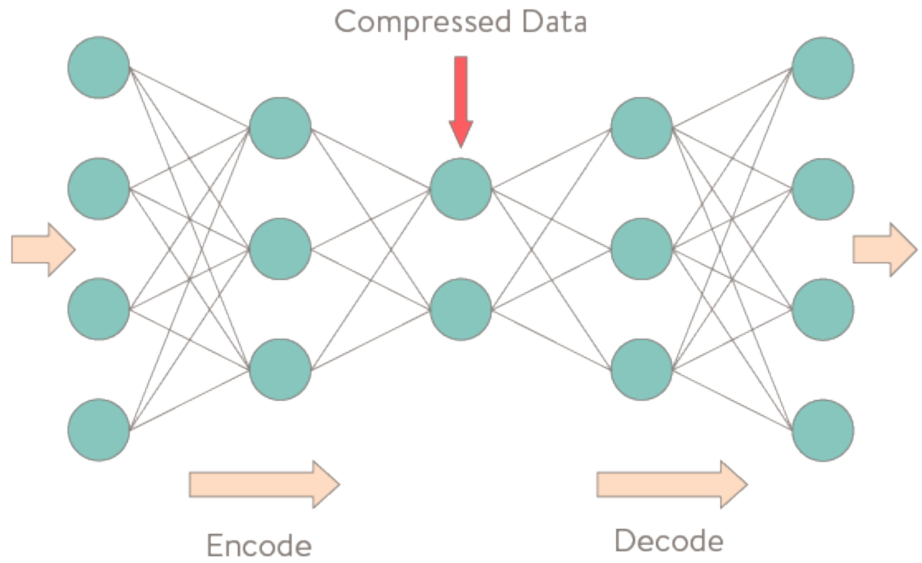
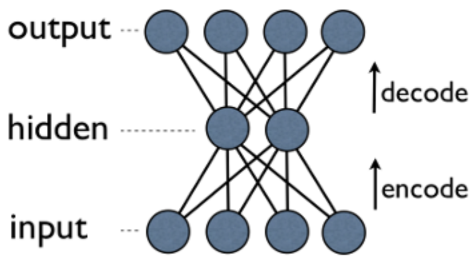


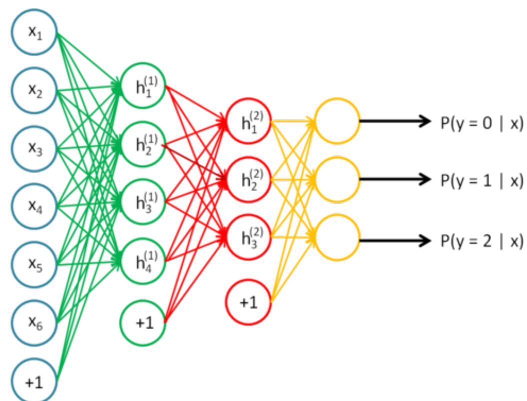
Autoencoders

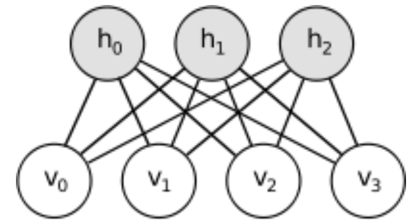
Qian-Yuan TANG
tangqianyuan@gmail.com

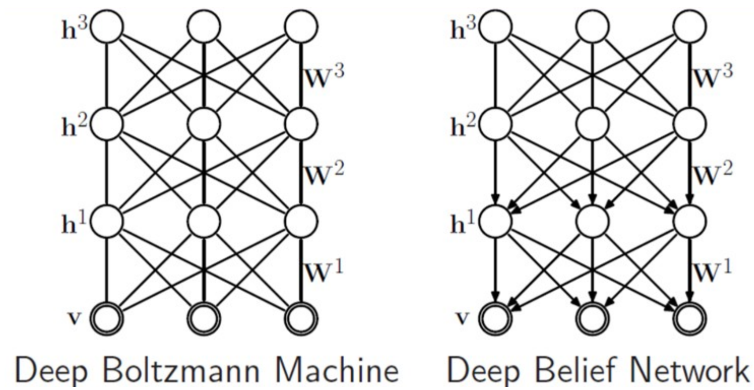
July 13 2017 @CSRC, Beijing

Autoencoders vs. Restricted Boltzmann Machine (RBM)









