Cooperative Energy Hierarchy

Jascha Sohl-Dickstein (jascha@berkeley.edu)
Advisor: Bruno Olshausen
Redwood Center for Theoretical Neuroscience
UC Berkeley

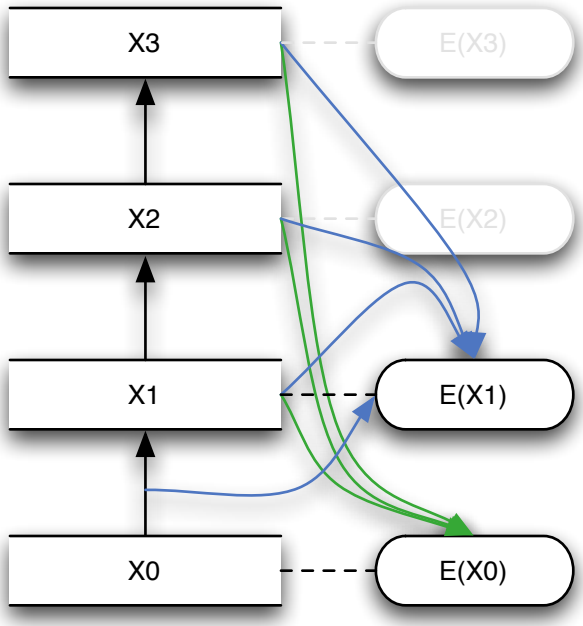
Canadian Institute for Advanced Research
Neural Computation & Adaptive Perception Summer School
Aug 10, 2007

Nice Properties for an Unsupervised Learning Algorithm

- Hierarchical
  - Greater abstraction at higher levels
- Discovery of independent/invariant aspects of input
- Local connectivity and learning
- Tractable learning
- Willingness to set aside structure/information once it has been successfully described

Brief (confusing) Overview

- Find a transformation of the data X0 into a space X1 such that the energy assigned to every point in X0 is a linear function of its representation in X1.
- Perform a change of variables, propagating both the data and the associated energy landscape forward into X1.
- Find a nonlinear transformation of the data X1 into a space X2, such that any desired modifications of the existing energy landscape in X1 can be written as a linear function of X2.
- Rinse. Repeat.



One Layer Case

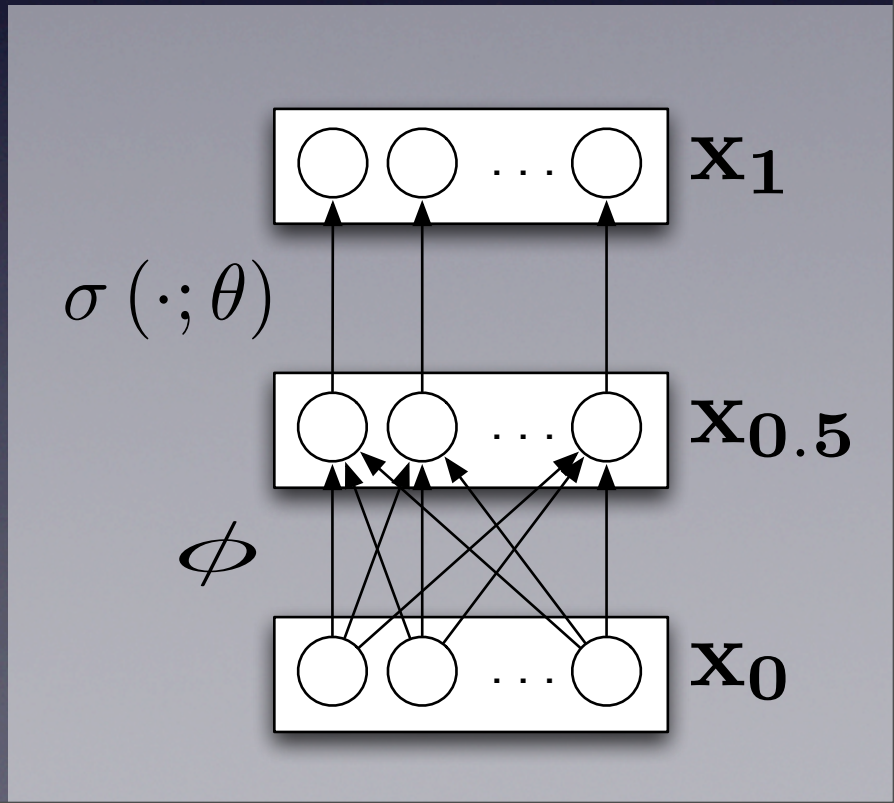
$$q\left(\mathbf{x_0};\theta\right) = \frac{e^{-E(\mathbf{x_0};\theta)}}{Z\left(\theta\right)}$$

