Bi-directional learning system

Shikui Tu

Department of Computer Science and Engineering, Shanghai Jiao Tong University

2018-05-17

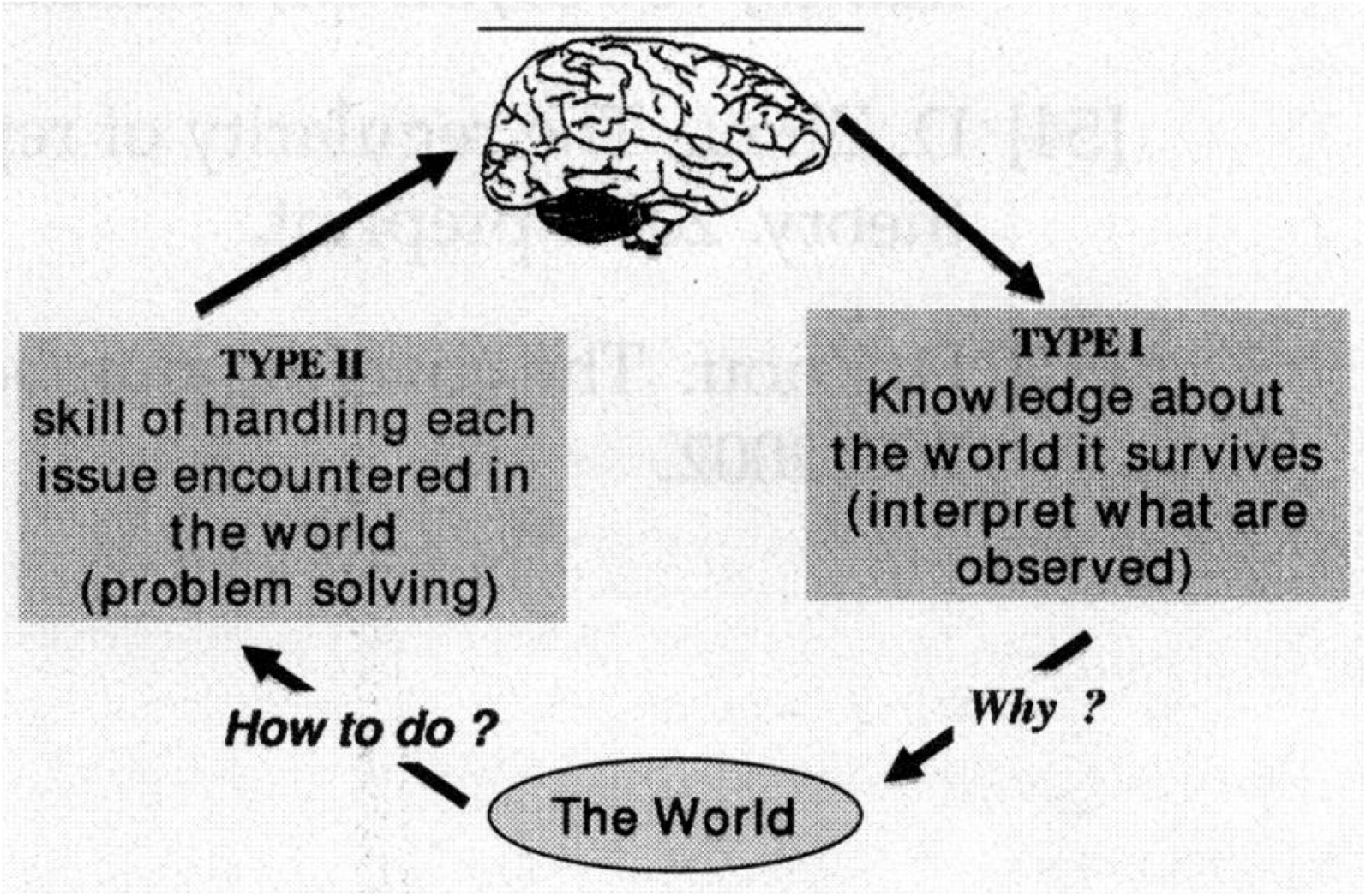
Outline

- **Fundamentals and challenges of learning**
  - Two types of intelligent abilities and learning
  - Three levels of inverse problems

