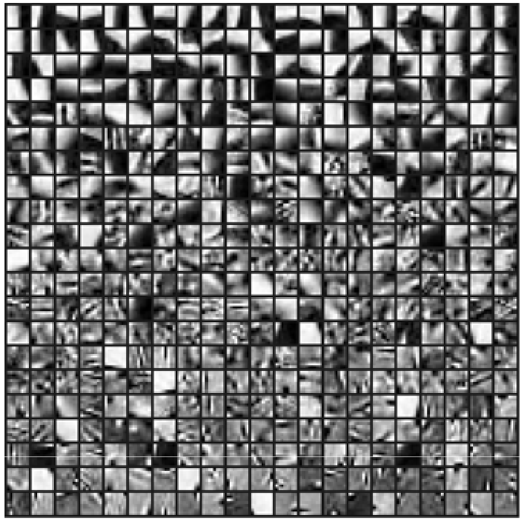
Matrix Factorization Applications

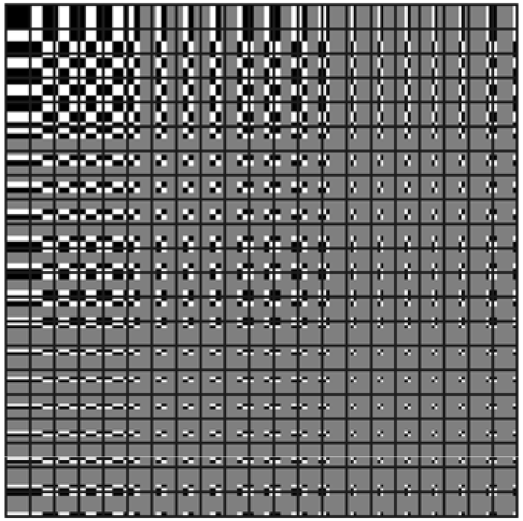
Angshul Majumdar

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

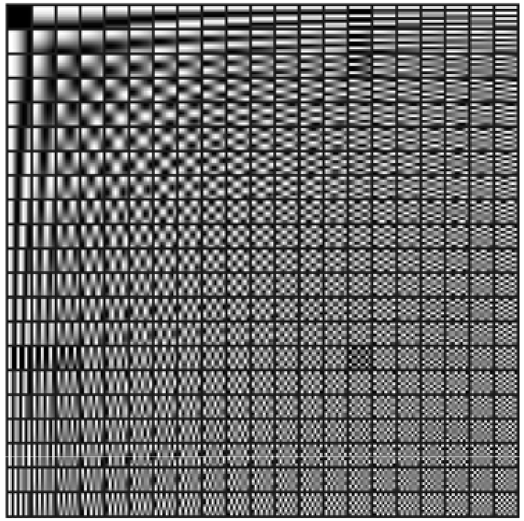
Learned Dictionary



Learned



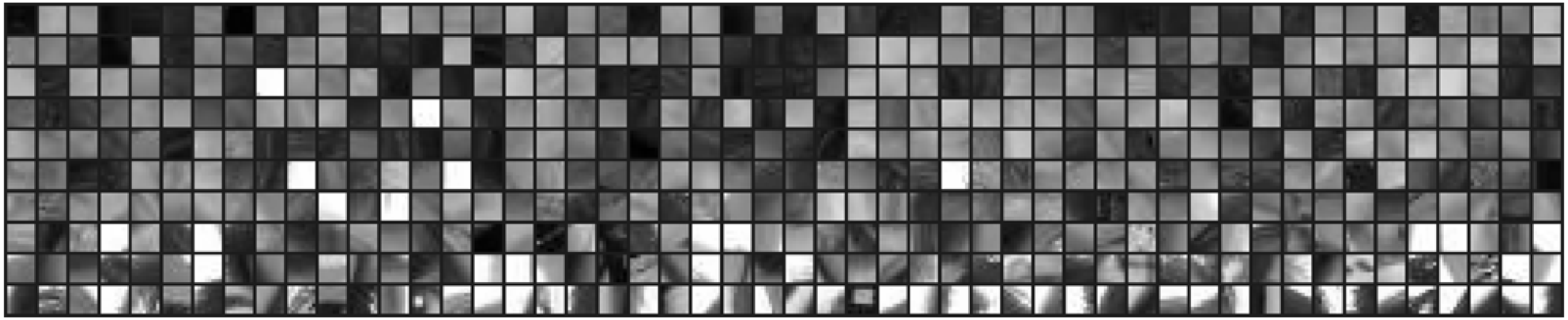
Haar



DCT

M. Aharon, M. Elad and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", IEEE Transactions on Signal Processing, Vol. 54 (11), pp. 4311–4322, 2006.

MOD and ML-DL



M. Aharon, M. Elad and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", IEEE Transactions on Signal Processing, Vol. 54 (11), pp. 4311–4322, 2006.

- Train a dictionary D to represent the training data X.
- MOD and ML-DL Solves: $\min_{D,Z} \|X - DZ\|_F^2$
- Standard Matrix Factorization Problem
- Dictionary atoms need to be normalized

KSVD and Others

- Train a dictionary D to represent the training data sparsely.

$$\min_{D,Z} \|X - DZ\|_F^2 + \lambda \|D\|_F^2 \; s.t. \|Z\|_0 \leq \tau$$

  - Sparse Coding
  - Codebook / Dictionary update

- KSVD is an elegant solution to the problem. But there are others.

Applications (A few of them)

