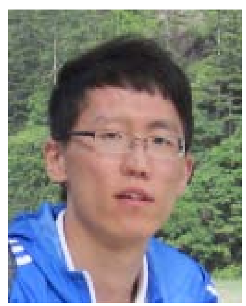
DeepVision: Deep Learning for Computer Vision 2014

Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction

[1]Wei Wang   [1]Yan Huang   [2]Yizhou Wang   [1]Liang Wang

[1]Center for Research on Intelligent Perception and Computing, CRIPAC

Nat'l Lab of Pattern Recognition, CASIA

[2]Nat'l Engineering Lab for Video Technology

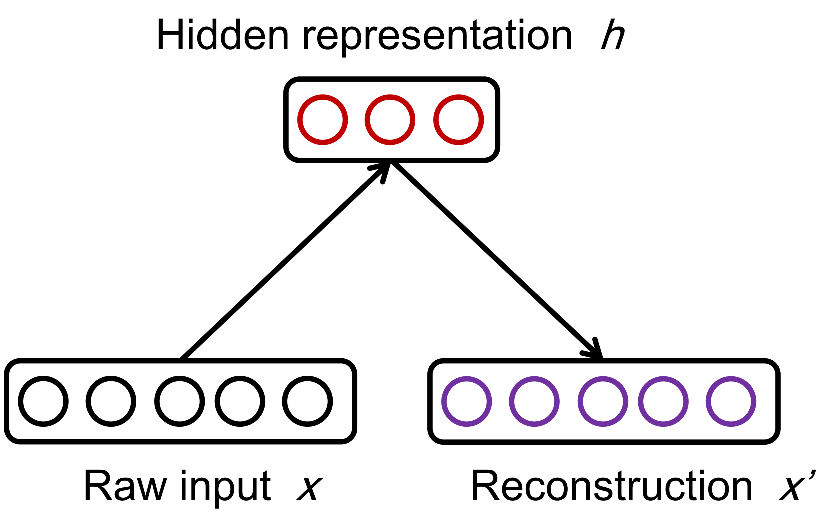
Key Lab. of Machine Perception (MoE), PKU, Beijing, China

Outline

- Motivation
- Related work
- Generalized autoencoder
- Experimental results
- Discussion and Conclusion

Motivation

- The autoencoder algorithm and its regularized variants are widely used in dimensionality reduction and manifold learning

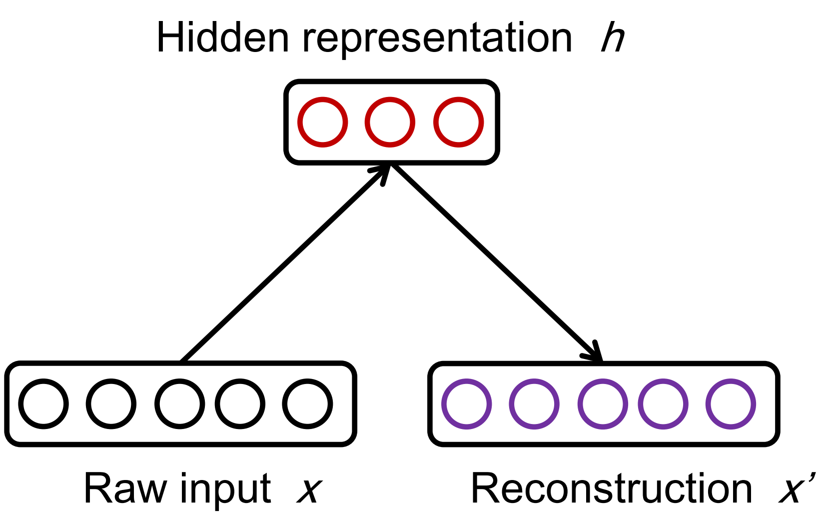


(a) Autoencoder

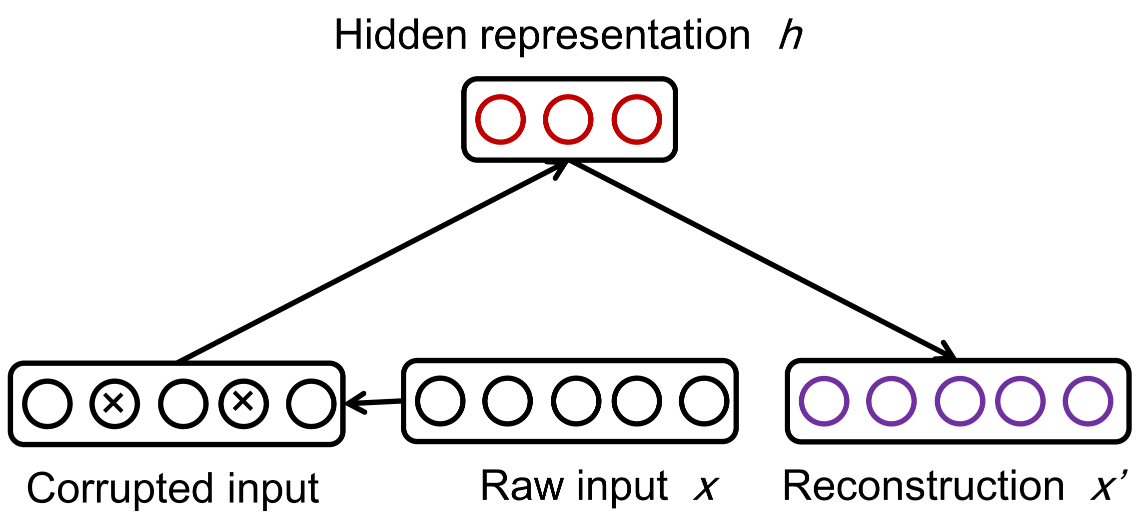
(a) G. Hinton et al. Reducing the dimensionality of data with neural networks. Science, 2006

Motivation

- The autoencoder algorithm and its regularized variants are widely used in dimensionality reduction and manifold learning



(a) Autoencoder

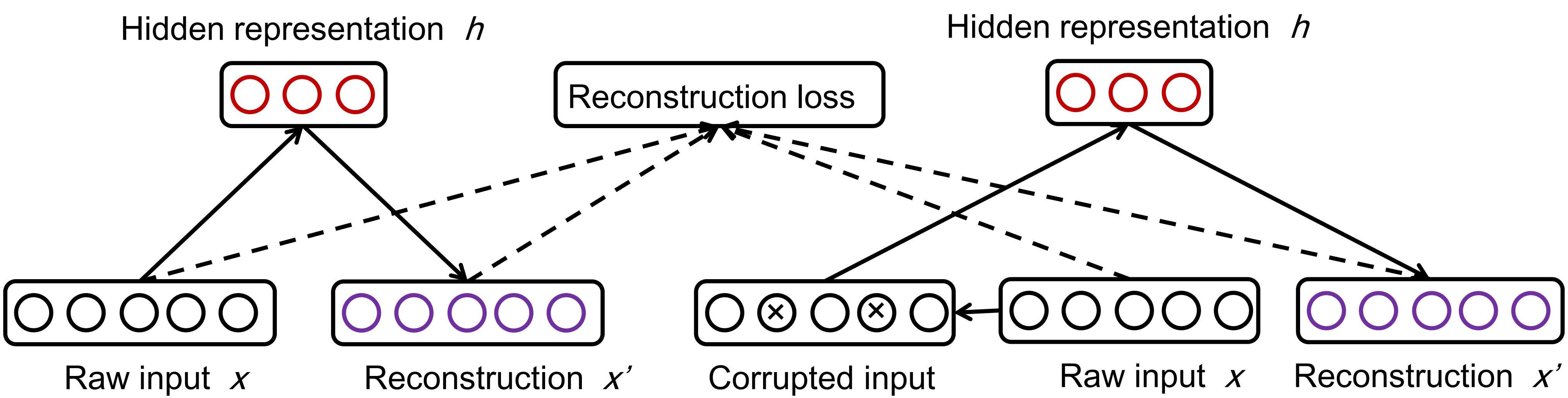


(b) Denoising Autoencoder

(a) G. Hinton et al. Reducing the dimensionality of data with neural networks. Science, 2006
(b) P. Vincent et al. Extracting and composing robust features with denoising autoencoders. ICML2008

Motivation

- The autoencoder algorithm and its regularized variants are widely used in dimensionality reduction and manifold learning



