Sparse coding

- Sparse coding [Olshausen and Field, 1997]
  - Objective: Given input data {**x**}, search for a set of bases {$\mathbf{b}_j$} such that

$$\mathbf{x} = \sum_j s_j \mathbf{b}_j$$

where s_j are mostly zeros.

- Main intuition:
  - Build compact/succinct representations.
  - Learn interpretable and discriminative features.

Two objectives in sparse coding

- Preserve information
  - Minimize the reconstruction error

$$||\mathbf{x}^{(i)} - \sum_j s_j^{(i)} \mathbf{b}_j||^2$$

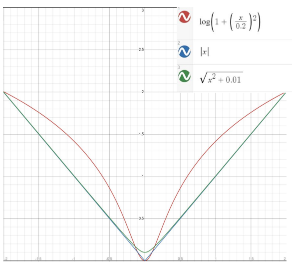
- Sparseness of coefficients
  - Minimize the sparsity penalty

$$\sum_{ij} \Psi\left(s_j^{(i)}\right)$$

Sparsity penalty

Many choices for inducing (approximately) sparse coefficients:

$$\Psi(s) = \begin{cases} I(s \neq 0) & \mathrm{L}_0 \text{ penalty} \\ \log(1+s^2) & \log \text{ penalty} \\ |s| & \mathrm{L}_1 \text{ penalty} \\ \sqrt{s^2+\epsilon} & \text{epsilon } \mathrm{L}_1 \text{ penalty} \end{cases}$$



Learning bases: optimization

Given input data {$\mathbf{x}^{(1)}$, …, $\mathbf{x}^{(N)}$}, we want to find good bases {$\mathbf{b}_1$, …, $\mathbf{b}_m$}:

$$\boxed{\min_{\mathbf{b},\mathbf{s}} = \underbrace{\sum_i ||\mathbf{x}^{(i)} - \sum_j s_j^{(i)} \mathbf{b}_j||_2^2}_{\text{Reconstruction error}} + \beta \underbrace{\sum_i ||\mathbf{s}^{(i)}||_1}_{\text{Sparsity penalty}} \quad \forall j: \ \|b_j\| \leq 1 \ \text{Normalization constraint}}$$

Tradeoff between “quality of approximation” and “sparsity” (compactness).

Sparse Coding Implementation

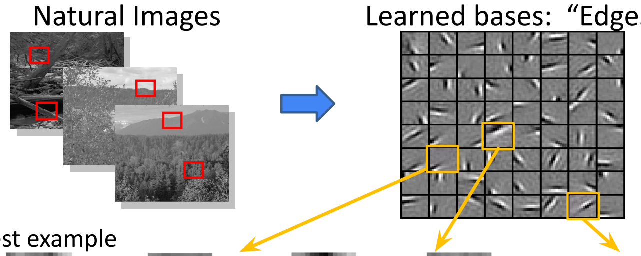
$$\min_{\mathbf{b},\mathbf{s}} = \underbrace{\sum_i ||\mathbf{x}^{(i)} - \sum_j s_j^{(i)} \mathbf{b}_j||_2^2}_{\text{Reconstruction error}} + \beta \underbrace{\sum_i ||\mathbf{s}^{(i)}||_1}_{\text{Sparsity penalty}}$$

$\forall j: \ \|b_j\| \leq 1$ Normalization constraint

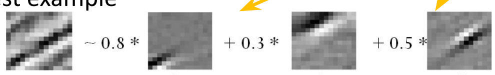
- Alternating optimization:
  - Optimize with either **b** (bases) and **s** (coefficients) while fixing others.
  - The problem is convex for each sub-problems, but not jointly convex
- Fast inference and learning algorithm
  - http://web.eecs.umich.edu/~honglak/softwares/nips06-sparsecoding.htm

Sparse coding for images

Natural Images Learned bases: “Edges”



Test example



$x \sim 0.8 * b_{36} + 0.3 * b_{42} + 0.5 * b_{65}$

[0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …] = coefficients (feature representation)

Compact & easily interpretable

Sparse coding: summary

- Sparse coding is a popular “dictionary learning” method in machine learning
- It can learn a large overcomplete set of bases; the coefficients are not a linear function of input.
- The coefficients can be used as features; Successful applications in object recognition, and many other tasks
- Limitation: computationally expensive; representation is unstable.

PCA, ICA and Sparse Coding Summary

- **PCA (Principal Component Analysis)** reduces dimensionality by identifying orthogonal directions of maximum variance to efficiently represent data.
- **ICA (Independent Component Analysis)** decomposes data into statistically independent components by maximizing non-Gaussianity.
- **Sparse coding** represents data as sparse linear combinations of basis functions, aiming for efficient encoding with minimal active components.

PCA and **ICA** both seek linear transformations of data, but **PCA** focuses on variance maximization, while ICA prioritizes statistical independence.

Sparse coding extends **ICA** by not only seeking statistically independent components but explicitly enforcing a sparsity constraint to ensure most coefficients in the representation are near-zero, promoting interpretability and efficiency.