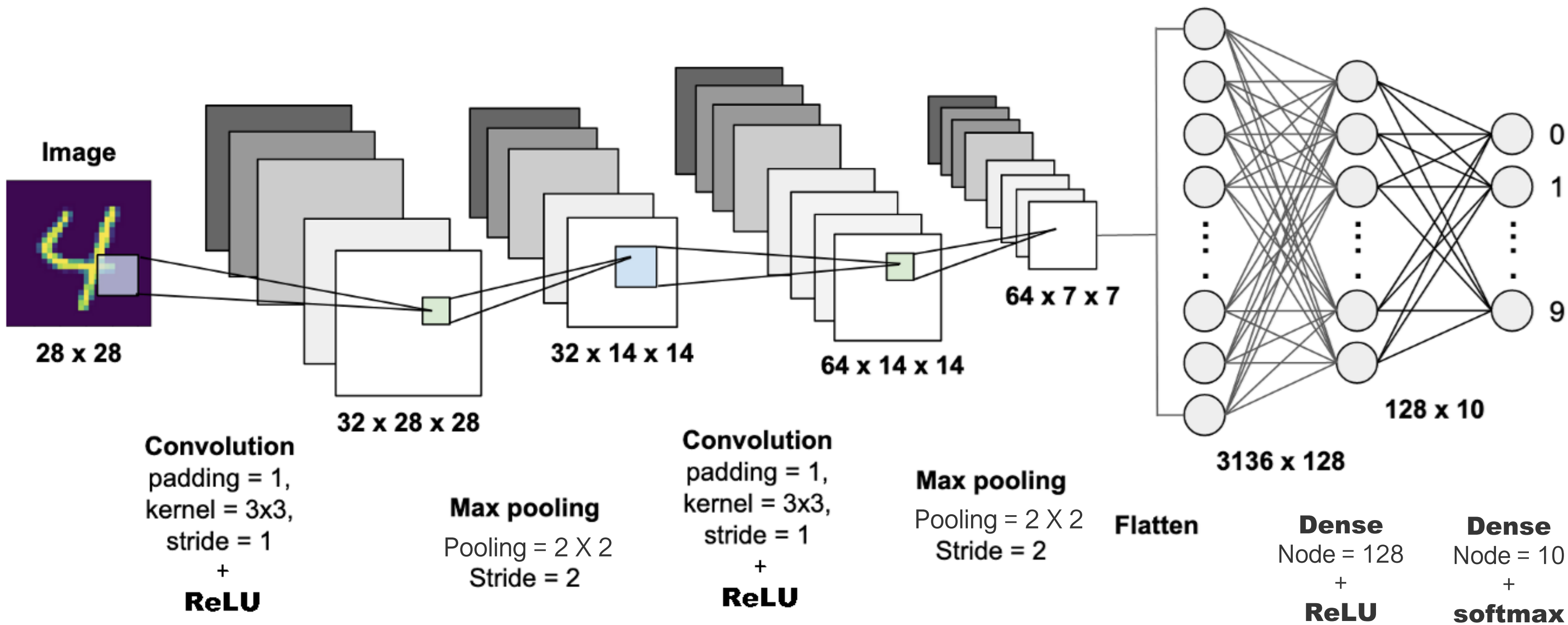
Time Series Anomaly Detection
CNN + AE

- CNN의 모든 내용을 설명하지는 않습니다.
- CNN의 기본 개념과 Conv1D에 대해서 간략하게 살펴보고, CNN+AE를 이용한 이상탐지를 수행해 봅니다.