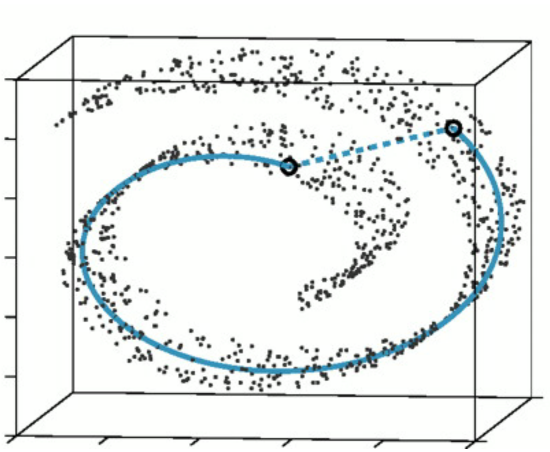
(a) Autoencoder

(b) Denoising Autoencoder

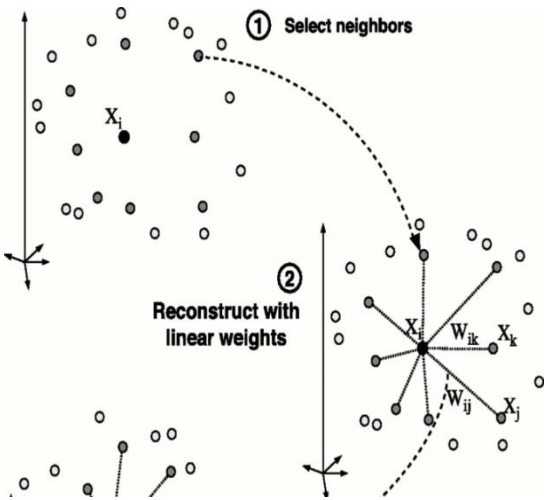
(a) G. Hinton et al. Reducing the dimensionality of data with neural networks. Science, 2006
(b) P. Vincent et al. Extracting and composing robust features with denoising autoencoders. ICML2008

Motivation

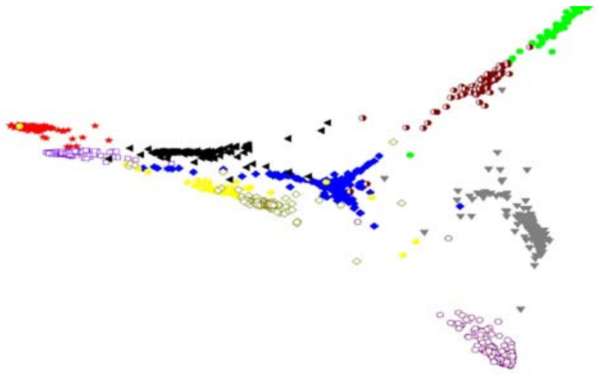
- Modeling data relation is missing, which is very important in dimensionality reduction and manifold learning



(a) Isomap



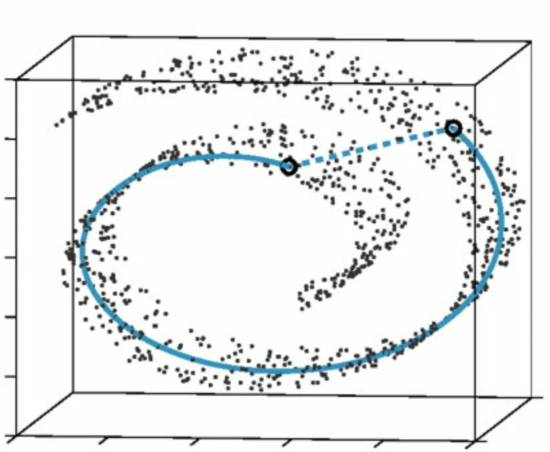
(b) Locally linear embedding



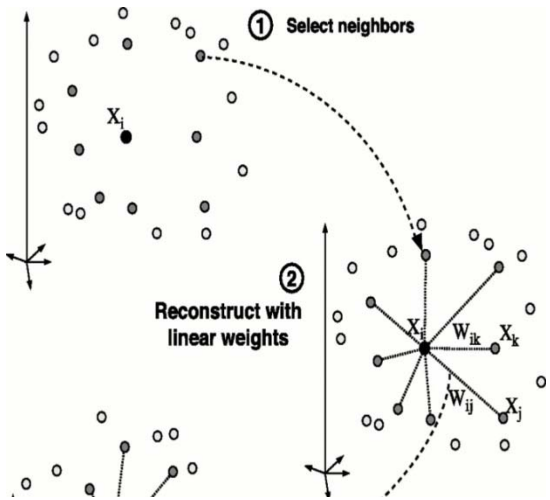
(c) Laplacian eigenmaps

Motivation

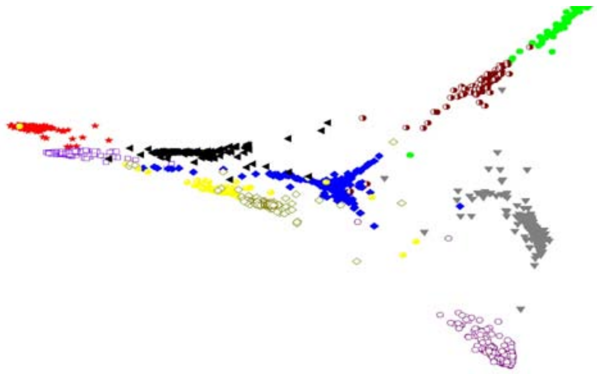
- Modeling data relation is missing, which is very important in dimensionality reduction and manifold learning



(**a**) Isomap



(**b**) Locally linear embedding

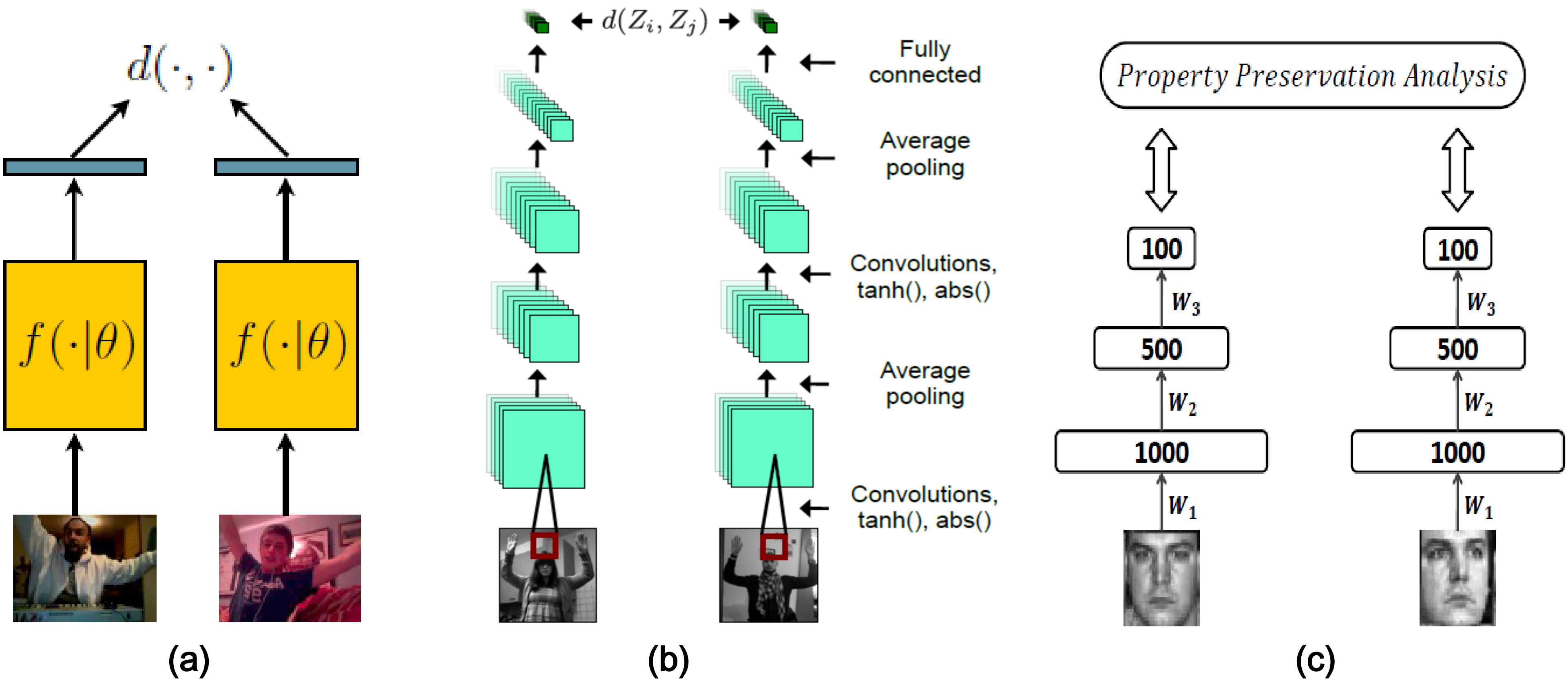


(**c**) Laplacian eigenmaps

How to model data relation in a neural network?

