Overcomplete models & Lateral interactions and Feedback

Teppo Niinimäki

April 22, 2010

Contents

Contents

Motivation

So far

- Sparse coding models: feature detector weights orthogonal
- Generative models: $\mathbf{A}$ invertible $\Rightarrow$ square matrix

$\Rightarrow$ no. of features $\leq$ no. of dimensions in data $\leq$ no. of pixels

Why more features?

- processing location independent
  $\Rightarrow$ same set of features for every location
- no. of simple cells in V1 $\gg$ no. of retinal gaglion cells
  ($\approx$ 25 times)

Overcomplete basis: Generative model

Generative model:

$$I(x,y) = \sum_{i=1}^{m} A_i(x,y)s_i$$

- basis vectors: A_i
- features: s_i
- no. of features: $m > |I|$ (or $m >$ dimension of data)

Overcomplete basis: Generative model

Generative model:

$$I(x,y) = \sum_{i=1}^{m} A_i(x,y)s_i + N(x,y)$$

- basis vectors: A_i
- features: s_i
- no. of features: $m > |I|$ (or $m >$ dimension of data)
- Gaussian noise: $N(x,y)$
  $\Rightarrow$ simplifies computations

Overcomplete basis: Computation of features

$$I(x,y) = \sum_{i=1}^{m} A_i(x,y)s_i + N(x,y)$$

How to compute the coefficients s_i for I?

- **A** not invertible
- more unknowns than equations
  $\Rightarrow$ many (infinite number of) different solutions

Find the sparsest solution (most s_i are close to 0):

- assume sparse distribution for s_i
- find the most probable values for s_i

Overcomplete basis: Computation of features

Aim: Find $\mathbf{s}$ which maximizes $p(\mathbf{s}|I)$.
By Bayes' rule we get

$$p(\mathbf{s}|I) = \frac{p(I|\mathbf{s})p(\mathbf{s})}{p(I)}$$

Ignore constant $p(I)$ and maximize logarithm instead:

$$\log p(\mathbf{s}|I) = \log p(I|\mathbf{s}) + \log p(\mathbf{s}) + \text{const.}$$

For prior distribution $p(\mathbf{s})$ assume sparsity and independence $\Rightarrow$

$$\log p(\mathbf{s}) = \sum_{i=1}^{m} G(s_i)$$

Overcomplete basis: Computation of features

$$I(x,y) = \sum_{i=1}^{m} A_i(x,y)s_i + N(x,y)$$
$$\log p(\mathbf{s}|I) = \log p(I|\mathbf{s}) + \log p(\mathbf{s}) + \text{const.}$$

Next compute $\log p(I|\mathbf{s})$.

Probability of $I(x,y)$ given $\mathbf{s}$ is Gaussian pdf of

$$N(x,y) = I(x,y) - \sum_{i=1}^{m} A_i(x,y)s_i.$$

Insert above into

$$p(N(x,y)) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} N(x,y)^2\right)$$

to get

$$\log p(I(x,y)|\mathbf{s}) = -\frac{1}{2\sigma^2}\left[I(x,y) - \sum_{i=1}^{m} A_i(x,y)s_i\right]^2 - \frac{1}{2}\log 2\pi.$$

Overcomplete basis: Computation of features

Because the noise is independent in pixels, we can sum over x, y to get the pdf for whole image

$$\log p(I|\mathbf{s}) = -\frac{1}{2\sigma^2}\sum_{x,y}\left[I(x,y) - \sum_{i=1}^{m} A_i(x,y)s_i\right]^2 - \frac{n}{2}\log 2\pi.$$

Combining above: Find **s** that maximizes

$$\log p(\mathbf{s}|I) = -\frac{1}{2\sigma^2}\sum_{x,y}\left[I(x,y) - \sum_{i=1}^{m} A_i(x,y)s_i\right]^2 + \sum_{i=1}^{m} G(s_i) + \text{const.}$$

$\Rightarrow$ numerical optimization
$\Rightarrow$ *non-linear* cell activities s_i

How about learning A_i?

Overcomplete basis: Basis estimation

Assume flat prior for the A_i

$\Rightarrow$ above $p(\mathbf{s}|I)$ is actually $p(\mathbf{s}, \mathbf{A}|I)$.

Maximize the probability (likelihood) of A_i over independent image samples $I_1, I_2, \ldots, I_3$:

$$\sum_{t=1}^{T} \log p(\mathbf{s}(t), \mathbf{A}|I_t) = -\frac{1}{2\sigma^2} \sum_{t=1}^{T} \sum_{x,y} \left[I_t(x,y) - \sum_{i=1}^{m} A_i(x,y) s_i \right]^2 + \sum_{t=1}^{T} \sum_{i=1}^{m} G(s_i(t)) + \text{const.}$$

At the same time we compute

- basis vectors A_i
- cell outputs $s_i(t)$.