Convolutional Neural Network

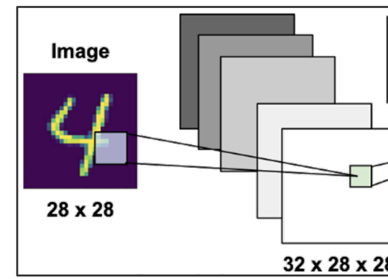


ConvNet①

```
Conv2D(32, kernel_size=(3, 3), input_shape=(5, 5, 1),
       padding='same', strides = (1,1), activation='relu'),
```

✓ **ConvNet(특히 Conv2D) : 이미지 분석에 주로 사용**

- 데이터에 담겨 있는 지역적(Local) 특징을 추출
- 필터(커널) : 개수 32개
  - 2차원으로 이동하며 Feature Map 구성



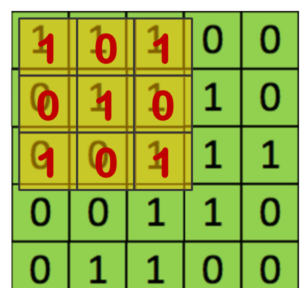
Input(Image)
(5, 5, 1)

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Filter(kernel)
(3, 3)

1	0	1
0	1	0
1	0	1

Convolution(합성곱)



$$(1\times1)+(1\times0)+(1\times1)$$
$$+(0\times0)+(1\times1)+(1\times0)$$
$$+(0\times1)+(0\times0)+(1\times1)$$

Convolved
Feature Map

4		

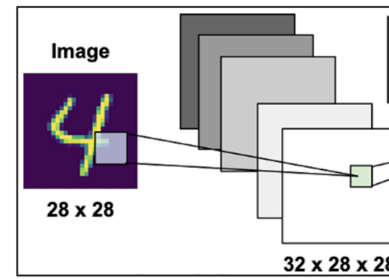
ConvNet②

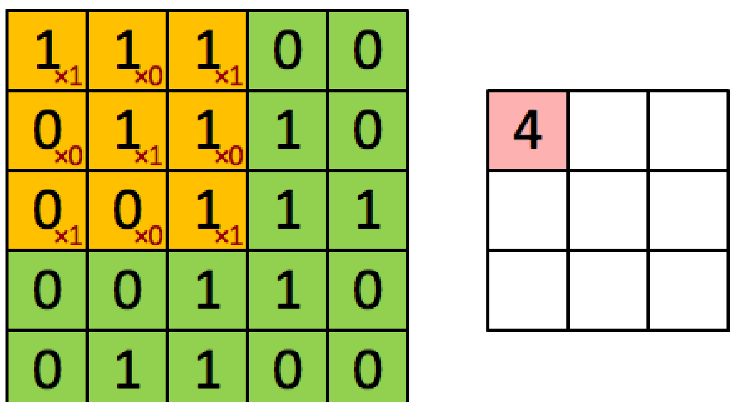
```
Conv2D(32, kernel_size=(3, 3), input_shape=(5, 5, 1),
         padding='same', strides = (1,1), activation='relu'),
```

✓ **Stride : 몇 칸 씩 이동할 것인지 지정**

✓ **Size 축소 5 X 5 ➔ 3 X 3**

- 만약 Size를 유지하려면 padding 옵션사용





Image

Convolved Feature

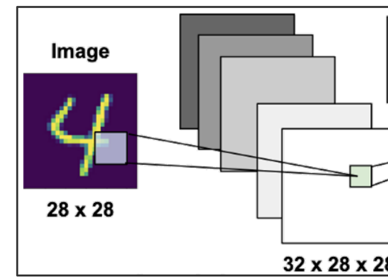
http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

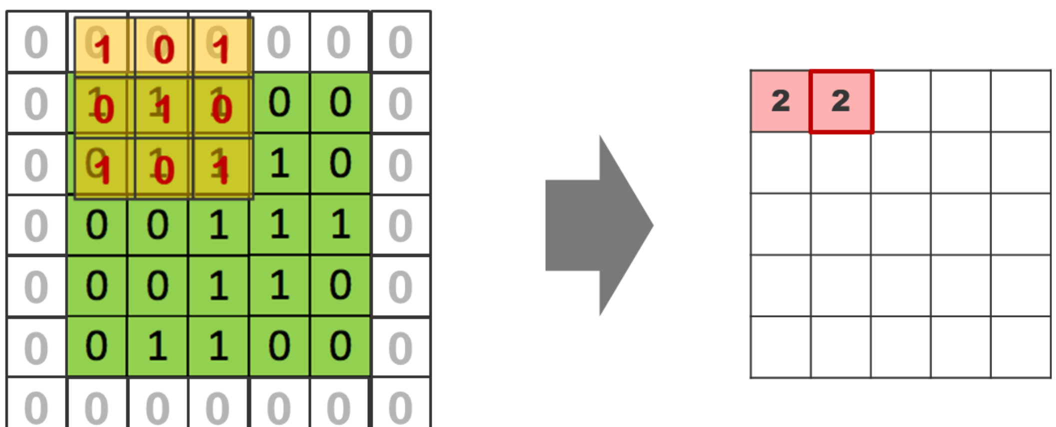
ConvNet③

```
Conv2D(32, kernel_size=(3, 3), input_shape=(5, 5, 1),
       padding='same', strides = (1,1), activation='relu'),
```