Related Work

- Modeling data relation with a siamese network



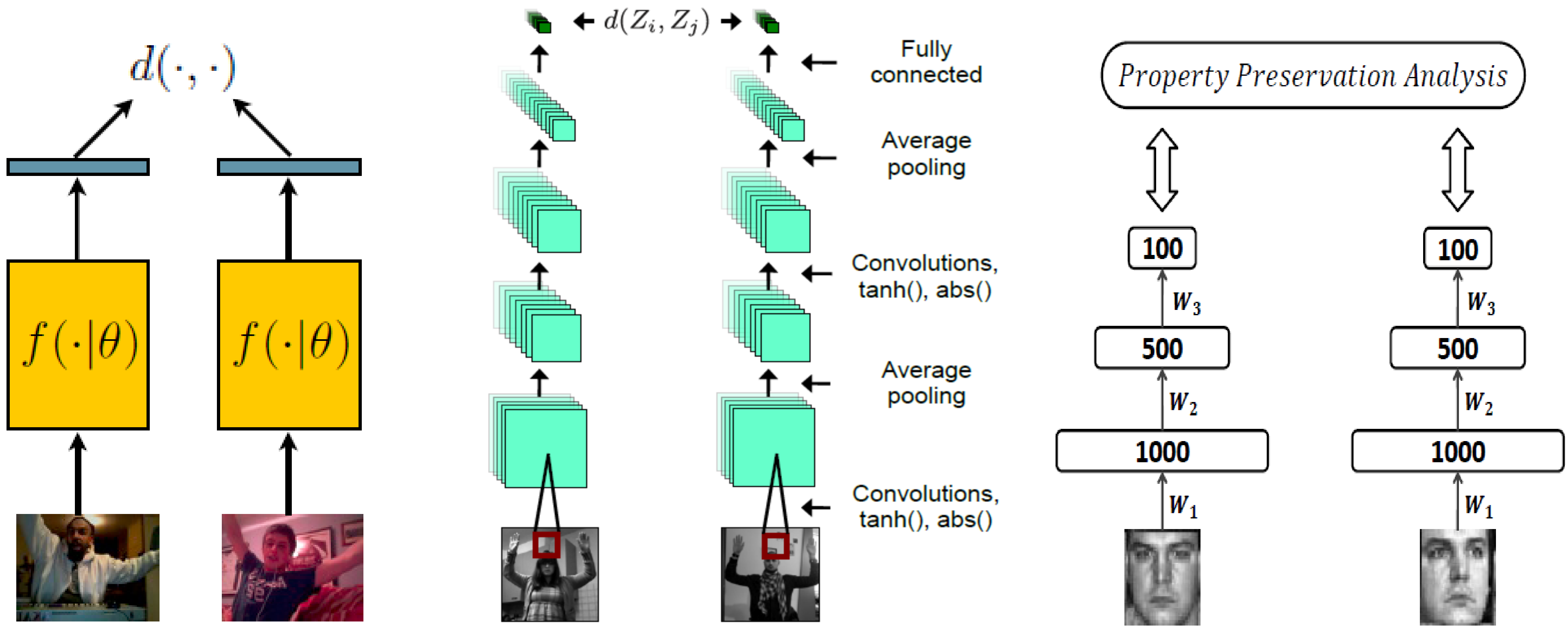
(a) Siamese network

(b) G.W. Taylor et al. Learning invariance through imitation. CVPR2011

(c) Y. Huang et al. A general nonlinear embedding framework based on deep neural network. ICPR2014

Related Work

- Modeling data relation with a siamese network



(a) (b) (c)

(a) Sia

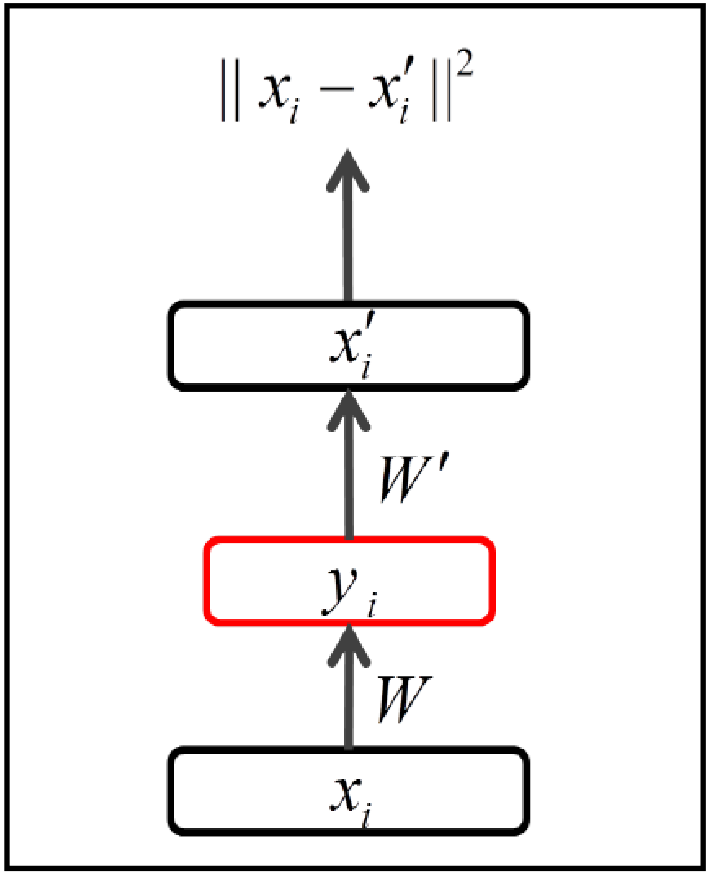
(b) G.

(c) Y. PR2014

How to model data relation in an autoencoder from a viewpoint of reconstruction?

Generalized Autoencoder (GAE)

Three key ingredients

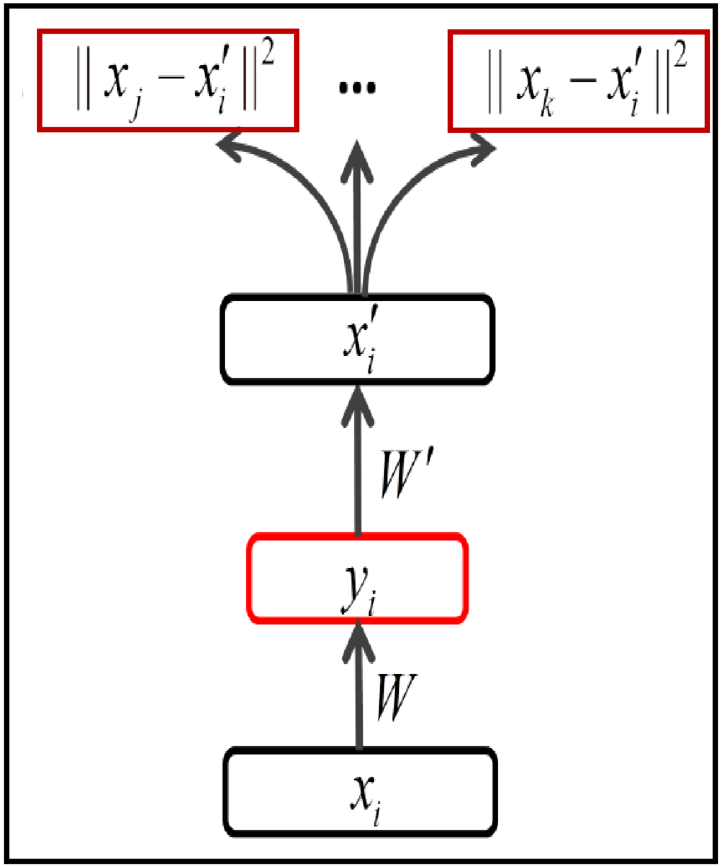


Generalized Autoencoder (GAE)

Three key ingredients

- Each instance x_i is used to reconstruct a set of instances $\{x_j\}$ rather than itself

Generalized Autoencoder

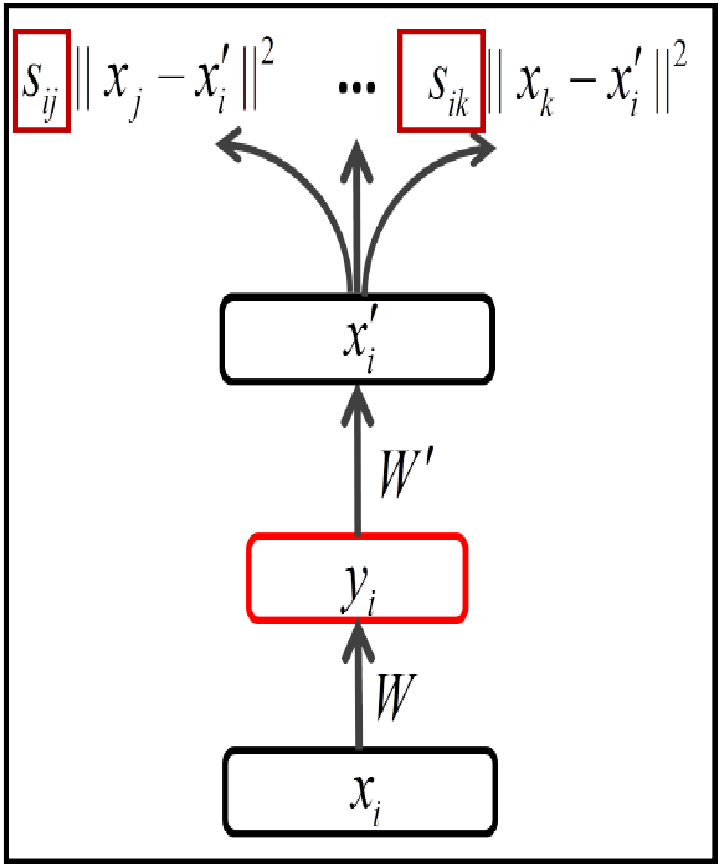


Generalized Autoencoder (GAE)

Three key ingredients

- Each instance x_i is used to reconstruct a set of instances $\{x_j\}$ rather than itself
- Each reconstruction error is weighted by a relational function of x_i and x_j