$$Z\left(\theta\right) = \int_{\mathbf{x}} e^{-E(\mathbf{x_0};\theta)}$$

$$E\left(\mathbf{x_0};\theta\right) = \sum_i \mathbf{x_{1,i}} + \epsilon||\mathbf{x_0}||^2$$

$$\mathbf{x_{1,i}} = \sigma_i\left(\phi_\mathbf{i}\mathbf{x_0};\theta\right)$$

Learn by maximizing the log-likelihood of model - details to follow



One Layer Case

In the case where

$$\sigma(t;\alpha) = \alpha \log\left(1+t^2\right)$$

this reduces to a product of student-t tests (Welling, Hinton, Osidero, 2003)

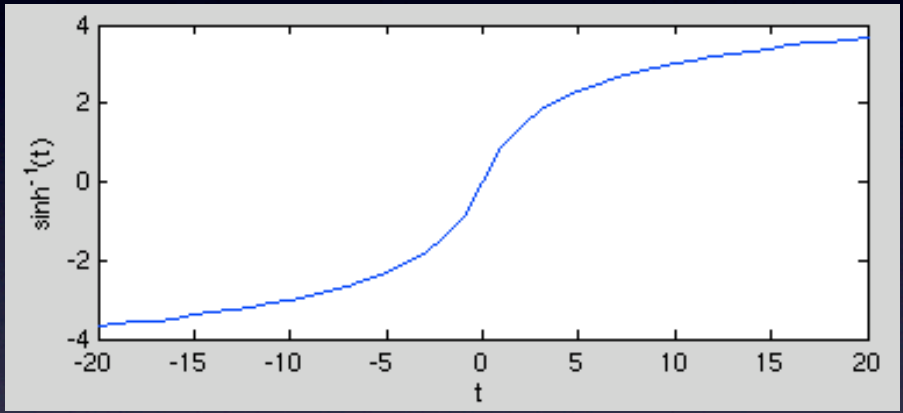
$$q(\mathbf{x_0};\theta) = \frac{e^{-\sum_i \alpha_i \log\left(1+(\phi_\mathbf{i}\mathbf{x})^2\right)}}{Z(\theta)} = \frac{\prod_i \left(1+(\phi_\mathbf{i}\mathbf{x})^2\right)^{-\alpha_i}}{Z(\theta)}$$

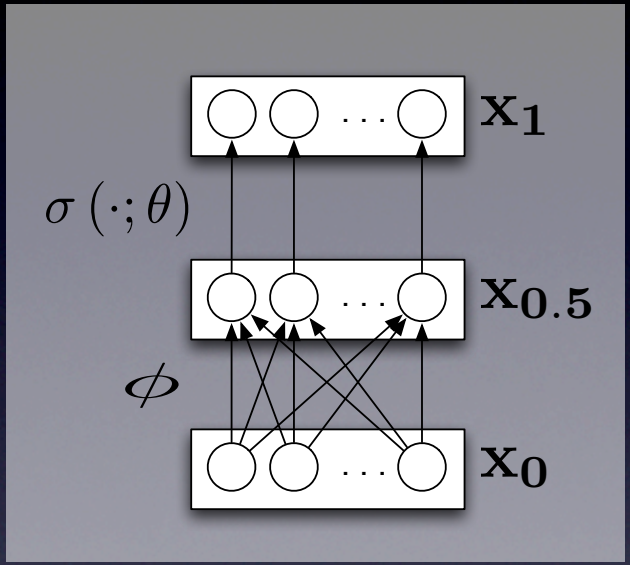
but what if we want a more flexible nonlinearity . . .?

