Fast Inference in Sparse Coding Algorithms with Applications to Object Recognition

Koray Kavukcuoglu, Marc'Aurelio Ranzato, Yann LeCun
Courant Institute of Mathematical Sciences
New York University

Overview

- Adaptive sparse coding methods learn an overcomplete set of basis functions for linearly representing an input signal
- Sparse features are more likely to be linearly separable in a high dimensional space
- Sparse coding methods learn good local feature extractors for natural images
- Limited application due to prohibitive cost of iterative optimization
- We propose Predictive Sparse Decomposition (PSD) that can both learn an overcomplete basis set and provide a smooth approximations to optimal sparse representations.

Sparse Coding Algorithms

$y \in \mathcal{R}^n$ (Signal) $\leftarrow$ $B \in \mathcal{R}^{n \times m}$ (Basis), $z \in \mathcal{R}^m$ (Code)

$$\min ||z||_0 \ \text{ s.t. } \ y = Bz$$

- Relax into a convex penalty function (Basis Pursuit)
- Reconstruction error + sparsity penalty

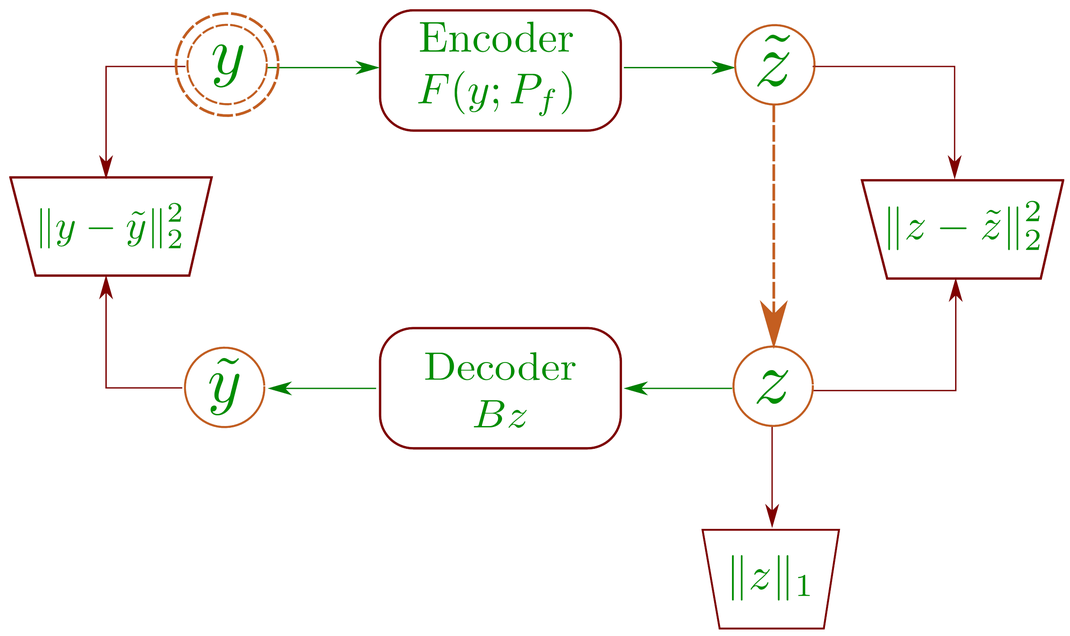
$$\frac{1}{2}||y - Bz||_2^2 + \lambda||z||_1$$

- Learn basis *B* and compute sparse representation *z*

The Algorithm

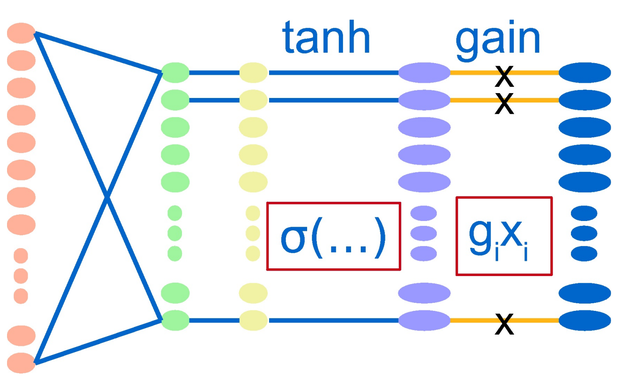
- Add an additional term in the penalty function to represent the encoder.
- Drive the system to prefer predictable representations among many possible ones

$$\|y - Bz\|_2^2 + \lambda\|z\|_1 + \alpha\|z - F(y; P_f)\|_2^2 \quad (*)$$



Optimal Inference

- Compute the optimal sparse representation *z* in (*). We use gradient descent with a diagonal approximation to the Hessian.