Generalized Autoencoder

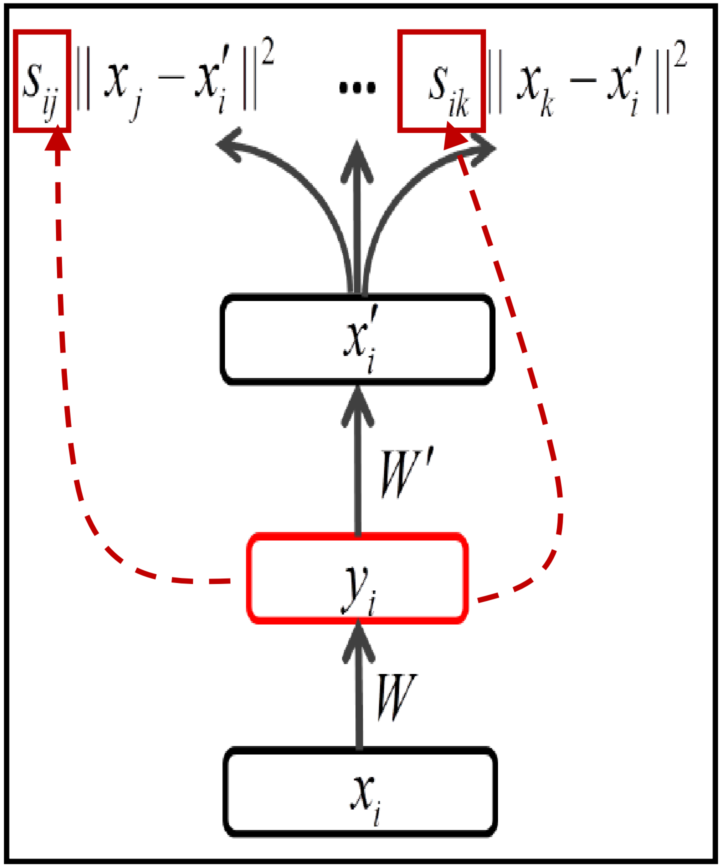


Generalized Autoencoder (GAE)

Three key ingredients

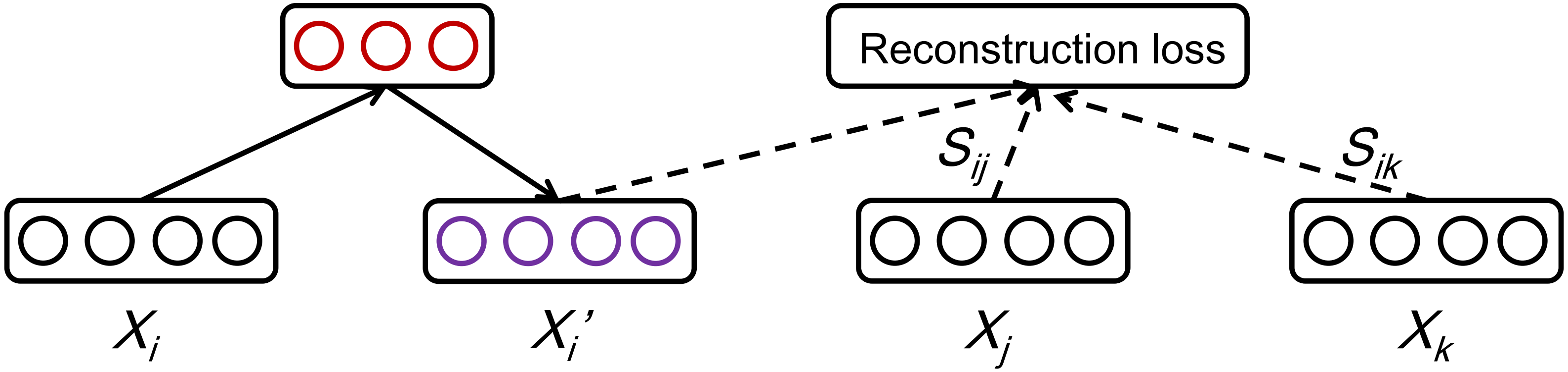
- Each instance x_i is used to reconstruct a set of instances $\{x_j\}$ rather than itself
- Each reconstruction error is weighted by a relational function of x_i and x_j
- Considering that fixed data relation defined on the original high-dimensional space may not be valid on the manifold, the data relation is iteratively updated

Generalized Autoencoder



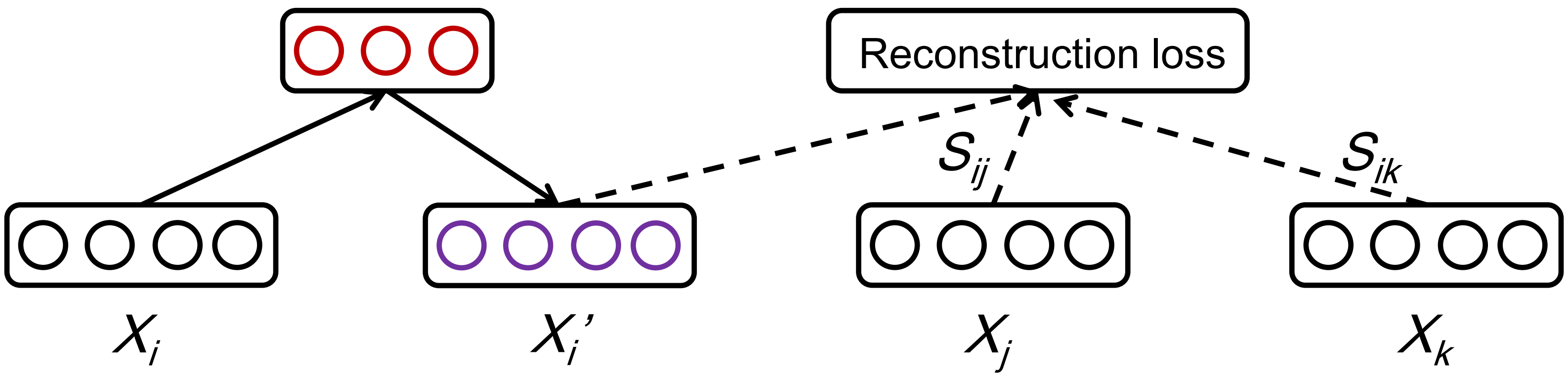
Two View Angles to GAE

- Preserve the relationship between reconstruction x_i' and other raw input x_j , x_k ,…

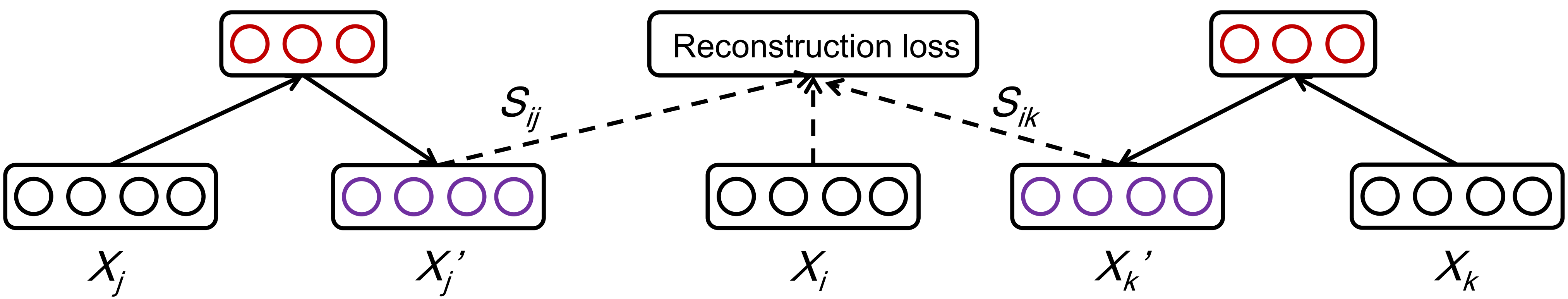


Two View Angles to GAE

- Preserve the relationship between reconstruction x_i' and other raw input x_j , x_k ,…



- Preserve the relationship between raw input x_i and other reconstructions x_j', x_k', …



The Formulation of GAE

- Objective function

$$E(W,W')=\sum_{i=1}^{n} e_i(W,W')=\sum_{i=1}^{n}\sum_{j\in\Omega_i} s_{ij}L(x_j,x_i')$$

where s_{ij} is reconstruction weight, Ω_i is the reconstruction set, $L(x_j,x_i')$ is the reconstruction error

for binary reconstruction

$$L(x_j,x_i') \;=\; -\sum_{q=1}^{d_x} x_j^{(q)} log(x_i'^{(q)}) + (1-x_j^{(q)}) log(1-x_i'^{(q)})$$

for linear reconstruction

$$L(x_j,x_i') \;=\; ||x_j - x_i'||^2$$

Iterative learning Procedure for GAE

Input: training set $\{x_i\}_1^n$

Parameters: $\Theta = (W, W')$

Notation: Ω_i: reconstruction set for x_i

S_i: the set of reconstruction weight for x_i

$\{y_i\}_1^n$: hidden representation

Iterative learning Procedure for GAE

Input: training set $\{x_i\}_1^n$
Parameters: $\Theta = (W, W')$
Notation: Ω_i: reconstruction set for x_i
S_i: the set of reconstruction weight for x_i
$\{y_i\}_1^n$: hidden representation

