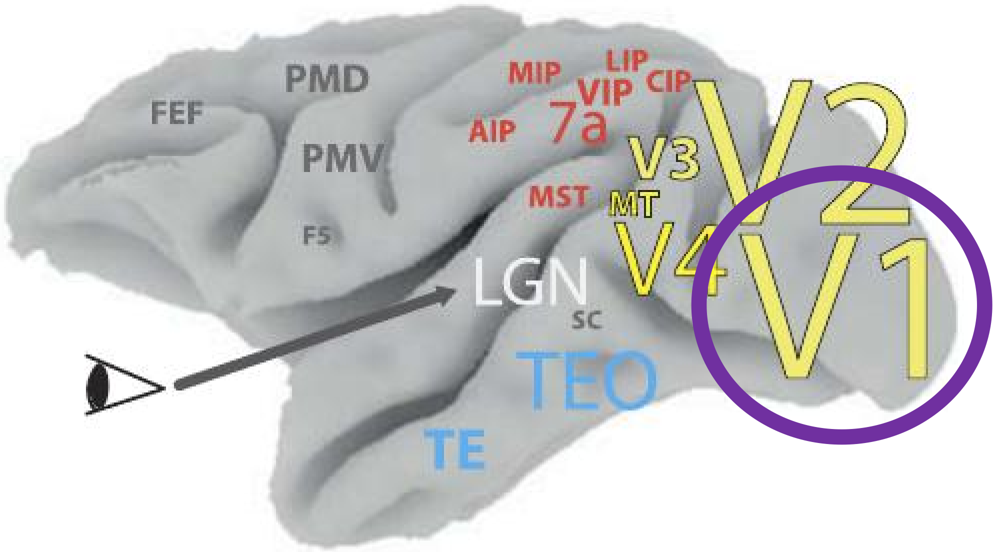
V1 Simple Cells Modeling

Daniela Pamplona

Contents

Contents

Where is V1?



PMD
FEF
PMV
FS
MIP
LIP
VIP
CIP
AIP
7a
V3
V2
MST
MT
V4
V1
LGN
SC
TEO
TE

Simple cells and complex cells in V1

Simple Cells

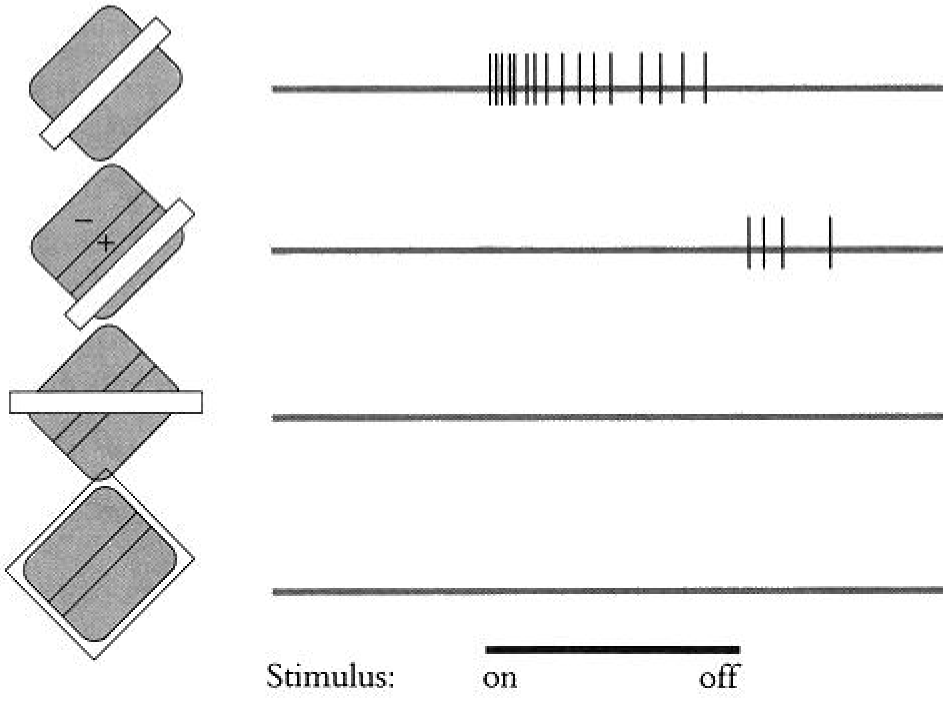
- have distinct excitatory and inhibitory regions within RF
- linearity of spatial summation within both the excitatory and inhibitory regions

