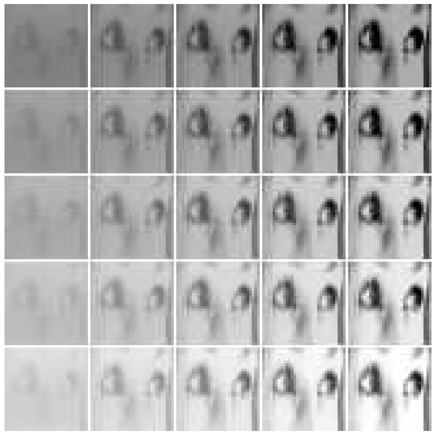
Learning a selectivity–invariance–selectivity feature extraction architecture for images

Michael U. Gutmann
University of Helsinki
michael.gutmann@helsinki.fi

Aapo Hyvärinen
University of Helsinki
aapo.hyvarinen@helsinki.fi

Motivation

- We are very good at detecting specific patterns while being invariant/tolerant to possible variations.
- It is the pairing of selectivity with invariance which is important. (“tolerant selectivity”)
- Tolerant selectivities occur at multiple levels



(a) “Low-level”

(b) “Higher-level”

Lower- and higher-level tolerant selectivities:

a) Same face, luminance and contrast vary
b) Same face, facial expression varies
(From “Facial Expressions - A Visual Reference for Artists” by Mark Simon.)

Question asked and methodology

- Basic hypothesis:
  Higher level tolerant selectivities emerge through a sequence of elementary *selectivity* and *invariance* computations.
  (see for example: Riesenhuber & Poggio, Nature 1999; Kouh & Poggio, NeCo 2008; Rust & Stocker, Curr Op Neurobiol, 2010)
- Question asked:
  In a system with three processing layers, what should be *selected* and *tolerated* at each level of the hierarchy?
- Methodology:
  - Learn the *selectivity* and *invariance* computations from images, using as few assumptions as possible.
  - Learning $\equiv$ fitting a probability density function

Data and preprocessing

- Tiny images dataset, converted to gray scale: complete scenes downsampled to 32 by 32 images

  (Torralba et al, TPAMI 2008)
- Preprocessing:
  - Removing DC component
  - Normalizing norm after whitening
  - Reducing the dimension from $32 \cdot 32 = 1024$ to 200
- Preprocessing can be considered a form of luminance and contrast gain control, followed by low-pass filtering.

Examples from the tiny images dataset before preprocessing.

Feature extraction architecture

- Let $\mathbf{x} \in \mathrm{R}^{200}$ be a vectorized image after preprocessing.
- Feature extraction with three processing layers:

$$y_i^{(1)} = \mathbf{w}_i^{(1)T}\mathbf{x} \qquad i = 1 \ldots 100$$

$$y_k^{(2)} = f_{\text{th}}\left(\ln\left[\sum_{i=1}^{100} w_{ki}^{(2)}(y_i^{(1)})^2 + 1\right] + b_k^{(2)}\right) \qquad k = 1 \ldots 50$$

$$\tilde{\mathbf{y}}^{(2)} = \text{gain control}\big(\mathbf{y}^{(2)}\big)$$

$$y_j^{(3)} = f_{\text{th}}\left(\mathbf{w}_j^{(3)T}\tilde{\mathbf{y}}^{(2)} + b_j^{(3)}\right) \qquad j = 1 \ldots n^{(3)}$$

Thresholding function $f_{\text{th}}(u)$: smooth version of $\max(u, 0)$

Gain control: centering, normalizing the norm after whitening, dimension reduction (similar to the preprocessing)

- Parameters of interest: feature vectors $\mathbf{w}_i^{(1)}$, pooling weights $w_{ki}^{(2)} \geq 0$, higher-order feature vectors $\mathbf{w}_j^{(3)}$

Other parameters: the thresholds $b_k^{(2)}$ and $b_k^{(3)}$

Learning

- First, learn the parameters of layers one and two. Keeping them fixed, learn the parameters of layer three.
- For layer one and two, fit the pdf

$$p(\mathbf{x}; \underbrace{\mathbf{w}_i^{(1)}, w_{ki}^{(2)}, b_k^{(2)}}_{\text{Parameters}}) \propto \exp\left[\sum_{k=1}^{50} y_k^{(2)}\right].$$

