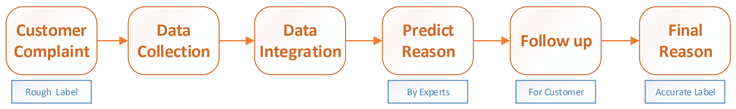
Supervised Deep Learning with Auxiliary Networks

Junbo Zhang, Guangjian Tian, Yadong Mu, Wei Fan

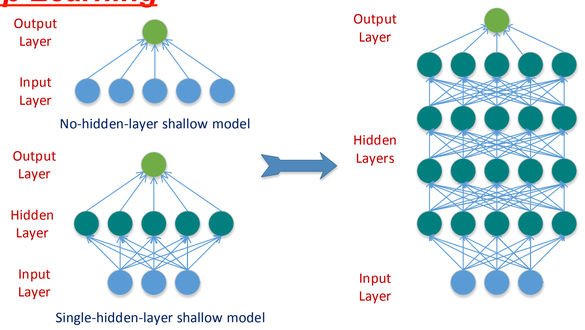
Huge data, but few labeled.

Labeling Data is Very Expensive & Time-consuming



Example: Find the reason why customers complain the quality of 2/3/4G networks

Deep Learning



Why Deep Learning?

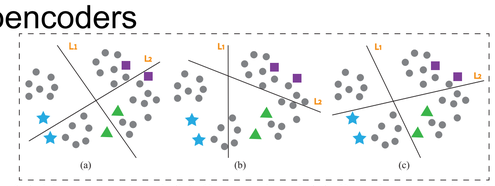
To model **high-level** abstractions by using architectures composed of multiple non-linear transformations.

Success in

- ✧ Computer vision: ImageNet, Face Recognition
- ✧ Speech recognition: Android Voice Recognition
- ✧ Natural language processing: Machine translation, Matching Short Text

Related work

- ✧ Supervised learning: Convolutional Neural Networks, Recurrent Neural Networks
- ✧ Unsupervised learning: Restricted Boltzmann Machines, Autoencoders
- ✧ Semi-supervised learning: Nonparametrically Guided Autoencoder, Semi-Supervised Recursive Autoencoders

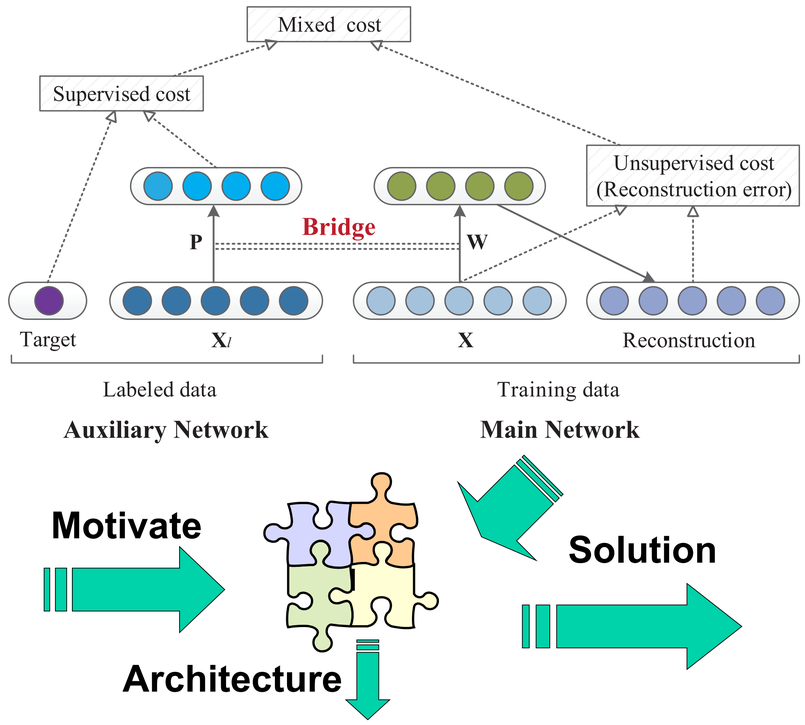


(a) Unsupervised (b) Supervised (c) Semi-supervised

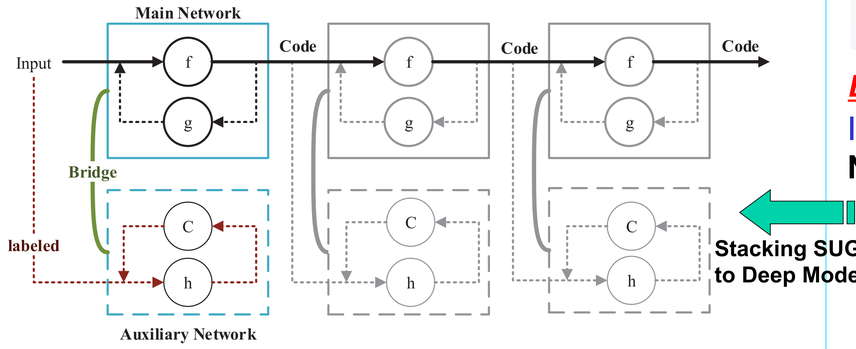
Problems and Shortcoming

- ✧ Ineffectively handle sparse side information
- ✧ Sample-specific annotations are always required

