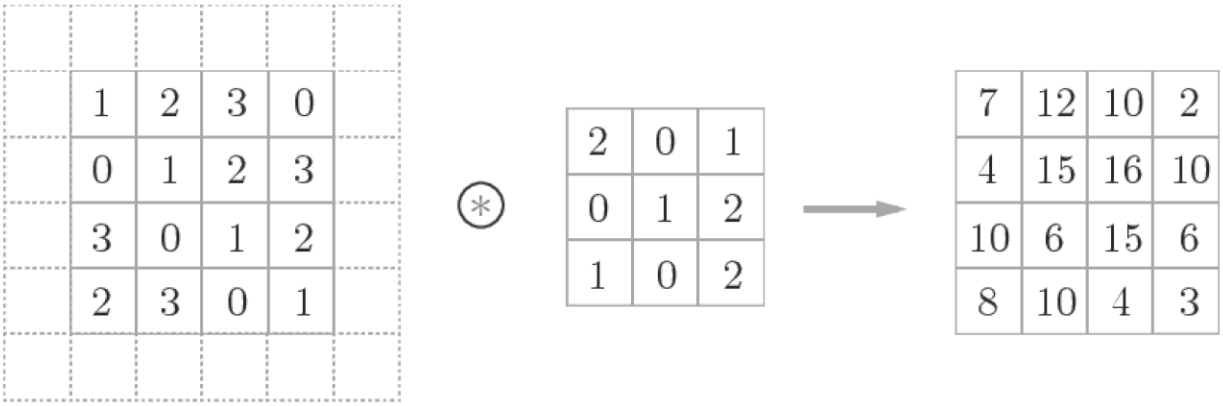


- 합성곱 연산은 이미지 처리에서 말하는 필터 연산에 해당 한다.
- 필터 = 커널
- 합성곱 연산은 필터의 Window를 일정 간격으로 이동해 가며 입력 데이터를 적용한다.
- 4×4 ∗ 3×3 → 2×2




1 2 3 0
0 1 2 3
3 0 1 2
2 3 0 1
2 0 1
0 1 2
1 0 2
15
1 2 3 0
0 1 2 3
3 0 1 2
2 3 0 1
2 0 1
0 1 2
1 0 2
15 16
1 2 3 0
0 1 2 3
3 0 1 2
2 3 0 1
2 0 1
0 1 2
1 0 2
15 16
6
1 2 3 0
0 1 2 3
3 0 1 2
2 3 0 1
2 0 1
0 1 2
1 0 2
15 16
6 15

패딩



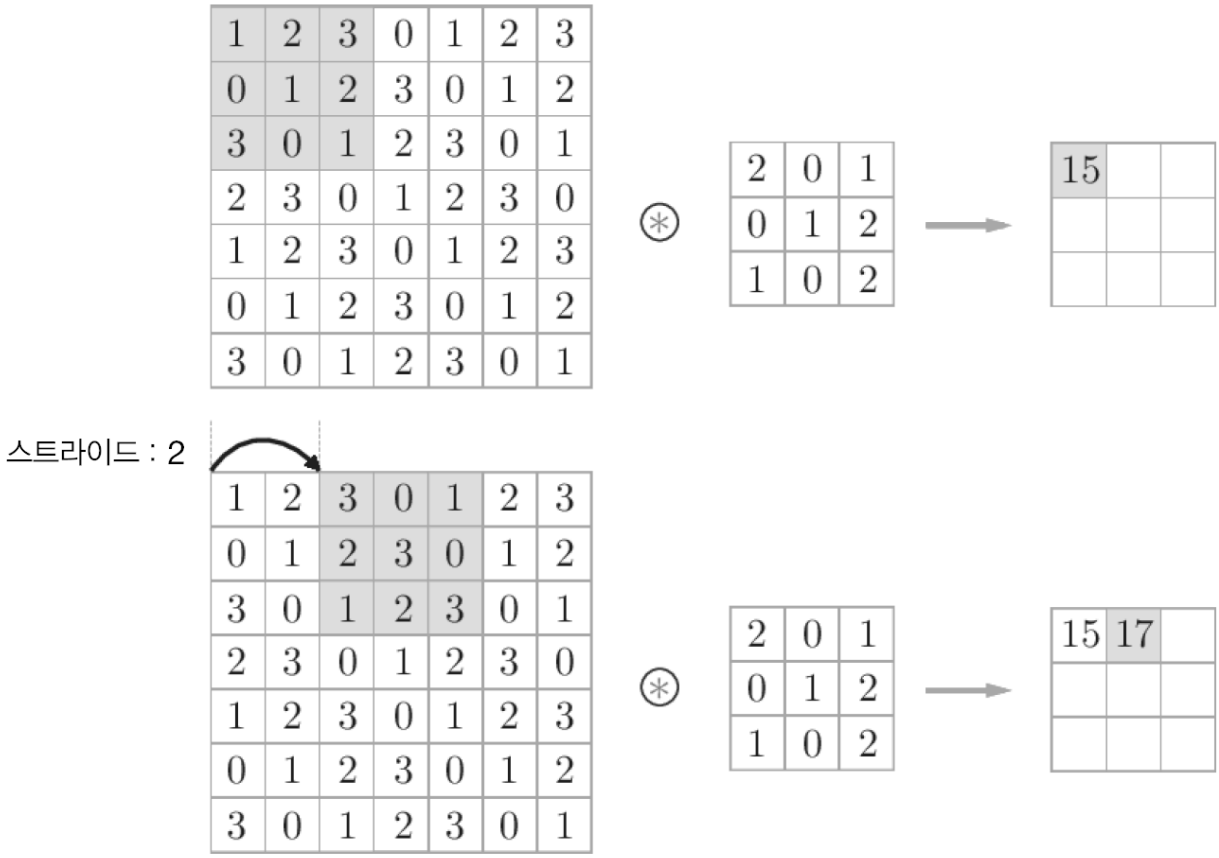
(4, 4)
입력 데이터(패딩 : 1)

