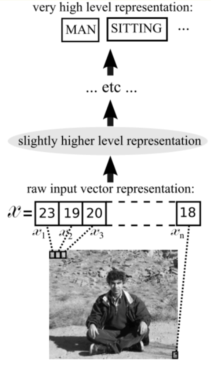
Supervised Deep Learning with Auxiliary Networks

Junbo Zhang[†,‡], Guangjian Tian[‡], Yadong Mu[‡], Wei Fan[‡]

[†]School of Information Science and Technology,
Southwest Jiaotong University, Chengdu 610031, China
[‡]Huawei Noah's Ark Lab, Hong Kong

August 24–27, 2014, New York, NY, USA

Deep Learning



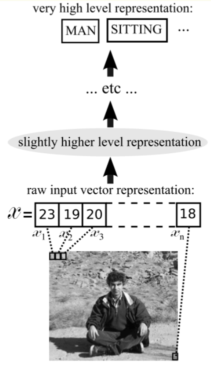
Y. Bengio, Learning Deep

Architectures for AI

Why Deep Learning?

To model high-level abstractions in data by using architectures composed of multiple non-linear transformations.

Deep Learning

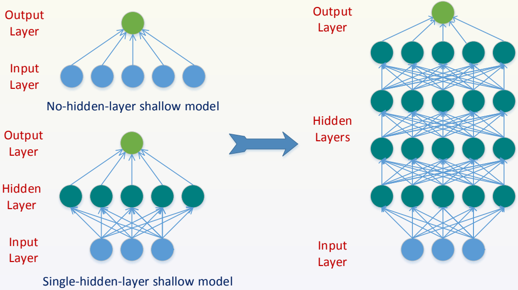


Y. Bengio, Learning Deep

Architectures for AI

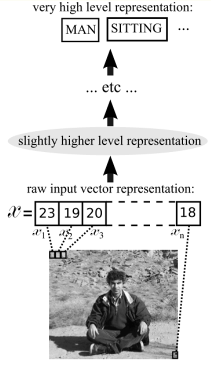
Why Deep Learning?

To model high-level abstractions in data by using architectures composed of multiple non-linear transformations.



Shallow Models **Deep Models**

Deep Learning



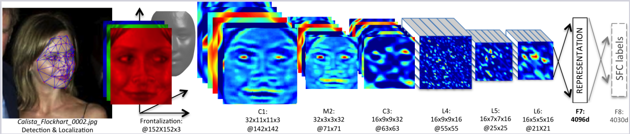
Y. Bengio, Learning Deep Architectures for AI

Why Deep Learning?

To model high-level abstractions in data by using architectures composed of multiple non-linear transformations.

Success in Computer Vision

- Computer vision



[a]

- Speech recognition: Android voice recognition (25% reduction) [b]
- Natural language processing: Machine translation, Matching short text

[a] Taigman et al., DeepFace: Closing the Gap to Human-Level Performance in Face Verification

[b] http://www.wired.com/2013/02/android-neural-network/

Existing Deep Learning Schemes

Manners

- Supervised
- Unsupervised
- Semi-supervised

Existing Deep Learning Schemes

Manners

- Supervised
- Unsupervised
- Semi-supervised

Models

- AutoEncoders (AE)
- Restricted Boltzmann Machines (RBM)
- Convolutional Neural Networks
- Recurrent Neural Networks
- ...

Existing Deep Learning Schemes

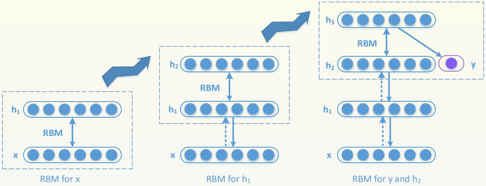
Manners

- Supervised
- Unsupervised
- Semi-supervised

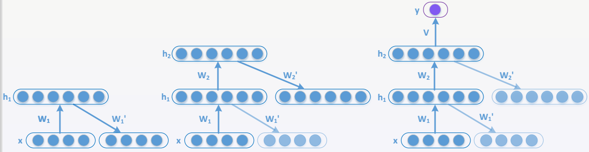
Models

- AutoEncoders (AE)
- Restricted Boltzmann Machines (RBM)
- Convolutional Neural Networks
- Recurrent Neural Networks
- ...