Complex cells

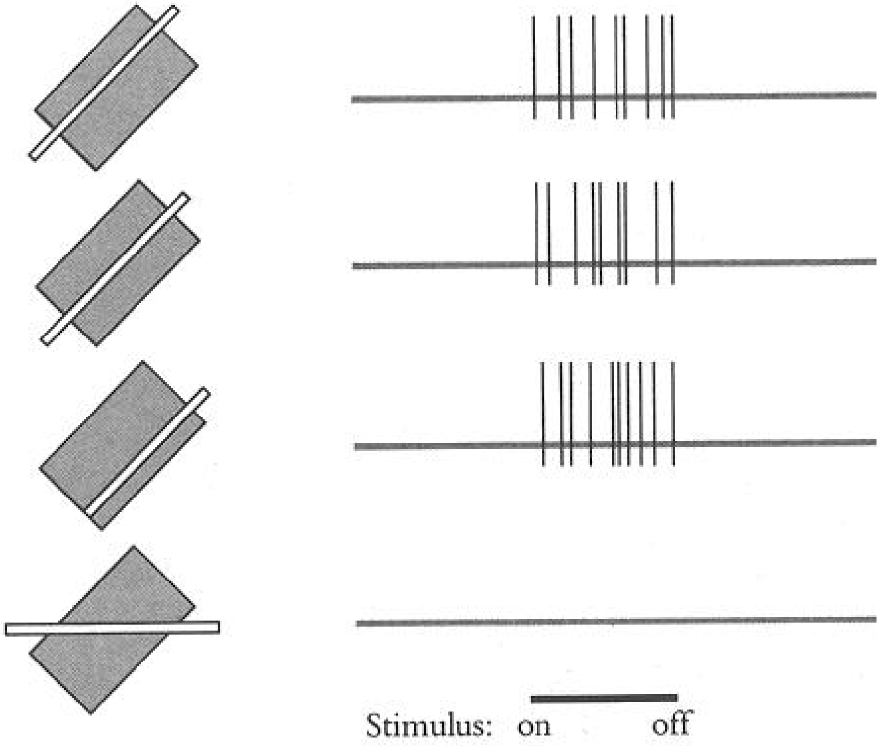
- have no clear division of excitatory and inhibitory regions inside their RFs
- a bar with width about one third of the RF width in the optimal orientation of the cell will evoke maximal response, independent of where it is placed