- For layer three, fit the pdf

$$p(\mathbf{x}; \underbrace{\mathbf{w}_j^{(3)}, b_j^{(3)}}_{\text{Parameters}}) \propto \exp\left[\sum_{j=1}^{n^{(3)}} y_j^{(3)}\right].$$

- Basic idea: the overall activity of the feature outputs determines how probable the input is.
- We do not know the partition functions: Likelihood is intractable. Use noise-contrastive estimation for the fitting.

  (Gutmann and Hyvärinen, JMLR2012)

Noise-contrastive estimation

(Gutmann and Hyvärinen, JMLR2012)

- Purpose: learn parameters $\boldsymbol{\theta}$ of a pdf $p_{\boldsymbol{\theta}}$ when you do not know the partition function.

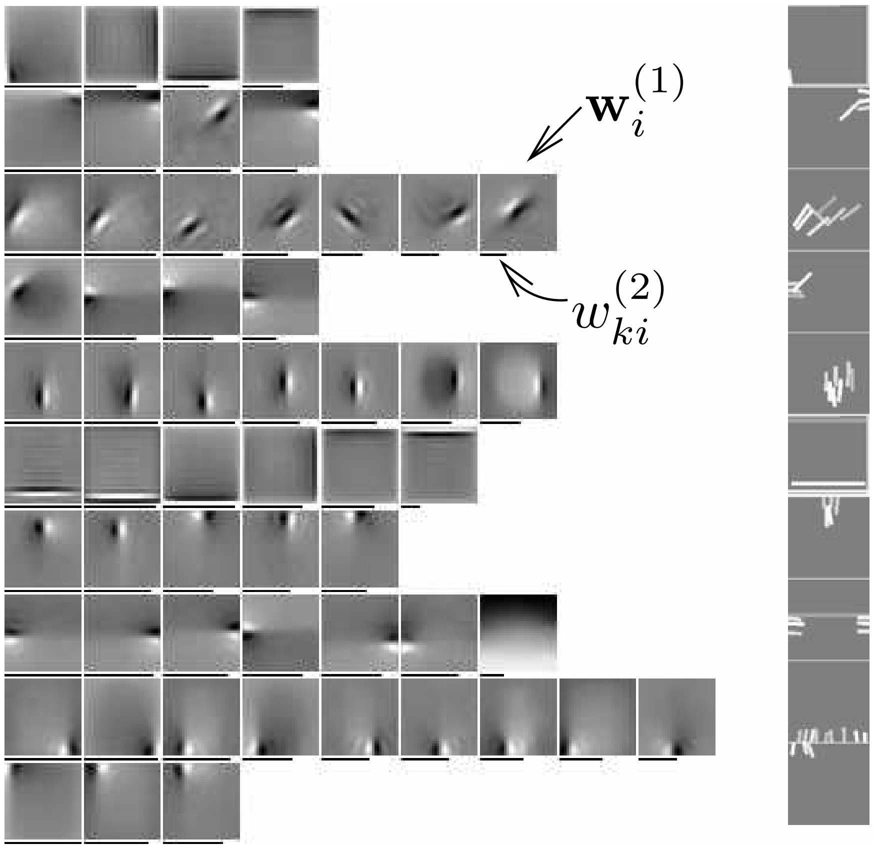
  Here: $p_{\boldsymbol{\theta}}(\mathbf{x}) = p(\mathbf{x}; \mathbf{w}_i^{(1)}, w_{ki}^{(2)}, b_k^{(2)})$ or $p_{\boldsymbol{\theta}}(\mathbf{x}) = p(\mathbf{x}; \mathbf{w}_j^{(3)}, b_j^{(3)})$
- Intuition: Learn the differences between the data and auxiliary "noise" whose properties you know. Deduce from the differences the properties of the observed data.
- More concrete:
  1. Choose a random variable $\mathbf{z}$ with known pdf $p_{\mathbf{z}}$ where sampling is easy.

