**What is the meaning of Conv1D, Conv2D and Conv3D?**

**1. Conv1D**

**1-1. overview**: Keras에서는 Conv1D, Conv2D 그리고 Conv3D의 기능을 제공한다. Conv1D는 1-dimension의 Convolutional Layer을 의미하고 Conv2D는 2-dimension의 Convolution layer, 그리고 Conv3D는 3-dimension의 Convolution layer를 의미한다.

이 중 Conv1D는 전체 데이터 세트의 고정 길이 세그먼트에서 특징의 위치가 별로 중요하지 않을 때, 그 특징을 도출하는데 매우 효과적이다. (A 1D CNN is very effective for deriving features from a fixed-length segment of the overall dataset, where it is not so important where the feature is located in the segment.)

따라서 Conv1D는 주로 아래와 같은 3가지 목적으로 사용되게 된다.

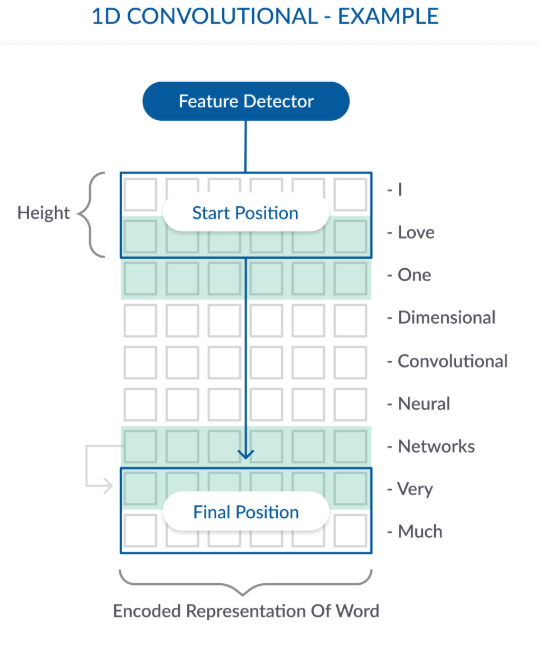
1) Analysis of a time series of sensor data

2) Analysis of signal data over a fixed-length period, for example, an audio recording

3) Natural Language Processing (NLP), although Recurrent Neural Network which leverage Long Short-Term Memory(LSTM) cells are more promising than CNN as they take into account the proximity of words to create trainable patterns

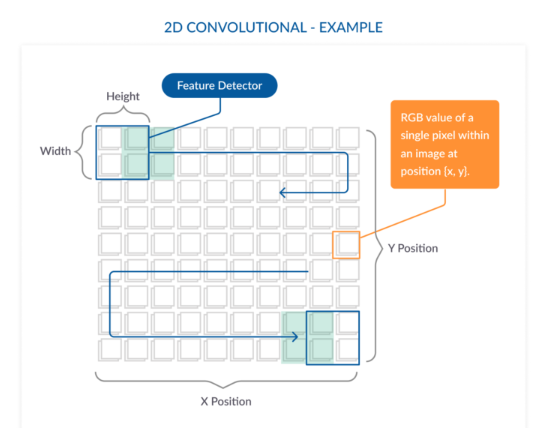
**1-2. What is the Difference Between a 1D CNN and a 2D CNN?**

CNN 알고리즘의 작동 방식은 1D든 2D든 3D이든 모두 같다. 하지만 차이점이 있다면 바로 입력 데이터의 구조와 필터가 데이터 상에서 어떻게 움직이느냐이다. (CNNs work the same way whether they have 1, 2, or 3 dimensions. The difference is the structure of the input data and how the filter, also known as a convolution kernel or feature detector, moves across the data.)



위 그림은 전형적인 Conv1D의 모습을 보여주고 있다. 특히 위의 그림은 NLP에서의 Conv1D를 보여주고 있으며 전체 문장은 “I Love One Dimensional Convolutional Neural Networks Very Much” 이다. Conv1D는 자연어 처리에 좀 더 적합한 모습을 보이며, 가로와 세로 방향 둘 다 움직이는 것이 아닌 한 방향으로 움직이게 된다.

다음 그림은 전형적인 Conv2D의 모습을 보여준다.