Simple cells and complex cells in V1

Simple Cells



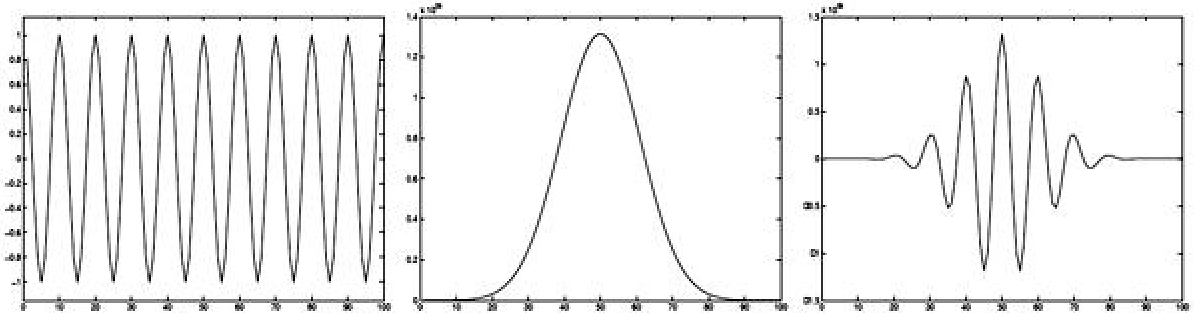
Complex cells



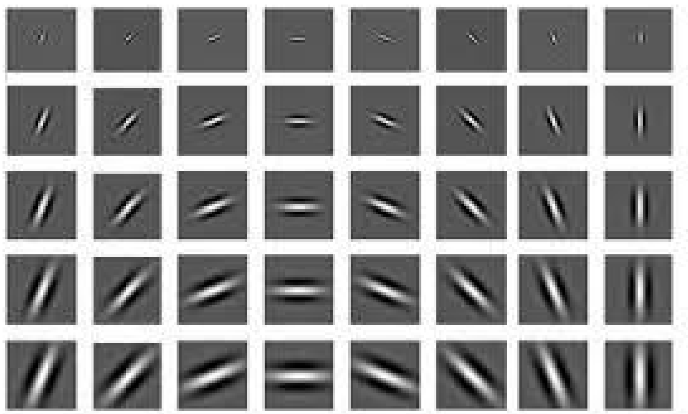
Gabor filters

sinusoidal wave multiplied by a Gaussian

1D

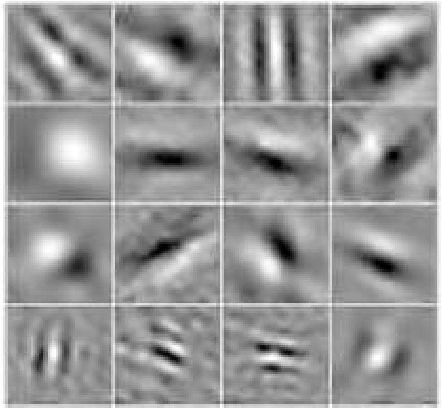


2D

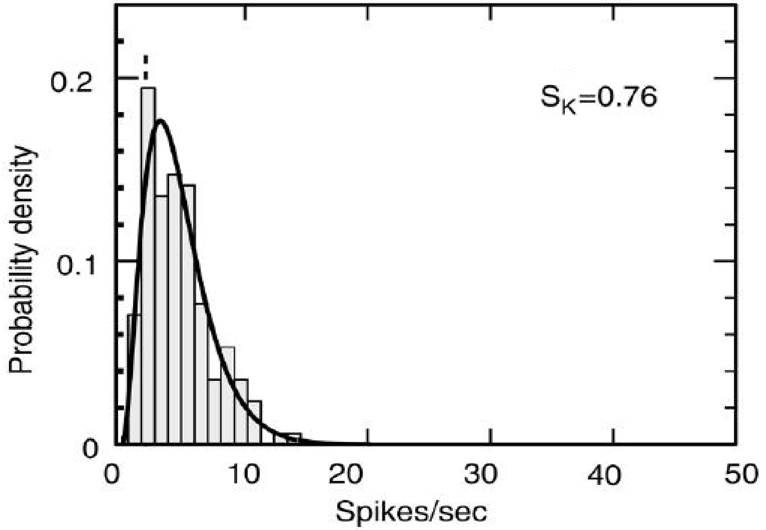


V1 simple cells' receptive fields and sparseness

- Their RF are similar to Gabors



- V1 simple cells fire sparsely



It therefore seems likely that, even in a strongly driven visual cortex, only a small fraction of neurons is working at any one time—between 1 in 25 and 1 in 63, with the latter the more probable value.

Lennie, 2003; Lehkya , 2005

Contents

Fixed point algorithms

Class of iterative algorithms

of shape $x_{t+1} = f(x_t)$ ⟵ update function

that stop when $x_{t+1} = x_t$

Algorithm Fixed point algorithms

---

initialize x randomly
define tol
define maxIter
for $t = 0$ **to** $t = maxIter$ **do**
    $x_{t+1} = f(x_t)$
    if $||x_t - x_{t+1}||^2 < tol$ **then**
        print `The algorithm converged`
        return x_{t+1}
    end if
    $t = t + 1$
end for
print `The algorithm did not converge`