(3, 3)
필터

(4, 4)
출력 데이터

- 합성곱 연산을 수행하기 전에 입력 데이터 주변을 0과 같은 특정 값으로 채우기도 하는 것
  - 보통 출력 크기를 조정하기 위해 사용한다.
- 출력 크기의 계산 : **패딩 없는 경우(No Padding)**:
  - 패딩 없이 필터를 적용하면, 출력 이미지의 크기는 입력 이미지의 크기보다 감소합니다.
  - **입력 크기**: $n \times n$
  - **필터 크기**: $f \times f$
  - **출력 크기**: $(n - f + 1) \times (n - f + 1)$
- 출력 크기의 계산 : **패딩 있는 경우(Padding)**:
  - 패딩을 추가하면, 출력 이미지의 크기를 입력 이미지의 크기와 동일하게 유지하거나, 특정한 출력 크기를 조절할 수 있습니다.
  - **패딩 크기**: p
  - **입력 크기**: $n \times n$
  - **필터 크기**: $f \times f$
  - **출력 크기**: $(n - f + 2p + 1) \times (n - f + 2p + 1)$

스트라이드



- 필터를 적용하는 위치의 간격
- **스트라이드(Strided Convolution)**:
  - **스트라이드 값**: s
  - 필터가 이동하는 간격을 의미한다. 예를 들어, $s = 2$ 라면 필터가 한 번에 두 칸씩 이동한다.
  - 출력 이미지의 크기를 줄이는 효과가 있다.
- **출력 크기 계산**:
  - **입력 크기**: $n \times n$
  - **필터 크기**: $f \times f$
  - **패딩 크기**: p

- **스트라이드 값**: s
- **출력 크기**:

  $$\left(\frac{n+2p-f}{s}+1\right)\times\left(\frac{n+2p-f}{s}+1\right)$$

- 이 식을 사용하여 주어진 입력 크기, 필터 크기, 패딩 크기, 스트라이드 값에 따라 출력 이미지의 크기를 계산할 수 있다.

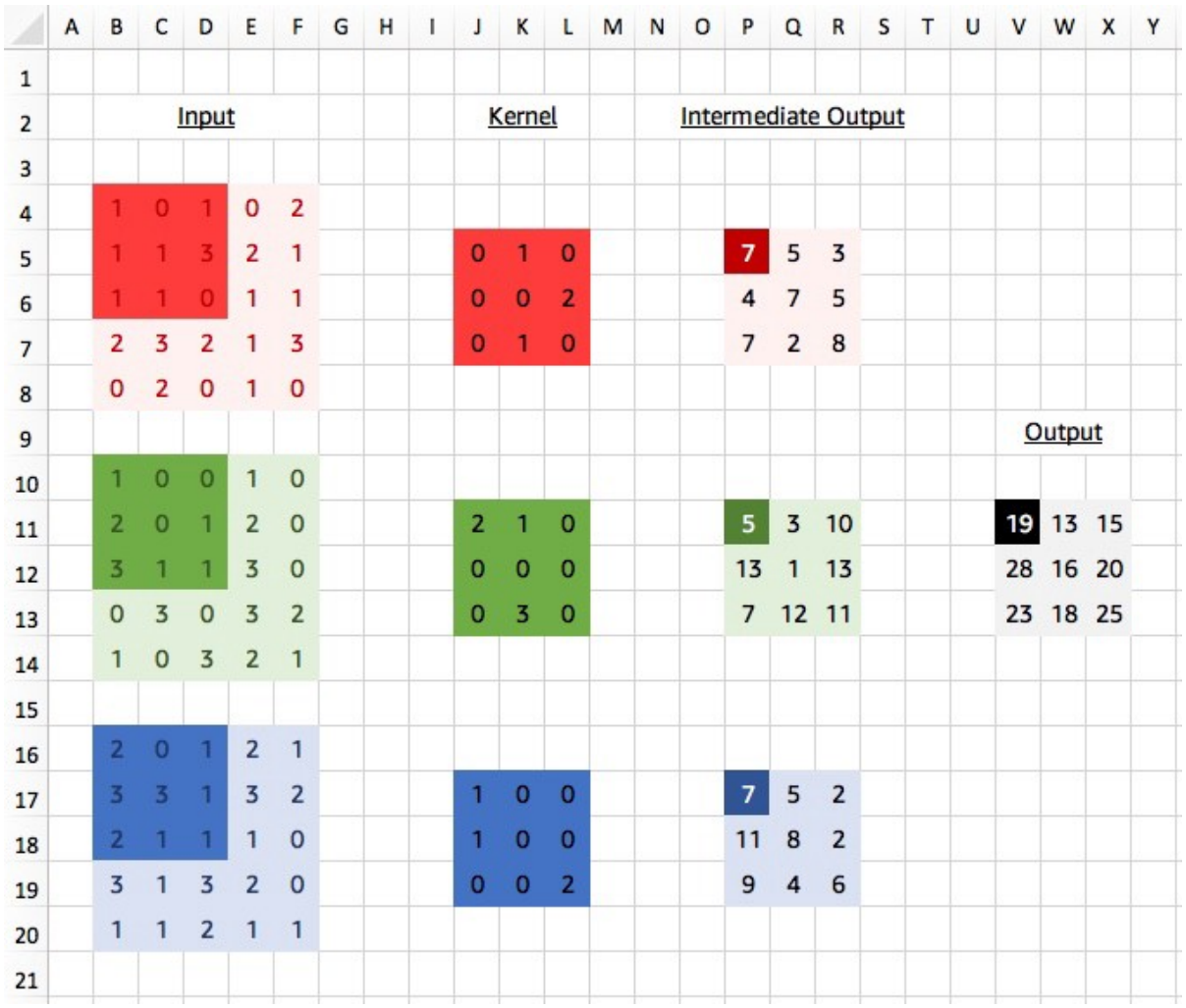
컬러 이미지는 어떻게 Convolution 되는걸까?