- Denoising
- Super-resolution
- Inpainting
- Demosaicing
- Inverse Half-toning

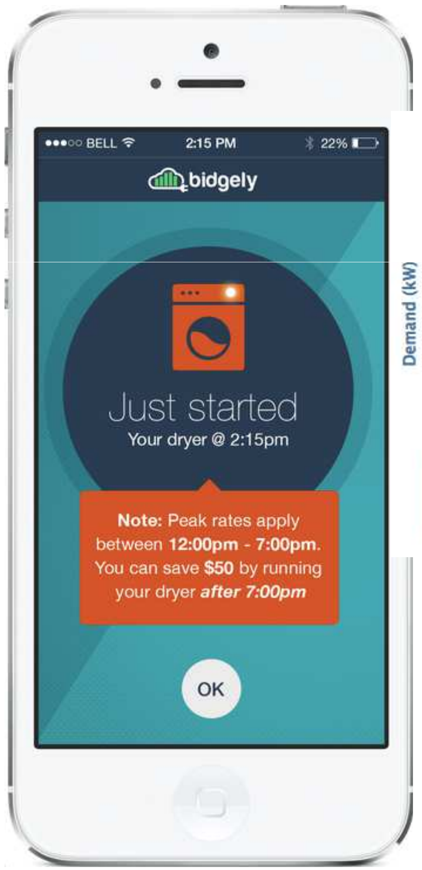
- Energy Disaggregation

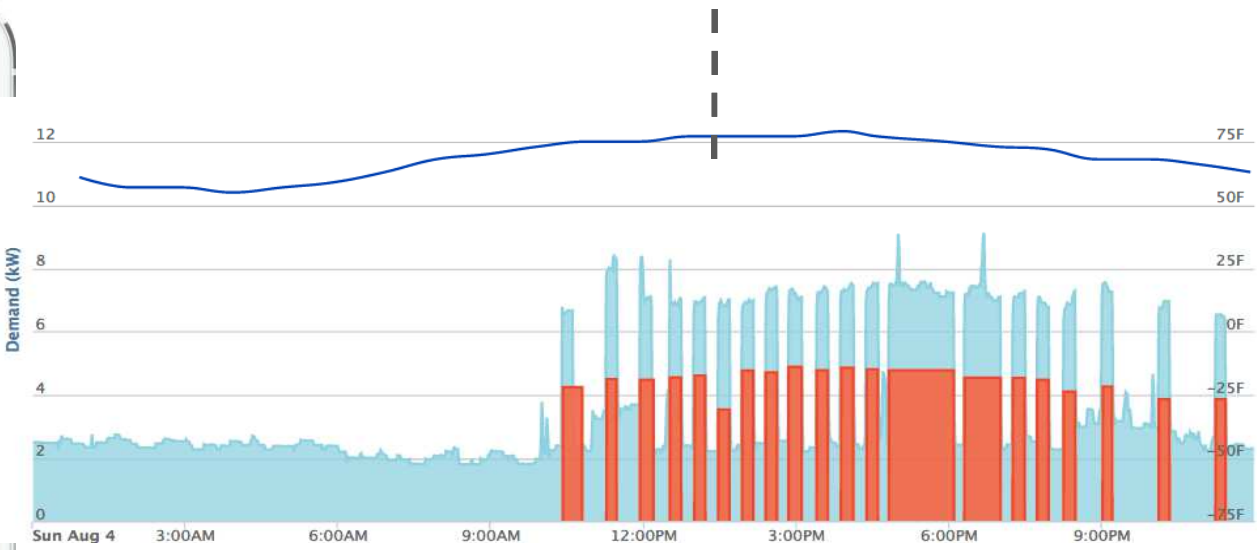
- Computer Vision

What is it?

- Extract appliance level energy consumption from aggregate data

Thermostat set point





Thermostat set point around 72°F

AC ran between 11am-9pm when a peak event in effect

!

Send mobile alerts when users turn on high load appliances during critical peak times of day

Disaggregation via sparse coding

- Learn a dictionary for each appliance.

Weeks / Households		Weeks / Households		Weeks / Households
Time Points $\{ X_1 \}$		Time Points $\{ X_2 \}$	…	Time Points $\{ X_N \}$
D_1		D_2		D_N

$$\min_{D,Z} \|X_i - D_i Z_i\|_F^2 + \lambda \|D_i\|_F^2 \; s.t. \|Z_i\|_0 \leq \tau, \forall i$$

- Can be solved using Standard KSVD

Disaggregation Contd.

- Assumption – Total power consumption follows a linear model: $X = \sum_i X_i$
- (Strictly speaking this is untrue!)
- For the aggregate data, the individual components are obtained as:

$$\min_Z \left\| X - [D_1 | \ldots | D_N] \begin{bmatrix} Z_1 \\ \ldots \\ Z_3 \end{bmatrix} \right\|_F^2 \; s.t. \|Z\|_0 \le \tau, \text{where } Z = [Z_1 | \ldots | Z_N]^T$$

$$\hat{X}_i = D_i Z_i$$

Energy Disaggregation – Typical Chart

Figure 1. Aggregate House 6, Day 1.