Steepest gradient descent

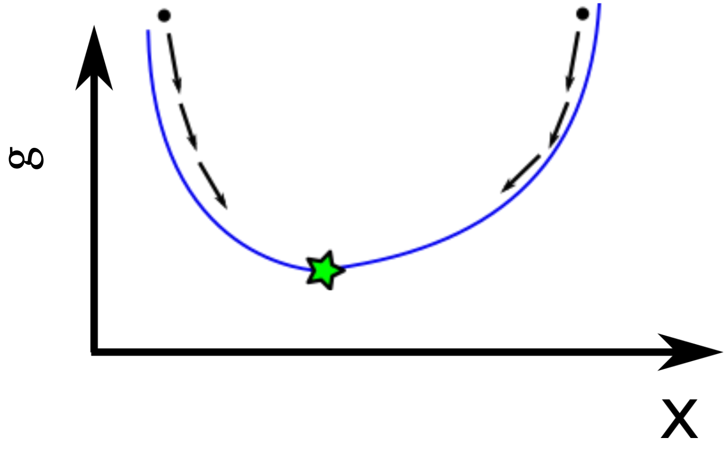
Goal: find minimum of g

Assumption: g is convex

$$\underbrace{}_{}f(x_t) = x_t - \gamma \nabla g(x_t)$$

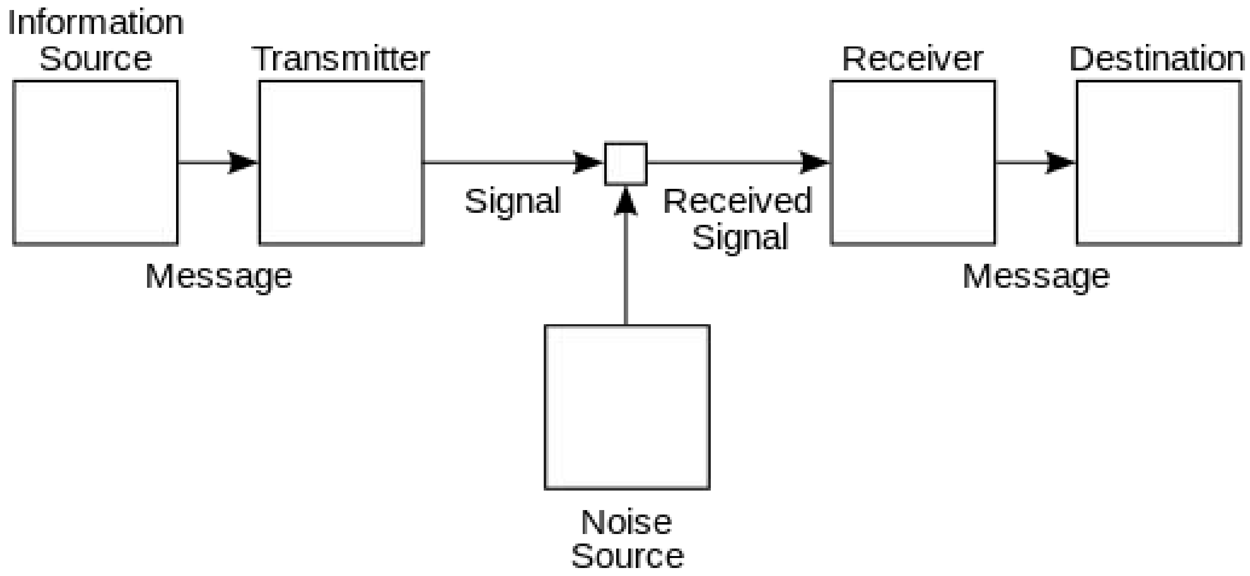
learning rate

gradient



How does it work: if $f(x_t) = x_t$ then $\nabla g(x_t) = 0 \Rightarrow x_t$ is a minimum of g.

Why efficient codes?



Example: Visual System

Information Source: Environment

Transmitter: Eye

Channel: Early visual system

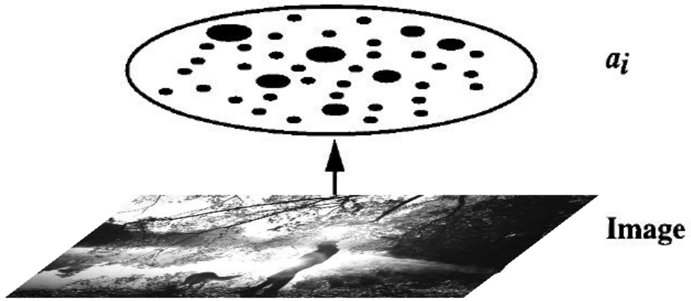
Noise: Unknown

Receiver: Higher areas (MT,TE,MIP,..)

Destination: Other brain areas (PMC,..)

Why sparse codes?

- Sparse codes are efficient, images are encoded representations that are most of the cases silent
- Firing is expensive, if each neuron fires/encodes sparsely images, then energy is saved



Sparseness weight

Energy function

$$E = \sum_t \left\{ x - \sum_i \alpha_i(x) W \right\}^2 + \lambda \sum_i S\left(\alpha_i(x)\right)$$

Sparseness measure

Sparse measurements

- $S(a_i) = (\prod (1+a_i^2))^{-1}$ (a~Cauchy distribution)
- $S(a_i) = -e^{-ai}$ (a~Exponential distribution)
- $S(a_i) = |a_i|$ (norm-1 of a)
- $S(a_i) = E[(a_i-E[a_i])^4/V[a_i]^2]$ (Kurtosis of a)

Sparse overcomplete learning
Gradient descent in 2 variables: a_i and W

Algorithm Pre-process

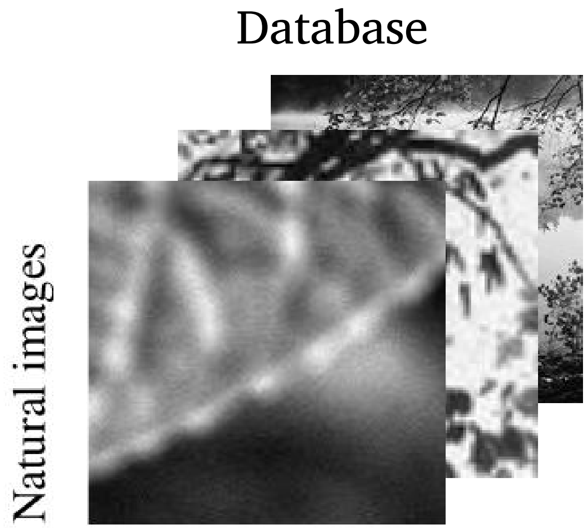
for all $x_i \in X$ **do**
  if $\dim(x_i) > 1$ **then**
    vectorize x_i
  end if
end for
for all $x_i \in X$ **do**
  $x_i = x_i - \mathbb{E}[x_i]$
end for
whiten X
return X

$$E = \sum_t \left\{ x - \sum_i \alpha_i(x) W \right\}^2 + \lambda \sum_i S\left(\alpha_i(x)\right)$$

