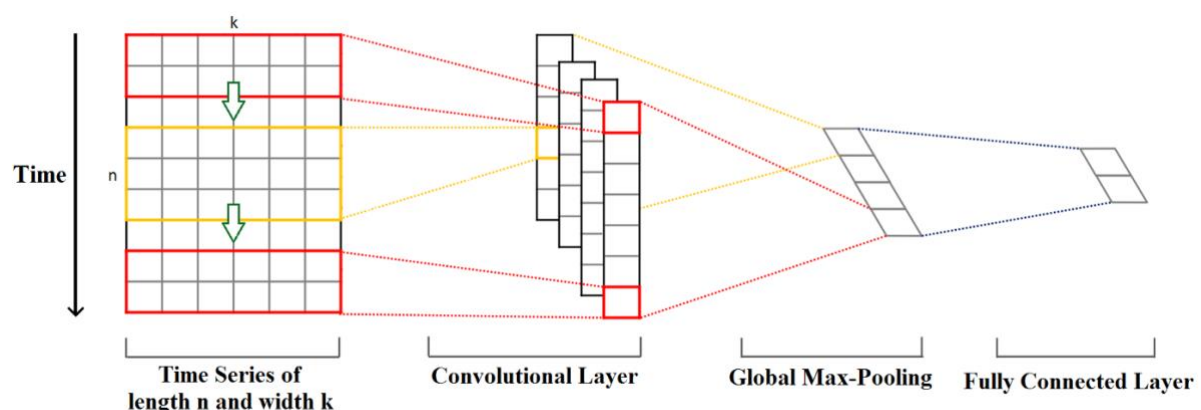


## Convolution and Pooling on 1D data for Time Series

There are many methods for time series classification. Most of them consist of two major stages: on the first stage you either use some algorithm for measuring the difference between time series that you want to classify.

Imagine a time series of length  $n$  and width  $k$ . The length is the number of timesteps, and the width is the number of variables in a multivariate time series. For example, for electroencephalography it is the number of channels (nodes on the head of a person), and for a weather time series it can be such variables as temperature, pressure, humidity etc.

The convolution kernels always have the same width as the time series, while their length can be varied. This way, the kernel moves in one direction from the beginning of a time series towards its end, performing convolution. It does not move to the left or to the right as it does when the usual 2-D convolution is applied to images.



The elements of the kernel get multiplied by the corresponding elements of the time series that they cover at a given point. Then the results of the multiplication are added together and a nonlinear activation function is applied to the value. The resulting value becomes an element of a new “filtered” univariate time series, and then the kernel moves forward along the time series to produce the next value.

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.