Learning

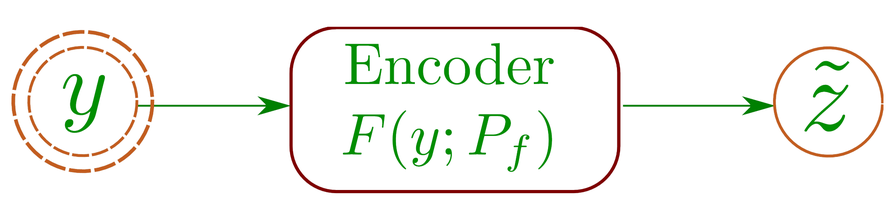
- Compute basis functions *B* and *encoder parameters* P_f.
  1) Keeping *B* and P_f fixed, minimize (*) with respect to *z*, starting from the initial value provided by $F(y;P_f)$.
  2) Using the optimal value of *z*, perform one step of online gradient descent to minimize (*) with respect to *B* and P_f.
  3) Normalize columns of *B* to unit norm.

Analogies

- For $\alpha = 0$ learning algorithm becomes similar to Olshausen and Field's sparse coding algorithm. A regressor can then be trained separately
- For $\alpha \in (0, +\infty)$ parameters are updated taking into account also the constraint on the representation, using the same principle employed by SESM training
- For $\alpha \to +\infty$ $z = F(y ; P_f)$, and the model becomes similar to an auto-encoder neural network with a sparsity regularization term acting on the internal representation *z*

Approximate Inference

- Compute an approximate representation *z*, as output by the feedforward encoder $F(y;P_f)$.