1. Compute the reconstruction weights S_i from $\{x_i\}_1^n$ and determine the reconstruction set Ω_i, e.g. by k-nearest neighbor

Iterative learning Procedure for GAE

Input: training set $\{x_i\}_1^n$
Parameters: $\Theta = (W, W')$
Notation: Ω_i: reconstruction set for x_i
S_i: the set of reconstruction weight for x_i
$\{y_i\}_1^n$: hidden representation

1. Compute the reconstruction weights S_i from $\{x_i\}_1^n$ and determine the reconstruction set Ω_i, e.g. by k-nearest neighbor

2. Minimize E in Eqn.4 using the stochastic gradient descent and update Θ for t steps

Iterative learning Procedure for GAE

Input: training set $\{x_i\}_1^n$
Parameters: $\Theta = (W, W')$
Notation: Ω_i: reconstruction set for x_i
S_i: the set of reconstruction weight for x_i
$\{y_i\}_1^n$: hidden representation

1. Compute the reconstruction weights S_i from $\{x_i\}_1^n$ and determine the reconstruction set Ω_i, e.g. by k-nearest neighbor

2. Minimize E in Eqn.4 using the stochastic gradient descent and update Θ for t steps

3. Compute the hidden representation $\{y_i\}_1^n$, and update S_i and Ω_i from $\{y_i\}_1^n$.

Iterative learning Procedure for GAE

Input: training set $\{x_i\}_1^n$
Parameters: $\Theta = (W, W')$
Notation: Ω_i: reconstruction set for x_i
S_i: the set of reconstruction weight for x_i
$\{y_i\}_1^n$: hidden representation

1. Compute the reconstruction weights S_i from $\{x_i\}_1^n$ and determine the reconstruction set Ω_i, e.g. by k-nearest neighbor

2. Minimize E in Eqn.4 using the stochastic gradient descent and update Θ for t steps

3. Compute the hidden representation $\{y_i\}_1^n$, and update S_i and Ω_i from $\{y_i\}_1^n$.

4. Repeat step 2 and 3 until convergence.

Connection to Graph Embedding

- The linearization extension of graph embedding is as follows

$$w^* = arg \min_{\substack{w^T XBX^T w=c \\ \text{or } w^T w=c}} \sum_{i,j} s_{ij} ||w^T x_i - w^T x_j||^2$$

- The linearization extension of GAE is as follows

$$w^* = arg \min_{w^T w=c} \sum_{i,j} s_{ij} (||w^T x_i - w^T x_j||^2 + (\frac{c}{2} - 1) y_i^2)$$

where $y_i = w^T x_i$ and $w^T w = c$

Connection to Graph Embedding

$$w^* = arg \min_{w^T w = c} \sum_{i,j} s_{ij} (||w^T x_i - w^T x_j||^2 + \boxed{(\frac{c}{2} - 1) y_i^2})$$

- The additional term $(\frac{c}{2} - 1) y_i^2$ controls different tuning behaviors over the hidden representation by varying c
  - When $c = 2$, GAE has the similar objective function to graph embedding
  - When $c > 2$, this term prevents the hidden representation from being too large, even if the norm of w could be large
  - When $c < 2$, this term encourages the hidden representation to be large enough when w is small

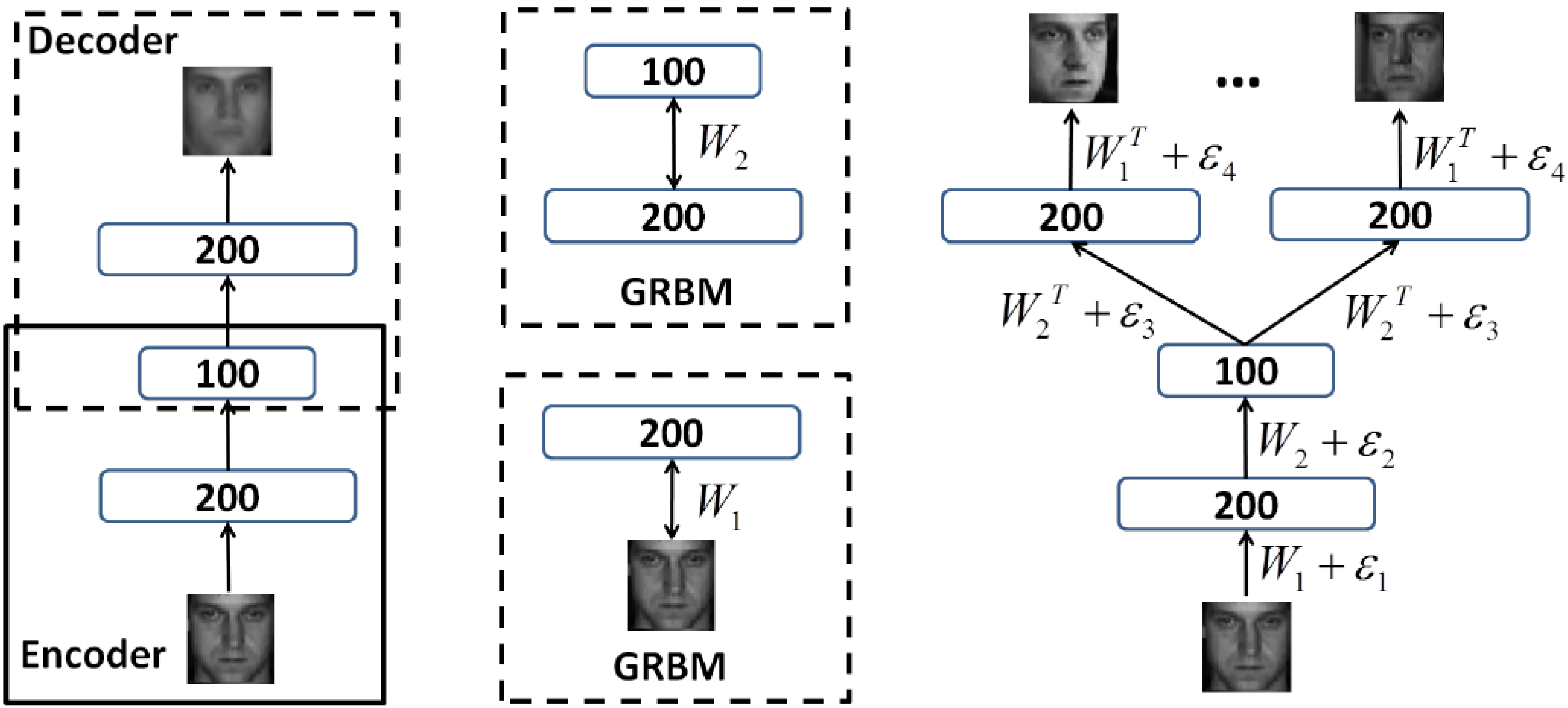
GAE Variants

- Six implementations of GAE inspired by PCA, LDA, ISOMAP, LLE, LE, MFA
  - define different reconstruction sets and weights
  - preserve various kinds of data relation

Method	Reconstruction Set	Reconstruction Weight
GAE-PCA	$j = i$	$s_{ij} = 1$
GAE-LDA	$j \in \Omega_{c_i}$	$s_{ij} = \frac{1}{n_{c_i}}$
GAE-ISOMAP	$j : x_j \in X$	$s_{ij} \in S = -H\Lambda H/2$
GAE-LLE	$j \in N_k(i)$,	$s_{ij} = (M + M^T - M^T M)_{ij}$ if $i \neq j$;
	$j \in (N_k(m) \cup m), j \neq i$ if $\forall m, i \in N_k(m)$	0 otherwise
GAE-LE	$j \in N_k(i)$	$s_{ij} = exp\{-\|\|x_i - x_j\|\|^2/t\}$
GAE-MFA	$j \in \Omega_{k_1}(c_i)$,	$s_{ij} = 1$
	$j \in \Omega_{k_2}(\bar{c}_i)$	$s_{ij} = -1$