✓ Padding

- Size 유지되도록 이미지 둘레에 0으로 덧대기(padding!)
- 5 X 5 ➔ 5 X 5
- padding = ‘same’



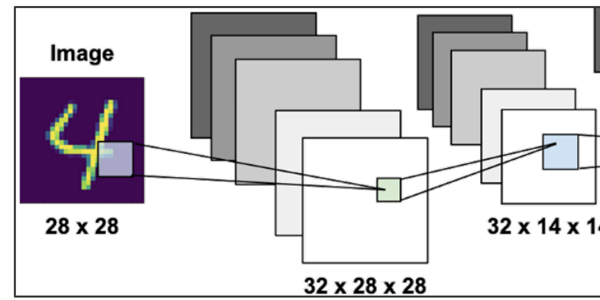


MaxPooling

```
MaxPooling2D(pool_size=(2, 2), strides=(2, 2)),
```

✓ 출력데이터(Feature Map)의 크기를 줄이거나 특정 데이터를 강조하기 위해 사용

- pooling_size : 풀링 크기 행 x 열
- strides : 옆, 아래 몇 칸 씩 이동할지
- 출력데이터크기 : Input Size // Pooling Size (나머지는 버림)



Feature Map

8	12	1	8
5	4	21	0
26	6	7	15
2	21	1	2

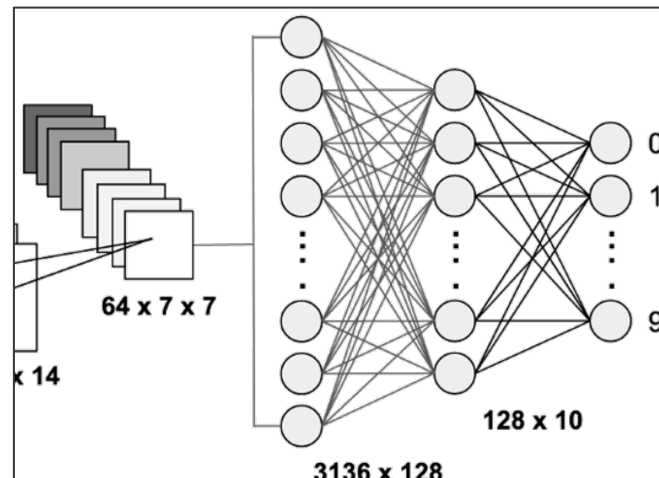
Max Pooling

12	21
26	15

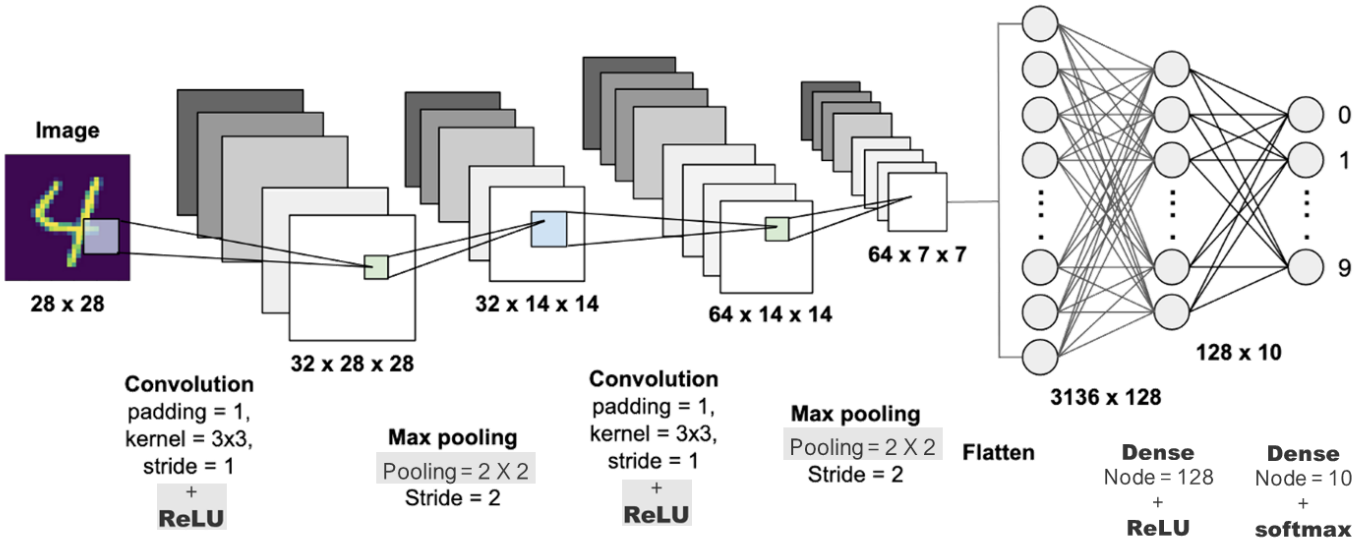
Flatten과 Dense Layer

✓ CNN + MaxPooling Layer로 특징을 추출한 후에, (3차원)
✓ 최종 예측 결과로 뽑기 위해서는 (1차원 혹은 단일 값)
✓ Dense Layer로 연결해야 합니다.

```
Flatten(),
Dense(128, activation='relu'),
Dense(10, activation='softmax')
```