Deep Architecture (Layer-wise Pre-training)



Stacked Restricted Boltzmann Machines (RBM) $\rightarrow$ Deep Belief Network (DBN)



Stacked Autoencoders: Unsupervised pre-training + supervised fine-turning

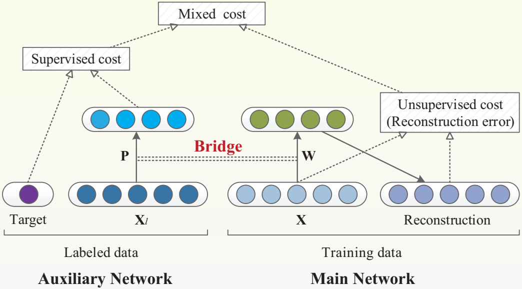
Problems & Shortcoming

1. Sample-specific annotations are always required
2. Ineffectively handle sparse side information

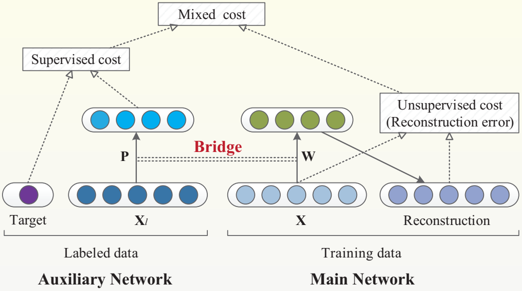
Side information

- More flexible: Similarity/dissimilarity constraints
- Greatly mitigates the workload of annotators

Solution: SUGAR

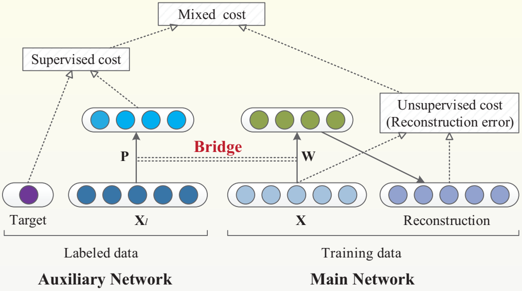


Solution: SUGAR



Main Network is used to reconstruct the input, *i.e.*, the unsupervised autoencoder;

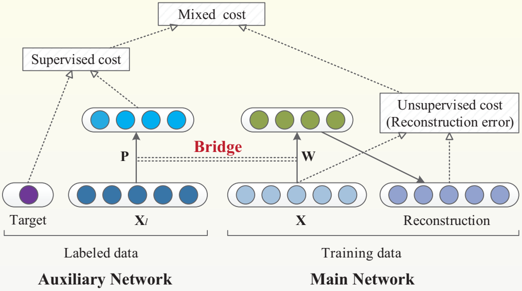
Solution: SUGAR



Main Network is used to reconstruct the input, *i.e.*, the unsupervised autoencoder;

Auxiliary Network is used to regularize the learnt network by pairwise similarity or dissimilarity constraints, *i.e.*, the supvervised hashing learning;

Solution: SUGAR



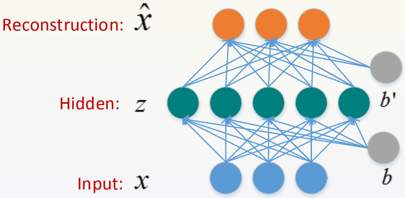
Main Network is used to reconstruct the input, *i.e.*, the unsupervised autoencoder;

Auxiliary Network is used to regularize the learnt network by pairwise similarity or dissimilarity constraints, *i.e.*, the supvervised hashing learning;

Bridge is used to connect *Main Network* and *Auxiliary Network* by enforcing the correlation of their parameters.