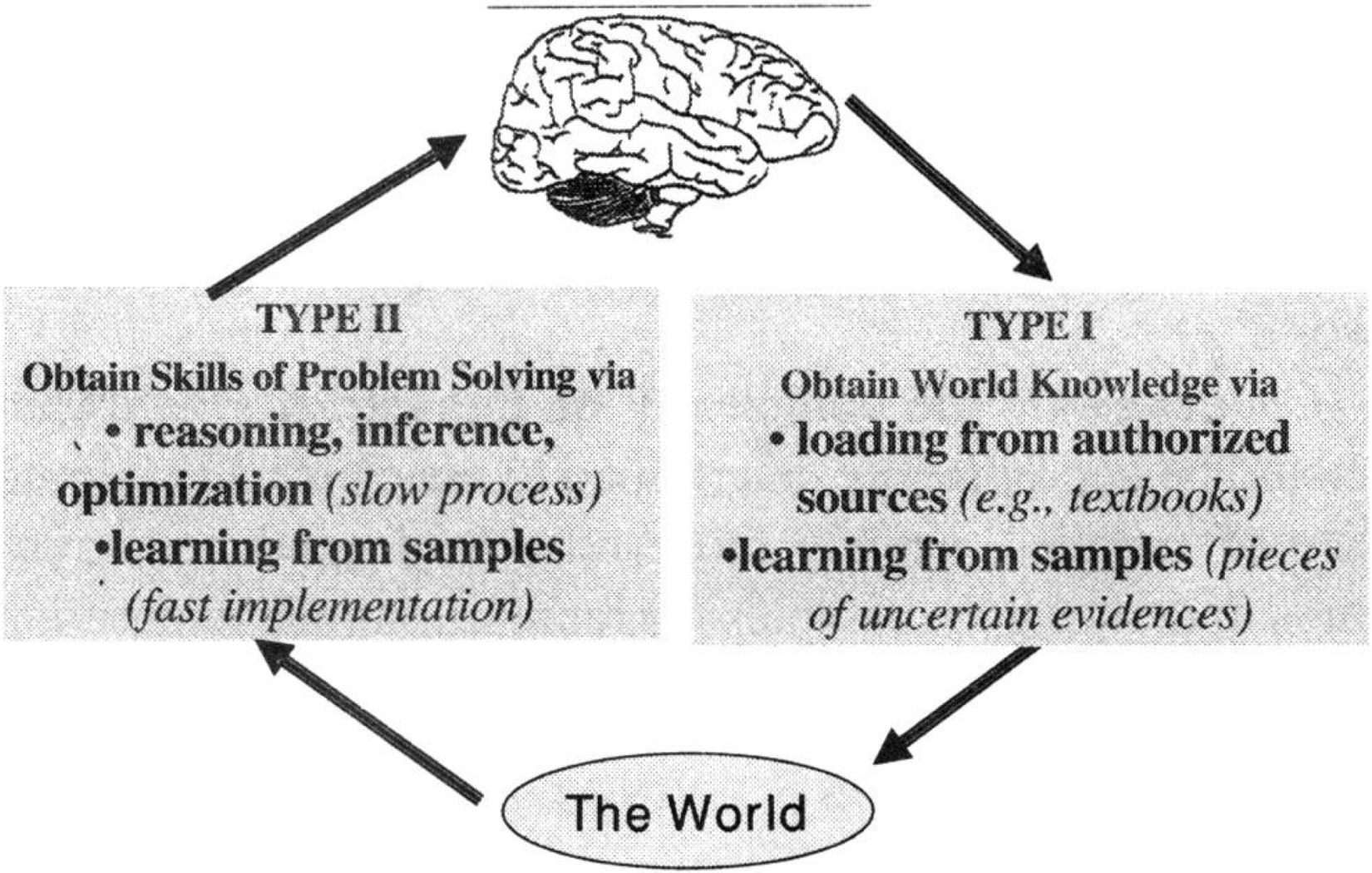
- Bi-directional learning
  - Inbound learning theory
  - Outbound learning theory
  - Bi-directional architectures