Algorithm Sparse Overcomplete Coding

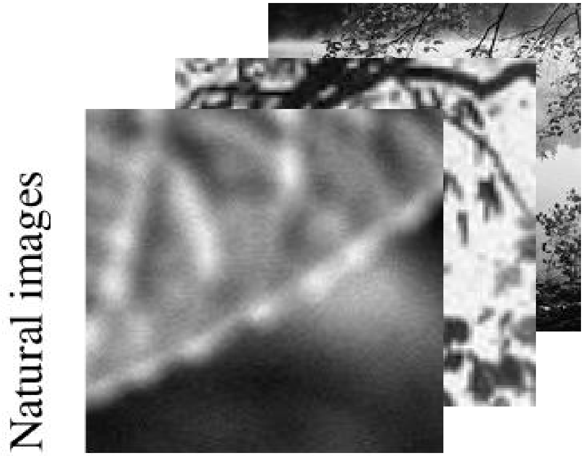
get X
pre-process X
define S, tol, maxIter
initialize W_0, α randomly
for $t = 0$ **to** $i = maxIter$ **do**
  $\alpha = \operatorname{argmin}_\alpha E$ (using gradient descent keeping W constant)
  $W_{t+1} = \operatorname{argmin}_W \mathbb{E}[E]$ (using gradient descent keeping α constant)
  if $||W_t - W_{t+1}||$ <tol **then**
    print `The algorithm converged`
    return W_{t+1}
  end if
  $t = t + 1$
end for
print `The algorithm did not converge`

Sparse overcomplete code of natural images

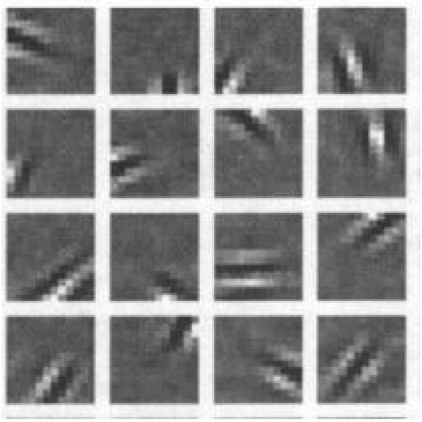


Sparse overcomplete code of natural images

Database



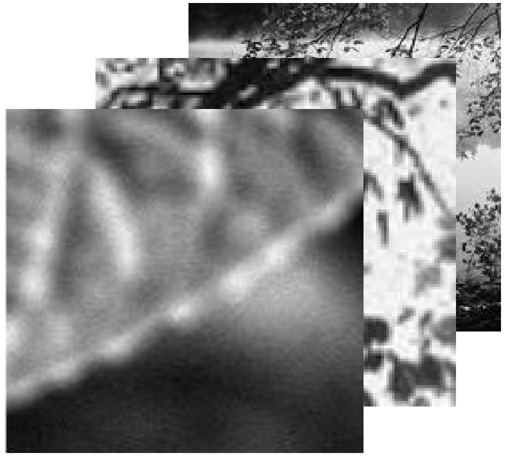
Sparse decoders



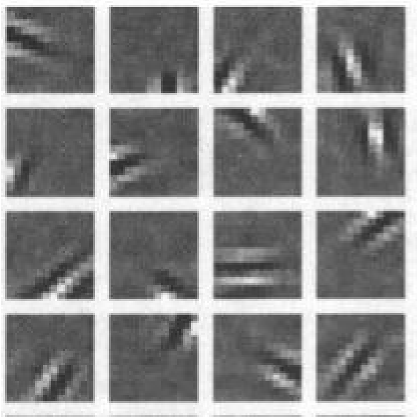
Sparse overcomplete code of natural images

Database

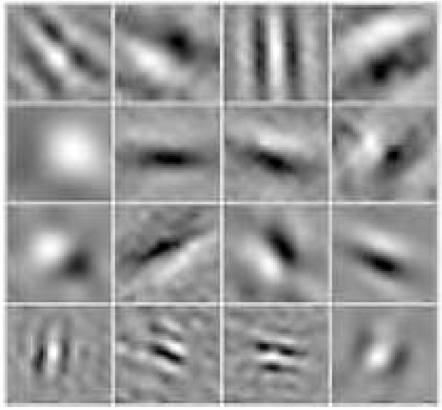
Natural images



Sparse decoders



V1 RFs



The receptive fields of V1 simple cells have such shape that they represent efficiently the natural images!