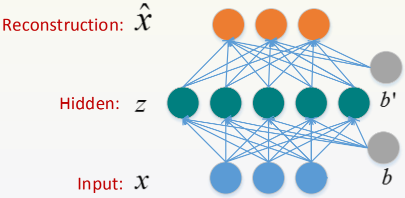
Main Network

A sparsity-encouraging variant of autoencoder.



Main Network

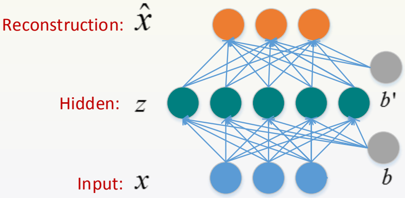
A sparsity-encouraging variant of autoencoder.



Encoder $\mathbf{z} = f(\mathbf{x}) = S_f(\mathbf{W}\mathbf{x} + \mathbf{b})$

Main Network

A sparsity-encouraging variant of autoencoder.

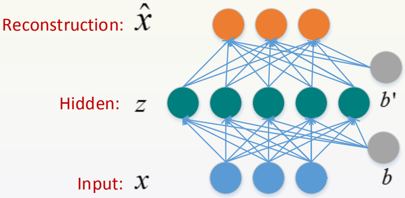


Encoder $\mathbf{z} = f(\mathbf{x}) = S_f(\mathbf{W}\mathbf{x} + \mathbf{b})$

Decoder $\widehat{\mathbf{x}} = g(\mathbf{z}) = S_g(\mathbf{W}'\mathbf{z} + \mathbf{b}')$

Main Network

A sparsity-encouraging variant of autoencoder.



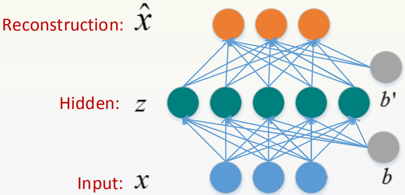
Encoder $\mathbf{z} = f(\mathbf{x}) = S_f(\mathbf{W}\mathbf{x} + \mathbf{b})$

Decoder $\widehat{\mathbf{x}} = g(\mathbf{z}) = S_g(\mathbf{W}'\mathbf{z} + \mathbf{b}')$

Reconstruction Error $\mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) = \|\mathbf{x} - \widehat{\mathbf{x}}\|^2$

Main Network

A sparsity-encouraging variant of autoencoder.



Encoder $\mathbf{z} = f(\mathbf{x}) = S_f(\mathbf{W}\mathbf{x} + \mathbf{b})$

Decoder $\widehat{\mathbf{x}} = g(\mathbf{z}) = S_g(\mathbf{W}'\mathbf{z} + \mathbf{b}')$

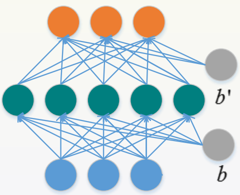
Reconstruction Error $\mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) = \|\mathbf{x} - \widehat{\mathbf{x}}\|^2$

Objective $\arg\min_{\phi} \sum_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) + \lambda\|\mathbf{W}\|_{\ell_1}$

$\phi = \{\mathbf{W}, \mathbf{b}, \mathbf{b}'\},\ \mathbf{W}' = \mathbf{W}^T.$

Main Network

A sparsity-encouraging variant of autoencoder.