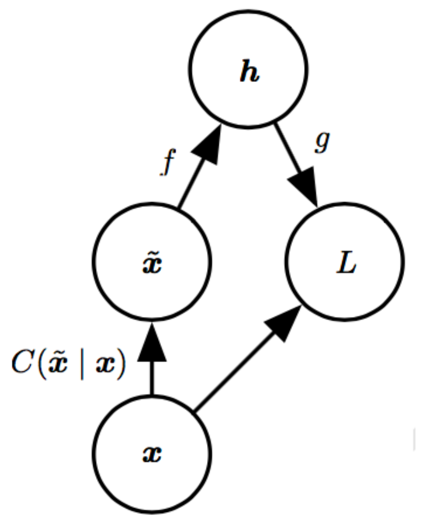
Autoencoders

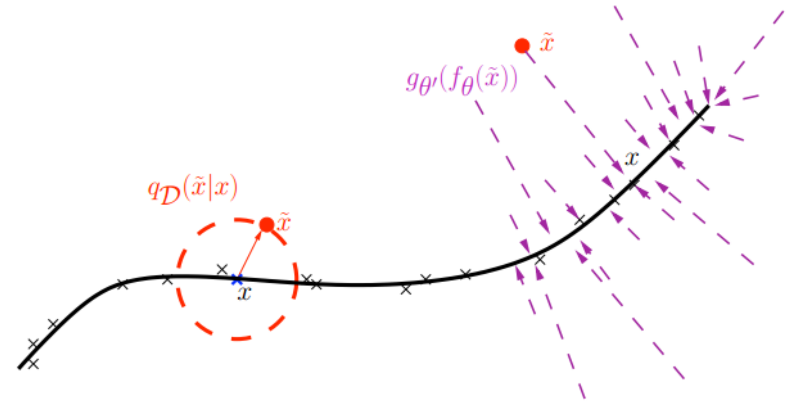
- Deterministic Model
- Training: Minimizing loss function by gradient descent (backpropagation)

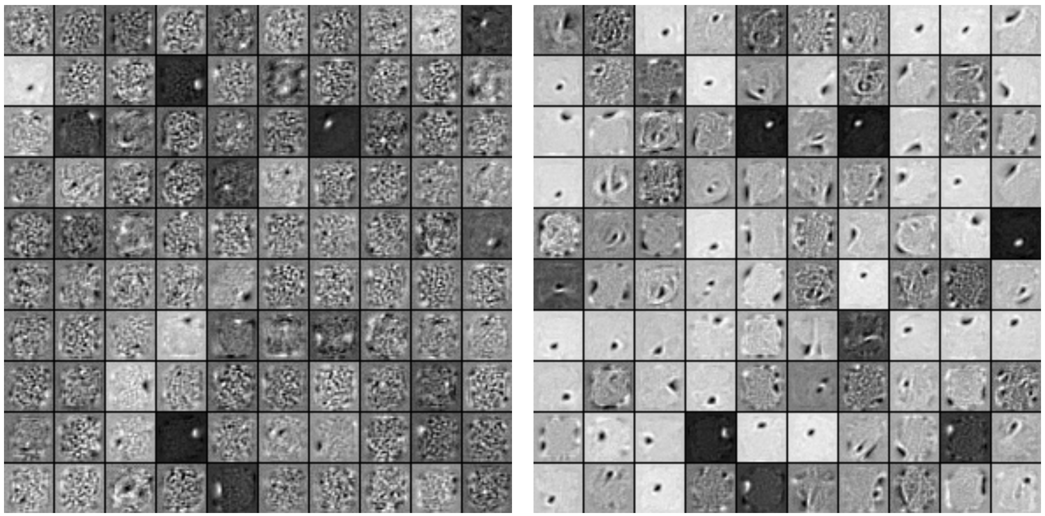
Restricted Boltzmann Machine

- Probabilistic Model (Energy-based model)
- Training: Contrastive divergence (CD)

Denoising Autoencoder (DAE)





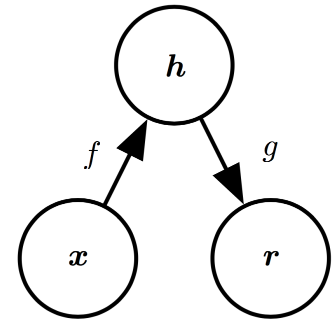


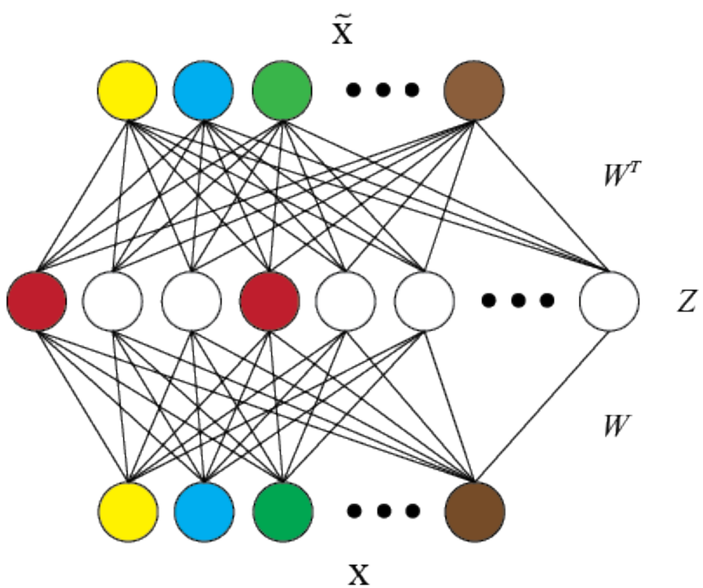
Normal AE: L(x, g(f(x))

DAE: L(x, g(f(x’))

- Perspective of human perception
- Perspective of manifold learning

Sparse Autoencoder





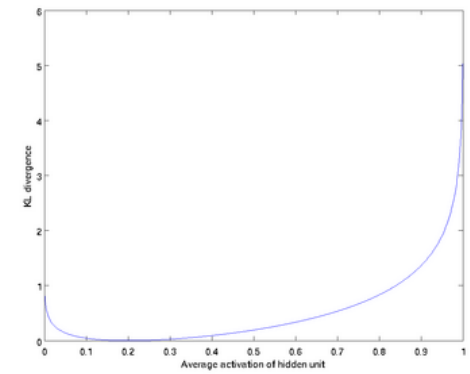
A diagram of a sparse autoencoder network. The input vector **x** is converted to a sparse representation on the hidden layer as **z** and then reconstructed as **x**'.