One Layer Case

- Choose a more flexible nonlinearity

$$\sigma(t;\alpha,\beta) = \alpha \sinh^{-1}(t+\beta)$$





(This choice is only mildly special. Learning might work better if a menagerie of pointwise nonlinearities are instead provided.)

- The energy function now looks like the output of a one layer neural net. (Reminiscent of probabilistic backprop by eg Neal, MacKay in early 90s . . . but their focus was on treatment of model parameters)

Adding Another Layer

- X1 is a good representation to build on
  - The space used to construct the energy landscape is likely a sensible one to use for future manipulations of the energy landscape
  - The nodes in X1 are struggling to be independent. (An energy function linear in X1 means a model distribution which is factorial in exp(X1), which is a reasonable stand-in for independence in a feed-forward model.)

$$q\left(\mathbf{x_0}\right) \propto \prod_i f\left(\mathbf{x_{1,i}}\right)$$

Adding Another Layer

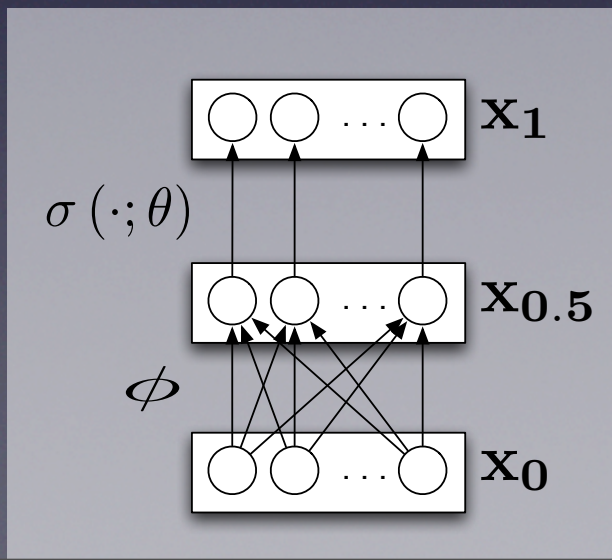
- Propagate both the data and the energy landscape up the hierarchy

$$\mathbf{x_{1,i}} = \sigma_i\left(\phi_\mathbf{i}\mathbf{x_0};\theta\right) \qquad q\left(\mathbf{x_1}\right) = \frac{q\left(\mathbf{x_0}\right)}{\left|\frac{\partial \mathbf{x_{1,i}}}{\partial \mathbf{x_{0,j}}}\right|}$$

$$\left|\frac{\partial \mathbf{x_{1,i}}}{\partial \mathbf{x_{0,j}}}\right| = |\phi| \prod_i \left|\frac{\partial \mathbf{x_{1,i}}}{\partial \mathbf{x_{0.5,i}}}\right|$$

$$E\left(\mathbf{x_0};\theta\right) = \sum_i \mathbf{x_{1,i}} + \epsilon||\mathbf{x_0}||^2$$

$$E\left(\mathbf{x_1};\theta\right) = \sum_i \left(\mathbf{x_{1,i}} - \log\frac{\partial \mathbf{x_{1,i}}}{\partial \mathbf{x_{0.5,i}}}\right) + \epsilon||\mathbf{x_0}||^2$$