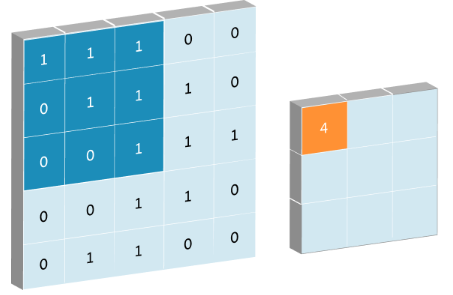


대략적으로 볼 수 있듯이 Conv2D에서는 X 위치와 Y 위치를 고려할 수 있으며 이미지의 RGB채널을 고려할 수 있다. 따라서 Conv2D는 이미지 데이터를 처리하는 것에 좀 더 특화되어 있다는 것을 알 수가 있다. Conv2D에서는 필터가 세로, 가로 두 방향 모두 움직일 수 있다.

이러한 특징 말로도 Conv1D와 Conv2D의 차이로는 Conv1D Networks는 좀 더 큰 필터 사이즈를 사용할 수 있다는 점이다. 예를 들자면 1D Network에서는 7 사이즈의 필터는 오로지 7개의 특징 벡터를 가지며 9 사이즈의 필터는 오로지 9개의 특징 벡터를 가진다. 하지만 2D Network에서는 7 사이지의 필터는 49개의 특징 벡터를 가지며 이것은 매우 큰 선택 범위를 주게 된다. 간단히 말하자면 1D에서는 N 사이즈 필터는 N개의 특징 벡터를 가지지만, 2D에서는 N 사이즈 필터는 N x N 개의 특징 벡터를 가진다는 것이다. 따라서 필터 사이즈가 커질수록 계산 속도 및 필요한 컴퓨팅 파워에서 차이가 생기게 된다. 이러한 특징은 Convolution windows도 마찬가지여서 1D에서는 2D보다는 더 큰 사이즈의 Convolution windows를 사용할 수 있다.

**2. Conv2D**

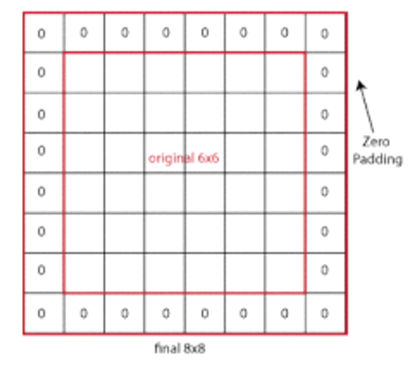
**2-1. Overview:** 입력 데이터를 3차원으로 받는 Convolutional Layers로서 X와 Y 그리고 채널(보통 RGB) 정보를 입력으로 받는다. 하나의 이미지 데이터를 모두 스캔할 할 때까지 지정한 사이즈의 Convolution Kernel(filter)가 이미지 상에서 움직이게 된다. 보통 Convolutional Layer를 애기할 때는 이 Conv2D를 가리키는 말이다. Convolution operation은 현재 필터 윈도우(아래 그림에서 찐한 파란색)가 위치한 곳의 픽셀 값들과 필터가 가지고 있는 가중치 값을 서로 dot product 하여 계산하게 된다.



이 Convolution Kernel은 Fine tuning 된 weight 값이 가지고 움직이며 이를 통해 하나의 결과 값을 만들어 낼 수 있는 것이다. 여기서 Output size(위 그림에서 오른쪽)는 [(입력 크기 – 필터 크기)/Stride] +1을 통해서 구할 수 있다. 여기서 Stride는 Filter window가 한번 움직일 때마다 얼만큼 움직일 것인가를 나타낸다. 예를 들어 만약 입력 사이즈가 7이고 필터 사이즈가 3, Stride의 크기가 2이면 출력 사이즈는 (7-3)/2 + 1 = 3이 되게 된다. 위 그림에서는 입력크기가 5이고 필터 사이즈가 3이고 출력 사이즈가 3이므로 Stride의 크기는 1이라는 것을 알 수가 있다.

