GSDPI

CONVOLUTIONAL LAYERS with Keras

Conv1D: Input and output shapes

Convolution operation

[1]

$$\mathbf{x}_l^{(m)} = \sigma\left(\sum_{c=1}^{C} \mathbf{W}_l^{(c,m)} * \mathbf{x}_{l-1}^{(c)} + \mathbf{b}_l^{(m)}\right)$$

In this equation we see that the convolution ($*$) of the channel c of the input $\mathbf{x}_{l-1}^{(c)}$ and the m^{th} filter of such channel $\mathbf{W}_l^{(c,m)}$ results in the m^{th} output feature map $\mathbf{x}_l^{(m)}$, being $\mathbf{b}_l^{(m)}$ the bias vector.

We perform the convolution operation by means of the Keras Conv1D layer, being c = channels and m = filters.

Conv1D layer

Input

3D tensor with shape
(batch, steps, channels)

Output

3D tensor with shape
(batch, new_steps, filters)

Example

```
Conv1D(filters=4, kernel_size=600, padding='valid')
```

input and output shapes

The batch dimension is the number of samples in the dataset. It is denoted as None, as it is not fixed.

Each input sample is a 3 channel window of size 800 steps.

Conv1D	input:	(None,800,3)
	output:	(None,201,4)

The layer returns 4 filters

The output step value may change due to padding or strides. In this case, we use: no padding ('valid'), unit strides and kernel of size 600.
So, new_steps = (800 - 600) + 1 = 201. [2]

convolution kernels

The Conv1D layer learns 12 (c·m = 12) kernels and returns 4 filters. Each output filter is the average of its 3 kernels.

[1] González-Muñiz, A., Díaz, I., & Cuadrado, A. A. (2020). DCNN for condition monitoring and fault detection in rotating machines and its contribution to the understanding of machine nature. *Heliyon, 6*(2), e03395.
[2] Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.