Contents

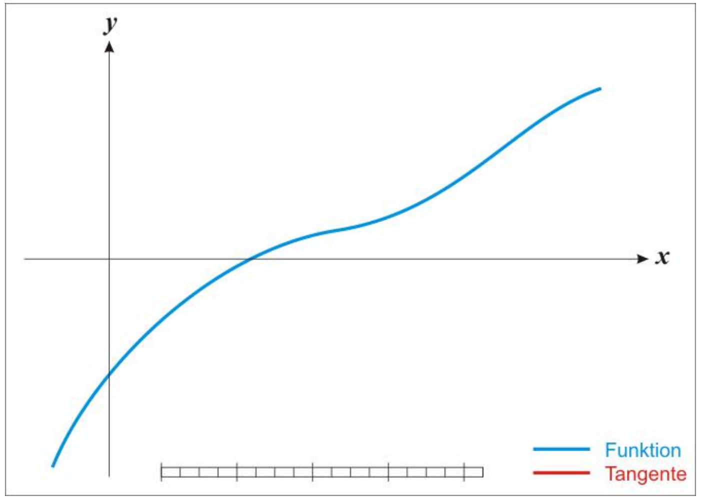
Newton method

Goal: find zero of g

Assumption: g' is non zero

$$f(x_t) = x_t - \frac{g(x_t)}{\nabla g(x_t)}$$

gradient



How does it work: if $f(x_t) = x_t$ then $g(x_t) = 0 \Rightarrow x_t$ is a zero of g.

Fixed point algorithms

steepest gradient descent

- linear convergence
- g must be convex
- more robust to errors

Newton method

- possible quadratic convergence
- more calculations
- g' must be non-zero
- sensitive to errors

Cocktail party problem

What is he saying?

Cocktail party problem

What is he saying?

sources

observations

$$x_1 = a_{11}s_1 + a_{12}s_2 + \cdots + a_{1k}s_k$$

$$x_2 = a_{21}s_1 + a_{22}s_2 + \cdots + a_{2k}s_k$$

$$\vdots$$

$$x_k = a_{k1}s_1 + a_{k2}s_2 + \cdots + a_{kk}s_k$$

linear
encoder

Cocktail party problem

What is he saying?

sources

observations ⟵ $x = A\,s$

linear
encoder

Cocktail party problem

What is he saying?

sources

observations $\longleftarrow x = A\, s$

linear encoder

linear independent decoded

Goal: to find W such $y = W\, x \; = s$

Non-Gaussianity is independence

Goal: to find W such $y = W x = WAs = s$

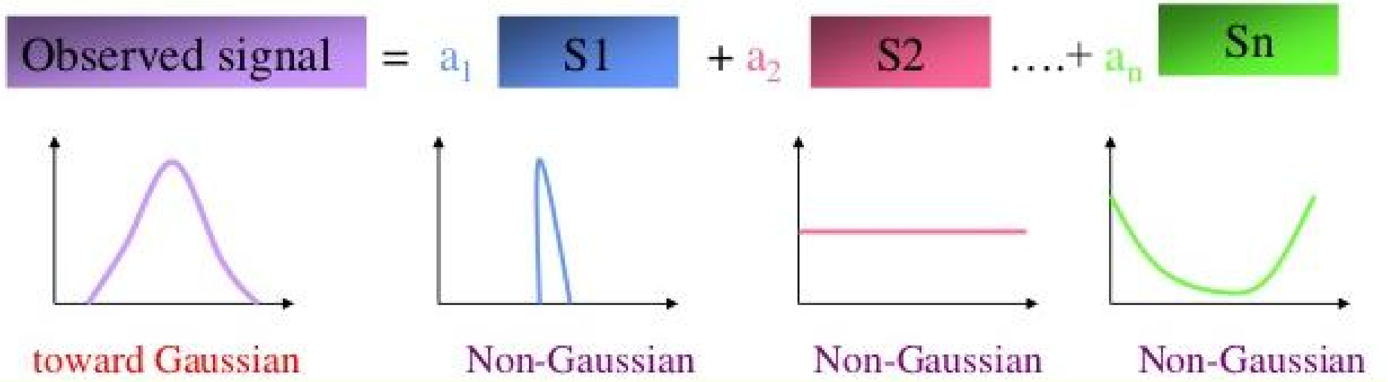
- $y = WAs$: y is a linear combination of s_i (independent variables)