- Early bi-directional deep learning

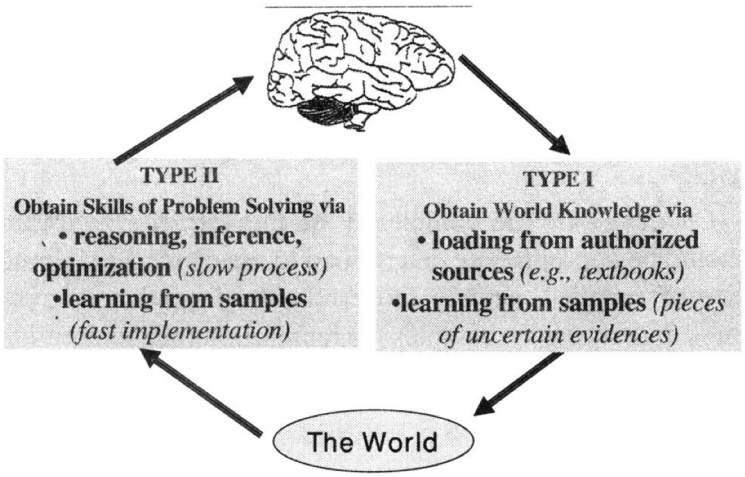
Two types of intelligent ability

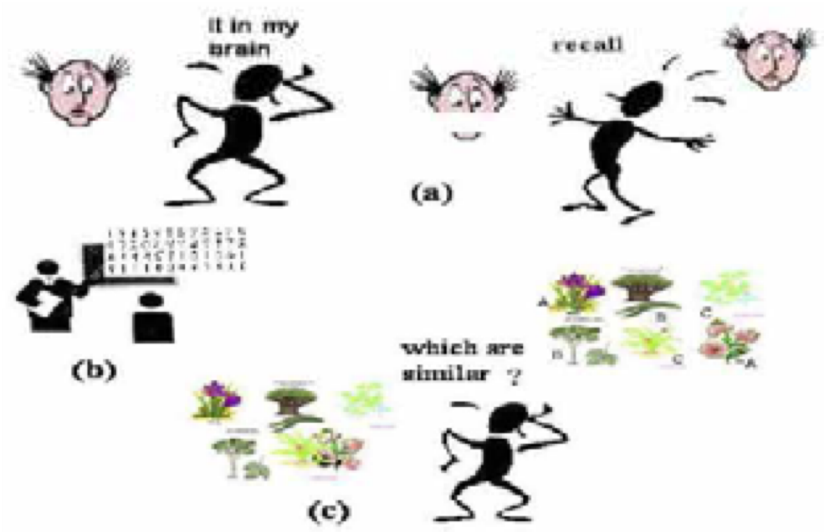


How to get the abilities

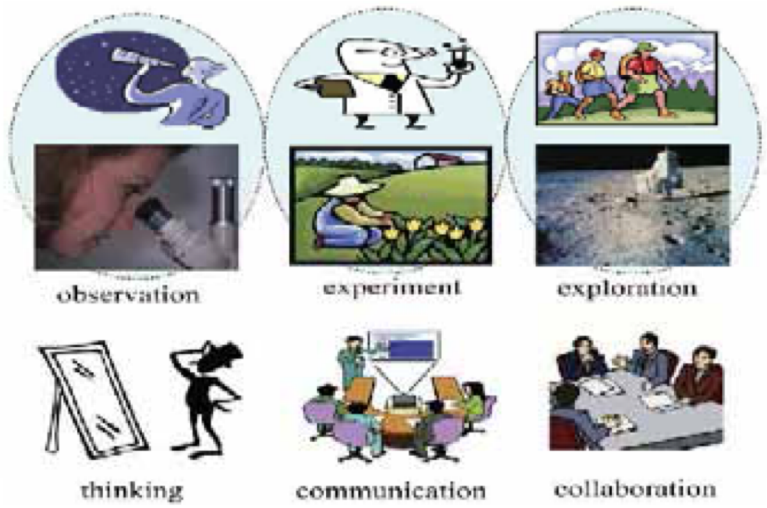


Abilities and problem-solving skills





(I) perceptron



Abilities of knowledge discovery

(II) control

Two types of learning

Type II

Training skills of problem-solving

via building up input-response type dependence per sample

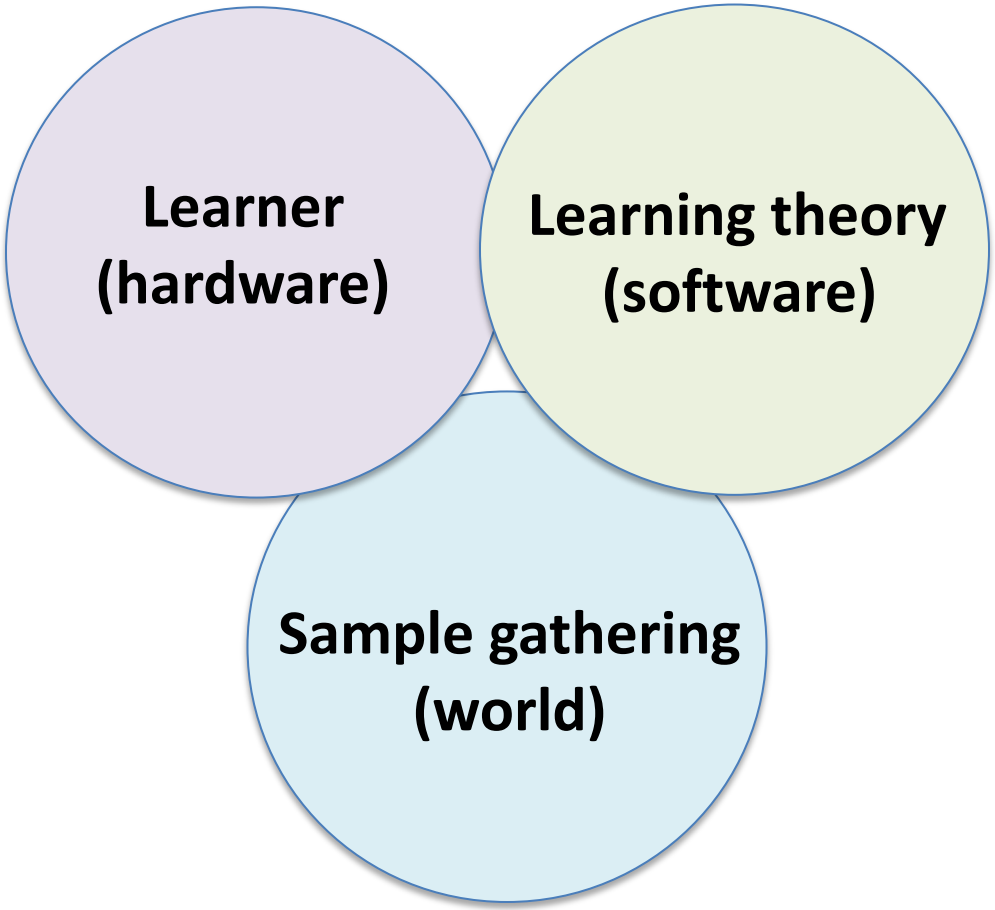
Type I

Discovering World Knowledge

via mining invariant dependence underlying a set of all samples

The world

Key ingredients of learning

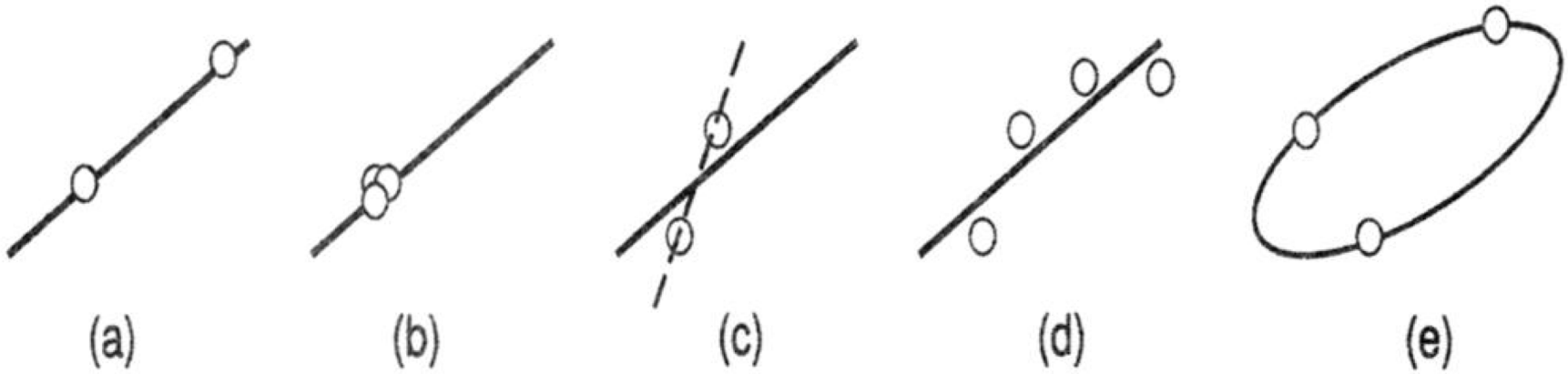


Two key learning challenges

- **Challenge I**
  - The learner's hardware should be designed not only be able to accommodate but also appropriately match the interested dependence structures underlying the world.
- **Challenge II**
  - The complexity of the learner's hardware should be appropriately determined to match usually a finite size samples, namely those reliable dependence structures underlying the samples for representing the underlying world.

