Convolution and Pooling on 1D data for Time Series

Say, we have a time series of length *n* and width *k*, i.e. with 'n' number of timesteps, and 'k' number of variables in a multivariate time series. Width of the convolution kernels will always be the same as the time series, while the length can be varied as per requirement. This is why the kernel performs convolution moving in one direction from the beginning to its end. It does not move to the left or to the right, unlike 2-D convolution that is applied to images. Each element of the kernel gets multiplied to the corresponding element of the time series that the kernel covers at a given point. Then the multiplication results are added together and a nonlinear activation function is applied to the value. The resulting value becomes an element of a new filtered time series, and then the kernel moves forward along the time series to produce the next value. This filtered time series will be 'univariate'. The number of such new filtered time series will be equal to the number of convolution kernels. Depending on the length of the kernel, different properties or features of the initial time series get captured in each of the new filtered series.

Next, pooling is applied to each of the filtered time series vectors. In case of Max-Pooling, the largest value is taken from each vector. A new vector is formed from these values, and this vector of maximum values is the final feature vector that can be used as an input to a regular fully connected layer.