Deep Generalized Autoencoder

- Stack multi-layers to form a deep GAE to handle more complex data



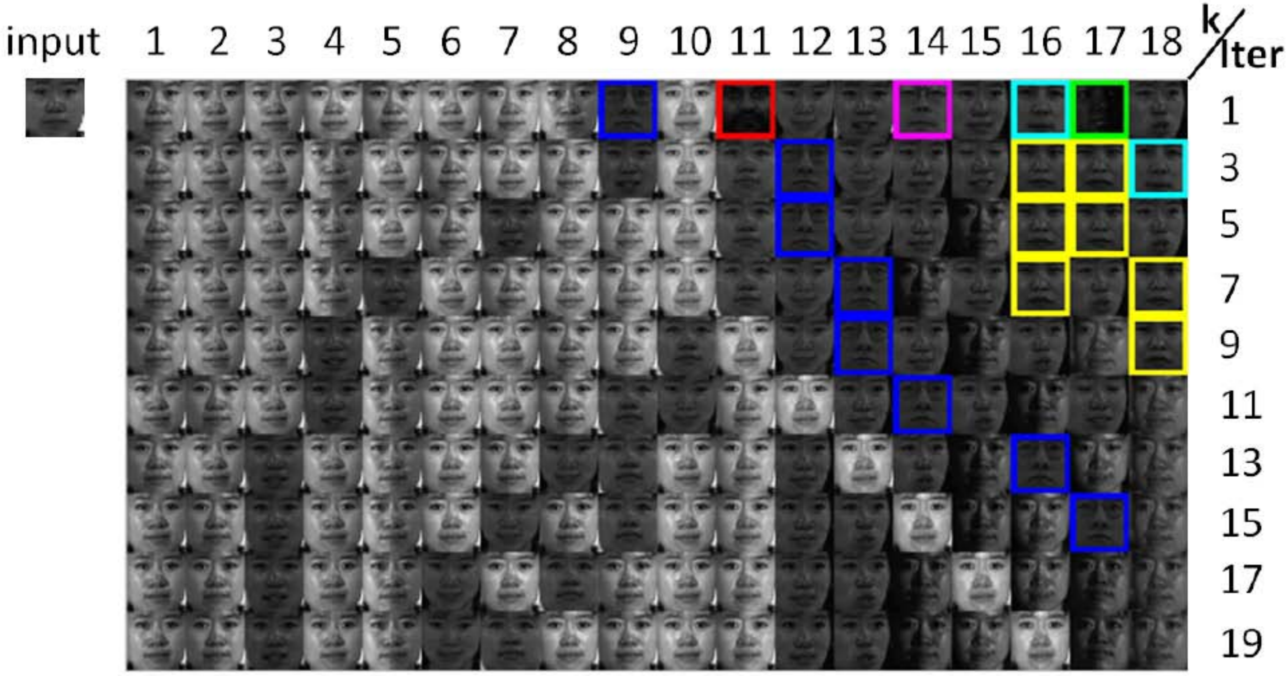
(a) dGAE (b) Pretraining (c) Fine-tuning

Experimental Results

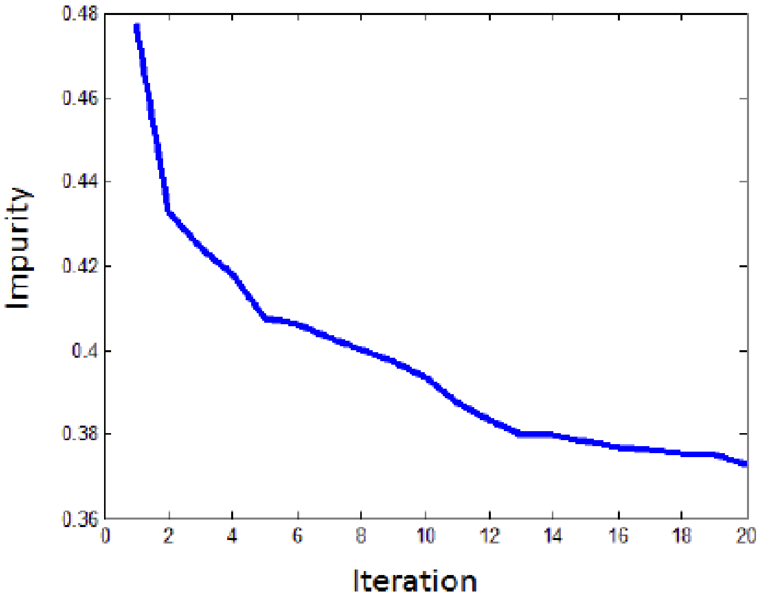
- A face dataset from a small video (F1)
  - 1965 grayscale face images
  - 20×28 pixels
- CMU PIE face dataset
  - 68 subjects in 41,368 face images
  - 32×32 pixels
- MNIST handwritten digits
  - Only 5000 training images and 5000 testing images used for computational cost consideration
  - 28×28 pixels

Manifold Learning

- (a) change of a face's 18 nearest neighbors during the first 20 iterations of dGAE-LE on CMU PIE dataset
- (b) change of impurity during the iteration learning



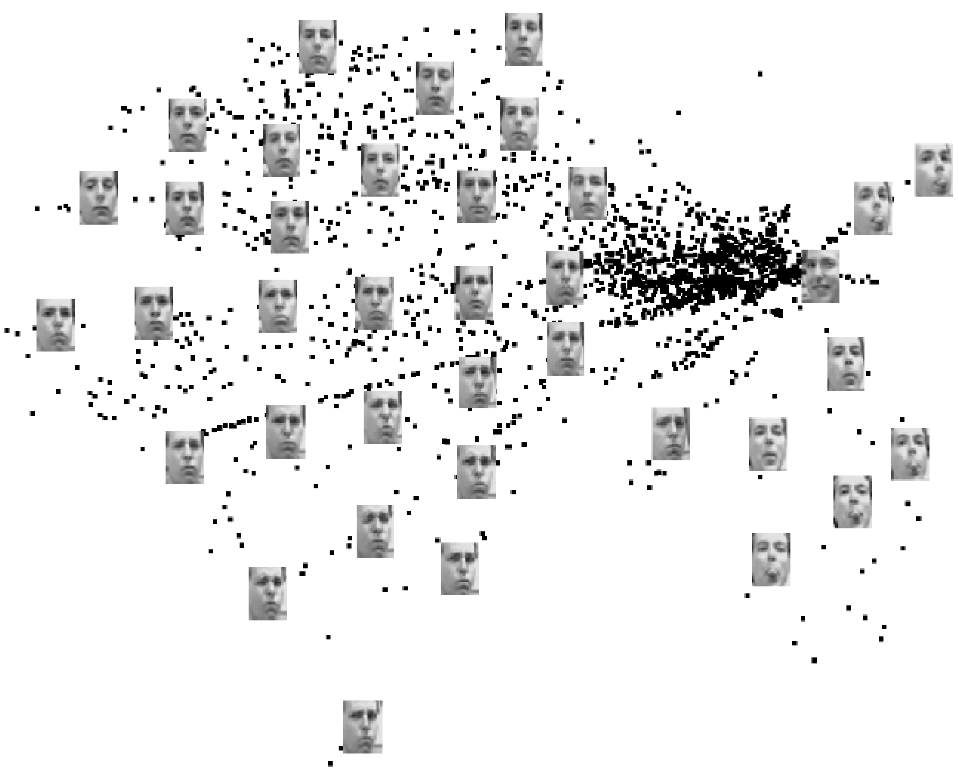
(a) Change of nearest neighbors



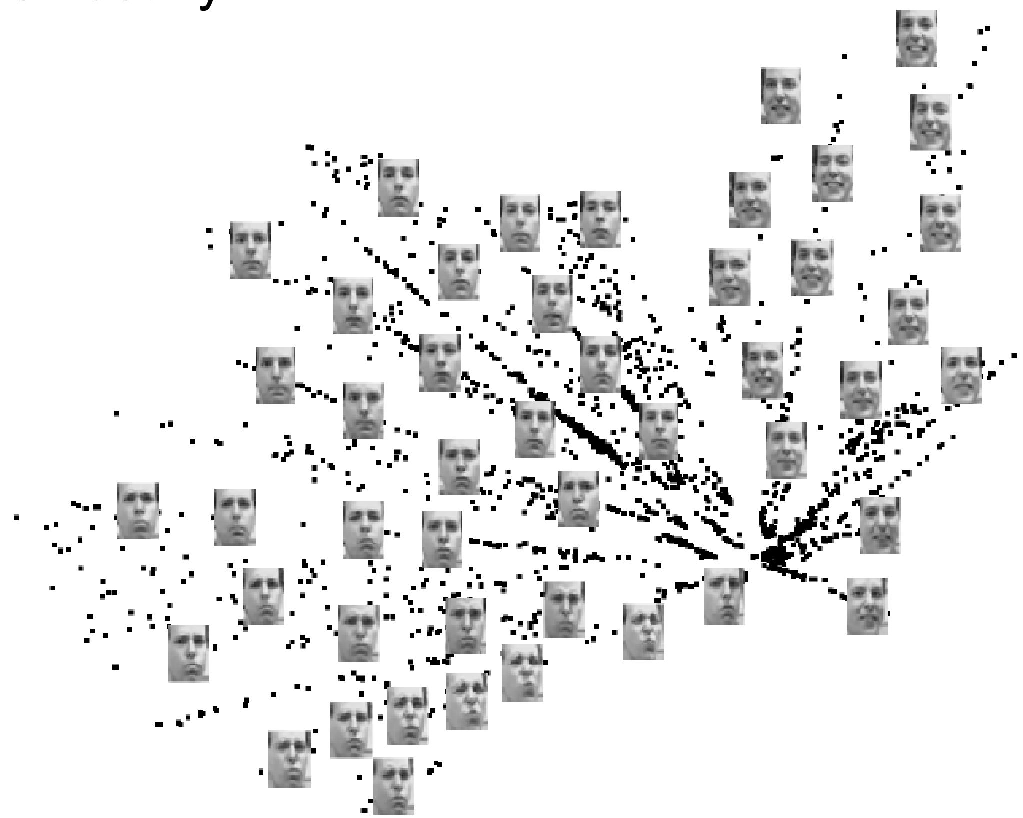
(b) Change of purity

Manifold Learning

- 2D visualization of the face image manifold on the F1
  - Radial patterns: along the radial axes and angular dimension, the facial expression and the pose change smoothly