Uncertainty in samples and randomly sampling

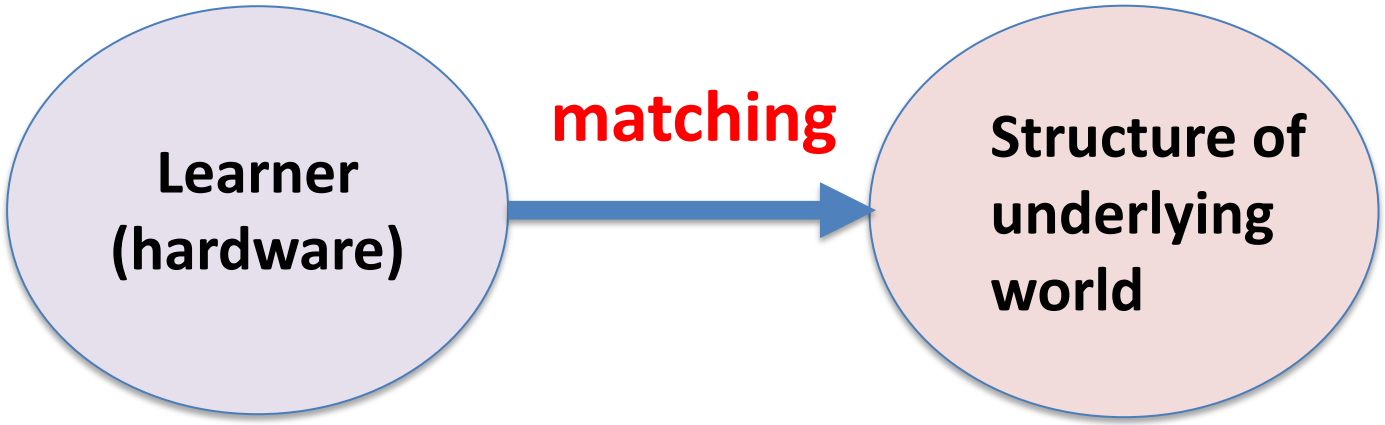
Consider a simple problem of learning a line from samples:



(a) Conceptual case
(b) Fail when two samples are the same
(c) Noise in gathering samples, quantization effects
(d) More samples to reduce uncertainty
(e) To find a more complicated structure, but not enough samples