https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215

- 1D Convolutions to 1 dimensional data (temporal)
- 2D Convolutions to 2 dimensional data (height and width)

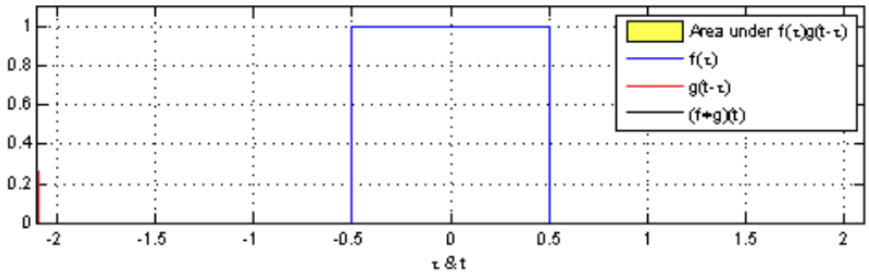
- 3D Convolutions to 3 dimensional data (height, width and depth)

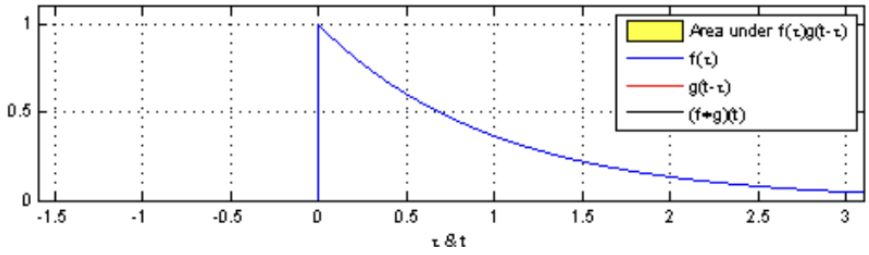
  https://medium.com/apache-mxnet/multi-channel-convolutions-explained-with-ms-excel-9bbf8eb77108



Convolution Equation

1. **1D 합성곱(1D Convolution)**:
   - **정의**: $G = H * F$
   - **계산식**: $G[t] = \int_{-\infty}^{\infty} H[t-\tau]F[\tau]d\tau$





2. **True Convolution Equation(Convolution)**
   - **정의:** $H = H * F$
   - **계산식**: $H[i,j] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} H[u-i, v-j]F[u,v]$
   - **구현**:
     - 필터를 두 차원 모두에서 뒤집는다: 하단에서 상단으로, 오른쪽에서 왼쪽으로
     - 그 후 교차 상관을 적용한다
   - **참고사항**: 인덱스가 범위를 벗어나는 경우 0으로 처리된다

3. **Deep learning convolution(Cross-correlation)**
   - **정의**: 두 입력(신호) 간의 유사성 측정
   - **계산식**: $H = H \otimes F$
   - **계산 예시**:

$$H[2,2] = F_1 \cdot H_1 + F_2 \cdot H_2 + ... + F_9 \cdot H_9$$

- **표현**: $H[i,j] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} H[u+i, v+j] F[u,v]$

4. **합성곱과 교차 상관의 비교**:
    - **합성곱**은 필터 반응을 나타내며, 필터를 뒤집은 후 적용된다.
    - **교차 상관**은 두 신호 간의 유사성을 측정하며, 필터를 뒤집지 않고 적용된다.
    - 가우시안 필터(Gaussian filter)와 같이 대칭적인 필터를 사용하는 경우, 합성곱과 교차 상관은 같은 결과를 나타낸다.
    - 입력이 임펄스 신호(impulse signal)인 경우, 합성곱과 교차 상관의 출력은 다를 수 있다.