Normal AE: L(x, g(f(x))

Sparse AE: L(x, g(f(x)) + Ω(h) Penalty term

$$\sum_{j=1}^{s_2} \mathrm{KL}(\rho \| \hat{\rho}_j) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}.$$

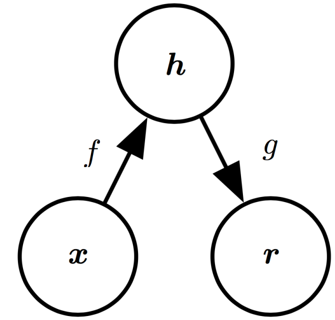
$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} \left[a_j^{(2)}(x^{(i)}) \right]$$



Sparse activation

- For a given hidden node, it's average activation value should be a small value close to zero, e.g., 0.05
- A term is added to the cost function which increases the cost if the above is not true.
- Learning features!

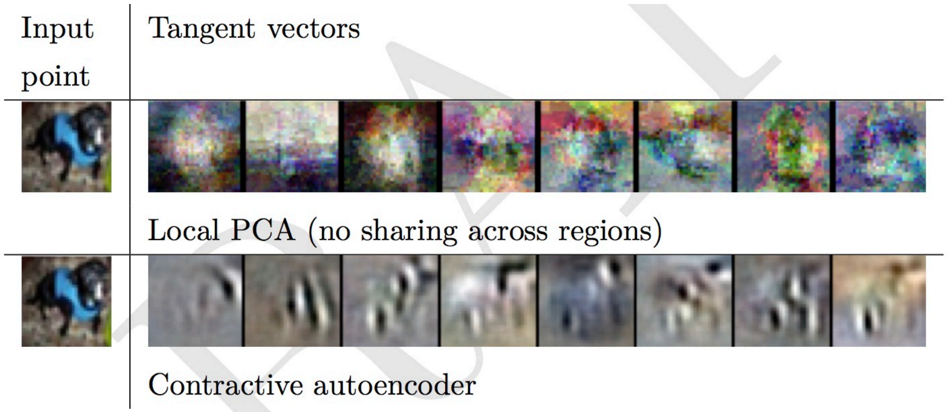
Contrastive Autoencoder



Normal AE: L(x, g(f(x))

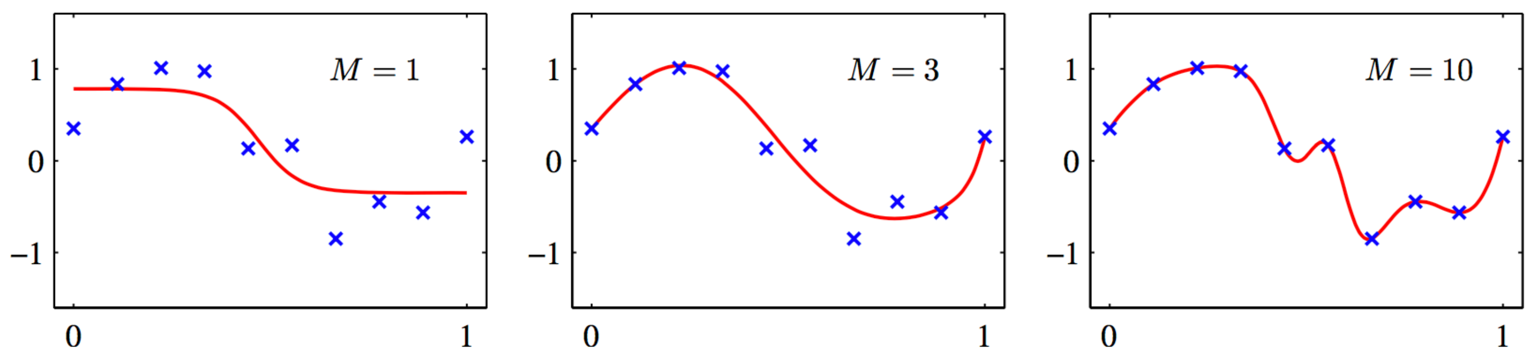
Contrastive AE: L(x, g(f(x)) + Ω(h)

$$\Omega(\boldsymbol{h}) = \lambda \left\| \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_F^2 .$$