→ Statistical learning

Learner represents data

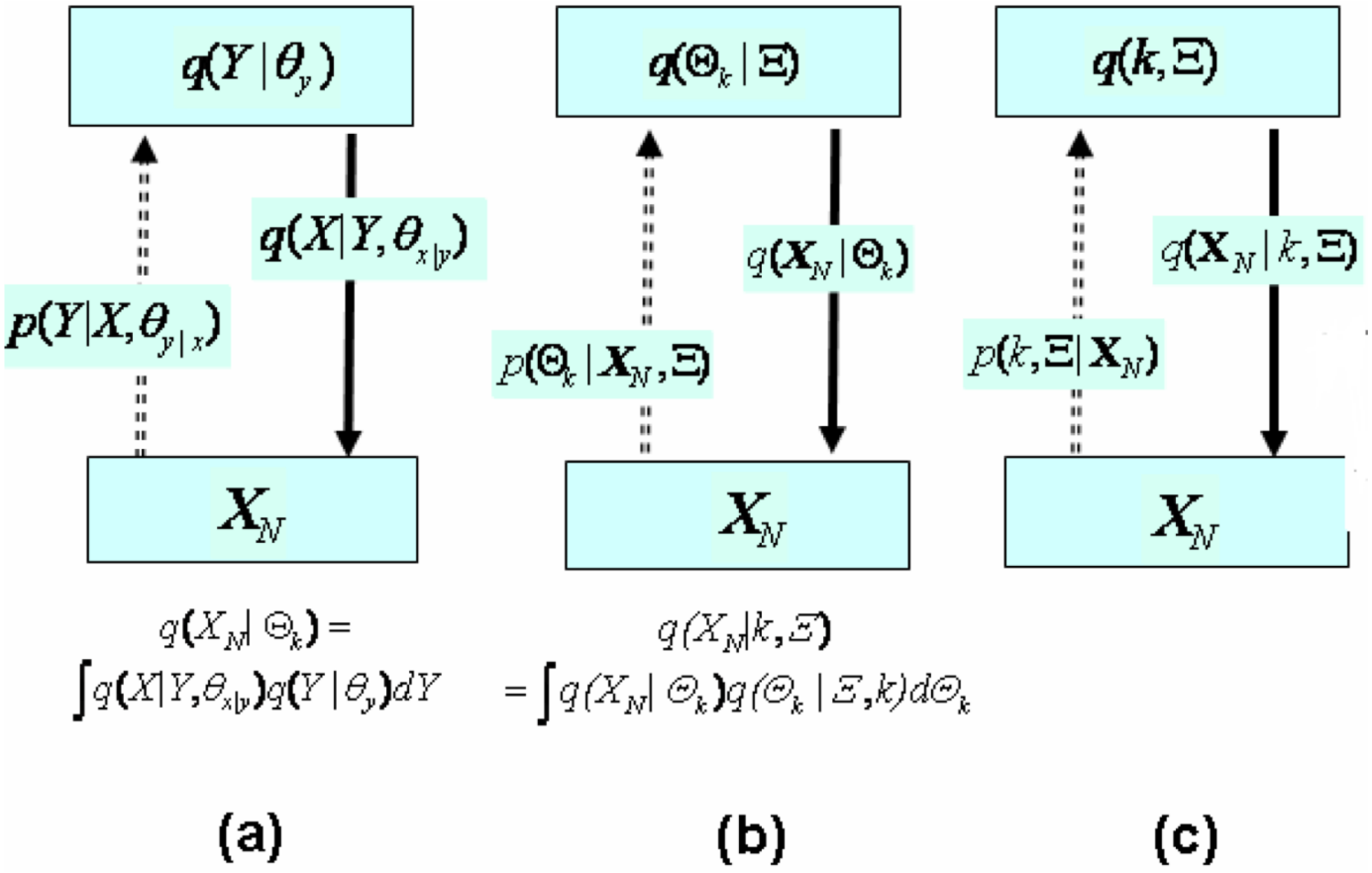


Learner's hardware appropriately represents dependence among data

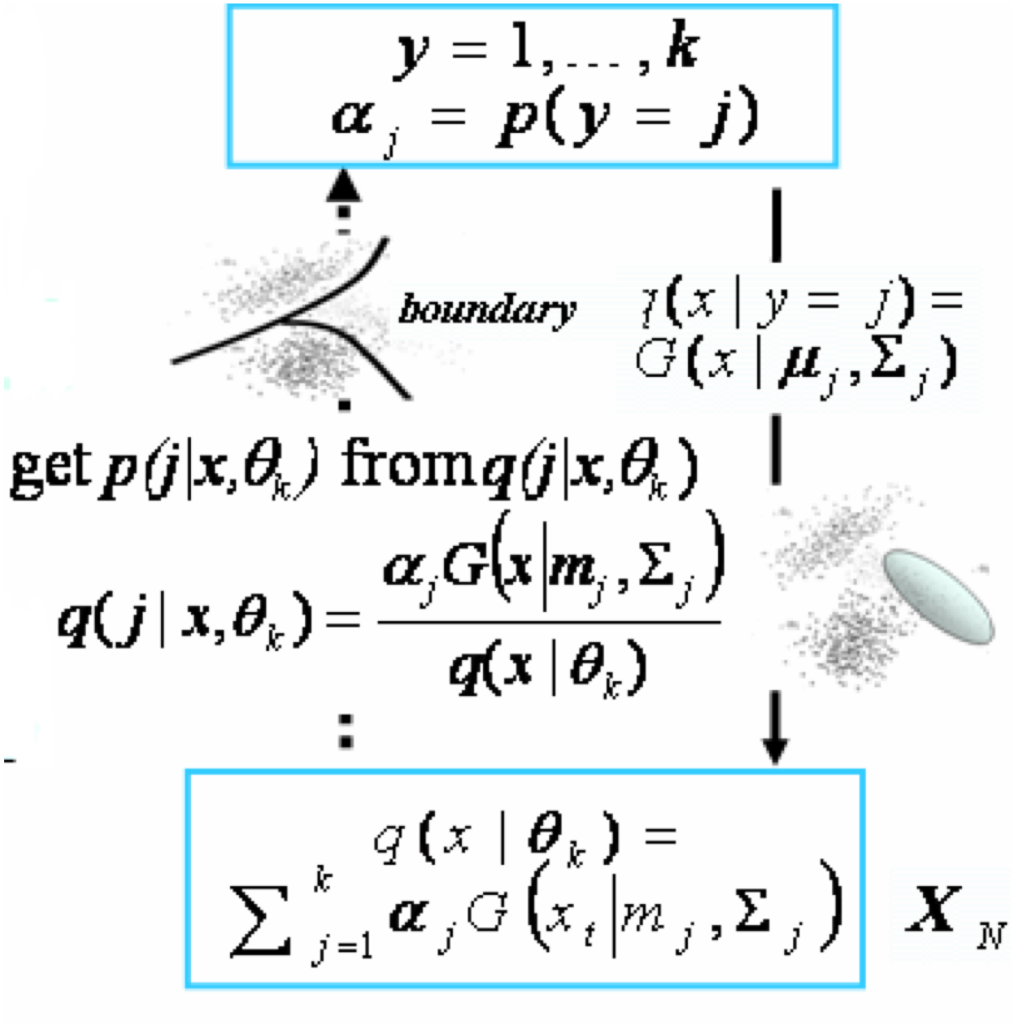
Outline

- **Fundamentals and challenges of learning**
  - Two types of intelligent abilities and learning
  - Three levels of inverse problems
- Bi-directional learning
  - Inbound learning theory
  - Outbound learning theory
  - Bi-directional architectures
- Early bi-directional deep learning

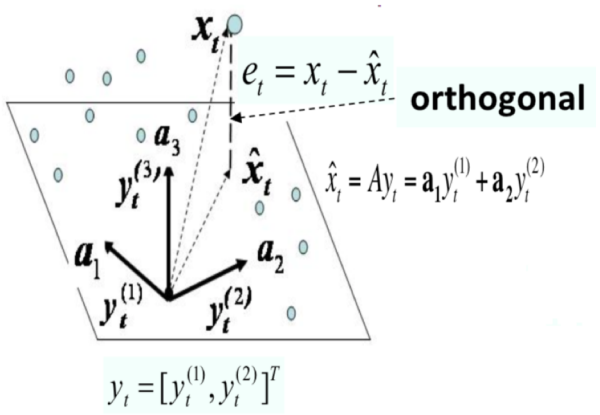
Three levels of inverse problems

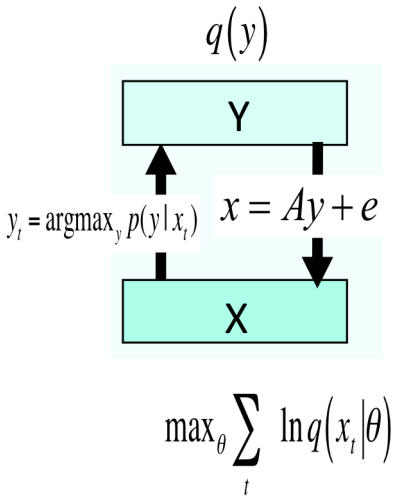


Learning Gaussian Mixture Models



Learning Factor Analysis





$$\max_\theta \sum_t \ln q(x_t|\theta)$$

$$q(y) = G(y|v, \Lambda)$$

Λ =diag[$\lambda_1, \cdots \lambda_m$] denotes a diagonal matrix with diagonal elements $\lambda_1, \cdots \lambda_m$.