Convolutional Neural Network-Review

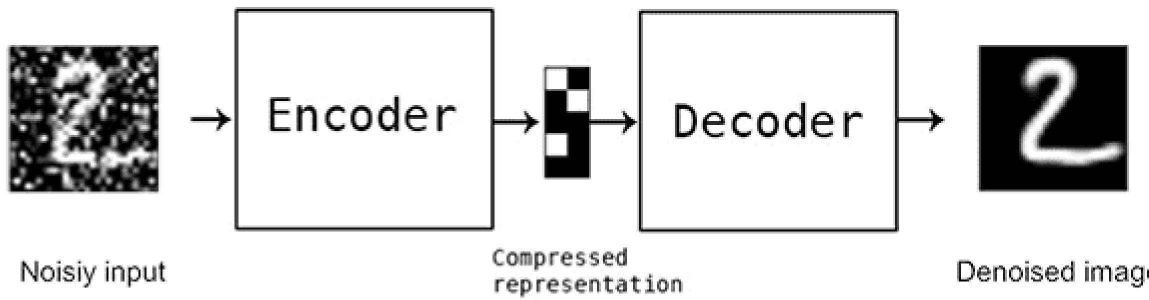



```
model = Sequential([
    Conv2D(32, kernel_size=(3, 3), input_shape=(28, 28, 1),
           padding='same', strides = (1,1), activation='relu'),
    MaxPooling2D(pool_size=(2, 2), strides=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), padding='same', activation='relu'),
    MaxPooling2D(pool_size=(2, 2), strides=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 128)	401536
dense_1 (Dense)	(None, 10)	1290

CNN + AE for Denoising