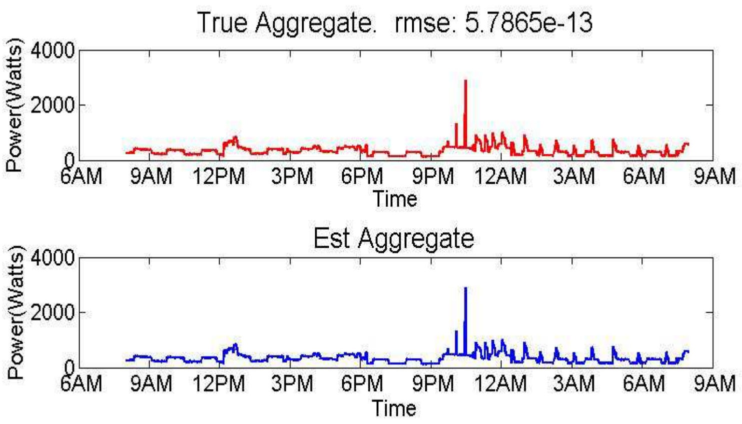


Figure 3. Estimate vs True device consumption

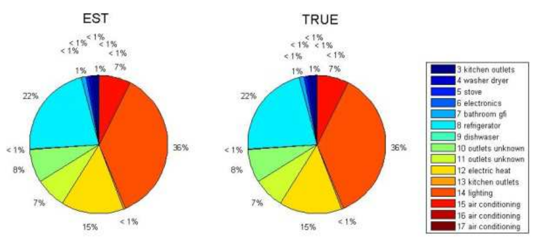
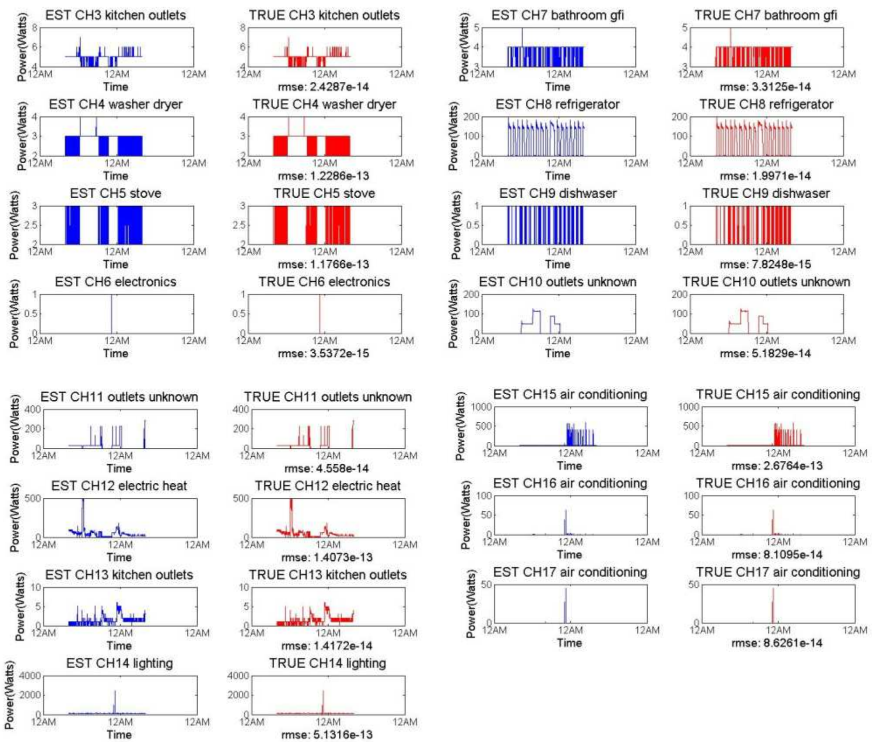
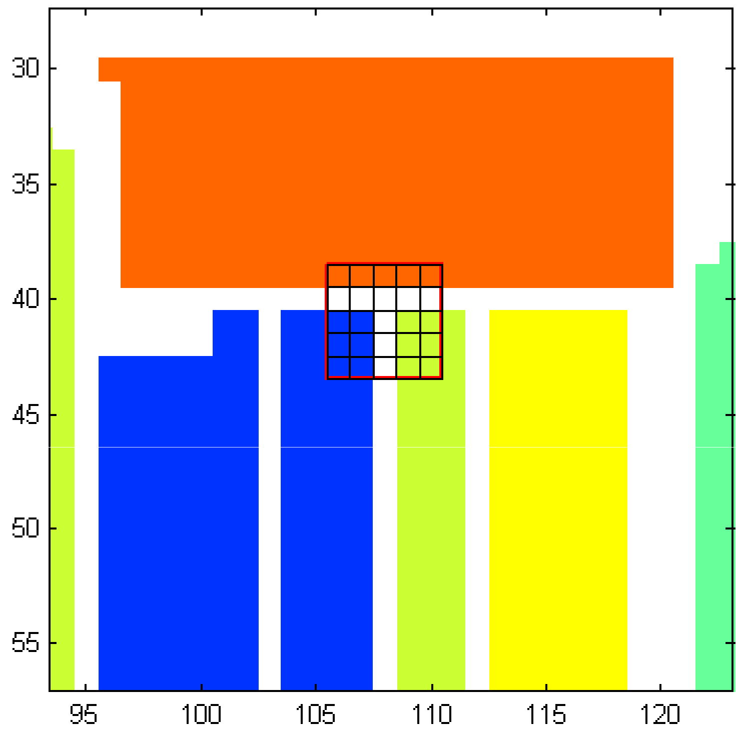


Figure 2. Individual Device Comparison. House 6, Day 1.



Spectral Unmixing

- Hyperspectral images – high spectral resolution, but low spatial resolution.
- Each ‘pixel’ corresponds to a mixture of several components.
- How to ‘unmix’?



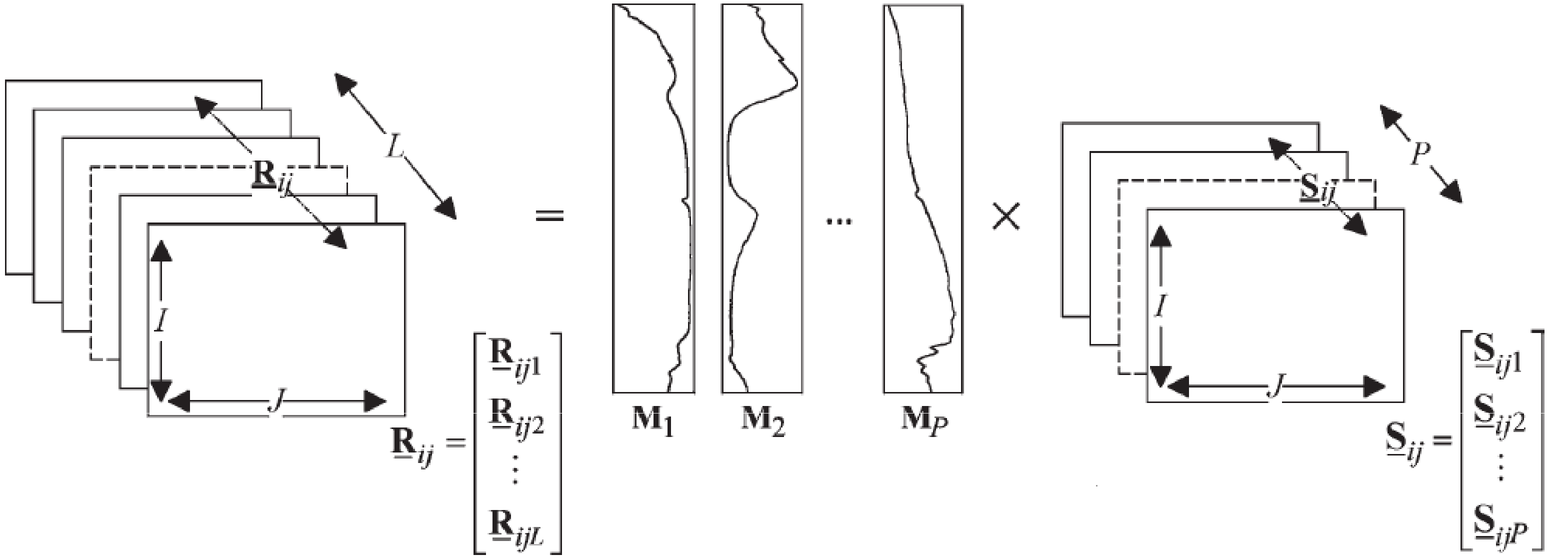
Endmember and Abundance

- The i^{th} pixel location is sampled at L-bands (say)

- Let P be the total number of possible materials. Every material will have a spectral signature at each band. This constitutes the endmember matrix of size LXP.

- The abundance specifies 'how much' of each material is present in the sampled pixel.

A schematic representation



A Linear Mixing Model (Approximation)

$$\text{Unmixing via solving: } \min_{M,S} \left\| R - MS \right\|_F^2$$

Sparsity

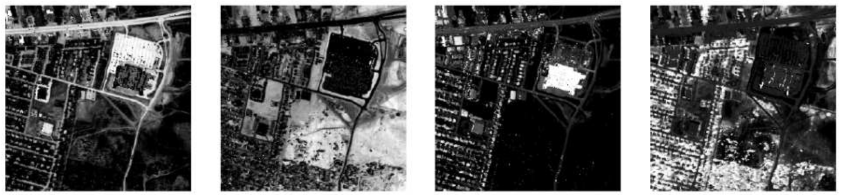
- All the endmembers cannot be present in all the pixels, only a few can.
- The abundance should be sparse.

Sparse NMF: $\min_{M,S} \|R - MS\|_F^2 + \lambda \|S\|_p$

Fig. 10. Urban HYDICE hyperspectral dataset at band 80.