Edge Detection

수직 가장자리 검출(Vertical Edge Detection):

- **예시**: 아래의 필터를 사용하여 수직 가장자리를 검출

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- **결과**: 수직 가장자리는 밝기 값이 급격하게 변하는 곳에서 높은 값으로 나타난다.

1. **수평 가장자리 검출(Horizontal Edge Detection)**:
    - **예시**: 아래의 필터를 사용하여 수평 가장자리를 검출

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

    - **결과**: 수평 가장자리는 밝기 값이 수평 방향으로 급격하게 변하는 곳에서 높은 값으로 나타난다.

2. **패딩(Padding)과 스트라이드(Strided Convolution)**:
    - **패딩**: 출력 이미지의 크기를 입력 이미지와 동일하게 유지하기 위해 사용.

- **스트라이드**: 필터를 적용할 때 몇 픽셀씩 건너뛸지 결정하는 값.

3. **유효 합성곱(Valid Convolution)과 동일 합성곱(Same Convolution)**:
   - **유효 합성곱**: 패딩 없이 적용되며, 출력 이미지의 크기가 입력 이미지보다 작아진다.
   - **동일 합성곱**: 패딩을 사용하여 출력 이미지의 크기를 입력 이미지와 동일하게 유지한다.
4. **출력 크기 계산**:
   - 출력 크기는 입력 이미지 크기, 필터 크기, 패딩, 스트라이드에 따라 결정된다.

Convolution 계층의 역할

- input data * kernel → feature 추출 → feature map 작성 → feature map 의 max pooling 값을 다음 계층으로 전달

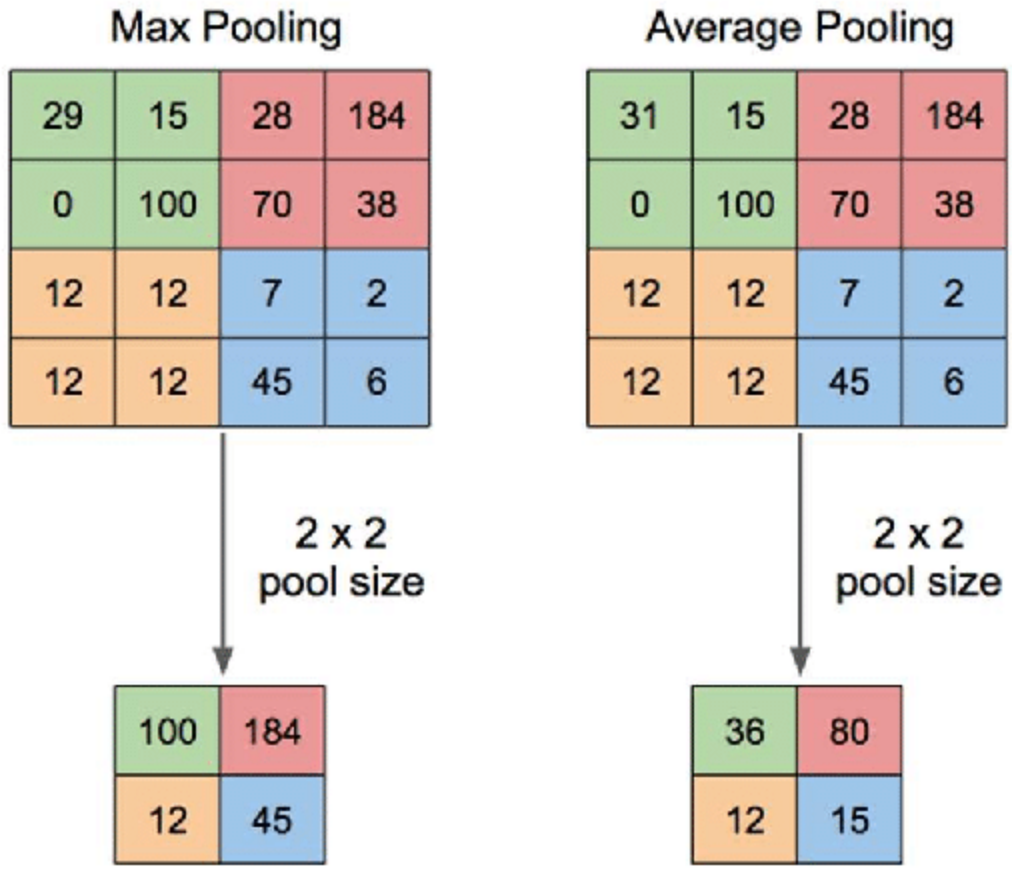
대뇌 시각 피질 실험과 결합

- 입력데이터
- 필터 : 가로 필터, 세로 필터, 대각선 필터 적용
- https://www.youtube.com/watch?time_continue=228&v=AgkfIQ4IGaM&feature=emb_title