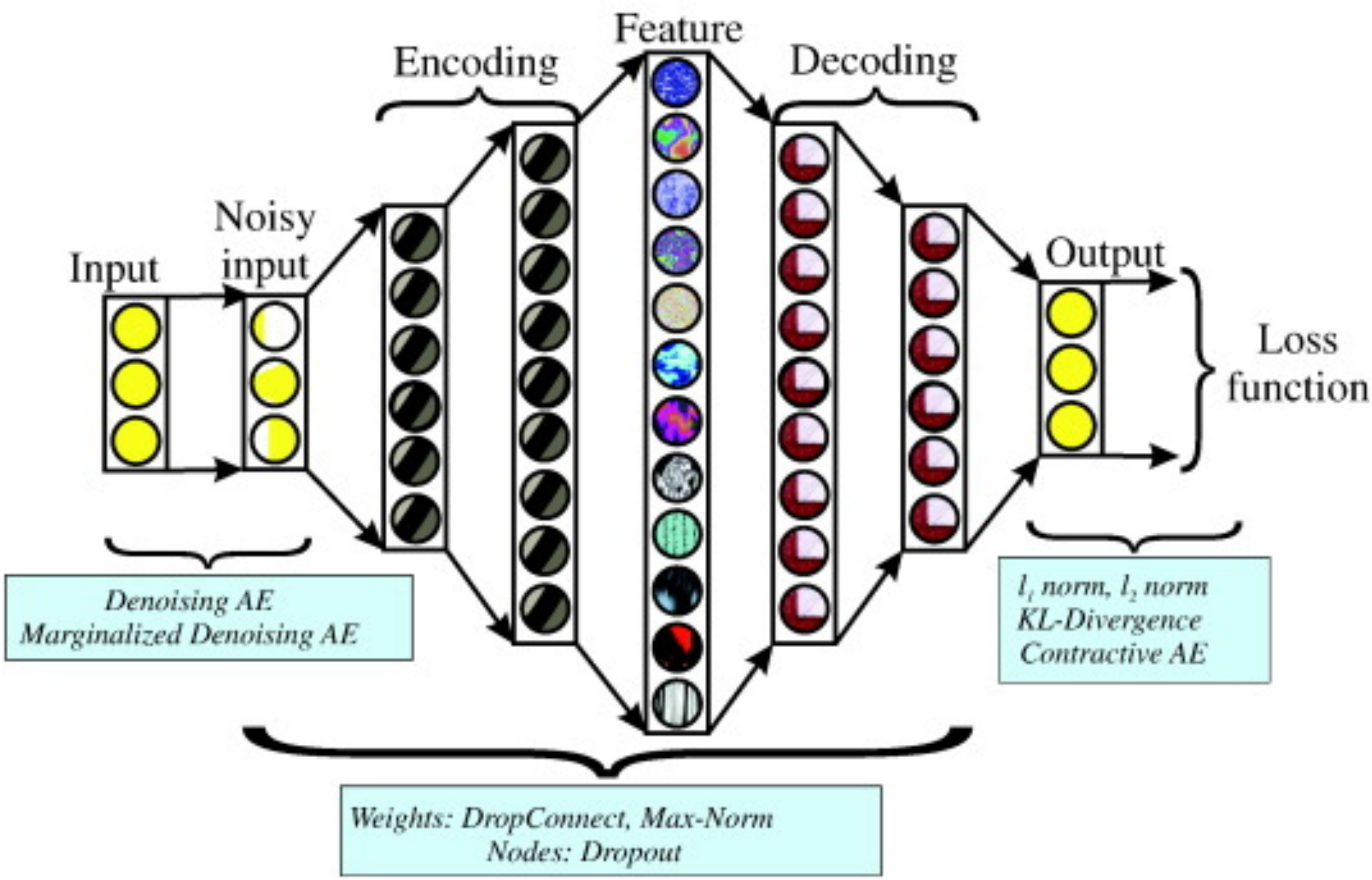
How to understand the regulation?

$$\widetilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}.$$



Rifai, Salah, et al. *(ICML-11)*. 2011.

More Complicated Applications



Sankaran A, Vatsa M, Singh R, et al. Image and Vision Computing, 2017, 60: 64-74.

Autoencoders vs PCA (1)