Encoder $\mathbf{z} = f(\mathbf{x}) = S_f(\mathbf{W}\mathbf{x} + \mathbf{b})$

Decoder $\widehat{\mathbf{x}} = g(\mathbf{z}) = S_g(\mathbf{W}'\mathbf{z} + \mathbf{b}')$

Reconstruction Error $\mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) = \|\mathbf{x} - \widehat{\mathbf{x}}\|^2$

Objective $\arg\min_{\phi} \sum_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) + \lambda\|\mathbf{W}\|_{\ell_1}$

$\phi = \{\mathbf{W}, \mathbf{b}, \mathbf{b}'\}, \ \mathbf{W}' = \mathbf{W}^T.$

L1 Regularization: Preventing Overfitting

Auxiliary Network

Hashing representation $\mathbf{h} = \mathbf{H}(\mathbf{x}) = \text{sgn}(\mathbf{P}\mathbf{x} + \mathbf{t})$

Auxiliary Network

Hashing representation $\mathbf{h} = \mathbf{H}(\mathbf{x}) = \mathrm{sgn}(\mathbf{P}\mathbf{x} + \mathbf{t})$

Original objective $\mathcal{J}(\mathbf{P}) = \sum_{k=1}^{K} \left\{ \frac{1}{|\mathcal{M}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} h_k(\mathbf{x}_i)h_k(\mathbf{x}_j) - \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} h_k(\mathbf{x}_i)h_k(\mathbf{x}_j) \right\}$

Auxiliary Network

Hashing representation $\mathbf{h} = \mathbf{H}(\mathbf{x}) = \mathrm{sgn}(\mathbf{P}\mathbf{x} + \mathbf{t})$

Original objective $$\mathcal{J}(\mathbf{P}) = \sum_{k=1}^{K} \left\{ \frac{1}{|\mathcal{M}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} h_k(\mathbf{x}_i)h_k(\mathbf{x}_j) - \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} h_k(\mathbf{x}_i)h_k(\mathbf{x}_j) \right\}$$

Relaxations $\mathbf{H}(\mathbf{X}_l) = \mathrm{sgn}(\mathbf{P}\mathbf{X}_l)$ is replaced by $\mathbf{P}\mathbf{X}_l$

$$\Omega_{ij} = \begin{cases} 1 \times \frac{1}{|\mathcal{M}|}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, \\ -1 \times \frac{1}{|\mathcal{C}|}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases}$$

Auxiliary Network

Hashing representation $\mathbf{h} = \mathbf{H}(\mathbf{x}) = \mathrm{sgn}(\mathbf{P}\mathbf{x} + \mathbf{t})$

Original objective $\mathcal{J}(\mathbf{P}) = \sum_{k=1}^{K} \left\{ \frac{1}{|\mathcal{M}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} h_k(\mathbf{x}_i) h_k(\mathbf{x}_j) - \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} h_k(\mathbf{x}_i) h_k(\mathbf{x}_j) \right\}$

Relaxations $\mathbf{H}(\mathbf{X}_l) = \mathrm{sgn}(\mathbf{P}\mathbf{X}_l)$ is replaced by $\mathbf{P}\mathbf{X}_l$

$$\Omega_{ij} = \begin{cases} 1 \times \frac{1}{|\mathcal{M}|}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}, \\ -1 \times \frac{1}{|\mathcal{C}|}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases}$$

Relaxed objective

$$\begin{aligned} \arg\max_{\mathbf{P}} \quad & \frac{1}{2} \mathbf{tr}\{\mathbf{P}\mathbf{X}_l \Omega \mathbf{X}_l^T \mathbf{P}^T\}, \\ \text{subject to} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I}. \end{aligned}$$

The balancing and pairwise decorrelation constraints can help generate good hash codes in which bits are independent and each bit maximizes the information by generating a balanced partition of the data. They are replaced by the orthogonality constraints.