Pooling 계층

- 풀링은 2차원 데이터의 세로 및 가로 방향의 공간을 줄이는 연산
- 풀링에는 최대 풀링(Max Pooling), 평균 풀링(Average Pooling)
- 최대 풀링은 대상 영역에서 최댓값을 취하는 연산이고, 평균 풀링은 대상 영역의 평균을 계산한다. 이미지 인식 분야에서는 주로 최대 풀링을 사용한다.
- 대푯값을 추출해내는 과정이다.

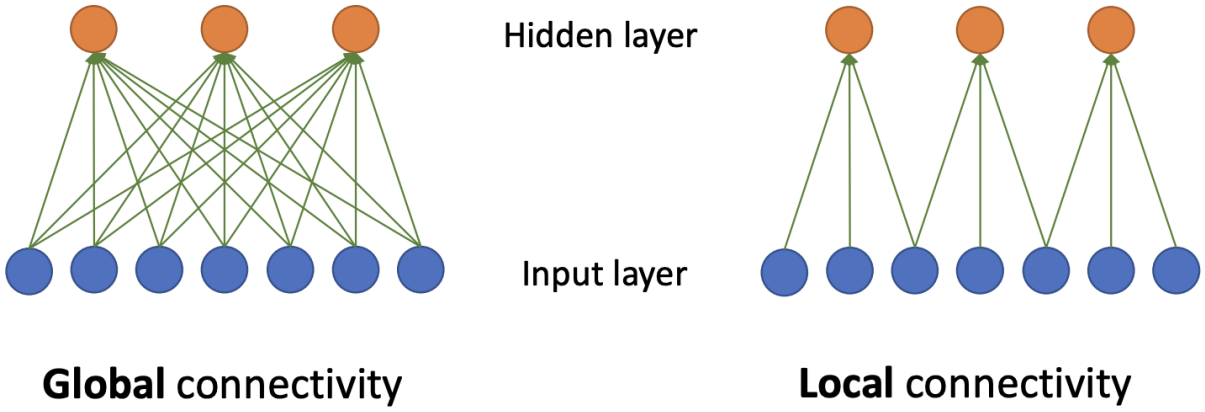
Max Pooling



Convolutional Neural Networks 의 특징

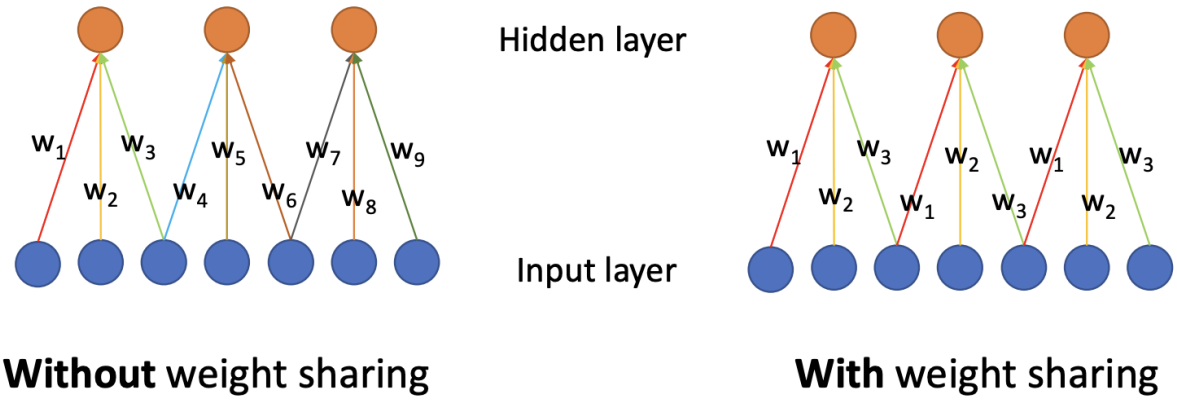
지역 연결성(Local Connectivity):

- **개념** : 모든 데이터에 대해 input으로 사용하지 않고 인접한 데이터에 대해서만 input으로 사용한다.
- 입력 유닛 수: 7
- 은닉 유닛 수: 3
- 매개변수 수:
- 전역 연결성(Global connectivity): 3 x 7 = 21
- 지역 연결성(Local connectivity): 3 x 3 = 9