     Here: Uniform distribution in the sphere where the data is defined
  2. Obtain an auxiliary sample of $\mathbf{z}$ ("noise").
  3. Perform logistic regression on the data and the auxiliary "noise"; use the ratio $p_{\boldsymbol{\theta}}/p_{\mathbf{z}}$ in the regression function.
- The procedure provides a consistent estimator of $\boldsymbol{\theta}$.

Results, layers one and two

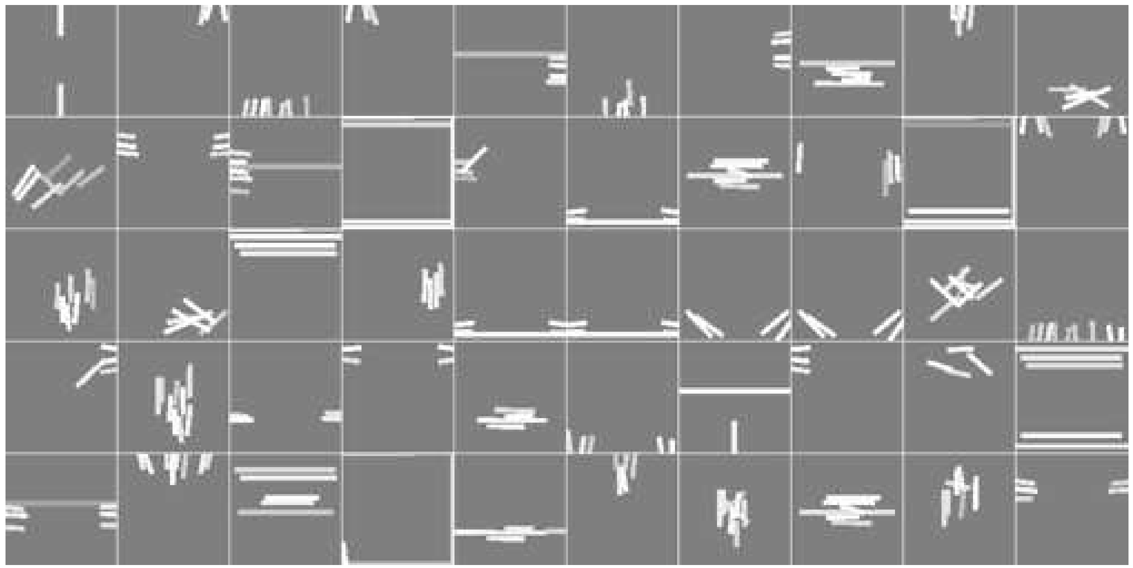
The $\mathbf{w}_i^{(1)}$ are Gabor-like, the $w_{ki}^{(2)}$ are sparse (94.5%: $< 10^{-6}$; 5.1%: > 10)
Mostly complex-cell like pooling

