An Introduction to Statistical Machine Learning - Parameter Sharing -

Samy Bengio

bengio@idiap.ch

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)

CP 592, rue du Simplon 4

1920 Martigny, Switzerland

http://www.idiap.ch/~bengio

Parameter Sharing

1. Introduction
2. Time Delay Neural Networks
3. LeNet for Images
4. Other Parameter Sharing Techniques

Why Should We Share Parameters?

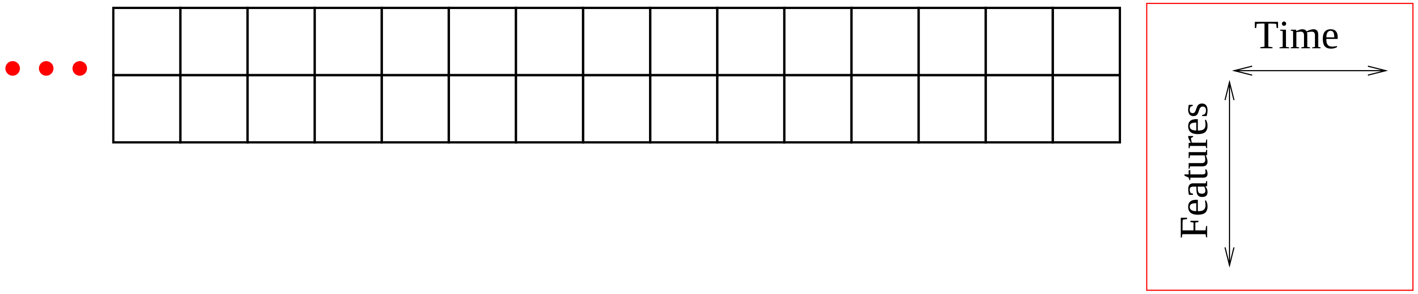
- There are various reasons why we could be interested in sharing parameters:
  - Reduces the number of parameters, hence controls the capacity
  - Searches for regularities, hence introduces prior knowledge

Time Delay Neural Networks (TDNNs)

- TDNNs are models to analyze sequences or time series.
- Hypothesis: some regularities exist over time.
- The same pattern can be seen many times during the same time series (or even over many times series).
- First idea: attribute one hidden unit to model each pattern
  - These hidden units should have associated parameters which are the same over time
  - Hence, the hidden unit associated to a given pattern p_i at time t will share the same parameters as the hidden unit associated to the same pattern p_i at time $t + k$.
- Note that we are also going to learn what are the patterns!

TDNNs: Convolutions

- How to formalize this first idea? using a convolution operator.
- This operator can be used not only between the input and the first hidden layer, but between any hidden layers.



Convolutions: Equations

- Let $s^l_{t,i}$ be the input value at time t of unit i of layer l.
  Let $y^l_{t,i}$ be the output value at time t of unit i of layer l.
  (inputs values: $y^0_{t,i} = x_{t,i}$).
  Let $w^l_{i,j,k}$ be the weight between unit i of layer l at any time t and unit j of layer l at time $t-k$.
  Let b^l_i be the bias of unit i at layer l.

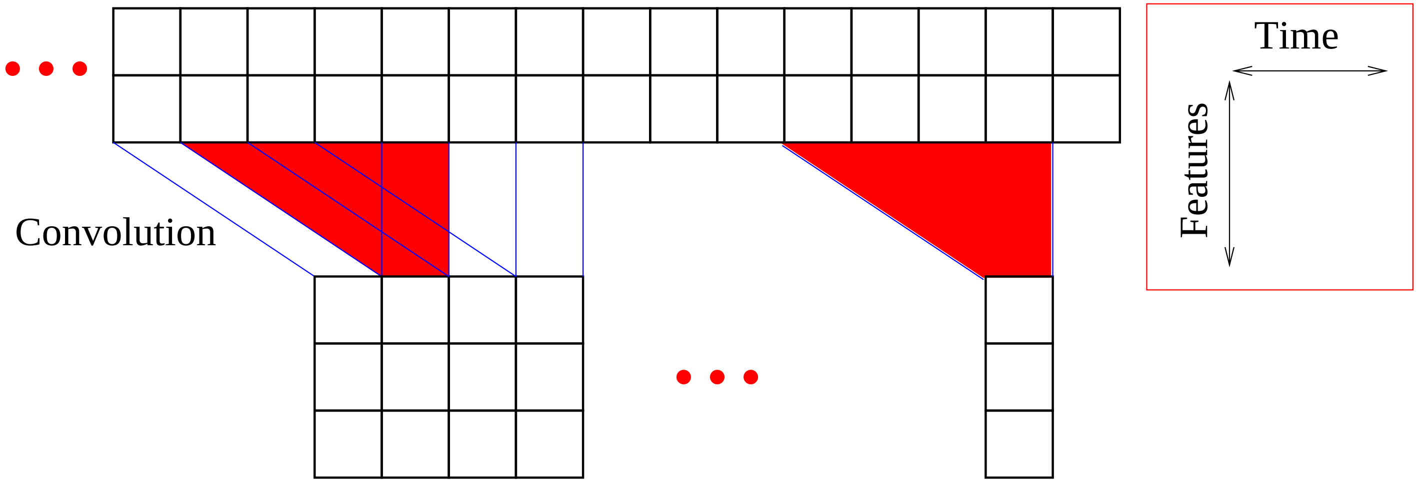
- Convolution operator for windows of size K:

$$s^l_{t,i} = \sum_{k=0}^{K-1} \sum_j w^l_{i,j,k} \cdot y^{l-1}_{t-k,j} + b^l_i$$