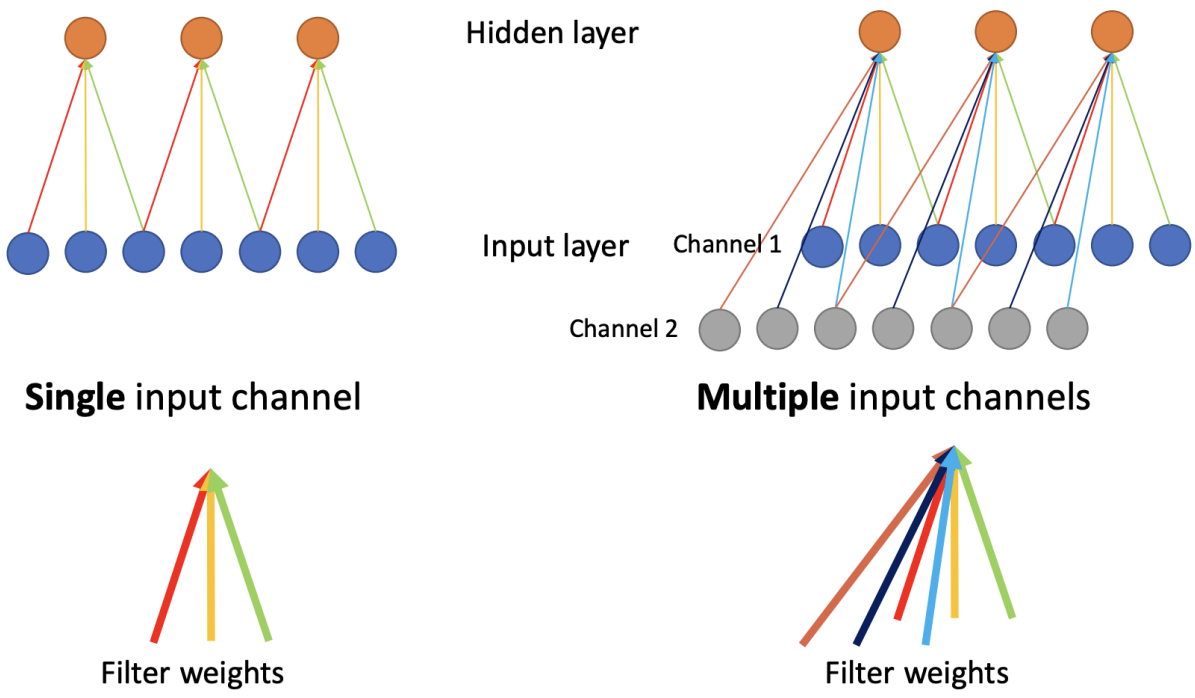
가중치 공유(Weight Sharing):

- **개념**: 같은 필터의 가중치가 입력 이미지의 다른 부분에 적용될 때, 동일한 가중치 값을 재사용
- 입력 유닛 수: 7
- 은닉 유닛 수: 3
- 매개변수 수:
  - 가중치 공유 없음: 3 x 3 = 9
  - 가중치 공유: 3 x 1 = 3
- 가중치 공유는 매개변수의 수를 줄이고, 모델의 일반화 능력을 향상시키는 데 도움을 줍니다.



다중 입력 채널(Multiple Input Channel):

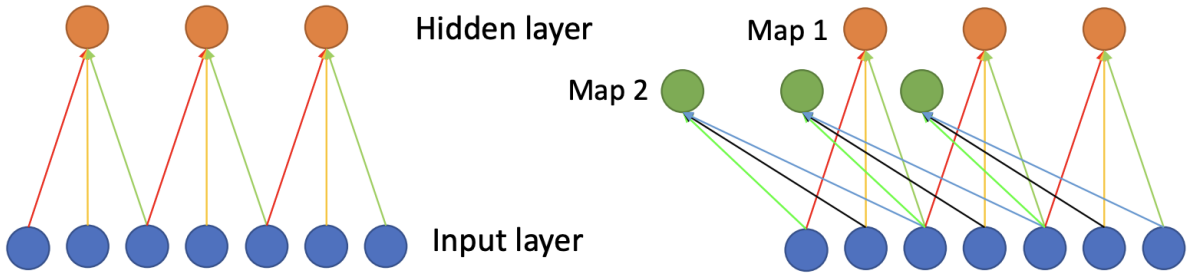
- **다중 채널 데이터 입력**: CNN은 다양한 채널(예: RGB 이미지의 Red, Green, Blue)을 가진 데이터를 입력으로 받을 수 있습니다.
- **가중치 공유**: 각 채널에 대한 가중치를 공유할 수 있으며, 이는 모델의 자유도를 높여줍니다.
- **필터의 필요성**: 각 입력 채널을 처리하기 위해 필터가 필요합니다. 입력 채널의 수가 증가함에 따라 필요한 필터의 수도 증가합니다.
- **결과물 처리**: 각 채널의 동일 위치 요소들을 요소별(element-wise)로 합하여 결과물을 생성합니다.