$$x = Ay + e$$

$$p(y|x) = q(y|x)$$

$$E y e^T = 0$$

$$q(y|x) = \frac{G(x|Ay+\mu, \Sigma)G(y|v, \Lambda)}{G(x|\mu, A\Lambda A^T + \Sigma)}$$

$$y = W(x - \mu) + \varepsilon$$

$$q(x|y) = G(e|\mu, \Sigma)$$

$$G(x|\mu, A\Lambda A^T + \Sigma) = \int G(x|Ay+\mu, \Sigma)G(y|v, \Lambda)dy$$

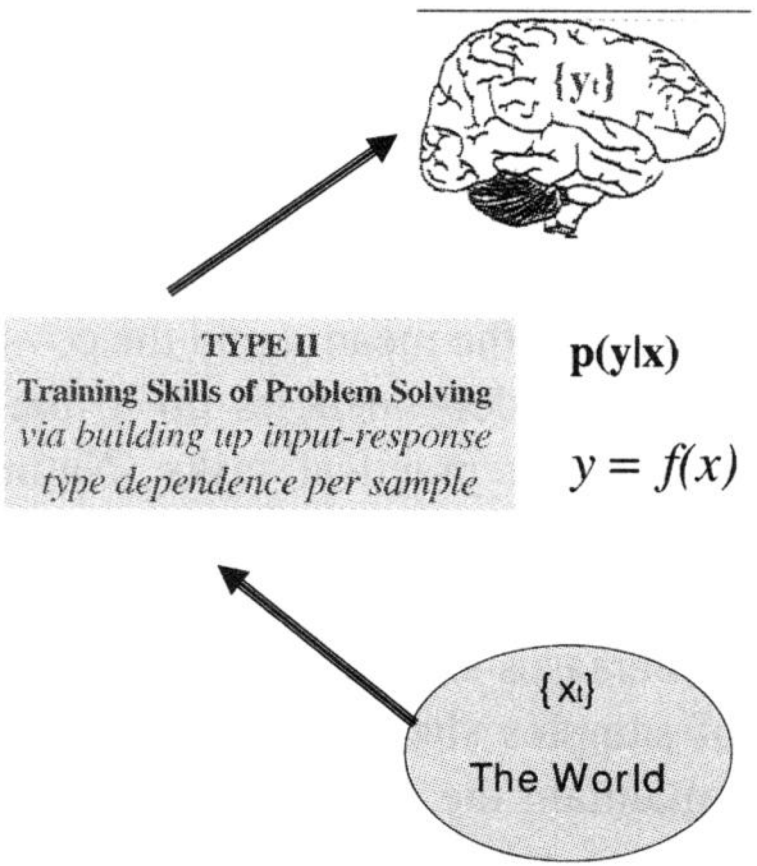
Outline

- **Fundamentals and challenges of learning**
  - Two types of intelligent abilities and learning
  - Three levels of inverse problems

- Bi-directional learning
  - **Inbound learning theory**
  - Outbound learning theory
  - Bi-directional architectures