Bridge: Mixed Objective

$$\begin{aligned} \arg\min_{\phi,\mathbf{P}} \quad & \alpha\mathcal{J}_{AE}(\phi) + (1-\alpha)\mathcal{J}_{SH}(\mathbf{P}) + \frac{\epsilon}{2}\|\mathbf{P}-\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_{\ell_1} \\ \text{subject to} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I}. \end{aligned}$$

where ϵ is a correlation coefficient between $\mathbf{P}$ and $\mathbf{W}$, λ is sparsity (L_1) penalty ratio, $\alpha \in [0,1]$ is a guiding coefficient, and linearly blends the following two objectives:

$$\mathcal{J}_{AE}(\phi) = \sum_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\mathbf{x},\widehat{\mathbf{x}}) = \frac{1}{2}\sum_{\mathbf{x}\in\mathbf{X}} \|\mathbf{x}-\widehat{\mathbf{x}}\|^2, \mathcal{J}_{SH}(\mathbf{P}) = -\frac{1}{2}\,\mathbf{tr}\{\mathbf{P}\mathbf{X}_l\boldsymbol{\Omega}\mathbf{X}_l^T\mathbf{P}^T\}.$$

Bridge: Mixed Objective

$$\begin{aligned} \arg\min_{\phi,\mathbf{P}} \quad & \alpha\mathcal{J}_{AE}(\phi) + (1-\alpha)\mathcal{J}_{SH}(\mathbf{P}) + \frac{\epsilon}{2}\|\mathbf{P}-\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_{\ell_1} \\ \text{subject to} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I}. \end{aligned}$$

where ϵ is a correlation coefficient between $\mathbf{P}$ and $\mathbf{W}$, λ is sparsity (L_1) penalty ratio, $\alpha \in [0,1]$ is a guiding coefficient, and linearly blends the following two objectives:

$$\mathcal{J}_{AE}(\phi) = \sum_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\mathbf{x},\widehat{\mathbf{x}}) = \frac{1}{2}\sum_{\mathbf{x}\in\mathbf{X}} \|\mathbf{x}-\widehat{\mathbf{x}}\|^2, \mathcal{J}_{SH}(\mathbf{P}) = -\frac{1}{2}\,\mathbf{tr}\{\mathbf{P}\mathbf{X}_l\boldsymbol{\Omega}\mathbf{X}_l^T\mathbf{P}^T\}.$$

Alternative Optimization with Stochastic Gradient Descent

- Fix ϕ, Update $\mathbf{P}$

$$\mathbf{P} \leftarrow \mathbf{P} - \eta\frac{\partial\mathcal{J}}{\partial\mathbf{P}}$$

$$\mathbf{P} \leftarrow (\mathbf{P}\mathbf{P}^T)^{-\frac{1}{2}}\mathbf{P} \quad \text{(Orthogonal projection)}$$

- Fix $\mathbf{P}$, Update ϕ

$$\phi \leftarrow \phi - \eta\frac{\partial\mathcal{J}}{\partial\phi}$$

Extensions: SUGAR with Various Autoencoder

SUGAR with Denoising Autoencoder

$$\begin{aligned} \arg\min_{\phi,\mathbf{P}} \quad & \alpha\mathcal{J}_{DAE}(\phi) + (1-\alpha)\mathcal{J}_{SH}(\mathbf{P}) + \frac{\epsilon}{2}\|\mathbf{P}-\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_{\ell_1}, \\ \text{subject to} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I}. \end{aligned} \tag{1}$$

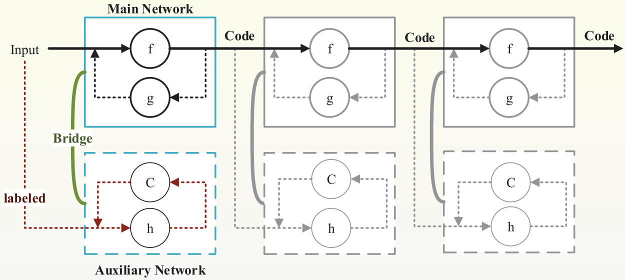
where $\mathcal{J}_{DAE}(\phi) = \sum_{\mathbf{x}\in\mathbf{X}} \mathbb{E}_{\widetilde{\mathbf{x}}\sim q(\widetilde{\mathbf{x}}|\mathbf{x})}\left[\mathcal{L}(\mathbf{x},\widehat{\widetilde{\mathbf{x}}})\right]$.