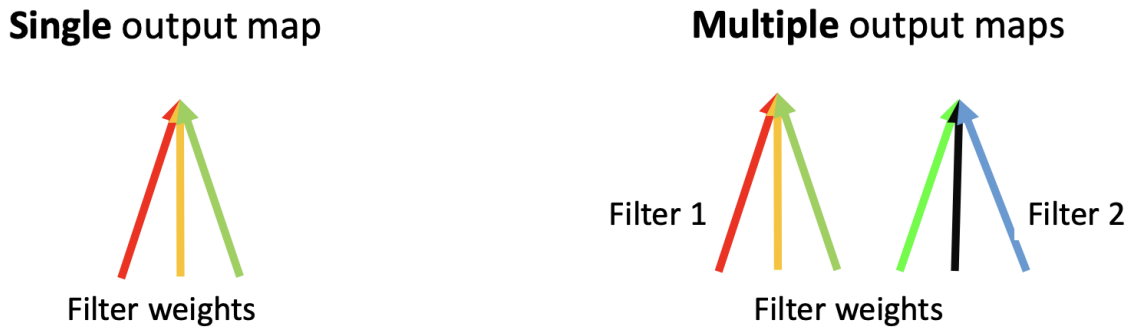


다중 출력 채널(Multiple Output Channel):

- **단일 채널 입력에서 다중 채널 출력**: 하나의 입력 채널로부터 다수의 출력 채널을 생성할 수 있습니다. 예를 들어, 흑백 이미지를 컬러 이미지로 변환하는 등의 작업이 가능합니다.

- **출력 채널 수의 증가**: 출력 채널의 수가 늘어날 경우, 은닉 유닛(hidden unit)의 수도 증가해야 합니다. 이는 모델의 복잡성이 증가한다는 것을 의미합니다.





▼ 참고

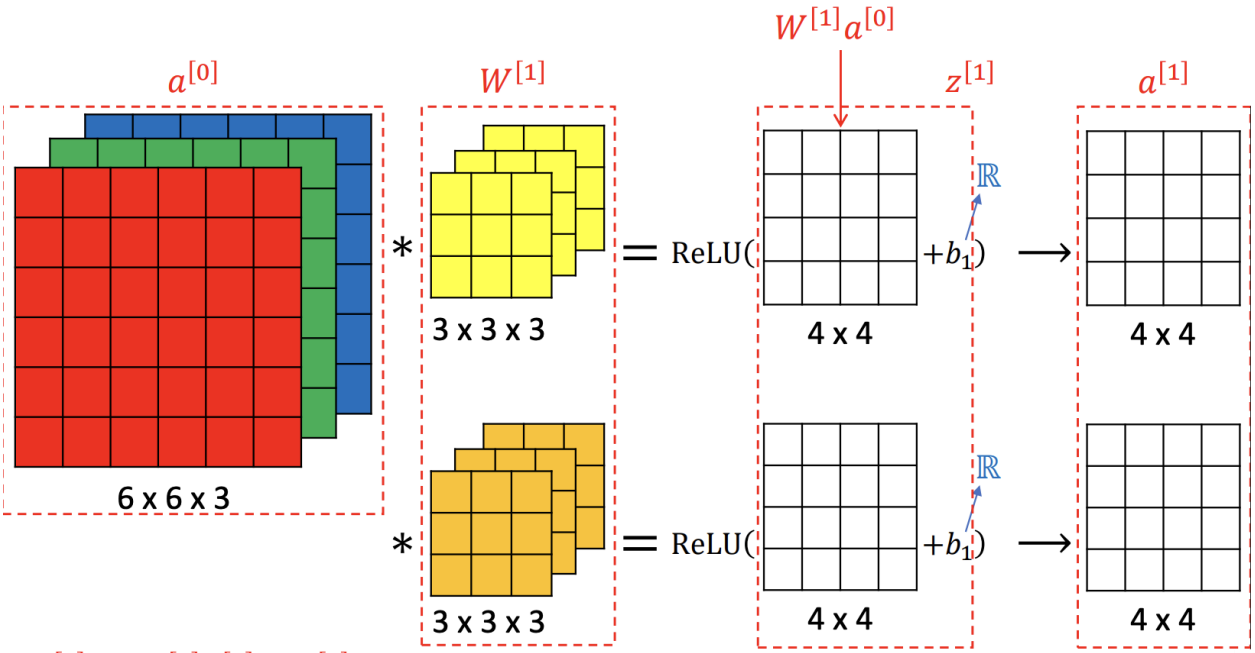
다중 입력 및 출력 채널을 사용하는 CNN은 이미지의 다양한 측면을 더 효과적으로 포착하고 처리할 수 있습니다. 다중 입력 채널은 다양한 유형의 정보를 동시에 처리할 수 있게 해주며, 다중 출력 채널은 데이터의 다차원적인 특징을 학습하고 표현하는 데 유용합니다. 그러나 이러한 구조는 모델의 복잡성을 증가시키며, 이에 따라 더 많은 계산 자원과 데이터가 필요할 수 있습니다.

다중 입력 채널(Multiple Input Channels):

- 단일 입력 채널과 다중 입력 채널 사이의 비교를 보여줍니다.
- 다중 입력 채널은 다양한 유형의 정보(예: RGB 색상 채널)를 동시에 처리할 수 있게 해줍니다.