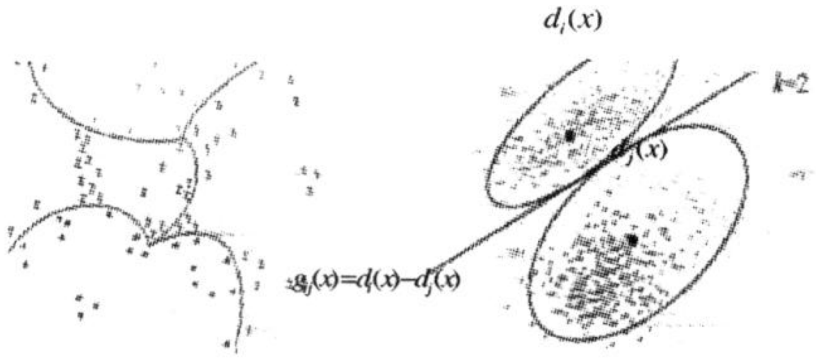
- Early bi-directional deep learning

Inbound architecture



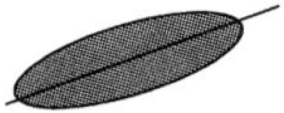
Pairwise structures

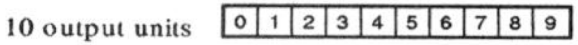
$\{(x_t, y_t)\}$

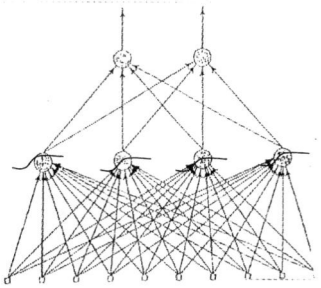


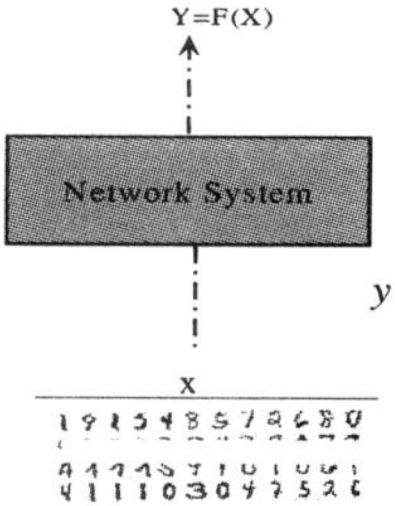
(a) classification

(b) regression



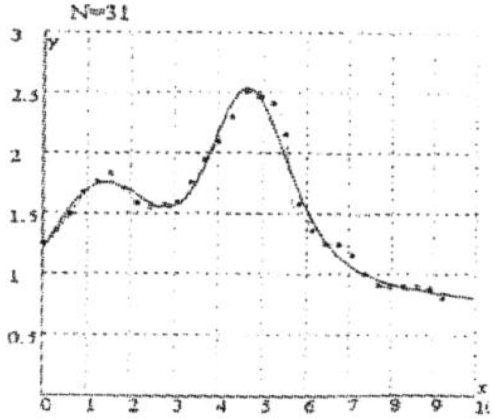




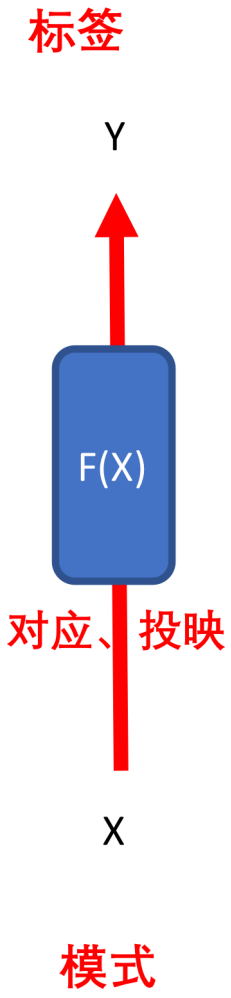


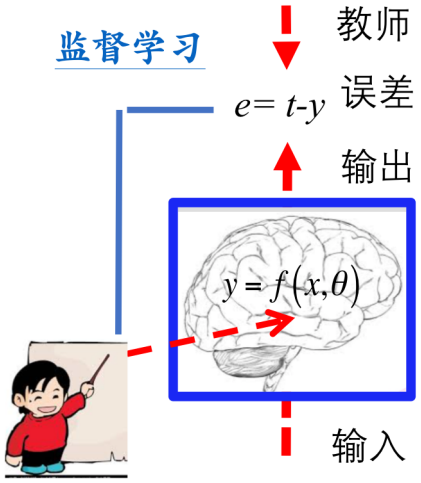
$y = f(x)$

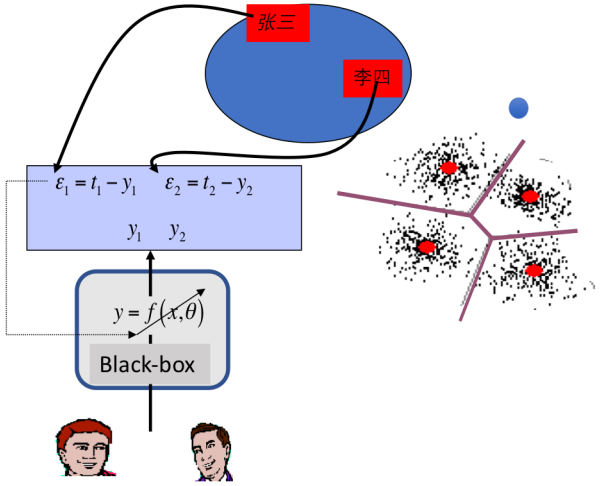
$E(y|x) = \int y p(y|x) dy$



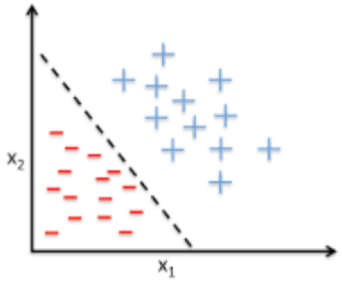
最佳拟合、最大区分、最小错误



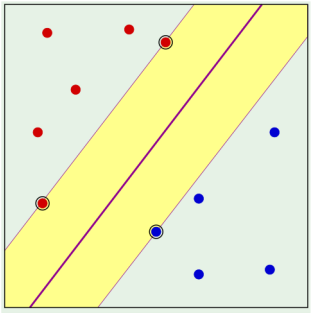




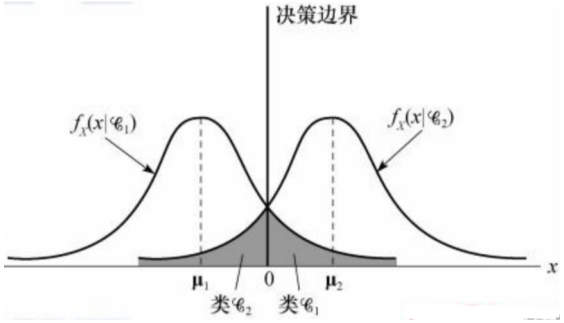
Perceptron



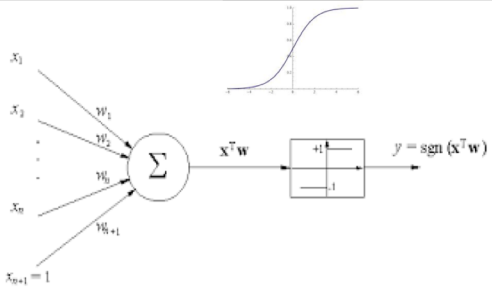
SVM

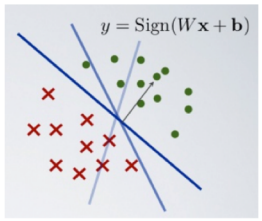


Statistical decision

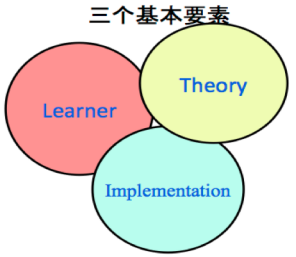


From perceptron to deep learning



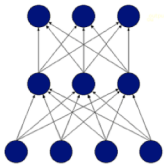


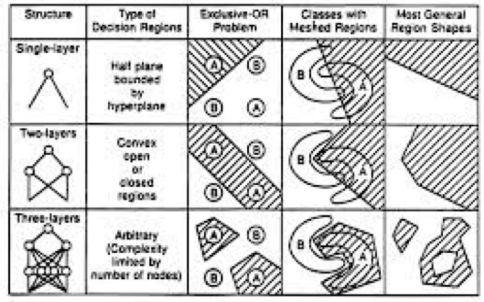
Frank Rosenblatt
(1928-1971)



Rosenblatt的Perceptron Learning(1957)

Minsky and Papert Perceptrons (1969)
(unable to learn an XOR function ???)