SUGAR with Contractive Autoencoder

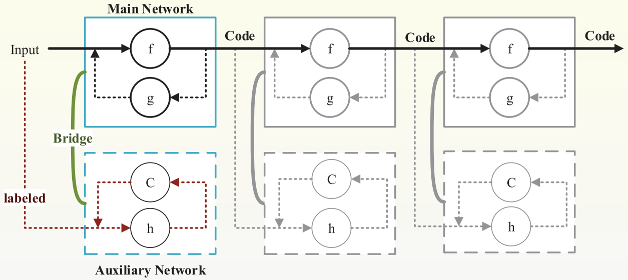
$$\begin{aligned} \arg\min_{\phi,\mathbf{P}} \quad & \alpha\mathcal{J}_{CAE}(\phi) + (1-\alpha)\mathcal{J}_{SH}(\mathbf{P}) + \frac{\epsilon}{2}\|\mathbf{P}-\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_{\ell_1}, \\ \text{subject to} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I}. \end{aligned} \tag{2}$$

where $\mathcal{J}_{CAE}(\phi) = \sum_{\mathbf{x}\in\mathbf{X}} \left(\mathcal{L}(\mathbf{x},\widehat{\mathbf{x}}) + \mu\|J_f(\mathbf{x})\|_F^2\right)$.

Deep SUGARs


Main Network
Input
f
g
Code
f
g
Code
f
g
Code
Bridge
labeled
C
h
C
h
C
h
Auxiliary Network

Deep SUGARs



Layer-wise Training

After training, the feedback decoding modules g and the encoder modules h with the corresponding classifier modules (all dashed lines) are discarded and the system is used to produce very compact representations by a feed-forward pass through the chain of encoders f.

Experiments: Datasets

- MNIST: well-known digit classification problem, http://yann.lecun.com/exdb/mnist
- Benchmark classification tasks: http://www.iro.umontreal.ca/~lisa/icml2007
  - Variations on MNIST
  - Discrimination between tall and wide rectangles
  - Recognition of convex sets

Table 1 : Datasets

Data Set	Train	Valid.	Test	Class
MNIST	50000	10000	10000	10
Rectangles	1000	200	50000	2
Rect_{Img}	10000	2000	50000	2
Convex	7000	1000	50000	2
MNIST_{Basic}	10000	2000	50000	10
MNIST_{Rot}	10000	2000	50000	10
MNIST_{Rand}	10000	2000	50000	10
MNIST_{Img}	10000	2000	50000	10
MNIST_{RotImg}	10000	2000	50000	10

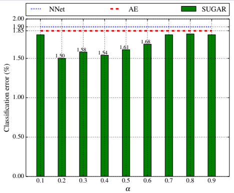
Baseline methods

- SVM
  - SVM-RBF: SVM with RBF kernels
  - SVM-Poly: SVM with polynomial kernels
- NNet: Feed-forward neural network
- GSM: Gated softmax classifier
- NonGSM: Non-factored gated softmax classifier
- SAA: Stacked Autoassociator Network
- RBM: Restricted Boltzmann Machine

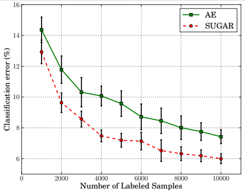
Performance Evaluation: Shallow Architecture on **MNIST**