(a) Asphalt (b) Grass (c) Roof (d) Tree

groundtruth



(a) Asphalt (b) Grass (c) Roof (d) Tree

estimated

Robustness

- The linear model is an approximation.
- Modelling the non-linearity as an error:

$$R = MS + E + N$$
$$E - nonlinearity \ (sparse)$$
$$N - white\ noise$$

  - Linear model holds most of the times, Therefore E is mostly zero.
  - Non-linearity arises for ‘few pixels’ owing to scattering.
- Solution similar to Robust PCA

Modified PCP

- No constraint on endmember (M)
- Abundance (S) is sparse
- E is group-sparse (certain rows of R, corresponding to pixels with non-linearity are non-zeroes)

$$\min_{M,S,E} \left\| R - MS - E \right\|_F^2 + \lambda \left\| S \right\|_1 + \mu \left\| E \right\|_{2,1}$$

$$\left\| E \right\|_{2,1} = \sum_j \left\| E^{j\rightarrow} \right\|_2$$

Information Retrieval

The $t = 6$ **terms**:

T1: bak(e,ing)
T2: recipes
T3: bread
T4: cake
T5: pastr(y,ies)
T6: pie

The $d = 5$ **document** titles:

D1: How to Bake Bread Without Recipes
D2: The Classic Art of Viennese Pastry
D3: Numerical Recipes: The Art of Scientific Computing
D4: Breads, Pastries, Pies and Cakes: Quantity Baking Recipes
D5: Pastry: A Book of Best French Recipes

The 6×5 term-by-document matrix before normalization, where the element $\hat{a}_{ij}$ is the number of times term i appears in document title j:

$$\hat{A} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The 6×5 term-by-document matrix with unit columns:

$$A = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1.0000 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$$

- *term-by-document* matrix A
  - Columns : Document Vectors
  - Rows : Term Vectors
- A(i,j) = weighted frequency of the **i**$^{\text{th}}$ **term** associated with the **j**$^{\text{th}}$ **document**

Query Evaluation

- *Query Matching – Finding most* geometrically close vectors in the matrix to the query vector
- Usually calculates the *cosine of the angle* between the vectors
- If *cosine between* ***query and single document*** vector > ***threshold*** → Relevant document found!
- Example
  - Suppose query *"baking bread"*
  - Query Vector **q = [1,0,1,0,0,0]T**
  - ***threshold = 0.5***
  - 1st and 4th documents retrieved

VSM

- Vector Space Model – 60’s and 70’s
- Gerard Salton’s Information Retrieval System
- Dubbed –

SMART: System for the Mechanical Analysis and Retrieval of Text

OR

Salton’s Magical Automatic Retriever of Text

Latent Semantic Indexing

- Noise in A - synonyms and polysems
- Reduce noise – low-rank approximation of A
- Introduced by Susan Dumais
- Two patents for Bell / Telcordia!
  - Computer information retrieval using latent semantic structure. U.S. Patent No. 4,839,853, June 13, 1989.
  - Computerized cross-language document retrieval using latent semantic indexing. U.S. Patent No. 5,301,109, April 5, 1994.
- Retrieval mechanism doesn't change (just change of basis)

NNMF

- Learn a basis to represent the term-document matrix A: A=WH
- W – dictionary/basis and H – coefficient
- Very similar to the SVD model. W can be seen as scaled version of the left singular vectors.
- Once W (explanatory variables) is learnt, it can be used to analyze new ‘documents’ .

$$\mathbf{doc}_5 \approx \overset{\mathbf{w}_9}{\begin{pmatrix} \text{fatty} \\ \text{glucose} \\ \text{acids} \\ \text{ffa} \\ \text{insulin} \\ \vdots \end{pmatrix}} .1646 + \overset{\mathbf{w}_6}{\begin{pmatrix} \text{kidney} \\ \text{marrow} \\ \text{dna} \\ \text{cells} \\ \text{nephr.} \\ \vdots \end{pmatrix}} .0103 + \overset{\mathbf{w}_7}{\begin{pmatrix} \text{hormone} \\ \text{growth} \\ \text{hgh} \\ \text{pituitary} \\ \text{mg} \\ \vdots \end{pmatrix}} .0045 + \cdots$$

Example on Med Dataset with 10 atoms

MRI Reconstruction Basics

- You might already know – MRI data is captured in K-space (Fourier frequency domain)

$$y = Fx + \eta$$

- The problem is to accelerate the K-space scan
- The K-space is under-sampled: $y = RFx + \eta$
- The problem is to solve the under-determined problem.
- Use Compressed Sensing

Dynamic MRI

- The K-space is acquired for each frame: $y_t = RFx_t + \eta$
- Compressed Sensing formulations can be used – exploits spatio-temporal redundancy in the form of sparse representation.
- Alternate Approach – Low Rank Model

$$X = \left[\underbrace{x_1 \,|\, \ldots \,|\, x_T}_{time \rightarrow} \right]$$

- Temporal correlations lead to rank deficiency

Recovery Techniques – Matrix Factorization

- Matrix Factorization (Halder at al):

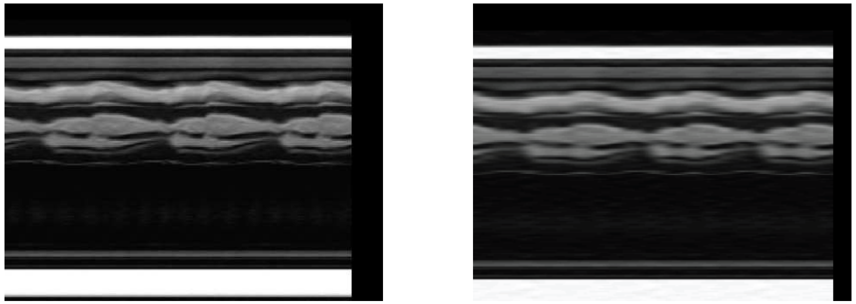
$$\min_{U,V} \left\|Y - FVU\right\|_F^2 + \lambda\left(\left\|U\right\|_F^2 + \left\|V\right\|_F^2\right)$$

$$X = UV$$

$$Interpretation -$$

U – temporal basis functions

V – coeffcients



Temporal evolution (1000 frames) of a vertical line passing through the left ventricle – Groundtruth and Rank-8 Reconstruction

Blind Compressed Sensing

- Interprets as a sparse regression problem.
  - U – basis (allows for more basis than MF; typically 40+)
  - V – coefficients (sparse – since only few basis are required)

$$\min_{U,V} \|Y - FVU\|_F^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_1$$