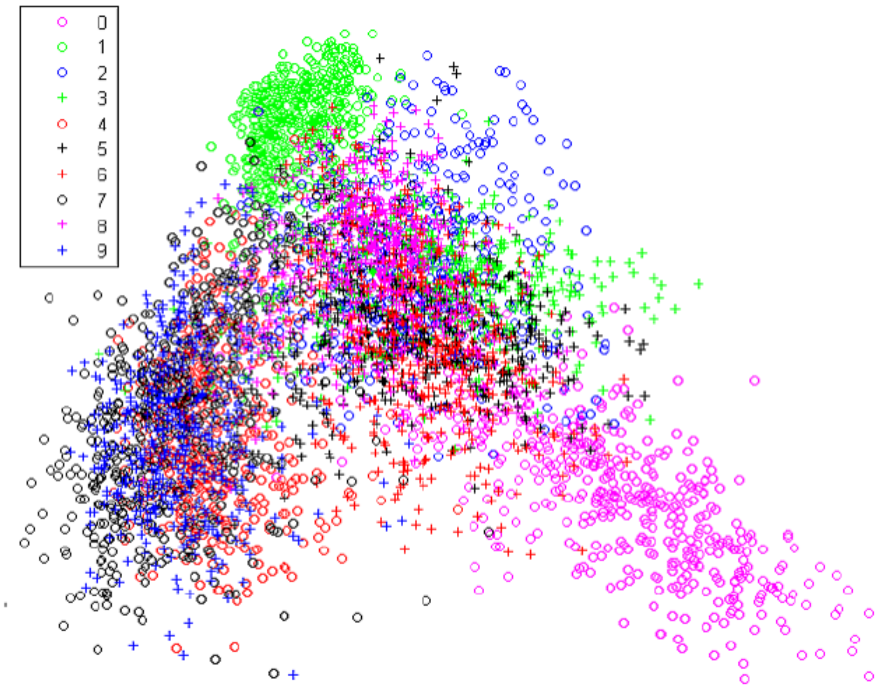
dGAE-PCA



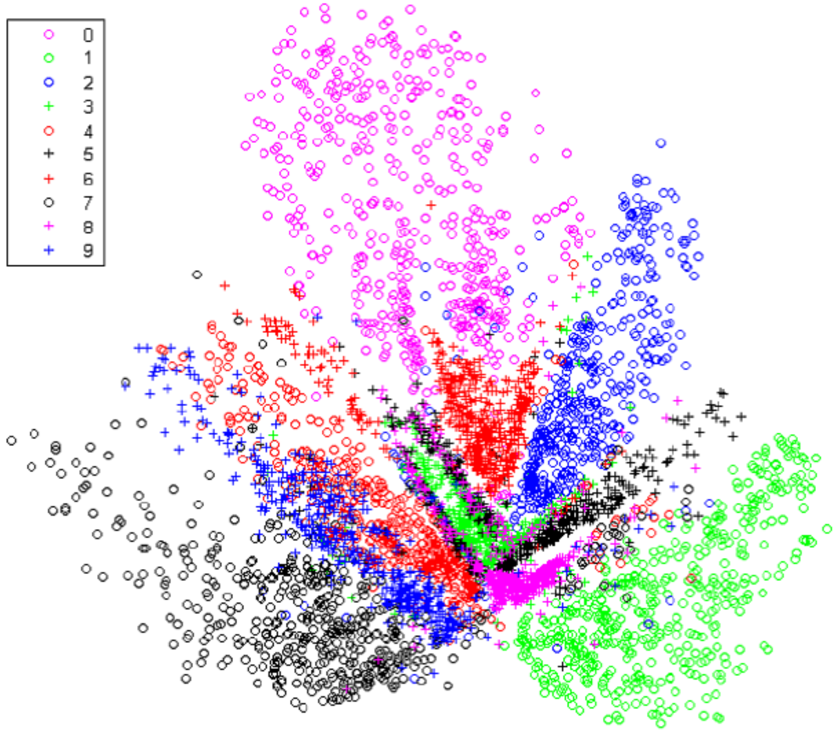
dGAE-LE

Manifold Learning

- 2D visualization of the learned digit image manifold on the MNIST



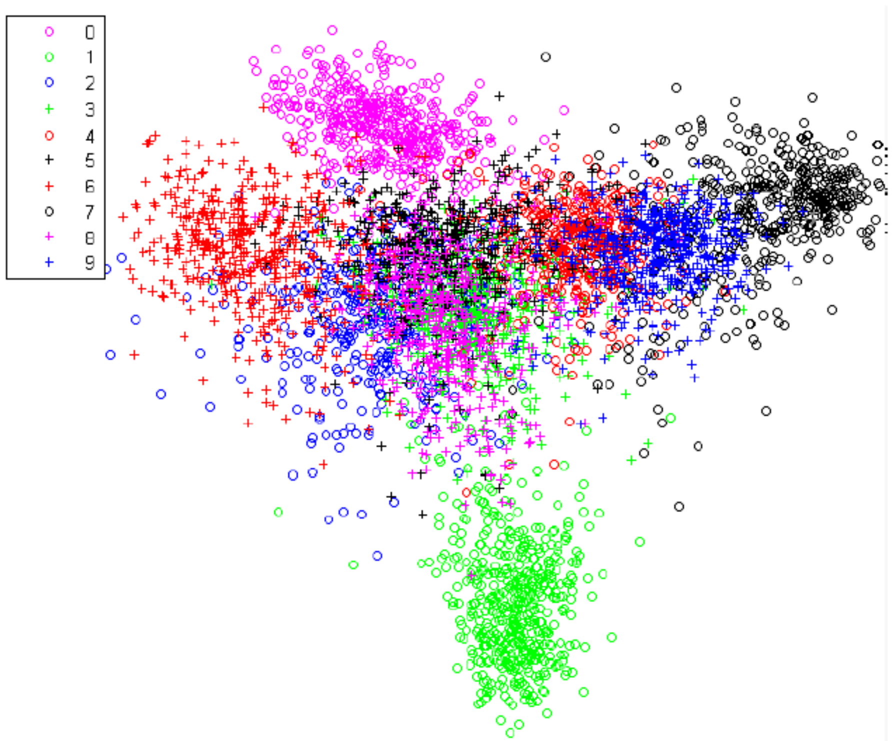
(a) LPP [4]



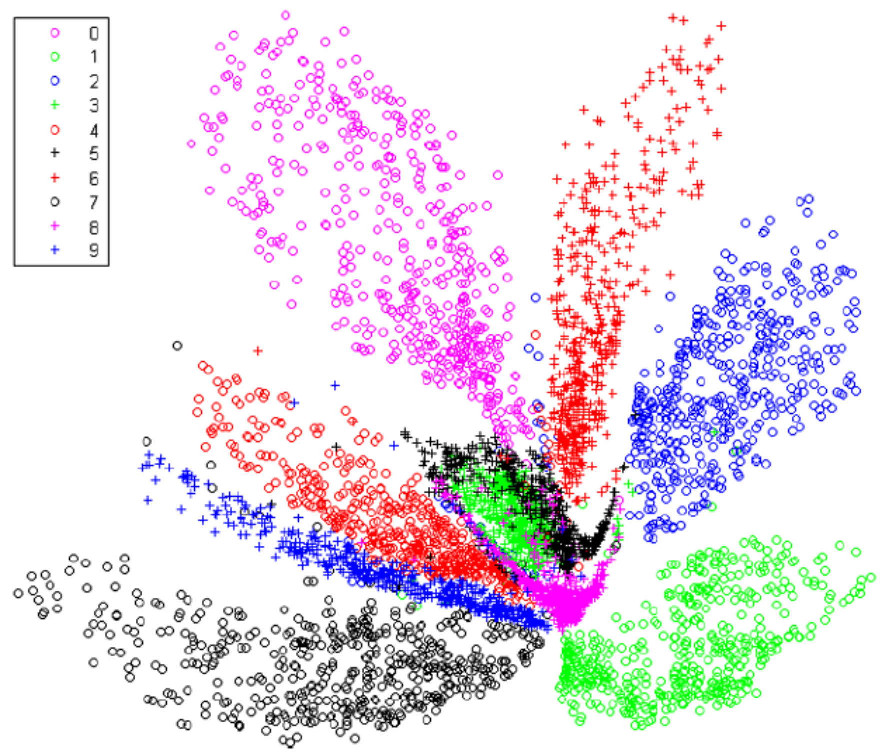
(d) GAE-LE

Manifold Learning

- 2D visualization of the learned digit image manifold on the MNIST



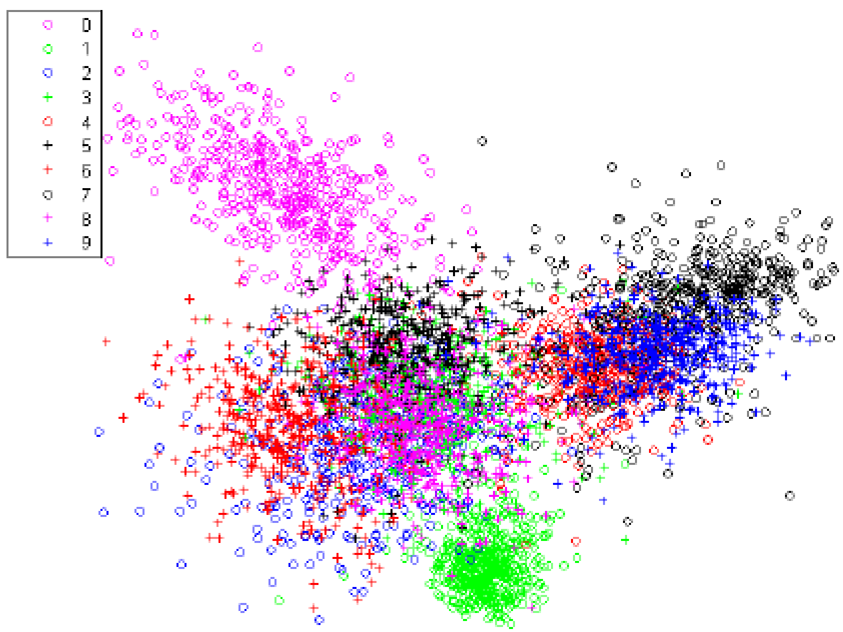
(b) MFA [20]