\*Zero Pad

위의 그림에서도 볼 수 있듯이 Convolution Layer를 통과하면 이미지의 사이즈가 작아진다는 사실을 알 수가 있다. 하지만 만약, 이미지의 크기에 변형을 가하고 싶지 않다면 zero padding을 하면 된다. Zero padding은 입력 데이터를 감싸는 0으로 된 벽을 만드는 것을 의미한다. 이 방법을 사용하면 앞서 말한 대로 입력 데이터와 출력 데이터의 크기를 같게 해주는 것 외에도 입력 데이터의 외각 부분 (입력 데이터의 가장 끝 부분)에 대한 Feature Extraction도 강화되는 효과를 얻을 수가 있다. 다시 말해 이미지 외곽의 특징을 잘 잡아 낼 수 있다는 의미이다. 아래 그림은 Zero padding의 한 예시로 zero padding의 크기는 1인 경우이다.

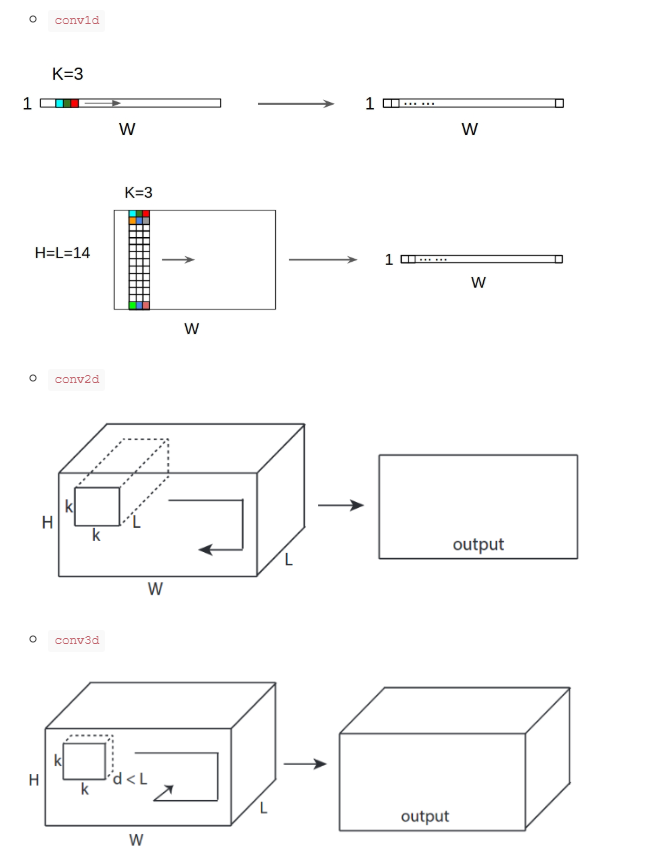


또한 필터는 채널의 수만큼 이미지 위에서 움직이게 되는데 만약 채널 수가 1이라면 (회색으로 바꿨으면) 1번만 움직이게 되지만 만약 채널 수가 3이라면(RGB) 총 3번 이미지 위에서 움직이게 된다. 컨벌루션이 끝나면 피쳐가 다운 샘플링되고 동일한 컨벌루션 구조가 다시 반복됩니다. 처음에 컨볼 루션은 원본 이미지 (예 : 고양이, 몸, 다리, 꼬리, 머리)의 특징을 식별 한 다음 이미지의 작은 부분 (예 : 머리, 귀)의 하위 기능을 식별합니다. , 수염, 눈). 결국이 프로세스는 이미지를 분류하는 데 도움이되는 필수 기능을 식별하기위한 것입니다. (After the convolution ends, the features are downsampled, and then the same convolutional structure repeats again. At first, the convolution identifies features in the original image (for example in a cat, the body, legs, tail, head), then it identifies sub-features within smaller parts of the image (for example, within the head, the ears, whiskers, eyes). Eventually, this process is meant to identify the essential features that can help classify the image.)

**3. Conv3D**

Conv3D는 거의 사용하지 않는 Convolutional Layer로서 Conv3D가 다른 Conv1D와 Conv2D와 가지는 차이점은 아래 와 같다.





따라서 목적에 맞게 1D와 2D 그리고 3D를 선택하면 된다. 1D와 2D를 섞는 경우도 있는 그 경우에는 1D를 사용하면 특징 벡터의 수를 줄일 수 있기 때문에 학습 속도를 증가 시키는 하나의 테크닉으로 사용되는 경우가 더러 있다.