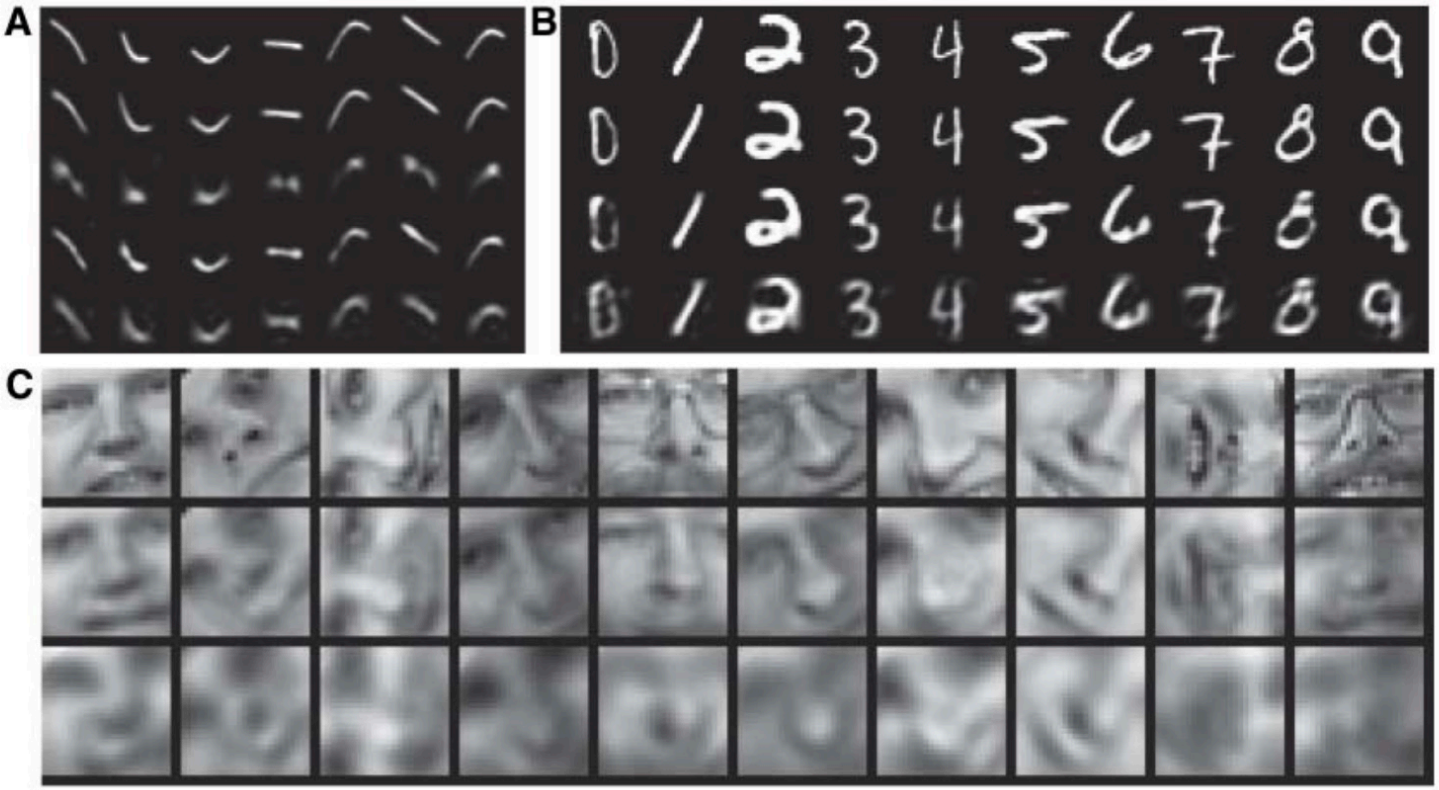
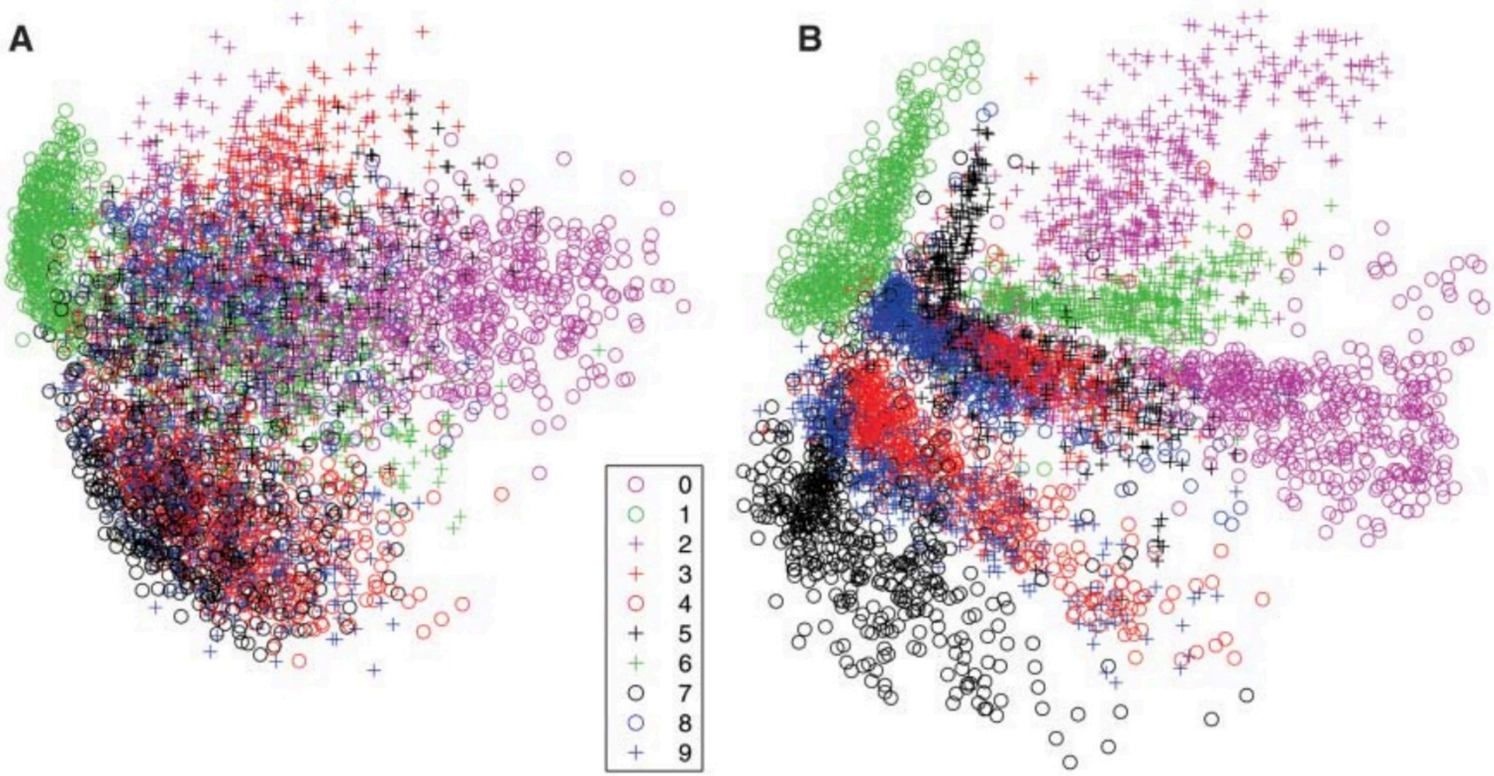


Fig. 2. (**A**) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by "logistic PCA" (*8*) using six components; reconstructions by logistic PCA and standard PCA using 18 components. The average squared error per image for the last four rows is 1.44, 7.64, 2.45, 5.90. (**B**) Top to bottom: A random test image from each class; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional logistic PCA and standard PCA. The average squared errors for the last three rows are 3.00, 8.01, and 13.87. (**C**) Top to bottom: Random samples from the test data set; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional PCA. The average squared errors are 126 and 135.