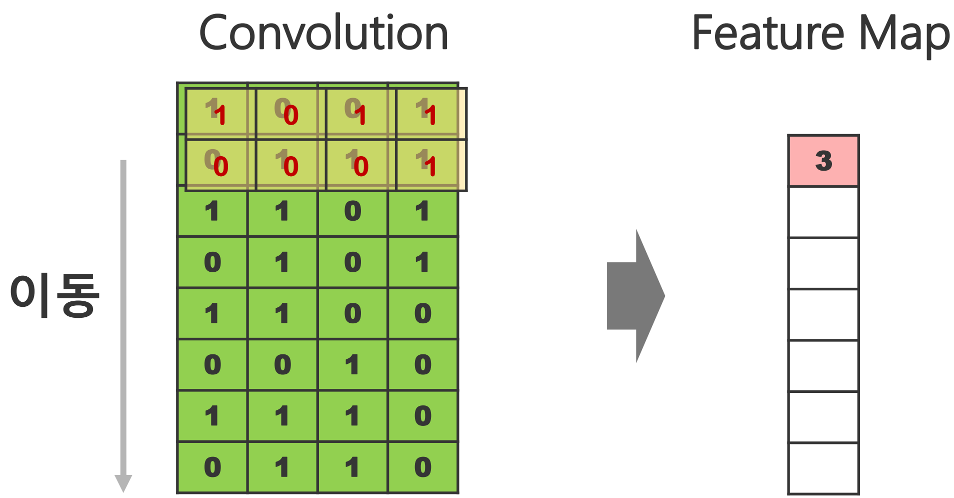
✓ CNN을 AE로 만들어 이미지의 노이즈를 제거하는 용도로 사용됩니다.



1D-Conv Net (Conv1D)

✓ 지역적 특징을 1차원 관점에서 추출

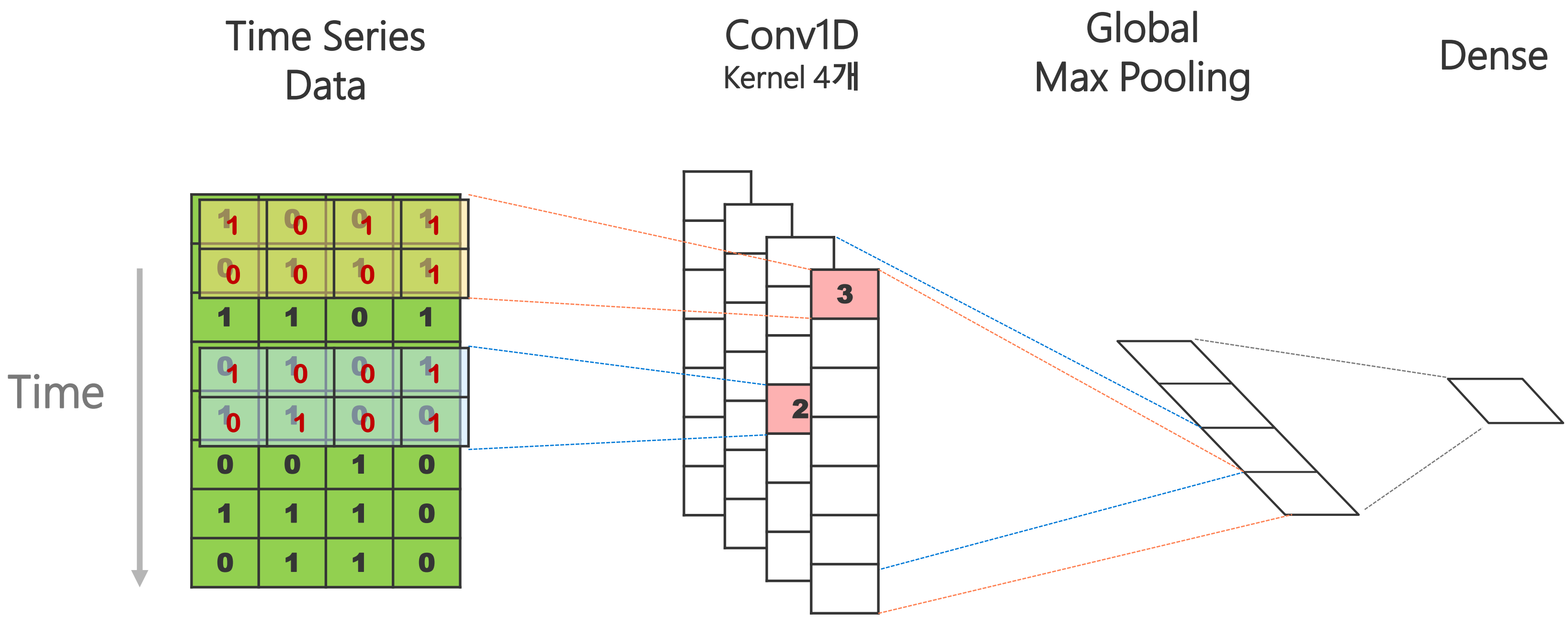
- 필터가 1차원으로 이동하며 Feature Map 구성
- 시퀀스 분석에 사용.



1D-Conv Net for Time Series

✓ **LSTM : 과거에서 불필요한 것은 잊고, 최근 정보를 더 중요하게**

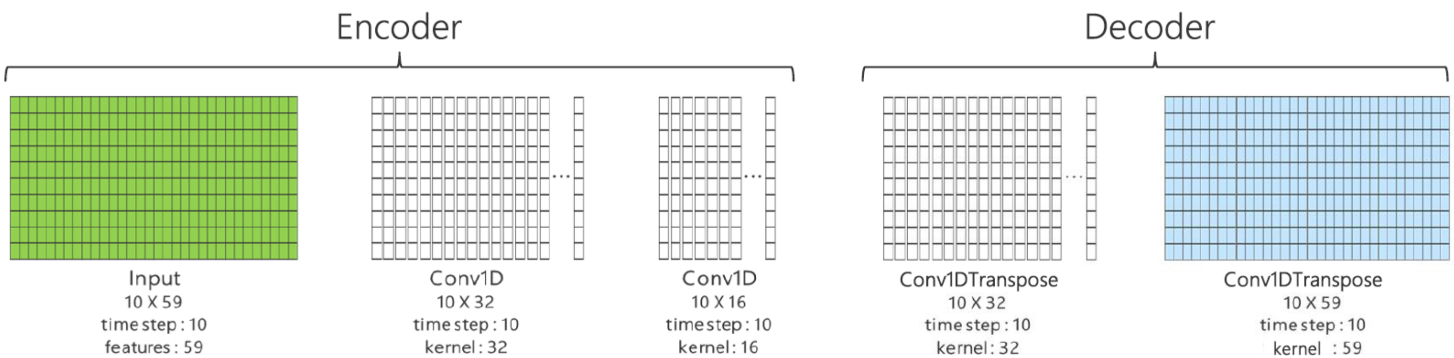
✓ **1D-Conv : 시간 흐름별로 중요한 정보 요약, 평활화.**



1D-Conv Net + AE for Anomaly Detection

✓ 코드 구성

- Conv1D의 filter 수를 32 → 16 → 32 로 줄였다가 늘려가면서 구성
- padding = 'same' : timesteps 유지하기 위해.
- Conv1DTranspose : deconvolution