Performance Evaluation: Shallow Architecture on **MNIST**

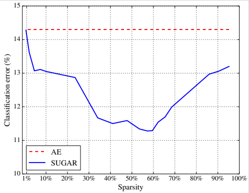
Effect of Different Factors



(a) Guiding Coefficient



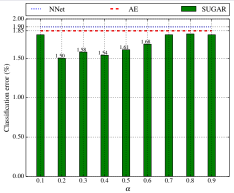
(b) Labeled Data



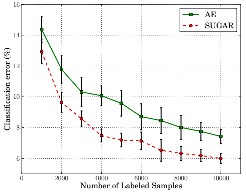
(c) Sparsity Penalty

Performance Evaluation: Shallow Architecture on **MNIST**

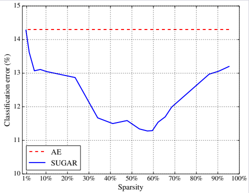
Effect of Different Factors



(a) Guiding Coefficient

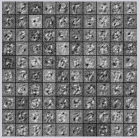


(b) Labeled Data

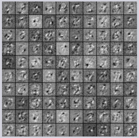


(c) Sparsity Penalty

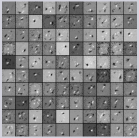
Filters learnt from **MNIST** with various sparsity



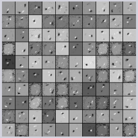
(d) 10%



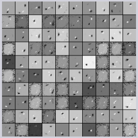
(e) 25%



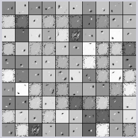
(f) 40%



(g) 50%



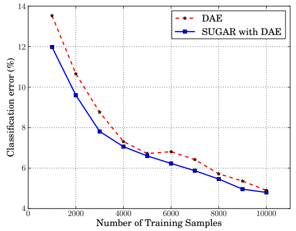
(h) 60%



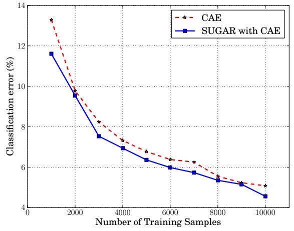
(i) 70%

Performance Evaluation: Shallow Architecture

Guidance to Autoencoder Variants (DAE and CAE)



(a) DAE vs. SUGAR



(b) CAE vs. SUGAR

Figure 2 : Guiding ability on autoencoder variants

Deep Architecture on Benchmark Classification Tasks

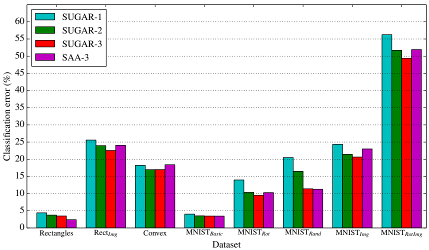


Figure 3 : Classification error rates

Dataset/Model:	SVM-RBF	SVM-Poly	NNet	GSM	NonGSM	SAA-3	RBM	SUGAR-3
Rectangles	02.15	02.15	07.16	0.83	**0.56**	02.41	04.71	03.49
$Rect_{Img}$	24.04	24.05	33.20	**22.51**	23.17	24.05	23.69	**22.55**
Convex	19.13	19.82	32.25	17.08	21.03	18.41	19.92	**17.00**
$MNIST_{Basic}$	**03.03**	03.69	04.69	03.70	03.98	03.46	03.94	03.47
$MNIST_{Rot}$	11.11	15.42	18.11	11.75	16.15	10.30	14.69	**9.53**
$MNIST_{Rand}$	14.58	16.62	20.04	10.48	11.89	11.28	**09.80**	11.40
$MNIST_{Img}$	22.61	24.01	27.41	23.65	22.07	23.00	**16.15**	20.65
$MNIST_{RotImg}$	55.18	56.41	62.16	55.82	55.16	51.93	52.21	**49.40**
Average	18.98	20.27	25.63	18.23	19.25	18.11	18.14	**17.19**

Conclusions

Proposed model: SUGAR

- SUGAR incorporates both week supervision (pairwise constraints) or strong supervision (labeled) into Autoencoder framework
- It is demonstrated that both semi-supervised and supervised SUGAR is consistently more accurate than unsupervised autoencoder

Potential Application Areas

1. Handwriting Recognition
2. Domain Adaptation
3. Telecommunication Data Mining
4. Others
   - Multi-source data
   - Few Labeled data

Codes will be available at `http://kdd2014.noahlab.com.hk/sugar`.

Q & A

Thanks