- Proposed by Jacobs et al

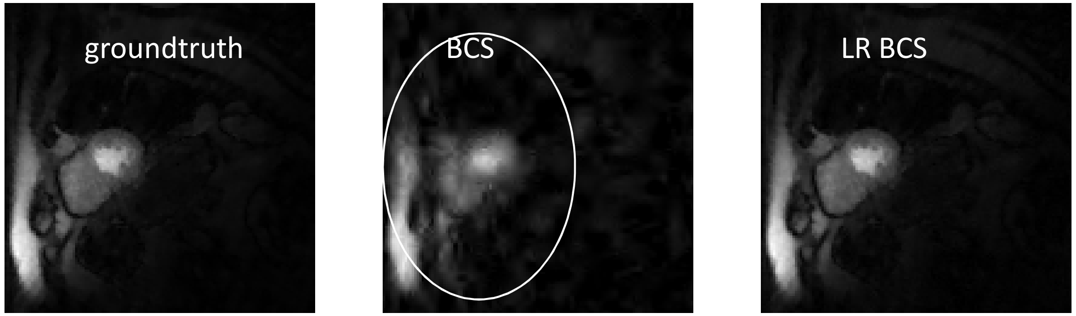
Low-rank BCS

- X-low rank (BCS accounts for it implicitly)
- Allow richer (over complete dictionary) similar to K-SVD.
- Analysis prior formulation:

$$\min_{D,X} \|Y - FX\|_F^2 + \lambda_1 \|D\|_F^2 + \lambda_2 \|DX\|_1 + \lambda_3 \|X\|_{NN}$$

$D -$ spatial dictionary

- Synthesis prior: $\min_{U,V} \|Y - FUV\|_F^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_1 + \lambda_2 \|V\|_{NN}$



FROM RECONSTRUCTION TO ANALYSIS

A Digression – Sparse Classification

- Any test sample belonging to a particular class can be approximately represented as a linear combination of training samples belonging to that class.

$$v_{k,test} = \alpha_{k,1} v_{k,1} + \alpha_{k,2} v_{k,2} + \ldots + \alpha_{k,n_k} v_{k,n_k} + \varepsilon$$

- The assumption can be written in terms of all the training samples

$$v_{k,test} = V\alpha + \varepsilon$$

where $V = [v_{1,1} | \ldots | v_{n,1} | \ldots | v_{k,1} | \ldots | v_{k,n_k} | \ldots v_{C,1} | \ldots | v_{C,n_C}]$

and $\alpha = [\alpha_{1,1} \ldots \alpha_{1,n_1} \ldots \alpha_{k,1} \ldots \alpha_{k,n_k} \ldots \alpha_{C,1} \ldots \alpha_{C,n_C}]^T$

Sparse Classification contd.

- According to the assumption, the vector α is sparse, since it has non-zero coefficients corresponding only to the correct group.
- Classification Algorithm
  - Find α: $\min_{\alpha} \|\alpha\|_1$ such that $\| v_{k,test} - V\alpha \|_2 < \varepsilon$
  - Compute Representative sample of each class

$$v_{rep}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j} \quad , \forall \mathrm{i}=1 \ldots \mathrm{C}$$

  - Assign test sample to class with minimum error

$$error(v_{test}, i) = \| v_{k,test} - v_{rep(i)} \|_2 , \forall i = 1 \ldots C$$

Dictionary Learning for Classification

- So far discussion was on recovery capacities of dictionary learning.
- The learned dictionary was largely used as a substitute for designed dictionaries like wavelet.
- But ‘learning’ offers more flexibility.
- Often ‘recovery’ is not the final goal. It is followed by some analysis, e.g. classification
- Dictionary learning allows the flexibility for incorporating such analysis constraints.

Metaface – a naive approach

- Use the SC framework but use dictionaries instead.
- For each class learn a dictionary from training samples of that class

$$\min_{D_i, Z_i} \left\| X_i - D_i Z_i \right\|_F^2 + \lambda \left\| Z_i \right\|_1, s.t. \left\| D_i^{j\downarrow} \right\|_2 = 1 \ \forall j$$

- The learnt dictionaries are concatenated in the SC formulation in place of the raw samples.
- Does not really learn discriminative dictionaries.

M. Yang, L. Zhang, J. Yang, and D. Zhang. metaface learning for sparse representation based face recognition. ICIP, 2010.

Structured Incoherence

- Dictionaries from different classes should not resemble each other.
- Incoherence term between dictionaries of classes i and j denoted as $\left\| D_i^T D_j \right\|_F^2$

$$\min_{D_i, Z_i} \sum_i \left\{ \left\| X_i - D_i Z_i \right\|_F^2 + \lambda \left\| Z_i \right\|_1 \right\} + \eta \sum_{i \neq j} \left\| D_i^T D_j \right\|_F^2$$

- This formulation yields dictionaries that 'look' different; but the representation can still be similar for different classes.

I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. CVPR , 2010

FisherDL

- Build a dictionary consisting of sub-dictionaries for each class. $D=[D_1|\ldots|D_C]$
- Training samples represented as:

$$X_i = DZ = \sum_i D_i Z_i$$

- The dictionary learning problem is framed as:

$$\min_{D,Z} C(X,D,Z)+\lambda_1\|Z\|_1+\lambda_2 f(Z)$$

- C(X,D,Z) – discriminative fidelity
- F(Z) – discriminative coefficient

M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. ICCV, 2011

Discriminative Fidelity

- The coefficients should only be sparse in the class specific dictionary.

$$C(X_i, D, Z_i) = \left\| X_i - DZ_i \right\|_F^2 + \left\| X_i - D_i Z_i^i \right\|_F^2 + \sum_{i \neq j} \left\| D_j Z_i^j \right\|_F^2$$

- First term – The full dictionary should represent the data (obvious)
- Second term – X_i should be well represent by D_i
- Third term – X_i should not be representable in dictionaries for other classes

Discriminative Coefficient

- Fisher criterion – coefficients from same class should be similar (low variance) and coefficients from different classes should be dissimilar (high variance)
- Scatters $S_W = \sum_{c} \sum_{z_i \in Z_c} (z_i - \mu_c)(z_i - \mu_c)^T$

$$S_B = \sum_{c} (\mu_c - \mu)(\mu_c - \mu)^T$$

- Discriminative coefficient term

$$f(Z) = tr(S_W) - tr(S_B) + \eta \|Z\|_F^2$$

Discriminative KSVD

- Learning a separate dictionary for each class requires lot of data.
- Second, how to use these dictionaries for classification is not obvious.
- Learning a single yet discriminative dictionary would be ideal.

$$\min_{D,W,Z} \|X - DZ\|_F^2 + \lambda_1 \|H - WZ\|_F^2 + \lambda_2 \|Z\|_1$$

- H consists of class labels (as indicator functions).

D KSVD Classification

- For a new test sample find the sparse code:

$$\min_{z} \|x - Dz\|_F^2 + \lambda_2 \|z\|_1$$

- Find the class by projecting it with W, i.e. Wz.
- Assign z to the class having highest magnitude in (Wz).