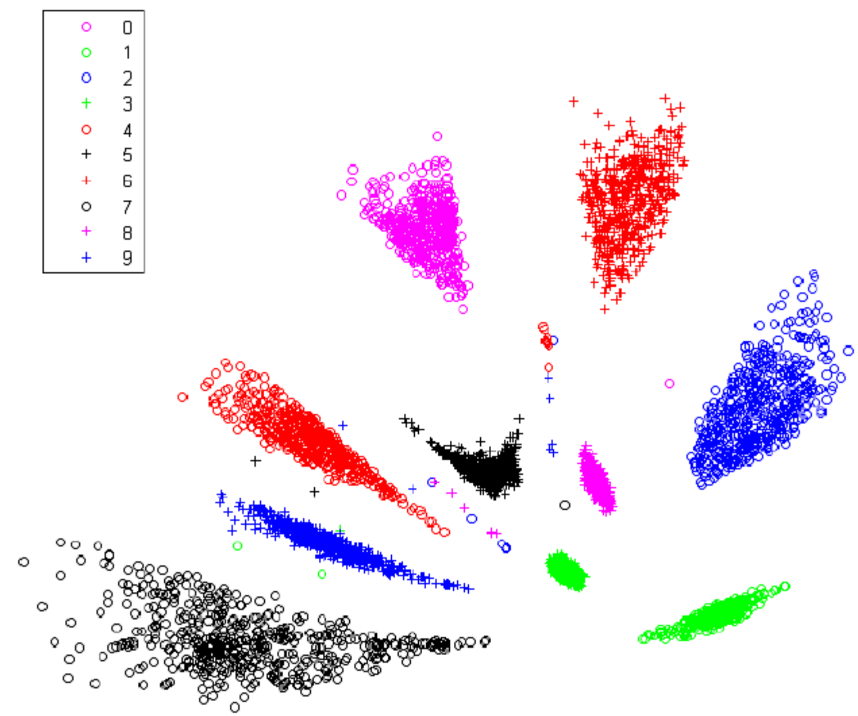
(e) GAE-MFA

Manifold Learning

- 2D visualization of the learned digit image manifold on the MNIST



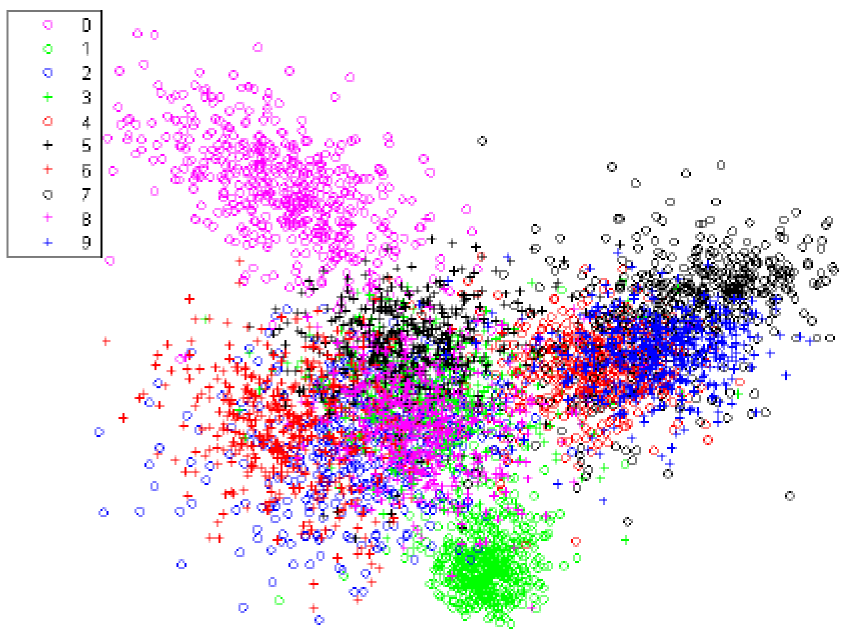
(c) LDA [2]

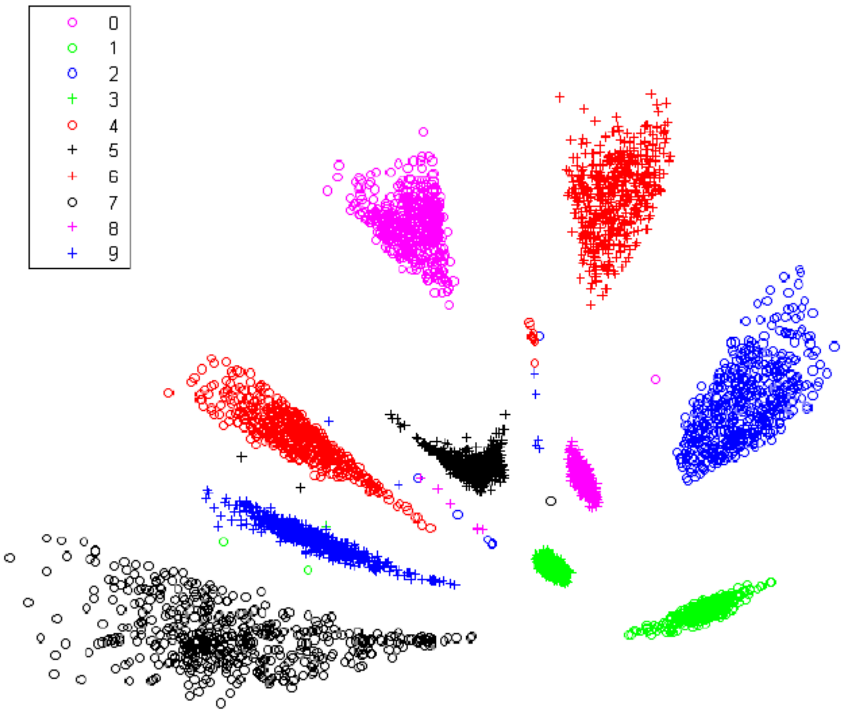


(f) GAE-LDA

Manifold Learning

- 2D visualization of the learned digit image manifold on the MNIST





Compared with other methods, the 2D data points from our GAE variants are more distinctive.

Face Recognition

- Face recognition on the CMU PIE
  - 85 training images and 85 testing images for each individual
  - 157-d features after PCA preprocessing
  - A 157-200-100 encoder network is used

Method	ER	Our Model	ER
PCA	20.6% (150)	dGAE-PCA	3.5%
Kernel PCA	8.1% (g)		
LDA	5.7% (67)	dGAE-LDA	1.2%
Kernel LDA	1.6% (pp)		
ISOMAP	–	dGAE-ISOMAP	2.5%
LLE	–	dGAE-LLE	3.6%
LPP	4.6%(110)	dGAE-LE	1.1%
Kernel LPP	1.7% (pp)		
MFA	2.6% (85)	dGAE-MFA	1.1%
Kernel MFA	2.1% (pp)		

Digit Classification

- Digit classification on the MNIST
  - 500 training images and 500 testing images for each digit
  - No PCA preprocessing
  - A 784-500-200-30 encoder network is used

Method	ER	Our Model	ER
PCA	6.2% (55)	dGAE-PCA	5.3%
Kernel PCA	8.5% (pp)		
LDA	16.1% (9)	dGAE-LDA	4.4%
Kernel LDA	4.6% (pp)		
ISOMAP	–	dGAE-ISOMAP	6.4%
LLE	–	dGAE-LLE	5.7%
LPP	7.9%(55)	dGAE-LE	4.3%
Kernel LPP	4.9% (pp)		
MFA	9.5% (45)	dGAE-MFA	3.9%
Kernel MFA	6.8% (pp)		

Discussion and Conclusion

- The relationship between GAE and denoising autoencoder
  - Denoising autoencoder can be a special case of GAE by defining the reconstruction set as the corrupted version of the input and the reconstruction weight as 1
- Easy to devise new algorithms
  - It is more flexible to construct the reconstruction set by containing the instances with various relationships, such as the same class, knn
- Reduce the computational cost in the large-scale dataset
  - Adopt sampling strategy to construct the reconstruction set

Thanks！ (Q&A)