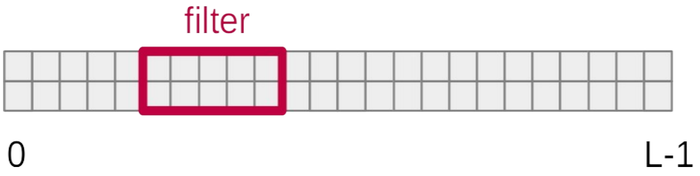
LCPB 21-22 exercise 3 (Convolutional Neural Network, CNN)

Chose if you prefer to work with the data generated during the lesson (and skip point 1) or to use new data.

1. *(Optional)* Set up data with two times series in parallel. The samples are now *L* by 2 dimensional. The kernels of the CNN will be *M* by 2 dimensional, to scan in parallel the two time series.

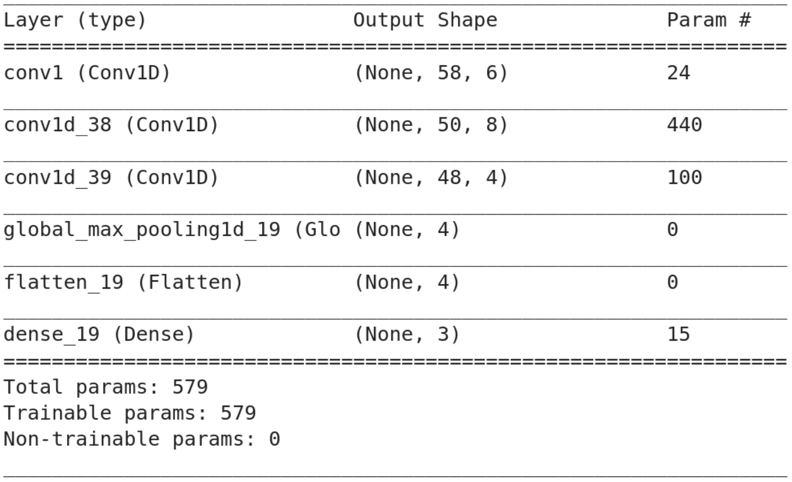


   Similarly to the lesson, introduce a pattern that affects both time series around the same position in the sample. For example, one can introduce a "sin"-like bump in series 1 followed by another one in series 2. The number of categories can also be increased depending on the chosen combinations. Describe your choice at the beginning of the jupyter.

2. Study the performance of the CNN by varying the amplitude *A* of the patterns while keeping *DX* fixed, namely by changing the signal-to-noise ratio.

3. Check if regularization may improve the performances by varying the parameter lambda (as usual in magnitude: 0, 10^{-5}, 10^{-4}, 10^{-3}, etc.) of the L1 (LASSO) or of the L2 (Ridge) regularization; see lambda in eqs.(43) and (52) in the review. There is also a mixed version (l1_l2) that can be tried.
   3.1 Are performances of the CNN are optimized at some intermediate value of lambda?
   3.2 Is there any improvement in the visualization and understanding of the weights in the filters?
   Note that the regularization we introduced acts on the w's, not on the biases. One can also try the equivalent procedure for biases or for the output of the relu units (see Keras doc.), if there is any reason for suspecting that it may help. In our case, the logic was to let the weights of the filters go to zero if not needed, hence that kind of regularization was selected.

4. *(Optional)*
   Try a CNN modeled around the best one of 2020-2021 year: three convolutional layers, no pooling, and the dense layer of the lesson replaced by a global max pooling. Is it working better? If yes, which could be the reason?



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1 (Conv1D)               (None, 58, 6)             24
_________________________________________________________________
conv1d_38 (Conv1D)           (None, 50, 8)             440
_________________________________________________________________
conv1d_39 (Conv1D)           (None, 48, 4)             100
_________________________________________________________________
global_max_pooling1d_19 (Glo (None, 4)                 0
_________________________________________________________________
flatten_19 (Flatten)         (None, 4)                 0
_________________________________________________________________
dense_19 (Dense)             (None, 3)                 15
=================================================================
Total params: 579
Trainable params: 579
Non-trainable params: 0
_________________________________________________________________
```