```
# Encoder
input_layer = Input(shape=(timesteps, n_features))
encoder = Conv1D(32, 3, activation='relu', padding = 'same')(input_layer)
encoder = Conv1D(16, 3, activation="relu", padding = 'same')(encoder)

# Decoder
decoder = Conv1DTranspose(32, 3, activation="relu", padding = 'same')(encoder)
decoder = Conv1DTranspose(n_features, 3, padding = 'same')(decoder)

conv1d_ae2 = Model(inputs=input_layer, outputs=decoder)
conv1d_ae2.compile(loss='mse', optimizer='adam')
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 10, 59)]          0

conv1d (Conv1D)              (None, 10, 32)            5696

conv1d_1 (Conv1D)            (None, 10, 16)            1552

conv1d_transpose (Conv1DTra  (None, 10, 32)            1568
nspose)

conv1d_transpose_1 (Conv1DT  (None, 10, 59)            5723
ranspose)

=================================================================
Total params: 14,539
Trainable params: 14,539
Non-trainable params: 0
```

Conv#DTranspose : deconvolution

- ✓ **ConvNet을 통해 압축해 갔던 feature map의 크기를 다시 늘려 줌**
  - 압축된 feature map의 각 cell 주의로 zero padding 추가
  - Filter를 통해 convolution 수행
- ✓ **Convolution Layer의 결과물을 반대로(원래 크기로) 돌려 놓을 때 사용.**