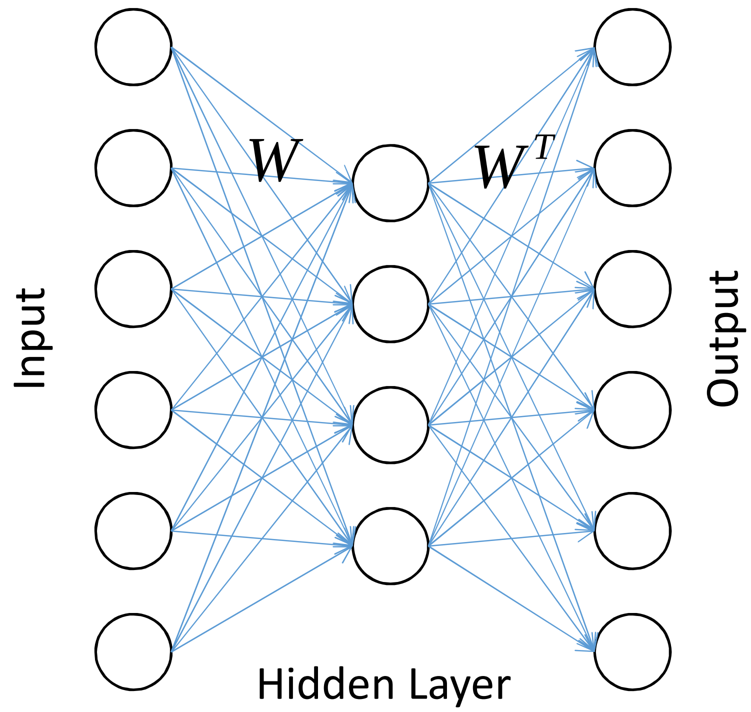
- No separate classifier required.
- Simple unified framework.

Some Classification Results

Method	YaleB	AR	Caltech
SRC (all)	97.2	97.5	70.7
SRC (limited samples)	80.5	66.5	48.8
KSVD (limited samples)	93.1	86.5	49.8
D KSVD (limited samples)	94.1	88.8	49.6

AUTOENCODER

A simple autoencoder



- X – input data
- Output same as input
- Learn the weights 'W' so that the reconstruction error is minimized.

$$\min_{W} \left\| X - W^T \varphi(WX) \right\|_F^2$$

- *WX* – representation at hidden layer
- φ activation function

Interpretation

- For the simple case where the activation function is linear:
  - W, W^T are just inverses of each other when the number of hidden nodes are the same as the number of input / output nodes
  - They act like the PCA when the number of hidden nodes are smaller.
- In practice the activation function is never linear. Consequently the weight is hard to interpret.
- AE is mostly used for automatic feature extraction.

Denoising Autoencoder

- The input is a noise corrupted sample and the output is a noise free sample.
- The denoising AE learns to encode the noisy samples to a latent feature space and the decode the latent features to a denoised sample.

  - Lots of papers – BUT only PSNR reported!
  - No images or other quality metrics like SSIM.
  - Actually results are quiet poor, results in blurred images (obviously high PSNR) but visually unsatisfactory.

Regularized Autoencoder

- Sparse AE - The latent representation should be sparse (not the weights – still a fully connected graph)

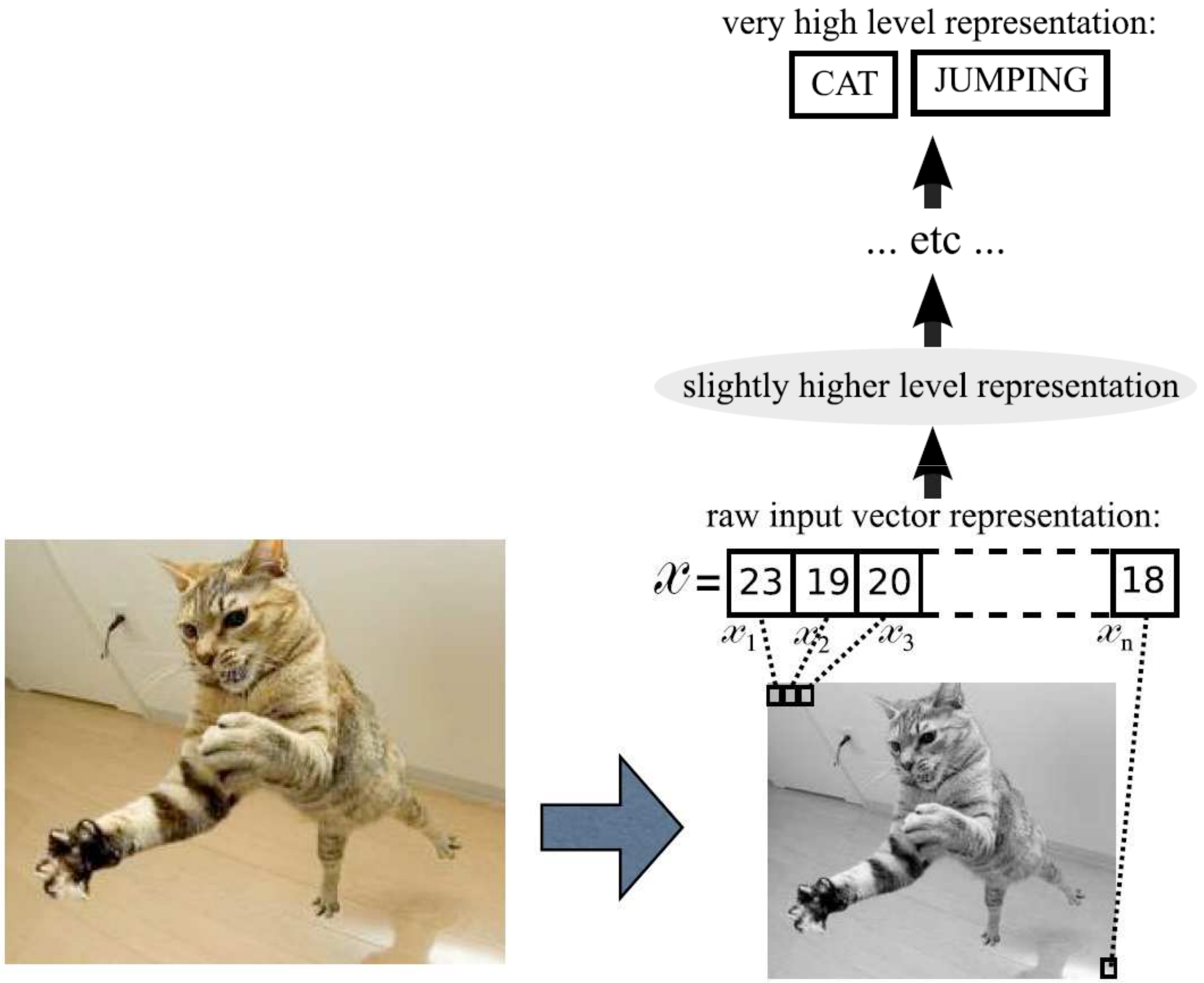
$$\min_{W} \left\| X - W^T \varphi(WX) \right\|_F^2 + \lambda \left\| WX \right\|_1$$

- Contractive AE –

$$\min_{W} \left\| X - W^T \varphi(WX) \right\|_F^2 + \lambda \left\| J\left(\varphi(W)\right) \right\|_F^2$$