Non-Gaussianity is independence

Goal: to find W such $y = Wx = WAs = s$

- $y = WAs$: y is a linear combination of s_i(independent variables)
- Central Limit Theorem:

$$\sum_{i=0}^{\infty} s_i \rightarrow \mathcal{N}(\mu, \sigma)$$

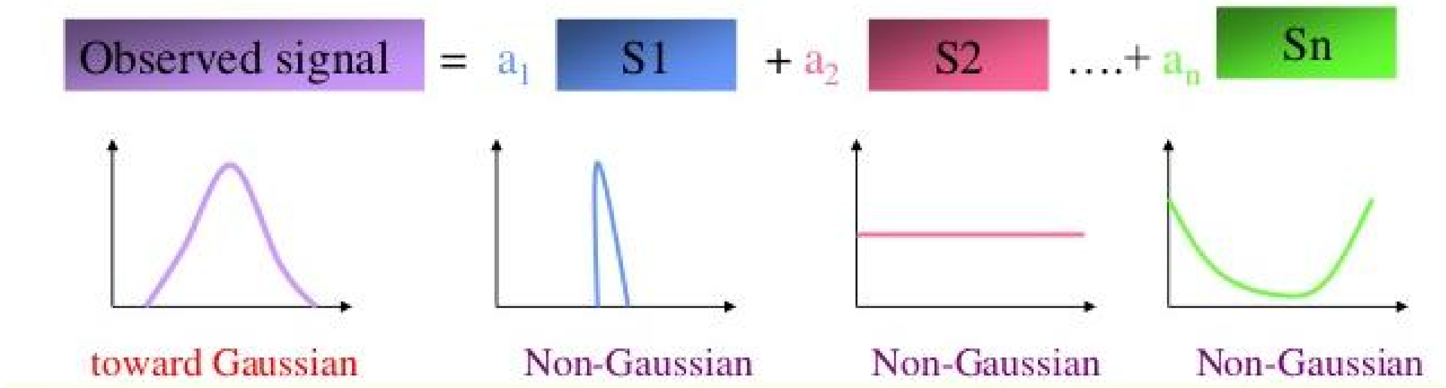


Non-Gaussianity is independence

Goal: to find W such $y = W x = WAs = s$

- $y = WAs$: y is a linear combination of s_i(independent variables)
- Central Limit Theorem:

$$\sum_{i=0}^{\infty} s_i \rightarrow \mathcal{N}(\mu, \sigma)$$



- y is the least Gaussian iff $y = s$

- New Goal: to find W such $y = Wx$ is non Gaussian

How to measure the non-Gaussianity?

- Kurtosis

- Neg entropy $J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$

Gaussian with the same variance as y

Non-quadratic function

Standard Gaussian

- Approximations of neg entropy $J(y_i) \approx c[E\{G(y_i)\} - E\{G(\nu)\}]^2$

Fast ICA algorithm

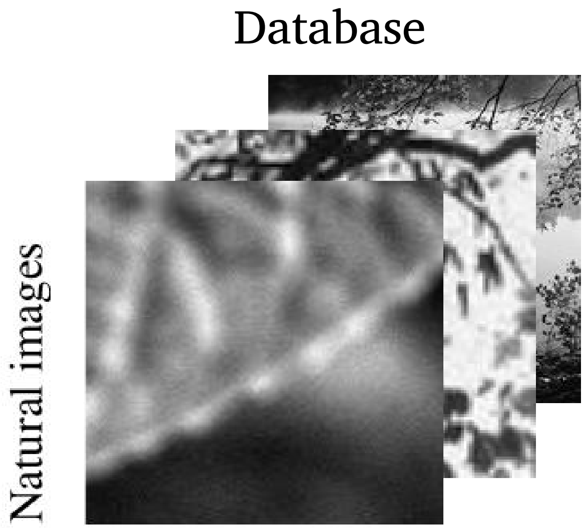
Algorithm Pre-process

for all $x_i \in X$ **do**
    if $\dim(x_i) > 1$ **then**
        vectorize x_i
    end if
end for
for all $x_i \in X$ **do**
    $x_i = x_i - \mathbb{E}[x_i]$
end for
whiten X
return X

Algorithm FAST ICA

get X
pre-process X
define G
initialize W randomly
$W = \text{argmax}_W \, J(WX)$ (using Newton method)
return W