Each row corresponds to a different $y_k^{(2)}$



Subset of the features and their icons

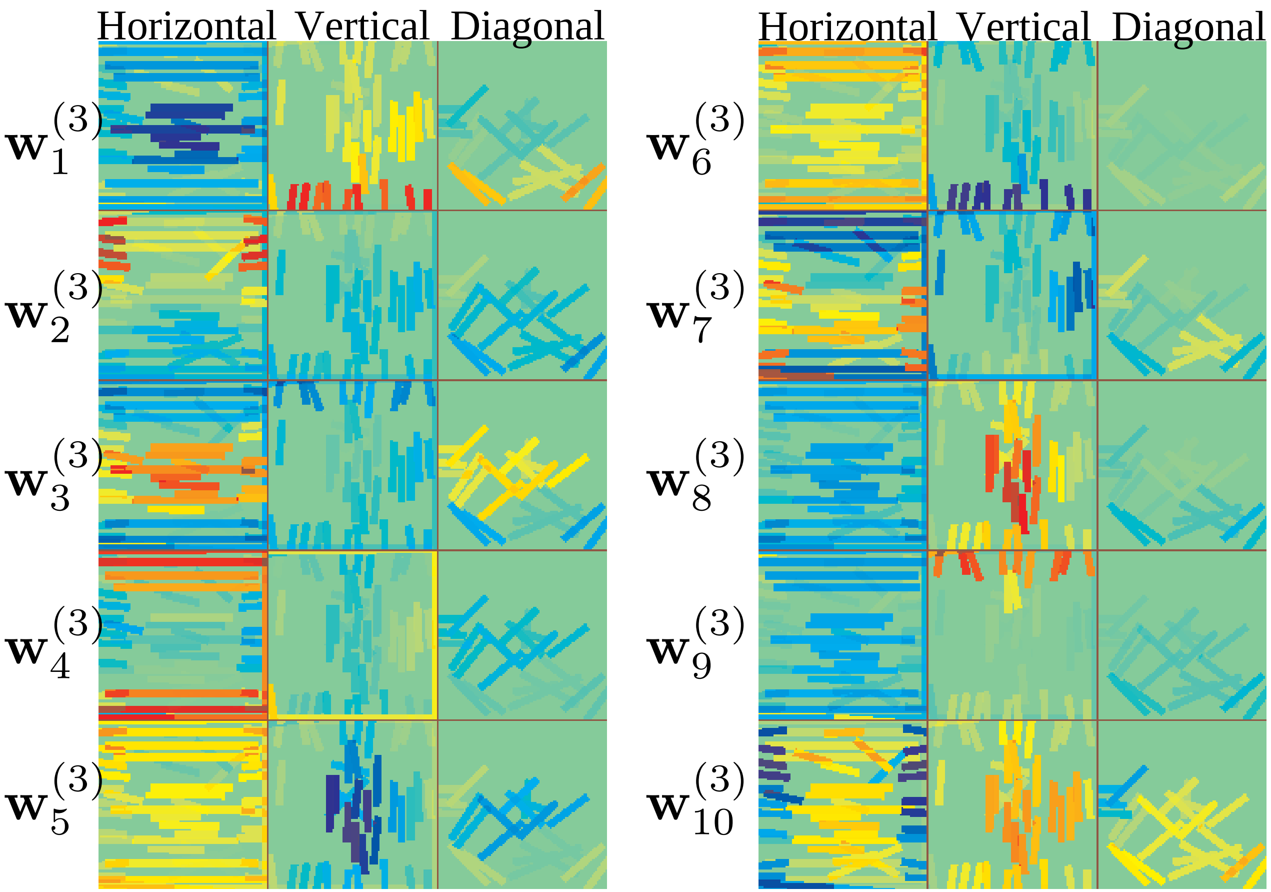
$$y_k^{(2)} = f_{\text{th}}\left(\ln\left[\sum_{i=1}^{100} w_{ki}^{(2)} (\mathbf{w}_i^{(1)T}\mathbf{x})^2 + 1\right] + b_k^{(2)}\right)$$



All the learned features for layer one and two

Results, layer three

Features with enhanced selectivity to orientation and space.



Complete set of $\mathbf{w}_j^{(3)}$ for $n^{(3)} = 10$. See paper for $n^{(3)} = 100$.

$$\tilde{\mathbf{y}}^{(2)} = \text{gain control}(\mathbf{y}^{(2)})$$
$$y_j^{(3)} = f_{\text{th}}\left(\mathbf{w}_j^{(3)\,T}\tilde{\mathbf{y}}^{(2)} + b_j^{(3)}\right)$$