Solution: SUpervision-Guided AutoencodeR (SUGAR)



Deep Architecture with SUGAR



Extensions: SUGAR with Various Autoencoders

SUGAR with Denoising Autoencoder

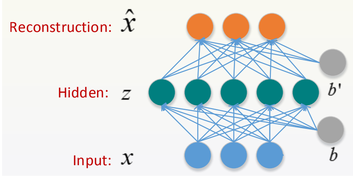
$$\arg\min_{\phi,\mathbf{P}} \quad \alpha\mathcal{J}_{DAE}(\phi) + (1-\alpha)\mathcal{J}_{SH}(\mathbf{P}) + \frac{\epsilon}{2}\|\mathbf{P}-\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_{\ell_1},$$
$$\text{subject to} \quad \mathbf{P}\mathbf{P}^T = \mathbf{I}.$$

SUGAR with Contractive Autoencoder

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

Main Network

It is used to reconstruct the input.
A sparsity-encouraging variant of autoencoder.



Encoder $\mathbf{z} = f(\mathbf{x}) = S_f(\mathbf{W}\mathbf{x} + \mathbf{b})$
Decoder $\hat{\mathbf{x}} = g(\mathbf{z}) = S_g(\mathbf{W}'\mathbf{z} + \mathbf{b}')$
Reconstruction Error $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$
Objective $\arg\min_{\phi} \sum_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda\|\mathbf{W}\|_{\ell_1}$

Auxiliary Network

It is used to regularize the learnt network by pairwise similarity or dissimilarity constraints.
The supervised hashing learning.

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

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 $\mathbf{H}(\mathbf{X}_l) = \text{sgn}(\mathbf{P}\mathbf{X}_l)$ 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

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

Bridge: Mixed Objective

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

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

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:

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

Alternative Optimization with Stochastic Gradient Descent

- Fix ϕ, Update $\mathbf{P}$

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

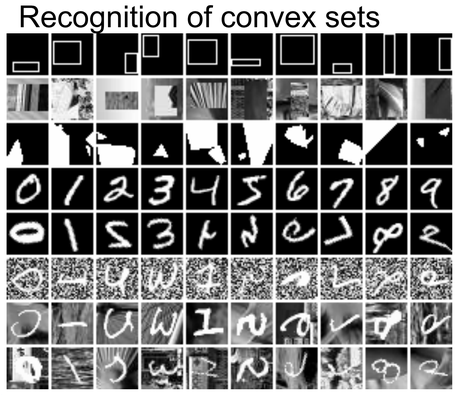
- Fix $\mathbf{P}$, Update ϕ

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

Experiments

Data Sets

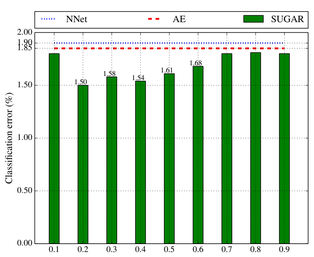
- ✧ MNIST: well-known digit classification problem
- ✧ Benchmark classification tasks
  - Variations on MNIST
  - Discrimination between tall and wide rectangles
  - Recognition of convex sets



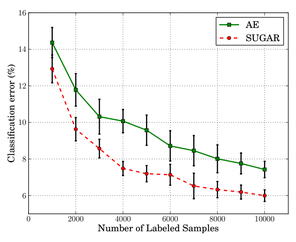
Baseline Methods

Support Vector Machines: SVM-RBF, SVM-Poly
Feed-forward neural network (Nnet)
Gated softmax classifier (GSN)
Stacked Autoassociator Network (SAA)
Restricted Boltzmann Machine (RBM)

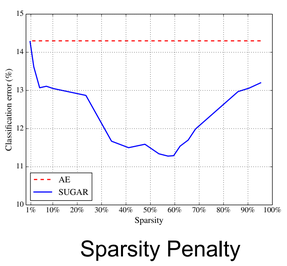
Parameter Sensitivity



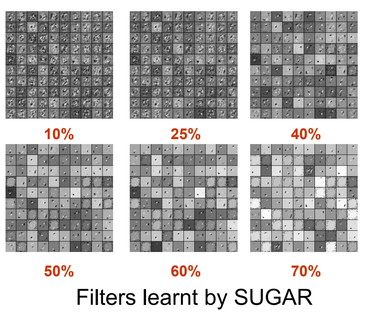
Guiding Coefficient



Labeled Data



Sparsity Penalty



Filters learnt by SUGAR with various sparsity

Classification error rates on the benchmark tasks

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**

Take away messages

SUpervision-Guided AutoencodeR (SUGAR) can effectively handle side information. It is a general model for representation learning from both unlabeled & labeled data.

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