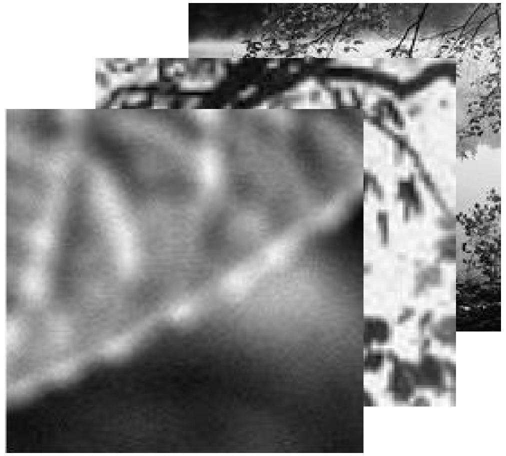
ICA natural images



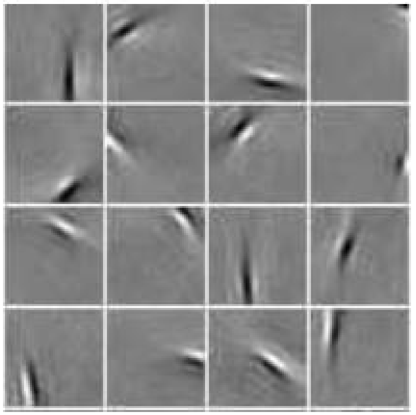
ICA natural images

Database

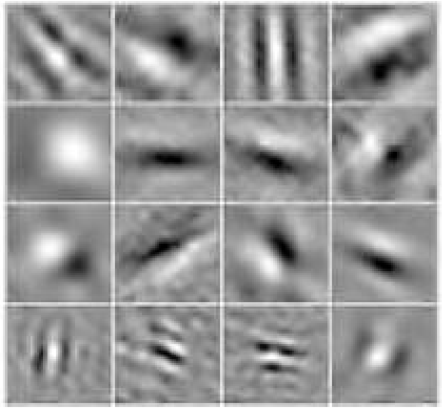
Natural images



Independent Components



V1 RFs



The receptive fields of V1 simple cells have such shape that they extract the independent sources of natural images!

V1-like methods

Sparse overcomplete coding

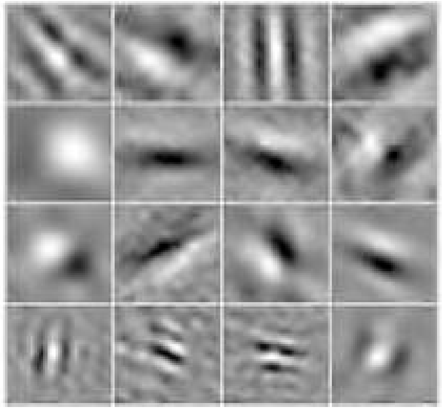
- Less simple to model
- Sparseness is specifically modeled
- Simple to implement
- No guaranty of convergence
- Encoders not on real time
- Decoders similar to RFs of V1 simple cells

Independent Components Analysis

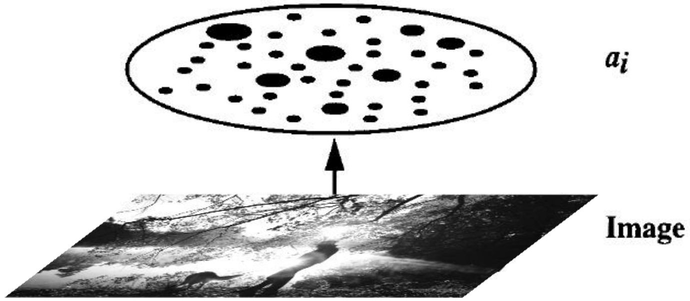
- Simple to model
- Sparseness emerges as consequence
- Simple to implement
- Mixing on real time
- Components similar to RFs of V1 simple cells

Summary

1. V1 simple cells receptive fields



2. Sparse overcomplete code of natural images



3. ICA of natural images

Bibliography

- Lennie, 2003, The Cost of Cortical Computation
- Lehkya et al, 2005, Selectivity and sparseness in the responses of striate complex cells
- Olshausen and Field, 1997, Sparse and overcomplete code: a strategy employed by V1?
- McDermott, 2009, The cocktail party problem
- Hyvärinen and Oja, 2000, Independent component analysis algorithms and applications
- Hyvärinen and Oja, 1997, A fast fixed-point algorithm for independent component analysis

Bibliography

- Hyvarinen et al, 2009, Natural Image Statistics
- Islam, 2013, A Comparative Study between ICA and PCA

https://pt.slideshare.net/SahidulIslam/a-comparative-study-between-ica-independent-component-analysis-and-pca-principal-component-analysis/8

- Bell and Sejnowski, 1997, The 'independent components' of natural scenes are edge filters