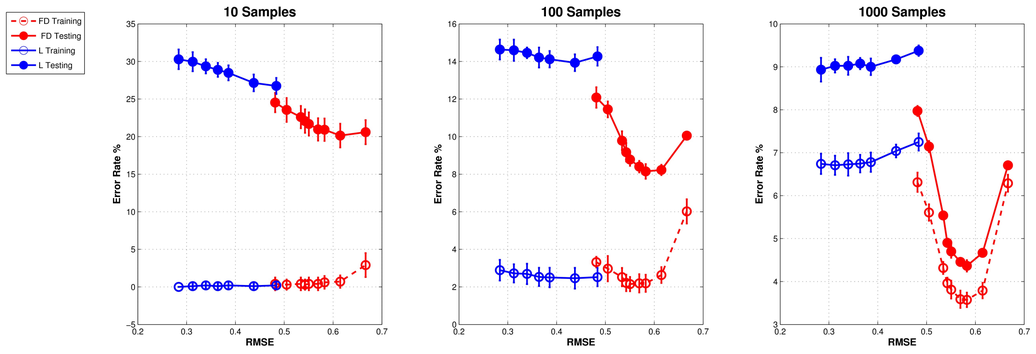


$$F(y; G, W, D) = G tanh(Wy + D)$$

Comparison on MNIST

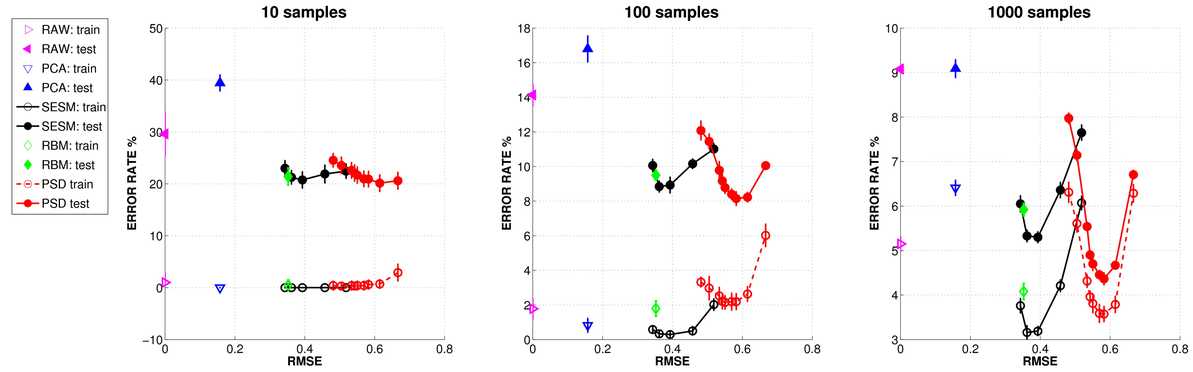
Encoder Selection

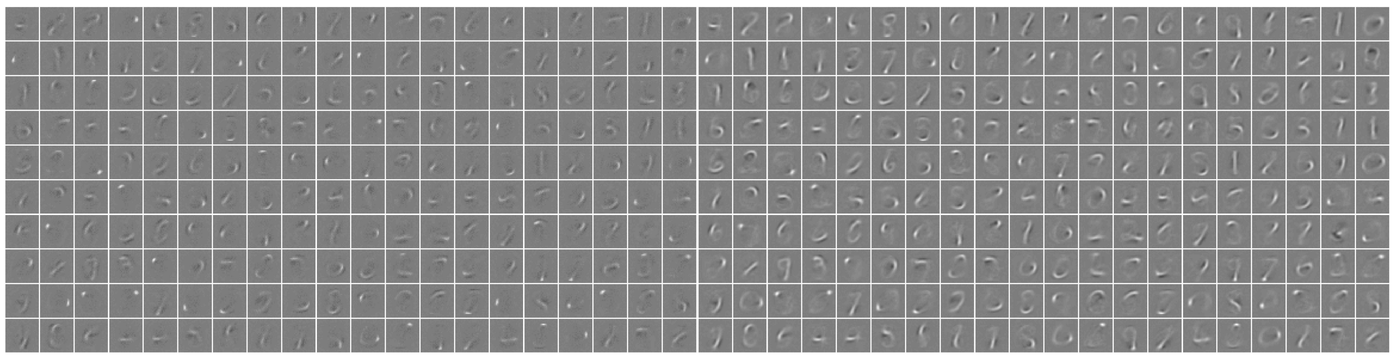
- Unsupervised machine is trained with 200 code units
- A linear classifier is trained on the features
- Classification accuracy is compared between linear encoder and our encoder



Recognition Performance

- Comparing recognition performance using raw pixel data, PCA, SESM and RBM.
- Our system achieves the worst RMSE but best recognition rate.





Unsupervised Training

- PSD is trained on 20,000 28x28 grayscale images from MNIST training dataset.
- The training and testing datasets are scaled by dividing each patch by the standard deviation of the training dataset