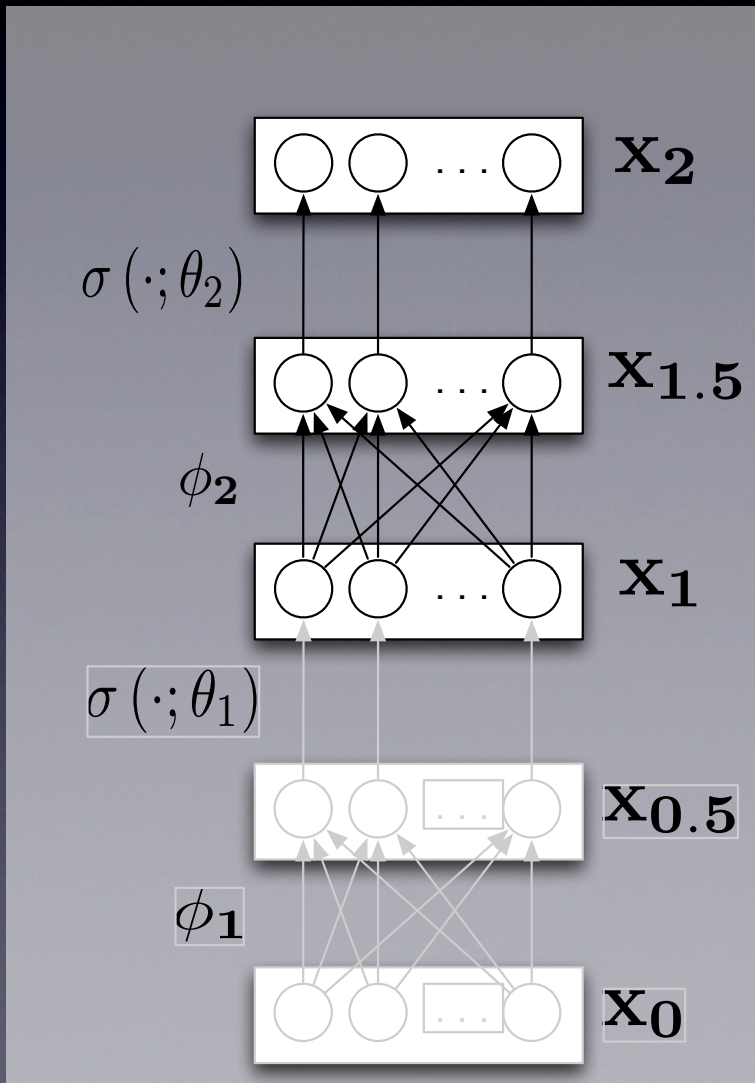
Adding Another Layer

$$q(\mathbf{x_1};\theta) = \frac{e^{-E(\mathbf{x_1};\theta)}}{Z(\theta)}$$

$$E(\mathbf{x_1};\theta) = \sum_i \left(\mathbf{x_{1,i}} - \log \frac{\partial \mathbf{x_{1,i}}}{\partial \mathbf{x_{0.5,i}}} \right) . + \sum_i \mathbf{x_{2,i}} + \epsilon ||\mathbf{x_0}||^2$$