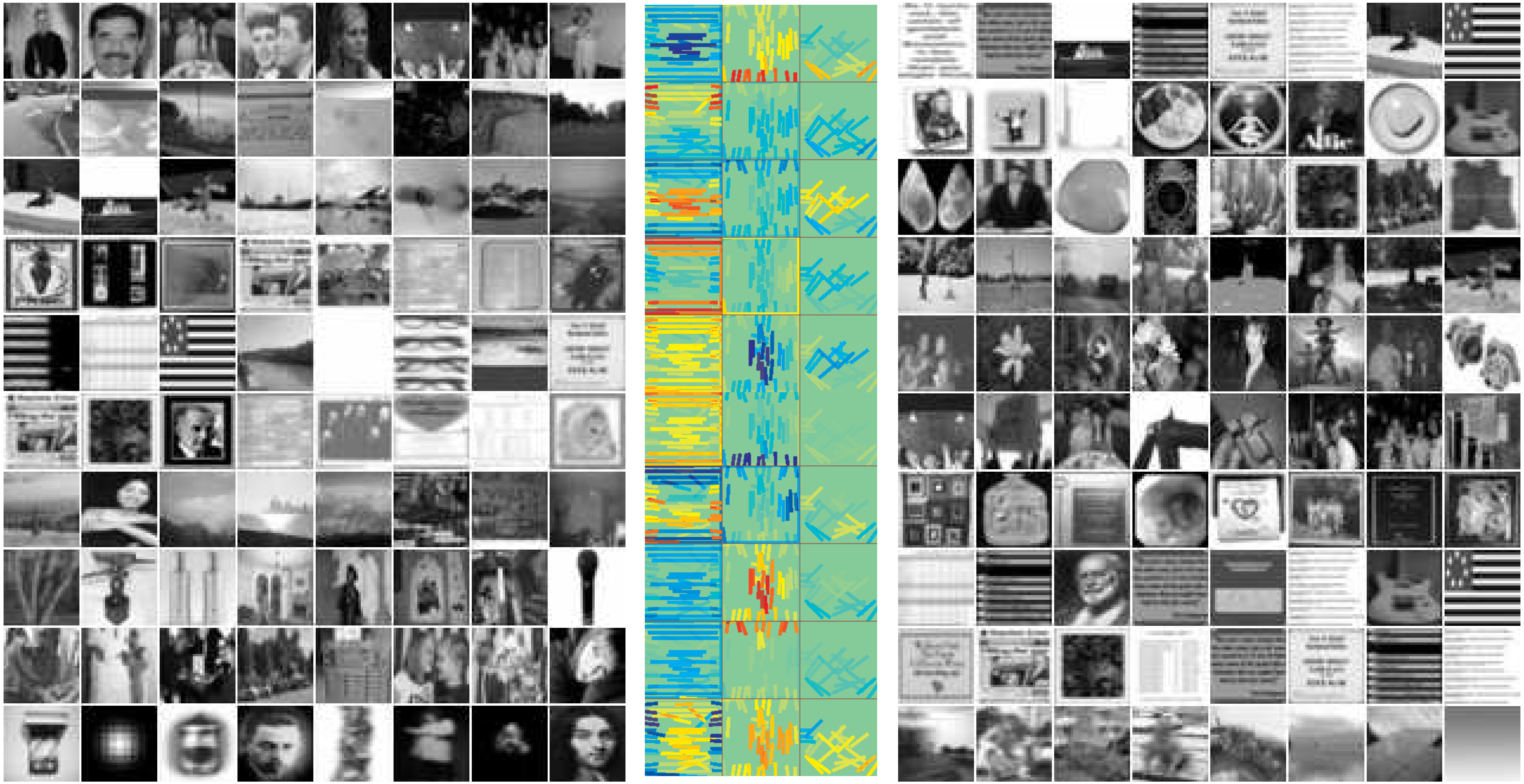
k-th element of $\mathbf{w}_j^{(3)}$ is positive: Activity of $y_k^{(2)}$ is detected. Corresponding icon is colored in red.

k-th element of $\mathbf{w}_j^{(3)}$ is negative: Inactivity of $y_k^{(2)}$ is detected. Corresponding icon is colored in blue.

Results, layer three

Descriptors of overall image properties?

Images giving maximal activation | Features | Images giving minimal activation



Feature outputs were computed for 10000 randomly chosen tiny images.

Summary

- Selectivity and invariance/tolerance are important for any feature extraction system.
- Question asked:
  In a system with three processing layers, what should be selected and tolerated at each level of the hierarchy?
- Looked for an answer by fitting probabilistic models to images:
  $\rightarrow$ First layer: Selectivity to Gabor-like image structure
  $\rightarrow$ Second layer: Tolerance to exact orientation or localization of the stimulus ("complex-cells")
  $\rightarrow$ Third layer: Enhanced selectivity to orientation and/or location of the stimulus