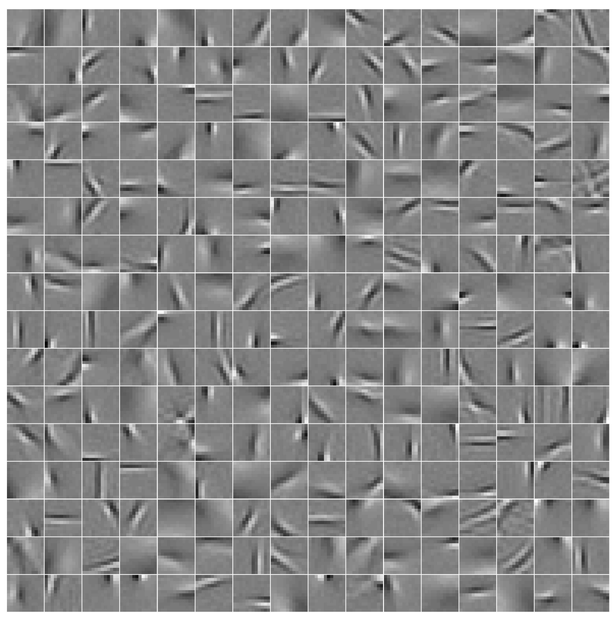
Comparison with Exact Algorithms

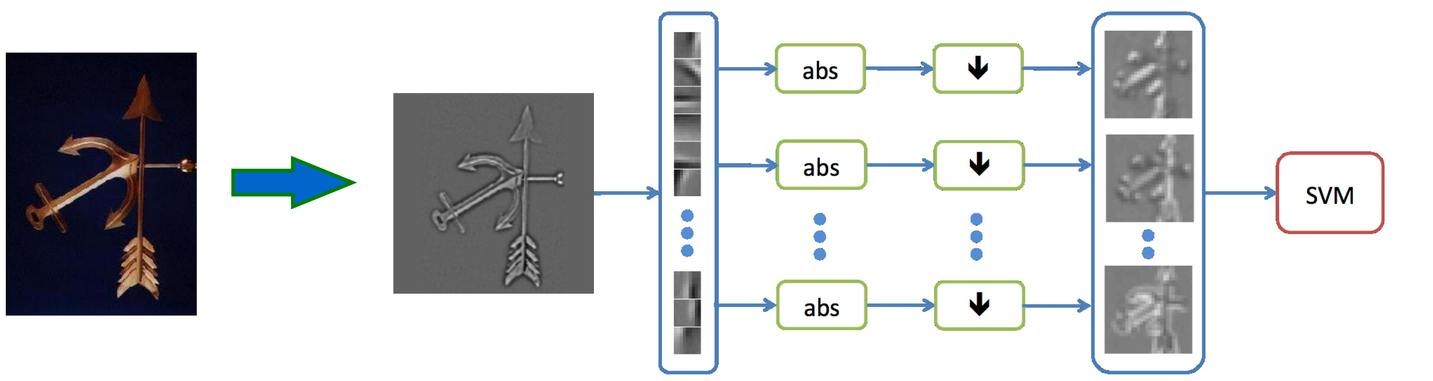
Caltech101 Preprocessing

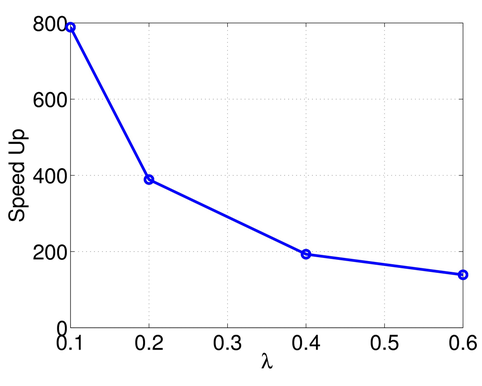
1) Each image is converted to grayscale
2) Downsampled and zero padded to 151x151
3) Centered by removing the image mean and scaling by the image standard deviation
4) Locally normalized using a Gaussian weighted 9x9 window

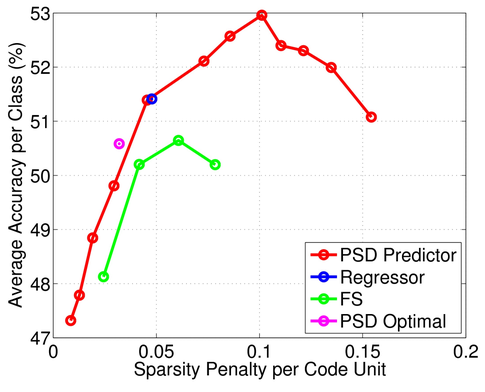
Caltech101 Recognition

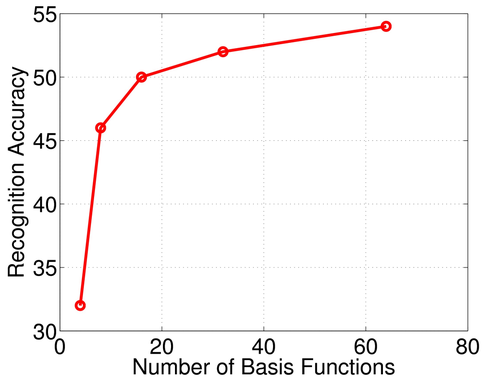
1) Extract 64 feature maps using PSD Predictor and Feature Sign
2) Rectify with absolute value
3) Average downsample to 30x30
4) Recognize the object in the image using a linear SVM









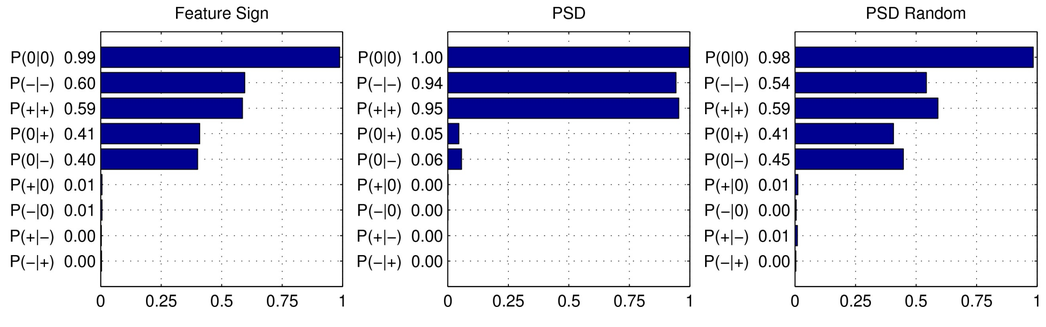


Caltech101 Results

- This simple architecture achieves a recognition rate of 53%
- At least 100 times faster than exact sparse coding algorithm, more than 800 times for best recognition performance
- Very inefficient for convolutional applications and the recognition accuracy saturates around 64 feature maps.

Stability

- Compare the stability of the representations under naturally changing input signals
- Extract 784 uniformly distributed patches from all 400 frames of the «*foreman test video*»
- Stability is measured by the number of times a unit of the representation changes its sign, either negative, zero or positive, between two consecutive frames
- PSD representation is thresholded in such a way that the number of zeros equals that of FS (roughly 96%).



Conditional probabilities for sign transitions between two consecutive frames (e.g. P (-|+) shows the conditional probability of a unit being negative given that it was positive in the previous frame)