Energy based models

Another approach:

- no generative model
- instead relax ICA to add more linear feature detectors W_i

$\Rightarrow$ not basis, but overcomplete representation

In ICA we maximized:

$$\log L(\mathbf{v}_1, \ldots, \mathbf{v}_m; \mathbf{z}_1, \ldots, \mathbf{z}_T) = T \log |\det(\mathbf{V})| + \sum_{i=1}^{m} \sum_{t=1}^{T} G_i(\mathbf{v}_i^T \mathbf{z}_t)$$

Recall $\mathbf{z}_t \sim I_t$, $\mathbf{v}_i \sim W_i$, $m = n$ and $G_i(u) = \log p_i(u)$.

If $m > n$ then $\log |\det(\mathbf{V})|$ is not defined.

Energy based models: estimation

Actually $\log|\det(\mathbf{V})|$ is a normalization constant.
Replace it and instead maximize:

$$\log L(\mathbf{v}_1, \ldots, \mathbf{v}_m; \mathbf{z}_1, \ldots, \mathbf{z}_T) = -T \log|Z(\mathbf{V})| + \sum_{i=1}^{m} \sum_{t=1}^{T} G_i(\mathbf{v}_i^T \mathbf{z}_t)$$

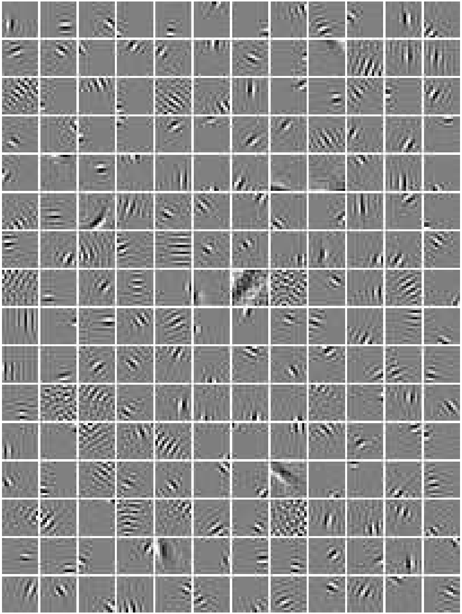
where

$$Z(\mathbf{V}) = \int \prod_{i=1}^{n} \exp(G_i(\mathbf{v}_i^T \mathbf{z})) d\mathbf{z}.$$

Above integral extremely difficult to evaluate. However

- it can be estimated or
- the model can be estimated directly: *score matching* and *contrastive divergence*

Energy based models: results



Estimated overcomplete representation with energy based model

- $G_i(u) = \alpha_i \log\cosh(u)$
- score matching
- patches of 16 x 16 = 256 pixels
- preprocessing ⇒ n = 128
- m = 512 receptive fields

(Fig 13.1: Random sample of W_i.)

Contents

Motivation

So far considered

- ”bottom-up” or feedforward frameworks

In reality there are also

- ”top-down” connections $\Rightarrow$ feedback
- lateral (horizontal) interactions

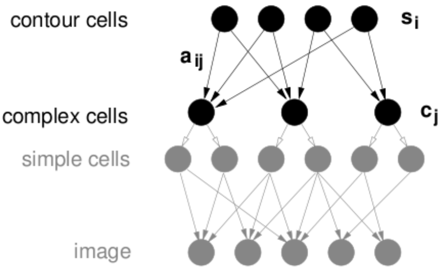
How to model them too?
$\Rightarrow$ using Bayesian inference!

Feedback as Bayesian inference: contour integrator

Why feedback connections?

- to enhance responses consistent with the broader visual context
- to reduce noise (activity inconsistent with the model)

$\Rightarrow$ combine bottom-up sensory information with top-down priors



Example: contour cells and complex cells

Define generative model:

$$c_k = \sum_{i=1}^{K} a_{ki} s_i + n_k$$

where n_k is Gaussian noise.

Feedback as Bayesian inference: contour integrator

$$c_k = \sum_{i=1}^{K} a_{ki} s_i + n_k$$

Now we just model the feedback!

First calculate **s** for given image:

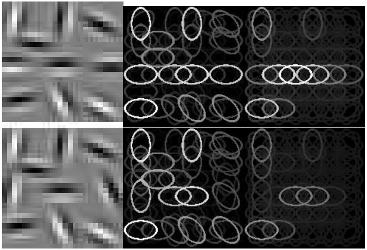
1. compute **c** normally (feedforward)
2. find $\mathbf{s} = \hat{\mathbf{s}}$ that maximizes $\log p(\mathbf{s}|\mathbf{c})$
   $\Rightarrow$ should be non-linear in **c** (why?)