$$\mathbf{x_{2,i}} = \sigma_{2,i}(\phi_{\mathbf{2,i}} \mathbf{x_1}; \theta)$$

Learn by maximizing the log-likelihood of model - details to follow



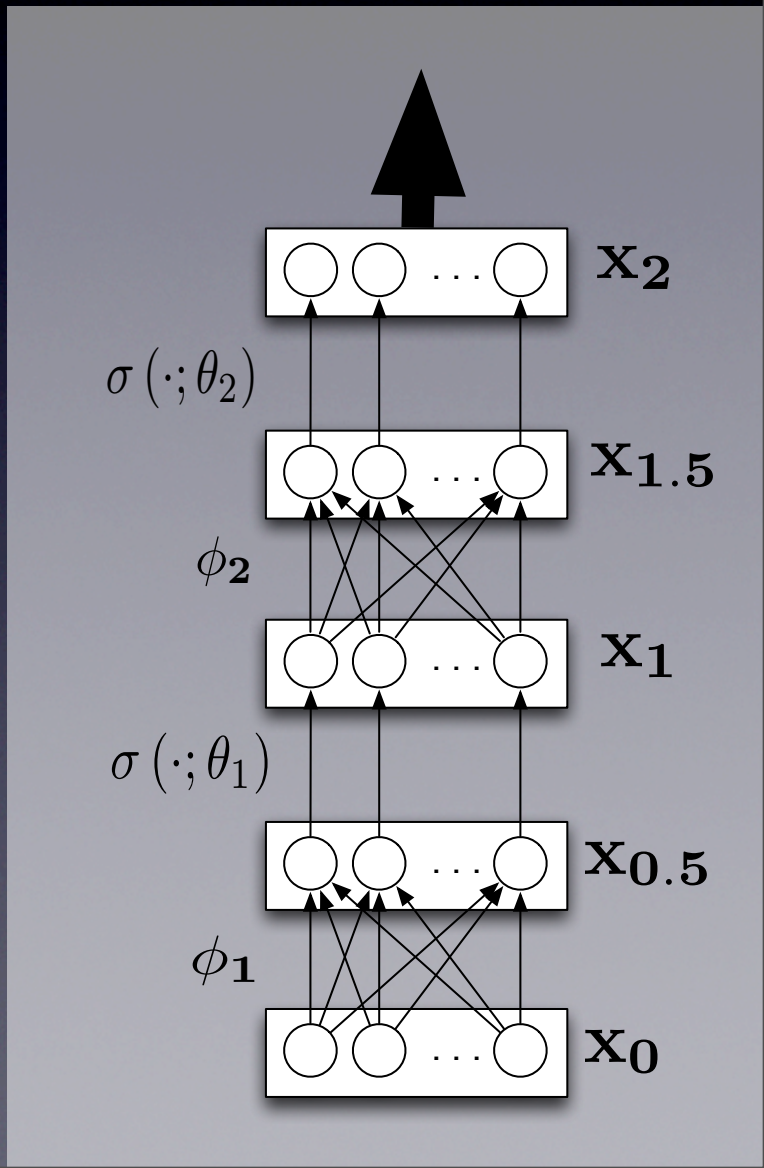
Adding Another Layer

- Instead of being relearned from scratch, only perturbations to the energy landscape are learned by higher levels. This means that higher levels don't waste effort rehashing structure which has already been described.
- The error function (log likelihood) is also computed in the new more sensible data space

Final Form

- The final energy assigned to an input is the sum of contributions from units at every level of abstraction

$$E\left(\mathbf{x_0};\theta\right)=\sum_i \mathbf{x_{1,i}}+\sum_i \mathbf{x_{2,i}}+\sum_i \mathbf{x_{3,i}}+\ldots+\epsilon||\mathbf{x_0}||^2$$



Learning

- Stochastic gradient descent on score matching objective function (Hyvärinen, 2005, or ask me for paper draft with alternative interpretation):

$$\hat{\theta} = \arg\max_{\theta} \left\langle \log q\left(\mathbf{x_0}; \theta\right) \right\rangle_{p(\mathbf{x_0})}$$