Autoencoders vs PCA (2)

Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).

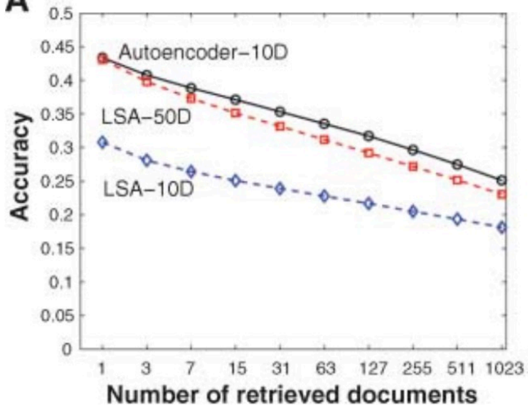


Hinton, Salakhutdinov, Science, 2006

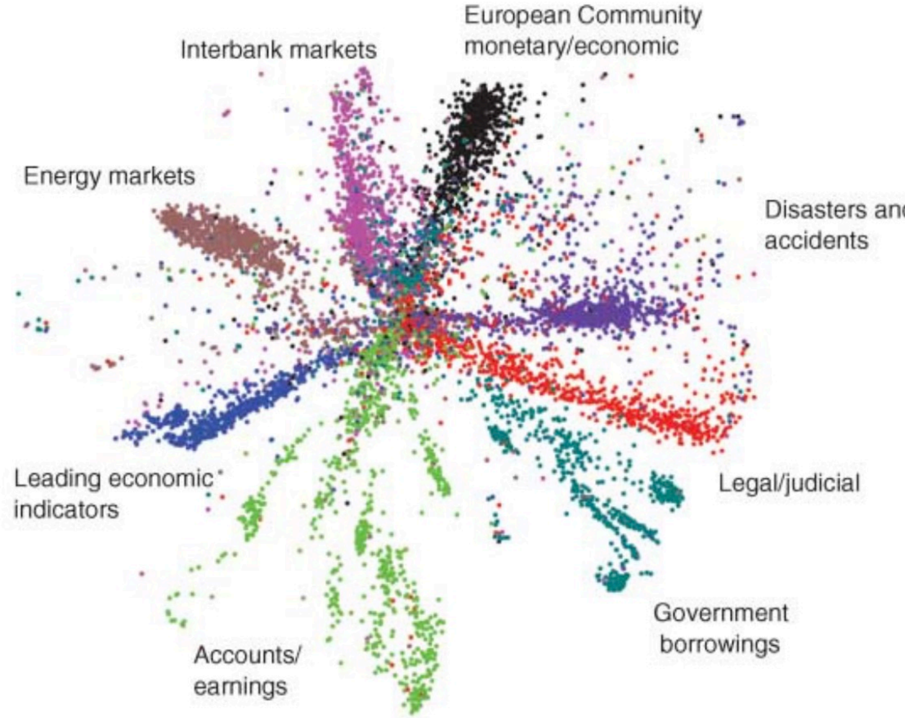
Autoencoders vs PCA (3)

Fig. 4. (**A**) The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. (**B**) The codes produced by two-dimensional LSA. (**C**) The codes produced by a 2000-500-250-125-2 autoencoder.

A



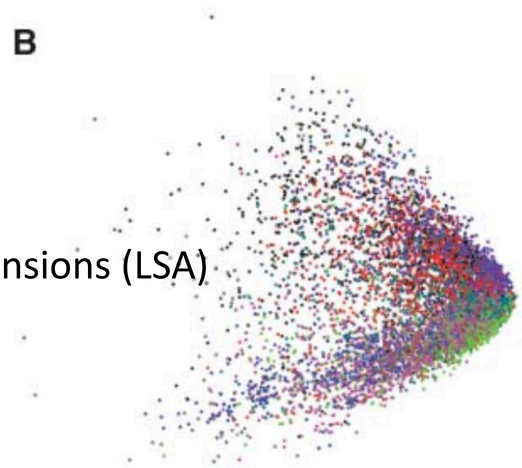
C



Word bag

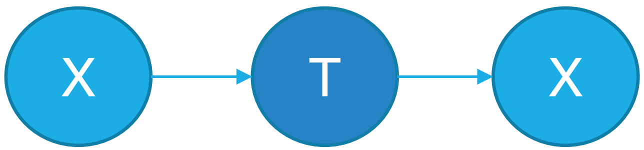
$$
\begin{array}{cc}
 & \mathbf{d}_j \\
 & \downarrow \\
\mathbf{t}_i^T \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix}
\end{array}
$$