- Transfer:

$$y^l_{t,i} = \tanh(s^l_{t,i})$$

Convolutions (Graphical View)



- Note: weights $w^l_{i,j,k}$ and biases b^l_i do not depend on time.
- Hence the number of parameters of such model is independent of the length of the time series.
- Each unit $s^l_{t,i}$ represents the value of the same function at each time step.

TDNNs: Subsampling

- The convolution functions always work with a fixed size window (K in our case, which can be different for each unit/layer).
- Some regularities might exist at different granularities.
- Hence, second idea: subsampling (it is more a kind of smoothing operator in fact).
  - In between each convolution layer, let us add a subsampling layer.
  - This subsampling layer provides a way to analyze the time series at a coarser level.

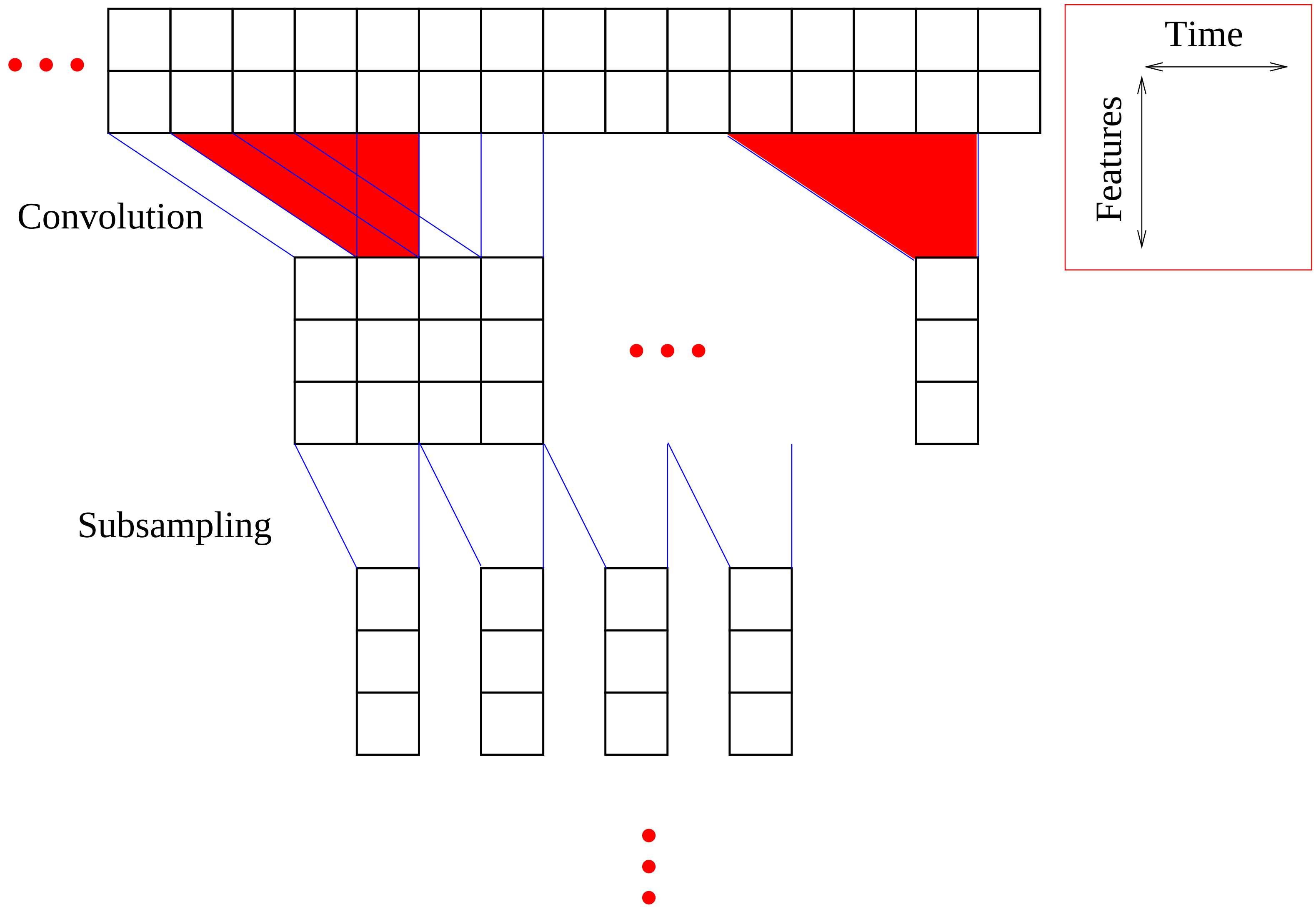
Subsampling: Equations

- How to formalize this second idea?
- Let $y_{t,i}^{l}$ be the output value at time t of unit i of layer l. (inputs values: $y_{t,i}^{0} = x_{t,i}$).
- Let r be the ratio of subsampling. This is often set to values such as 2 to 4.
- Subsampling operator:

$$y_{t,i}^{l} = \frac{1}{r} \sum_{k=0}^{r-1} y_{rt-k,i}^{l-1}$$

- Only compute values $y_{t,i}^{l}$ such that $(t \mod r) = 0$.
- Note: there are no parameter in the subsampling layer (but it is possible to add some, replacing for intance $\frac{1}{r}$ by a parameter and adding a bias term).

TDNNs (Graphical View)



Learning in TDNNs

- TDNNs can be trained by normal gradient descent techniques.
- Note that, as for MLPs, each layer is a differentiable function.
- We just need to compute the local gradient:
- Convolution layers:

$$\frac{\partial C}{\partial w^l_{i,j,k}} = \sum_t \frac{\partial C}{\partial s^l_{t,i}} \cdot \frac{\partial s^l_{t,i}}{\partial w^l_{i,j,k}} = \sum_t \frac{\partial C}{\partial s^l_{t,i}} \cdot y^{l-1}_{t-k,j}$$

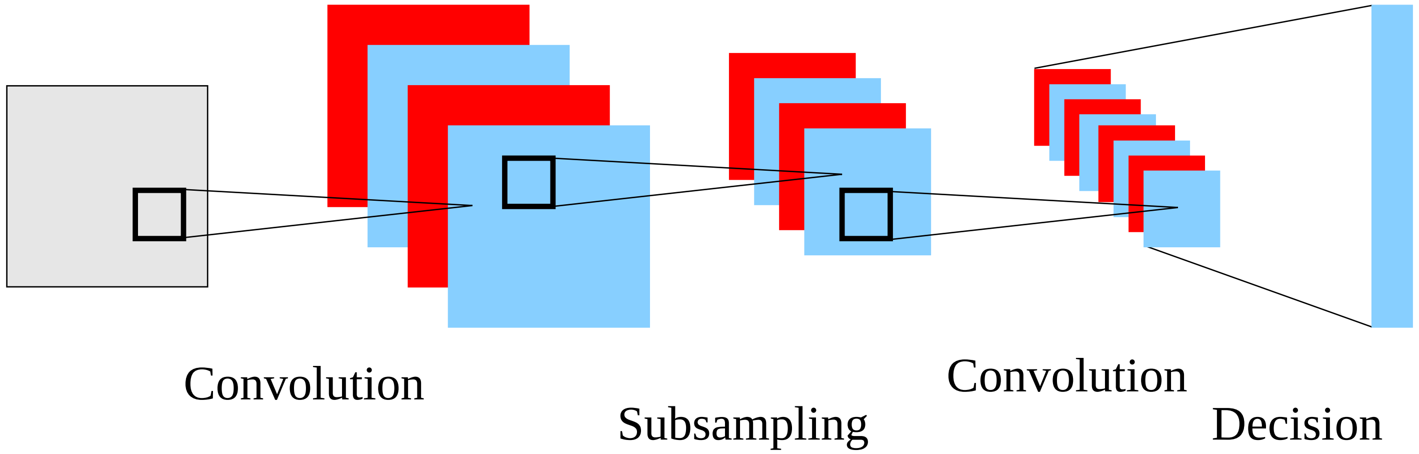
- Subsampling layers:

$$\frac{\partial C}{\partial y^{l-1}_{t-k,i}} = \frac{\partial C}{\partial y^l_{t,i}} \cdot \frac{\partial y^l_{t,i}}{\partial y^{l-1}_{rt-k,i}} = \frac{\partial C}{\partial y^l_{t,i}} \cdot \frac{1}{r}$$

LeNet for Images

- TDNNs are useful to handle regularities of time series
  ($\rightarrow$ 1D data)
- Could we use the same trick for images
  ($\rightarrow$ 2D data)?
- After all, regularities are often visible on images.
- It has indeed been proposed as well, under the name LeNet.

LeNet (Graphical View)



Other Parameter Sharing Techniques

- There are various other ways of sharing parameters.
- In speech recognition, word models are concatenations of phoneme models, hence words sharing the same phoneme will also share the same corresponding parameters!
- Soft weight tying:
  - Instead of sharing exactly the same value of the parameters, add a constraint such that some parameters should be near each other.
  - For instance, assume they were generated from the same Gaussian distribution.
  - Just fix the desired number of parameters, and each real parameter will have to be near one of the Gaussians of a Gaussian mixture trained simultaneously with the target problem.
- Multi-task parameter sharing...