▼ 참고

이러한 특징들은 CNN이 복잡한 이미지 데이터를 효과적으로 처리하고 특징을 추출하는 데 도움을 줍니다. 지역 연결성은 필요한 매개변수의 수를 줄이면서도 중요한 정보를 포착할 수 있게 해주고, 가중치 공유는 학습해야 할 매개변수의 수를 줄여줍니다. 다중 입력 채널은 이미지의 다양한 측면을 처리할 수 있는 능력을 제공합니다.



$$z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$$

$$a^{[1]} = g\left(z^{[1]}\right)$$

1. **필터와 매개변수의 계산**:
   - **예시**: 한 층에 $3 \times 3 \times 3$ 크기의 필터가 10개 있는 경우
   - 각 필터에는 $3 \times 3 \times 3 = 27$ 개의 가중치가 있습니다.
   - 따라서, 10개의 필터에 대한 총 매개변수 수는 $27 \times 10 = 270$개입니다.
   - 이에 더해, 각 필터에 대해 하나의 편향(bias) 매개변수가 있을 수 있으므로, 총 매개변수 수는 $270 + 10 = 280$ 개가 됩니다.
2. **합성곱 층의 기타 표기법**:
   - F_l : 필터 크기

- P_l : 패딩
- S_l : 스트라이드
- 입력 크기: $H_{l-1} \times W_{l-1} \times C_{l-1}$
- 출력 크기: $H_l \times W_l \times C_l$
- 출력 높이: $H_l = \frac{H_{l-1}+2P_l-F_l}{S_l} + 1$
- 출력 너비: $W_l = \frac{W_{l-1}+2P_l-F_l}{S_l} + 1$
- 출력 채널 수: C_l 은 필터의 수와 동일

3. **활성화 함수(Activations)**:
    - **ReLU(Rectified Linear Unit)**: 음수 값을 0으로 만들고, 양수 값은 그대로 유지하는 활성화 함수입니다.
    - 이는 비선형성을 도입하여 신경망이 더 복잡한 패턴을 학습할 수 있게 도와줍니다.

Why Convolution?

1. Sparse Interactions (Sparse Weight)

정의

- **Sparse Interactions**은 각 뉴런이 입력 데이터의 전체 영역이 아니라 일부 영역에만 연결되는 것을 의미합니다.
- 전통적인 완전 연결 네트워크(Fully Connected Network)와는 달리, CNN에서는 각 뉴런이 입력의 작은 부분집합과만 상호작용합니다.

특징 및 장점

- **계산 효율성**: 모든 입력과 출력 뉴런 간에 연결이 존재하지 않기 때문에 계산량이 줄어듭니다.
- **지역적 특징 포착**: 국소적인 입력 패턴에 집중할 수 있으며, 이는 이미지와 같은 데이터에서 중요한 시각적 패턴을 효과적으로 감지하는 데 도움이 됩니다.

- **과적합 감소**: 더 적은 수의 매개변수를 사용함으로써 과적합 위험을 줄일 수 있습니다.

2. Parameter Sharing (Tied Weight)

정의

- **Parameter Sharing**은 같은 가중치(파라미터)를 네트워크의 여러 부분에 걸쳐 공유하는 것을 말한다.
- CNN에서는 동일한 필터(커널)가 전체 입력 이미지에 걸쳐 재사용되어, 같은 종류의 특징을 여러 위치에서 감지할 수 있습니다.

특징 및 장점

- **매개변수 수 감소**: 한 세트의 파라미터를 여러 위치에서 재사용함으로써 전체적으로 필요한 파라미터 수가 줄어듭니다.(메모리의 효율성 증가)
- **공간적 불변성 학습**: 필터가 이미지의 다른 위치에서 유사한 특징을 감지할 수 있으므로, 위치에 상관없이 객체를 인식할 수 있습니다.
- **일반화 능력 향상**: 파라미터 공유는 모델이 특정 위치에 과도하게 의존하지 않도록 하여, 일반화 성능을 개선합니다.

3. Equivariant Representations

정의

- **Equivariant Representations**는 입력 데이터에 특정 변환(예: 이동, 회전)이 적용될 때, 출력도 동일한 방식으로 변화하는 특성을 의미한다.
- CNN에서는 이동에 대해 이러한 동등성(equivariance)을 갖는다.

특징 및 장점

- **변환에 대한 강인성**: 입력 이미지가 이동, 회전 등으로 변형되더라도, CNN은 해당 변환에 따라 특징을 일관되게 감지할 수 있습니다.
- **효과적인 특징 학습**: 입력의 변환에 따라 특징의 표현이 동일하게 변화함으로써, 더 효과적으로 학습이 이루어질 수 있습니다.

- **데이터 확장(Data Augmentation)에 대한 자연스러운 적응**: 이동 불변성은 CNN이 이미지의 위치 변화에 민감하지 않게 만들어, 데이터 확장 기법이 효과적으로 적용될 수 있도록 합니다.