- Boils down to Ridge Regression (weight loss in ML literature) for linear case

Abstraction

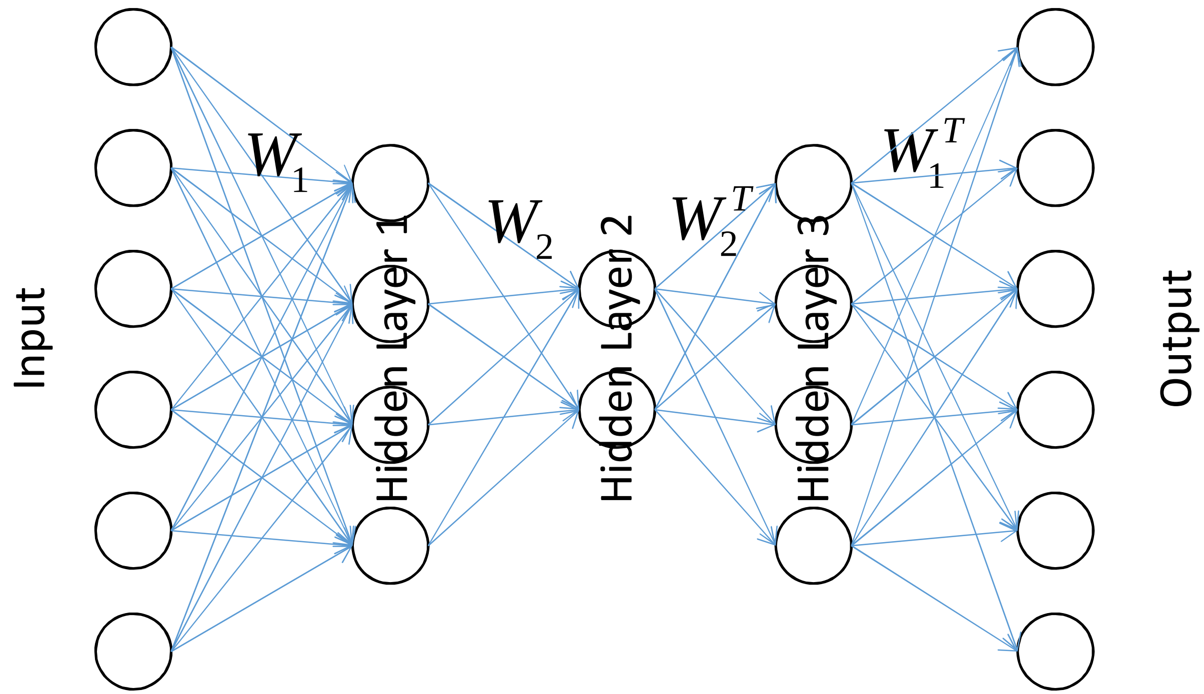


Deep Learning

- The goal is to learn arbitrary functional relationships.
- Shallow (single layer) architectures can achieve that – but ... The number of nodes (in hidden layer) will increase exponentially.
- Statistically ... More sensible to learn stacked architectures with fewer nodes (fewer parameters)

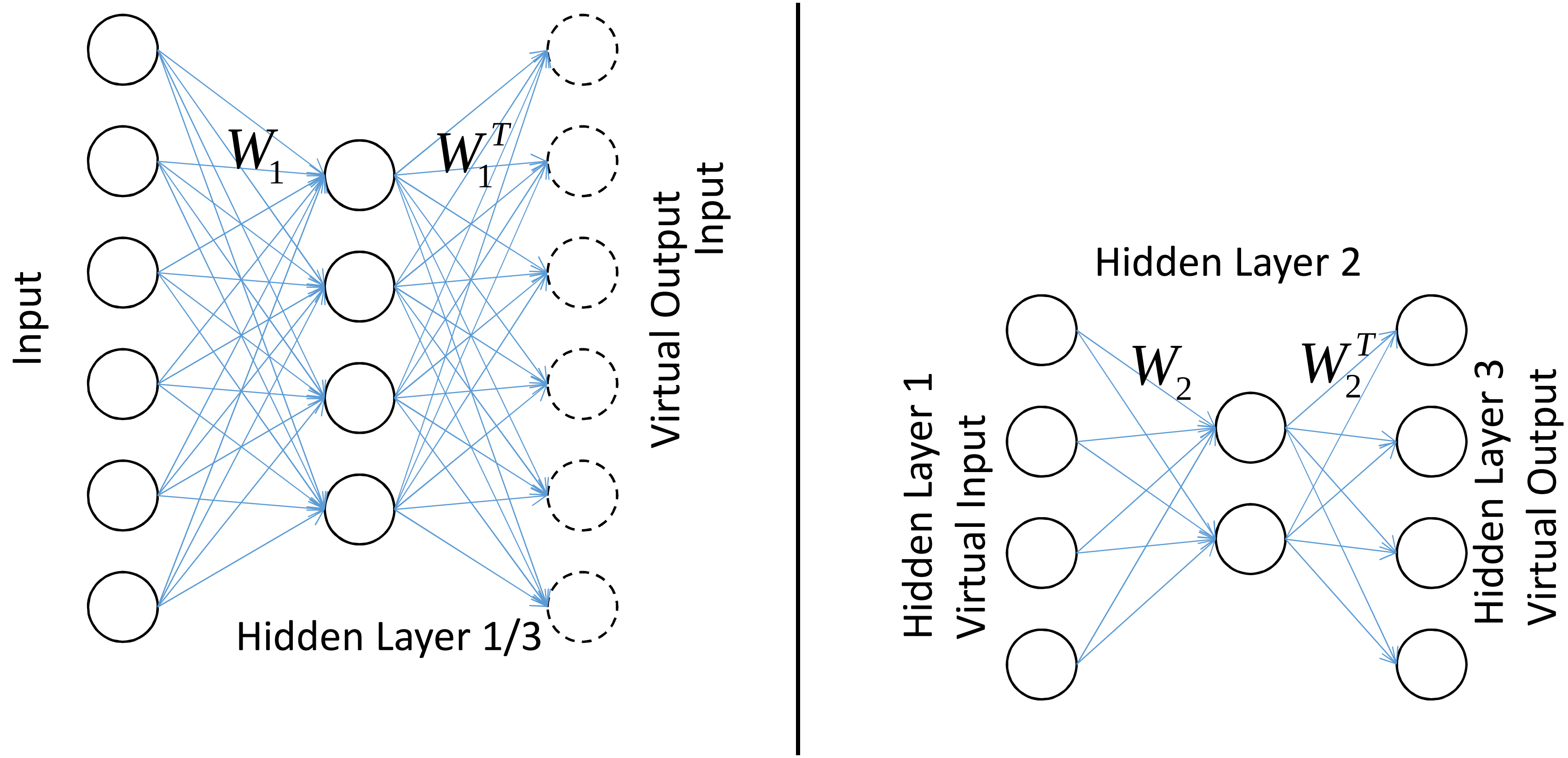
Stacked Autoencoder

$$\min_{W_1,W_2}\left\| X - W_1^T\varphi\left(W_2^T\varphi\left(W_2\varphi\left(W_1X\right)\right)\right)\right\|_F^2$$