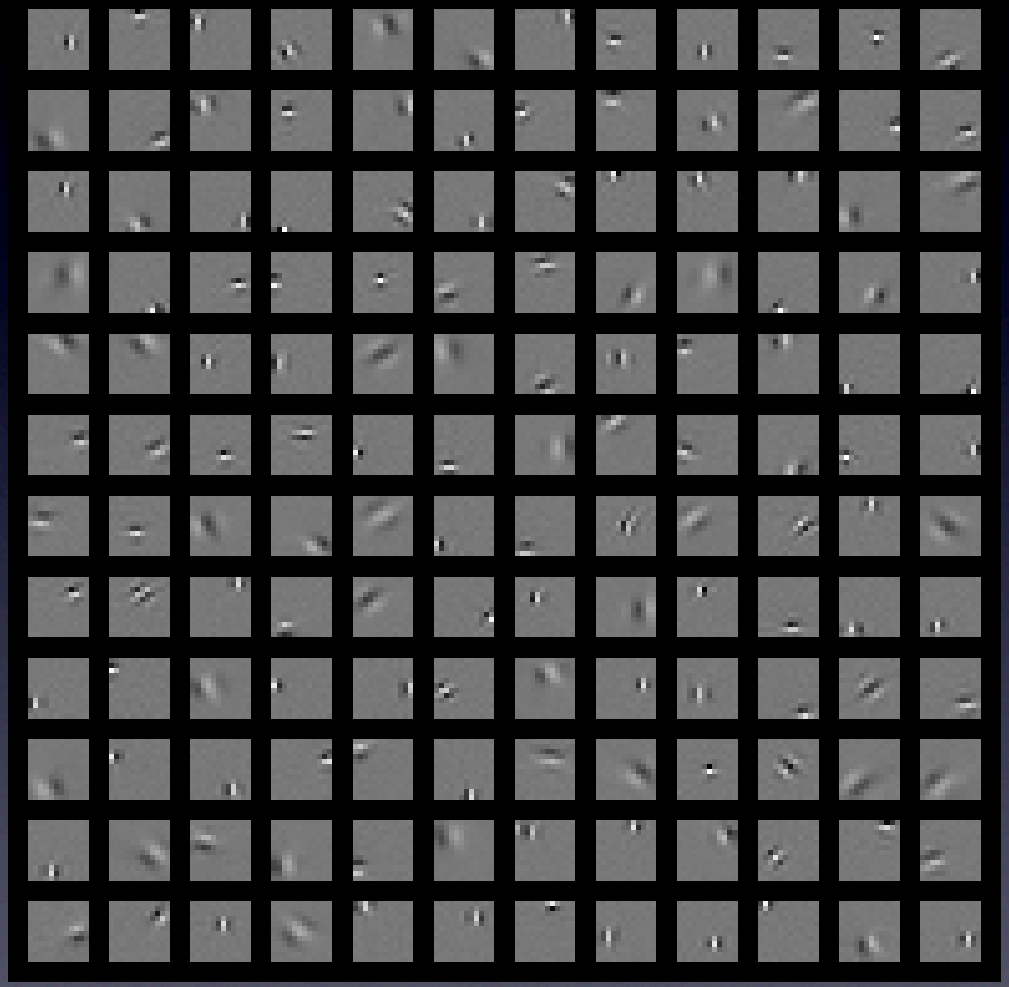
$$\hat{\theta} = \arg\min_{\theta} \left\langle \frac{1}{2} \nabla_X \cdot \nabla_X E\left(\mathbf{x_0}; \theta\right) - \nabla_X^2 E\left(\mathbf{x_0}; \theta\right) \right\rangle_{p(\mathbf{x_0})}$$

  - For SM learning, propagating the energy landscape up the hierarchy only involves accumulating the first spatial derivative of the energy function
  - When model and data distributions agree, both score matching and maximum likelihood share a global minima.
  - Score matching is equivalent to calculating the log learning gradient only in infinitesimal hyperspheres around the data points rather than over the full data space.

Additional Learning Tricks

- Stochastic gradient descent using a diagonal, partition free, approximation to the natural gradient
- Learning performed in whitened data space

Preliminary Results



12x12 receptive fields learned by first layer nodes after training on natural images

5 first layer nodes with (top) most positive and (bottom) most negative influence on 10 randomly chosen second layer nodes

Future Work

- Make hierarchy conditional on auxiliary information (eg, stacked RBMs learned in parallel)
- Allow energy landscape to propagate down as well as up hierarchy during learning
- Allow overcomplete higher levels (making incoherence assumption about manifold)
- Tapestry of Experts - unroll energy hierarchy over full image
- Temporal dynamics
- Intelligent guesses for initial energy landscape