The “Independent Components” of Natural Scenes are Edge Filters

Anthony J. Bell and Terrence J. Sejnowski

Motivation

• Neurons with line and edge selectivities found in primary visual cortex of cats and monkeys
• Edges are suspicious coincidences in an image
• Unsupervised learning algorithm \(\rightarrow\) find a factorial code of independent visual features

This paper:
• Non-linear infomax network \(\rightarrow\) sets of visual filters \(\rightarrow\) localized and oriented
• Independent filters
• Resembles the receptive fields of simple cells in visual cortex
How filtering works?

First filter

Second filter

*https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Blind Source Separation

Cocktail Party Problem

$s \in \mathbb{R}^M$ speakers
$A \in \mathbb{R}^{N \times M}$ mixing matrix
$x \in \mathbb{R}^N$ observations
$W \in \mathbb{R}^{M \times N}$ demixing matrix
$u \in \mathbb{R}^M$ recovered sources

Can we recover $s$?

$N$ speakers $M$ microphones $x = As$ $u = Wx$
Blind separation

Linear image synthesis model: \( x = As \)

If \( A \) was known \( \rightarrow s = A^{-1}x \)

Recovering: \( u = Wx \)
\( W = A^{-1} \)

Blind Separation

Goal: We want \( u_1, u_2, \ldots, u_m \) to be independent.

An easier solution: Uncorrelated \( u_1, u_2, \ldots, u_m \) \( \rightarrow \) Whitening

Definition: \( W \) is decorrelating matrix if \( u_1, u_2, \ldots, u_m \) are uncorrelated \( \rightarrow \) \( E[uu^T] \) is diagonal

\( E[uu^T] = WE[xx^T]W^T \rightarrow \text{TWO MANY CHOICES of } W \)

Special Case: \( E[uu^T] = I \rightarrow W^TW = E[xx^T]^{-1} = C^{-1} \)

Covariance Matrix: \( C = E[xx^T] \)
Baseline Models

### Principal Component Analysis (PCA) Solution

- Eigenvalue decomposition of $C \rightarrow \text{ED}_E^{-1} \rightarrow$ columns of $E$ are the eigenvectors
- Choose $W = D^{-1/2}E^T$
- Filters are orthogonal $\rightarrow WW^T = D^{-1}$ scaling matrix

---

### ZCA Solution

- $W^T = W \rightarrow C^{1/2}$
- not orthogonal filters $\rightarrow$ symmetric filters
- similar to how the biological eye (the retina) processes images

Both PCA and ZCA are based on second order statistics.

$C = E\{xx^T\}$ gets us **uncorrelated outputs** but is not enough to determine **independence**!
ICA - Preliminaries

**Independence:** \( f_\mathbf{u}(\mathbf{u}) = \prod_i f_{u_i}(u_i) \)

**Mutual Information:** always non-negative \( \Rightarrow \) zero iff the variables are statistically independent.

\[
I(u_1, u_2, ..., u_m) = \sum_{i=1}^{m} H(u_i) - H(\mathbf{u})
\]

---

**The Infomax Principle**
Maximizing the output entropy (or information flow) of a network with non-linear outputs.

\[
L = H(\sigma(u_1), \sigma(u_2), ..., \sigma(u_m))
\]

\( \sigma(u_i) \): output

\( \sigma \): non-linear scalar functions

---

Architecture

We are trying to maximize the joint entropy of \( y_i \)'s.

**Sigmoid function**

\[
\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}
\]
ICA - Preliminaries

Theorem: ICA can be performed exactly, by finding the maximum, with respect to \( W \), of the joint entropy, \( H(y) \), of an output vector, \( y \), which is the vector \( u \), except that each element is transformed by a sigmoidal function which is a Cumulative Distribution Function (CDF) of sources which we are looking for. In this case ‘infomax’ is equivalent to minimization of the mutual information between \( u_i \).*

\[
y^T = [y_1, y_2, \ldots, y_m] = [\sigma(u_1), \sigma(u_2), \ldots, \sigma(u_m)] = \sigma(u^T) = \sigma \left( (Wx)^T \right)
\]

Relationship of probability density functions (PDF): \( f_y(y) = \frac{f_x(x)}{|J|} \) \( J = \det \left[ \frac{\partial y_i}{\partial x_j} \right] \)

**Our cost function:** \( H(y) = -E \left[ \ln \left( f_y(y) \right) \right] = E \left[ \ln (|J|) + H(x) \right] \) \( H(x) \) is constant.


The Algorithm for ICA

**Goal:** Finding \( W \in \mathbb{R}^{M \times N} \) s.t. \( u = Wx \) has independent components

**Data:** 17595 12x12 sample in the training set created from 4 natural scene

**Preprocessing:** \( x \leftarrow 2W_z (x - \bar{x}) \)
\( \rightarrow \) removes first and second order statistics
The Algorithm for ICA

**Input:** $x^{(1)}, x^{(2)}, \ldots, x^{(k)} \in \mathbb{R}^N$

**Cost function:** Joint Entropy of the outputs $y$: $H(y) = H(\sigma(u))$

- Initialization: $W = I$
- After 30 sweeps through the data shuffling
- Batch size of 50 patches
- Do stochastic gradient ascent to increase the cost function

  - Entropy gradient:

    $$
    \Delta W \propto \frac{\partial H(y)}{\partial W} = E \left[ \frac{\partial \ln |J|}{\partial W} \right] \quad \text{where } J = \det \left[ \frac{\partial y_i}{\partial x_j} \right]_{ij}
    $$

The Algorithm for ICA

- In stochastic gradient ascent, $E[\cdot]$ is removed
  - Evaluating derivative

  $$
  \Delta W \propto [W^T]^{-1} + \hat{y}x^T
  $$

  where $\hat{y}_i = \frac{\partial}{\partial u_i} \ln \frac{\partial y_i}{\partial u_i}$

- Use of natural gradient for fast convergence and avoiding of matrix inverse:
  - Multiply with $W^T W$

  $$
  \Delta W \propto \frac{\partial H(y)}{\partial W} W^T W = (I + \hat{y}u^T)W
  \quad \Delta w_{ij} \propto w_{ij} + \hat{y}_i \sum_k w_{kj} u_k
  $$
Results

PCA: Basis functions and filters are the same
ZCA: First 6 are filters, last 6 are the basis functions $\rightarrow$ columns of $W^{-1}$
W: ICA filters learned on whitened data
ICA: ICA filters $W_{i} = W_{2} W_{i}$
A: Basis functions of ICA filters $\rightarrow$ Columns of $W_{i}^{-1}$ $\rightarrow$ Only stimulate their corresponding filter

Filters as rows of matrix $W$
- 1 DC filter
- 106 oriented filters
  - 35 diagonal
  - 37 vertical
  - 34 horizontal
- 37 localized checkerboard patterns
Summary

**Problem:** Learning linear filters based on natural images

**Results:**
- Localised edge detectors
- Algorithm is sensitive to higher-order statistics
- Whitening $\rightarrow$ second-order statistics are not enough
- Remaining information $\rightarrow$ higher-order statistics
- ICA takes higher order statistics into account
- For further levels of invariance (shifting, rotating, scaling, lighting) $\rightarrow$ multi-layer non-linear networks