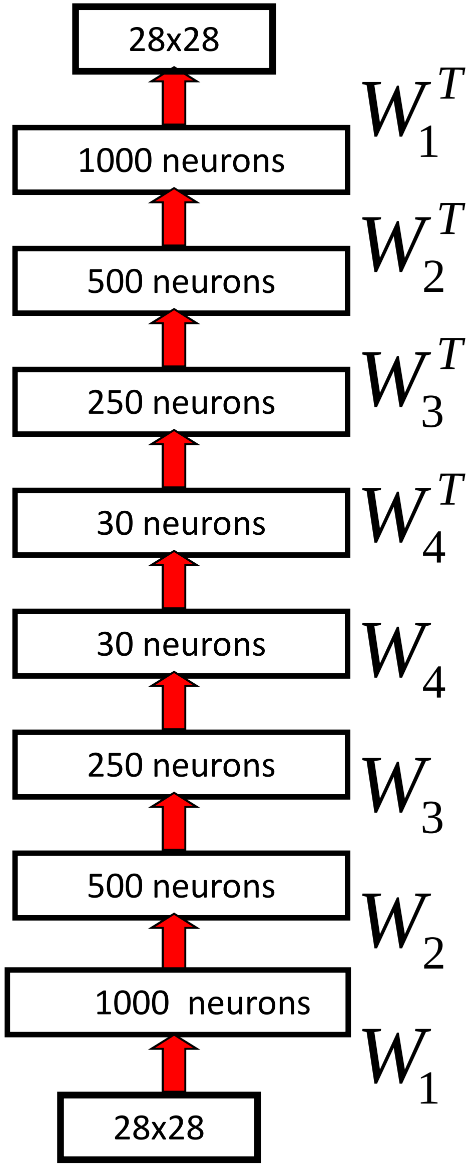


Greedy Learning (Bengio & Hinton)

- Since the aim is to reconstruct (almost) perfectly. Therefore without much loss, each of the layers can be learnt independently.



Representation Capability



Original

30D - DAE

30D PCA

Supervised Encoding

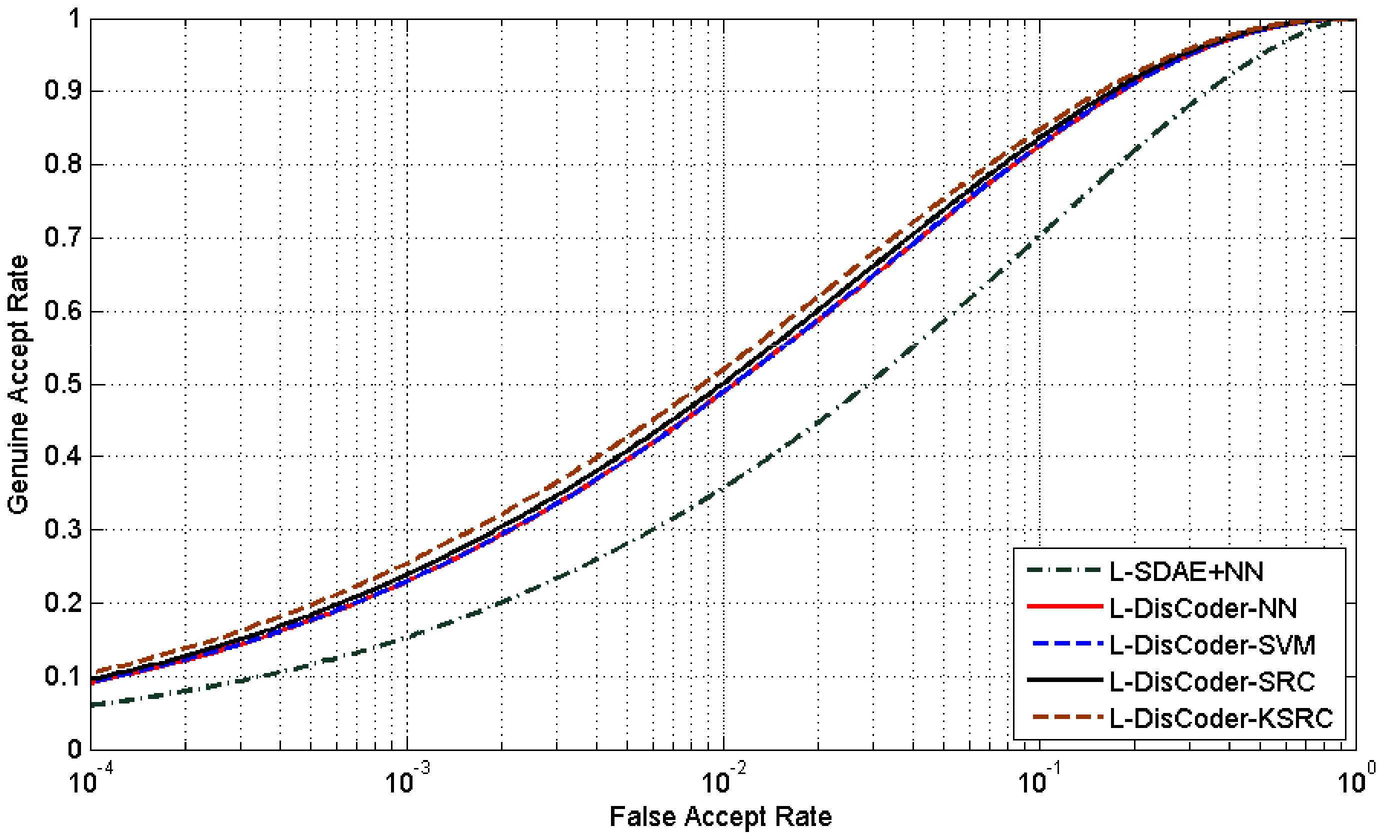
$$X = \left[\underbrace{x_{1,1} | \dots | x_{1,n_1} |}_{X_1 = class\ 1} \quad \underbrace{x_{2,1} | \dots | x_{1,n_2} |}_{X_2 = class\ 2} \quad \dots \quad \underbrace{x_{C,1} | \dots | x_{C,n_C}}_{X_C = class\ \mathrm{C}} \right]$$

- Learn the features (at hidden layers) in supervised fashion.

$$\min_W \left\| X - W^T \varphi(WX) \right\|_F^2 + \lambda \sum_{c=1}^{C} \left\| WX_c \right\|_{2,1}$$

- Assume that the features have a common sparse representation (apply to Bottleneck layer only)

Some Results



AE and DL

- Assuming a linear AE models: $X = W^T \varphi(WX)$
- The feature used for representation is: $\varphi(WX)$

$$Z = \varphi(WX) \Rightarrow X = DZ \text{ where } D = W^T$$

- This is exactly the same formulation as the Dictionary Learning problem (albeit a linear one)

THANK YOU