B



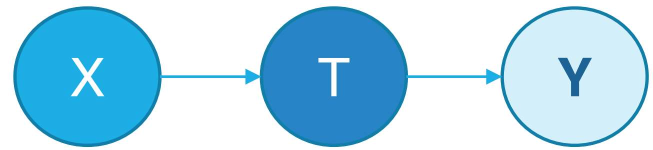
latent semantic dimensions (LSA)

Information Bottleneck Method



Autoencoder

Learning to reconstruct the input data X



Information Bottleneck

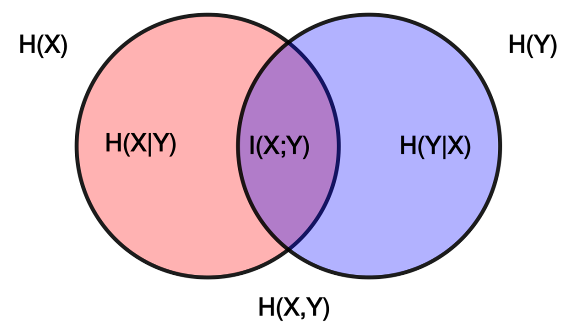
Learning to have a better prediction of Y

Example A: The behaviors of internet users
- X: Demographic and past behaviors
- T: Cluster (community) ID
- Y: Future purchases or click behaviors

$$\min_{q(t|x)} L[q(t|x)] = I(X;T) - \beta I(Y;T)\,,$$

- T compresses (extracts) information from X;
- T has the ability to predict Y.

Example B: Attention and memory
- X: Sensory input
- T: Neural activity and synaptic changes
- Y: Future sensory input

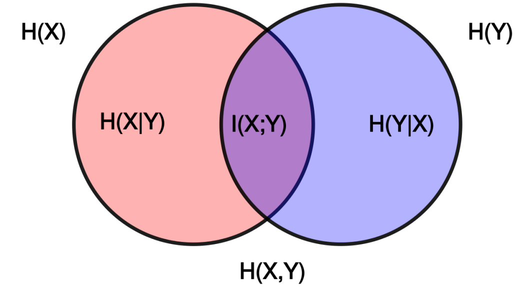


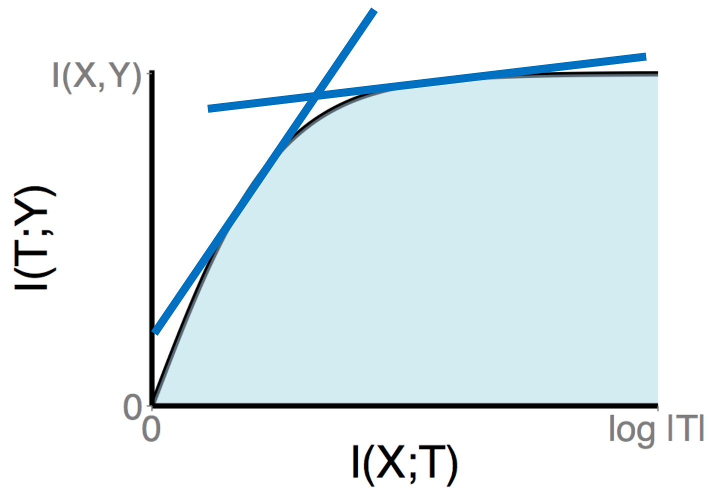
Tishby N, Pereira F C, Bialek W. arXiv preprint physics/0004057, 2000.

Information Bottleneck Method

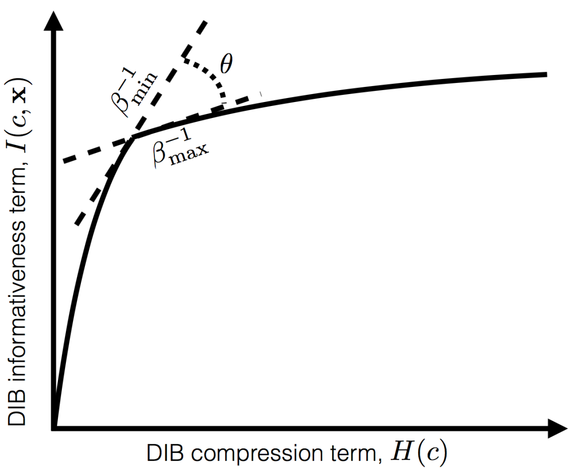
IB Method: $$\min_{q(t|x)} L[q(t|x)] = I(X;T) - \beta I(Y;T),$$

Deterministic IB: $$L_\alpha \equiv H(T) - \alpha H(T \mid X) - \beta I(T;Y).$$





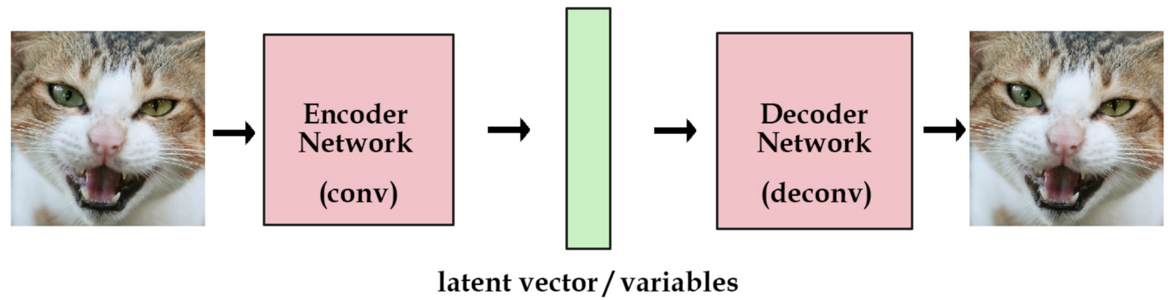
Information Bottleneck (IB)