Then reconstruct complex cell outputs using the linear generative model, but ignoring the noise:

$$\hat{c}_k = \sum_{i=1}^{K} a_{ki} \hat{s}_i$$

(for instance by sending feedback signal $u_{ki} = \left[\sum_{i=1}^{K} a_{ki} \hat{s}_i\right] - c_k$)

Feedback as Bayesian inference: contour integrator example



(Fig. 14.1)

Example results:

- left: patches with random Gabor functions (three collienar in upper case)
- middle: c_k
- right: $\hat{c}_k$ (based on contour-coding unit activities s_i)

$\Rightarrow$ noise reduction empasizes collinear activations but suppresses others

Feedback as Bayesian inference: higher-order activities

How to estimate higher order activities $\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} p(\mathbf{s}|\mathbf{c})$?

Like before, using Bayes' rule we get

$$\log p(\mathbf{s}|\mathbf{c}) = \log p(\mathbf{c}|\mathbf{s}) + \log p(\mathbf{s}) + \text{const.}$$

Again we assume that $\log p(\mathbf{s})$ is sparse. Analogously to overcomplete basis:

$$\log p(\mathbf{s}|\mathbf{c}) = -\frac{1}{2\sigma^2} \sum_{k=1}^{K} \left[c_k - \sum_{i=1}^{m} a_{ki} s_i \right]^2 + \sum_{i=1}^{m} G(s_i) + \text{const.}$$

Next assume $\mathbf{A}$ is invertible and orthogonal
$\Rightarrow$ multiplying $\mathbf{c} - \mathbf{A}\mathbf{s}$ by $\mathbf{A}^T$ in above square sum $\|\mathbf{c} - \mathbf{A}\mathbf{s}\|$ we get $\|\mathbf{A}^T\mathbf{c} - \mathbf{s}\|$ without changing the norm:

$$\log p(\mathbf{s}|\mathbf{c}) = -\frac{1}{2\sigma^2} \sum_{i=1}^{m} \left[\sum_{k=1}^{K} a_{ki} c_k - s_i \right]^2 + \sum_{i=1}^{m} G(s_i) + \text{const.}$$

Feedback as Bayesian inference: higher-order activities

Maximize separately each:

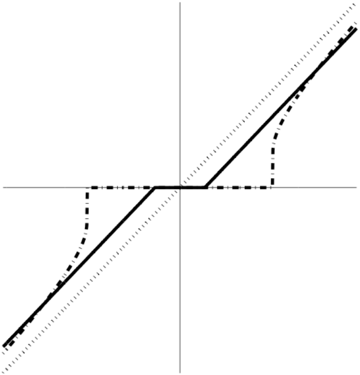
$$\log p(s_i|\mathbf{c}) = -\frac{1}{2\sigma^2}\left[\sum_{k=1}^{K} a_{ki}c_k - s_i\right]^2 + G(s_i) + \text{const.}$$

Maximum point can be represented as

$$\hat{s}_i = f\left(\sum_{k=1}^{K} a_{ki}c_k\right)$$

where f depends on $G = \log p(s_i)$.

ex. for Laplacian distribution $f(y) = \text{sign}(y)\max(|y| - \sqrt{2}\sigma^2, 0)$.



(Fig. 14.3)

Sparseness leads to shrinkage/tresholding.

Left image: f for

- Laplacian distribution (solid line)
- highly sparse distribution [7.22 in the book] (dash-dotted line)

$\Rightarrow$ cell activities considered noise are lowered to zero

Feedback as Bayesian inference: Categorization

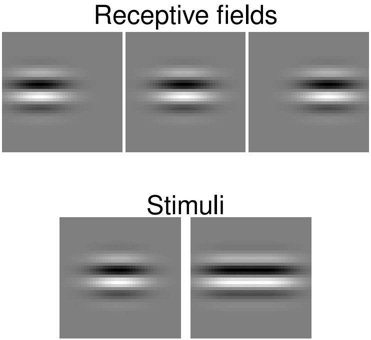
Generative model applicable to any two cell groups.

Example: category variables

- $s_i \in \{0, 1\}$
- value = 1 if the object in visual field in certain category

$\Rightarrow$ jumpy behaviour

Overcomplete basis and end-stopping



(Fig. 14.4)

End-stopping: some (simple) cells reduce firing rate if Gabor stimulus is elongated
$\Rightarrow$ receptive fields not linear?

How to model?

- overcomplete basis and bayesian inference

$\Rightarrow$ competition between overlapping cells

Predictive coding

- upper level predicts activity in the lower level
- lower level sends errors back to the upper level

In noisy generative model, the prediction is implicit. $\Rightarrow$ estimating noisy generative model $\approx$ minimization of prediction error

To infer the most likely s_i repeat above steps and update the model using gradient method with

$$\frac{\partial \log p(\mathbf{s}|\mathbf{c})}{\partial s_i} = \frac{1}{\sigma^2} \sum_k a_{ki} \left[c_k - \sum_{i=1}^{m} a_k i s_i \right] + G'(s_i).$$

Summary

Overcomplete models:

- overcomplete basis
- energy based models

Interactions:

- noisy model and Bayesian inference $\Rightarrow$ feedback
- overcomplete basis $\Rightarrow$ end-stopping
- predictive coding