Training Convolution RBMs

Pascal Lamblin, Université de Montréal
Why?

- RBMs work well for initializing layers of deep networks
- Convolution filters in Neural Networks give better performances on image data
  - Invariance by translation is easily learned
  - Takes advantage of the local informations
- Experiments (Yann Le Cun & Co.) have shown that learning convolution filters unsupervisedly can give good results
- So, why not try with RBMs?
A small introduction to convolution layers

The connections of a convolution layer represent a **linear transformation**, which implements:

- Sparse connectivity
- Capture of local interactions
- Weight sharing
More details on the 1D case

- If we denote by $f$ the convolution filter (set of weights):

$$h_i = \sum_{j=1}^{k} f_j v_{i+j}$$

- We can also represent the matrix corresponding to the linear transformation $h = Wv$

- The transposed transformation is also sparse, local, and linear, and is indeed a variant of the convolution
Convolutions in RBMs

- Defined by an Energy function:

$$\mathcal{E}(v, h) = c'v + h'Wv + h'b$$

- Update of $W_{ij}$ proportional to:

$$h^0_i v^0_j - h^1_i v^1_j$$

- We can easily replace $Wv$ by a convolution and $h'W$ by a transposed convolution

- It is also possible to compute the gradient wrt $W_{ij}$ and to “forward” the update to the right $f_k$, but in fact the update formula of $f$ is quite simple
The usual things we do with RBMs, but on images:

- Reconstruct images, possibly by also adding an explicit reconstruction error term in the gradient
- Sample images, to see how the generative model behaves
- Initialize weights of a LeNet-like architecture
- Understand better how Contrastive Divergence works
- Take over the world
Results

Ask him! ;)

Pascal Lamblin, Université de Montréal

CIAR Summer School – August 2007
Ask him! ;)}
Subsampling in usual convolution nets

Different techniques are used to limit the size of convolution layers, by collapsing a local area in a hidden layer into one single pixel.

- Parametrized mean

- Discarding pixels

- Taking the maximal value
We can train the RBM ignoring the fact that there will be a subsampling layer afterwards.

But it would be more interesting to incorporate this knowledge inside the RBM.

It is possible to tie the activity of several units in an RBM hidden layer, so that only one is active at a time (they act as a multinomial distribution).

We can do that over non-overlapping local areas of the hidden layers, achieving a form of sparsity.

Hopefully, taking the max after that will give interesting results.
Issues and future development

Lots of things to try... 

- Different variants of convolutions (to avoid problems at the edges)
- Sparse, local, untied weights
- Comparison with auto-associators or other models
- Other specially parametrized linear transformations...
- ...like a weight matrix parametrized by the output of another RBM, or Neural Net, etc.
Questions?
Questions?