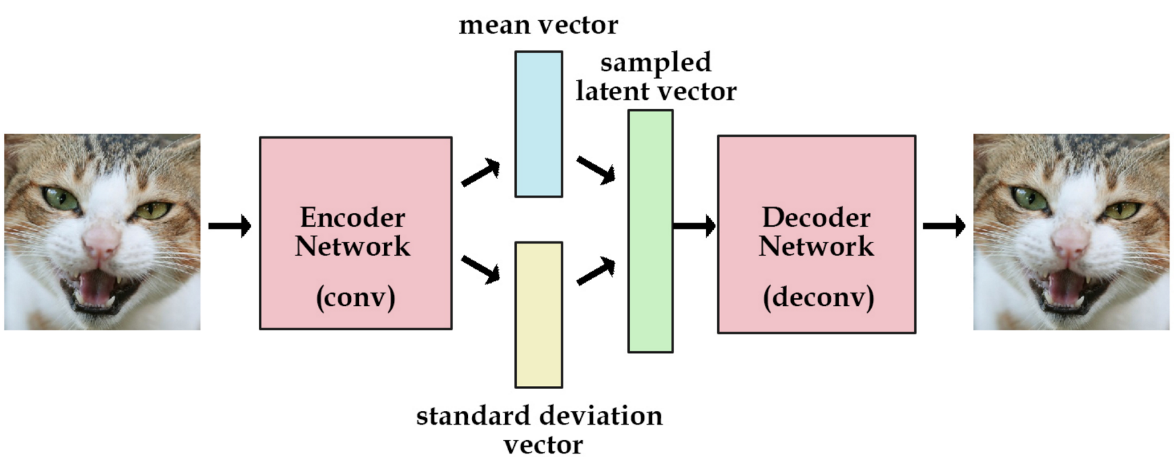


Deterministic IB

Can the idea of information bottleneck method help to design a better autoencoder?

Autoencoders as Generative Models (I): Variational Autoencoder (VAE)



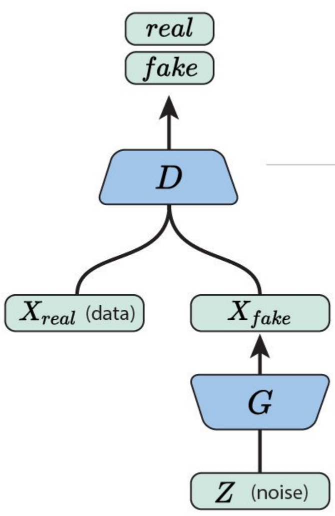


VAE generated

Original

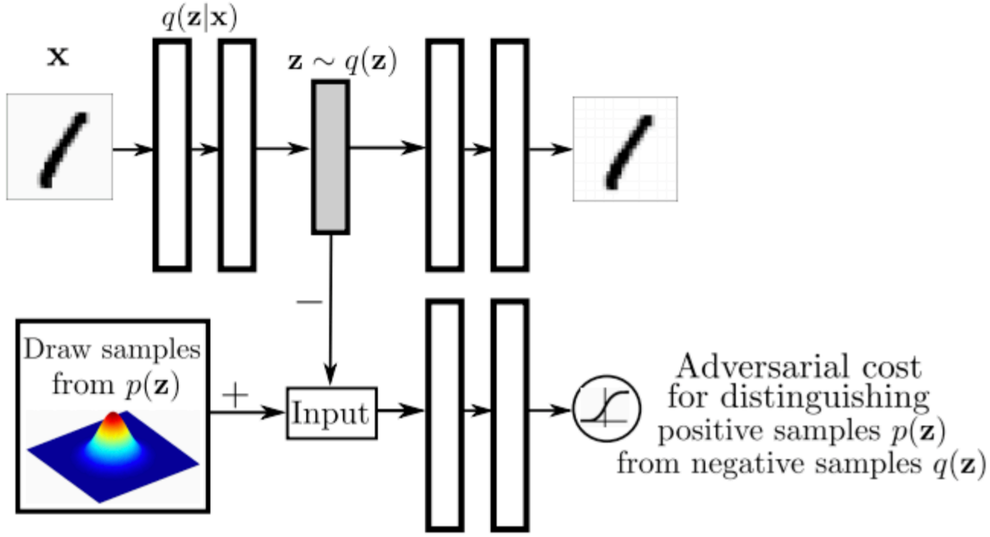
Autoencoders as Generative Models (II): Adversarial Autoencoders(AAE)

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.



Makhzani A, Shlens J, Jaitly N, et al. arXiv preprint arXiv:1511.05644, 2015.

Summary

Background: Dimensionality Reduction

- High Dimensional Descriptions vs Low Dimensional Descriptions
- Principle Component Analysis (PCA)
- The Limits of PCA

Autoencoders

- Basic Introductions
- Stacked Autoencoder and Deep Learning
- Sparse Autoencoder
- Denoising Autoencoder (DAE)
- Contrastive Autoencoder (CAE)
- Applications of Autoencoders

Additional Discussions

- Information Bottleneck Method
- Autoencoders as generative models (VAE and AAE)

Thank you!