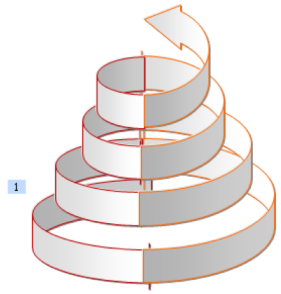


到Rumelhart, McClelland & Hinton 的BP-learning(1986)

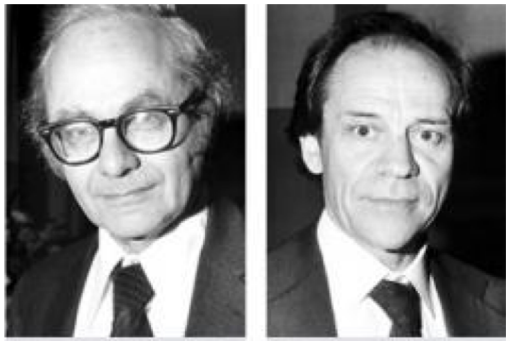
再到Hinton 的 Deep learning (2006),

突破的主要原因是计算技术和信息技术的发展积累所产生的高速计算能力和大数据的获得和处理能力

完成了螺旋上升之肯定－否定－再肯定－再否定－出现突破

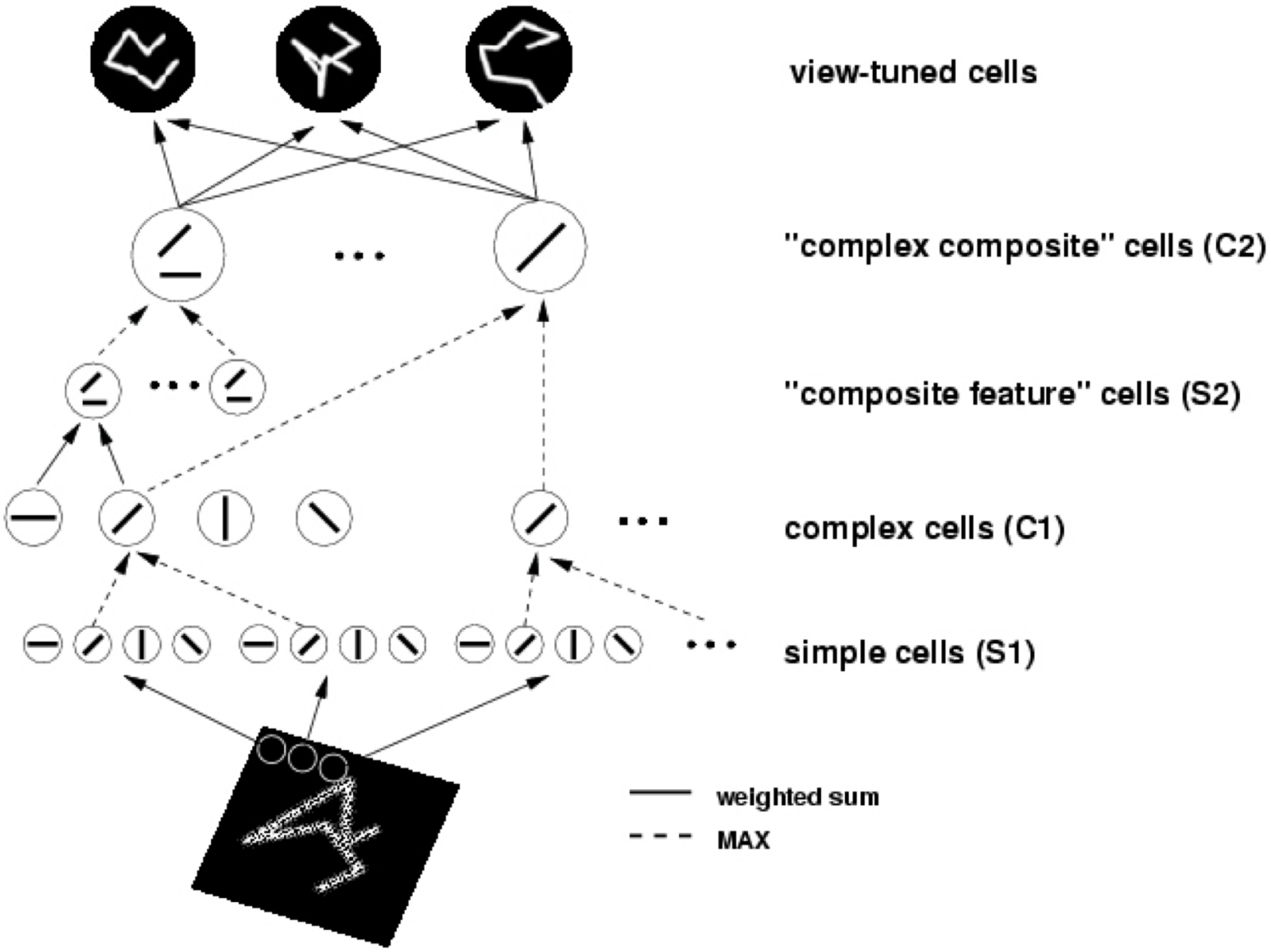


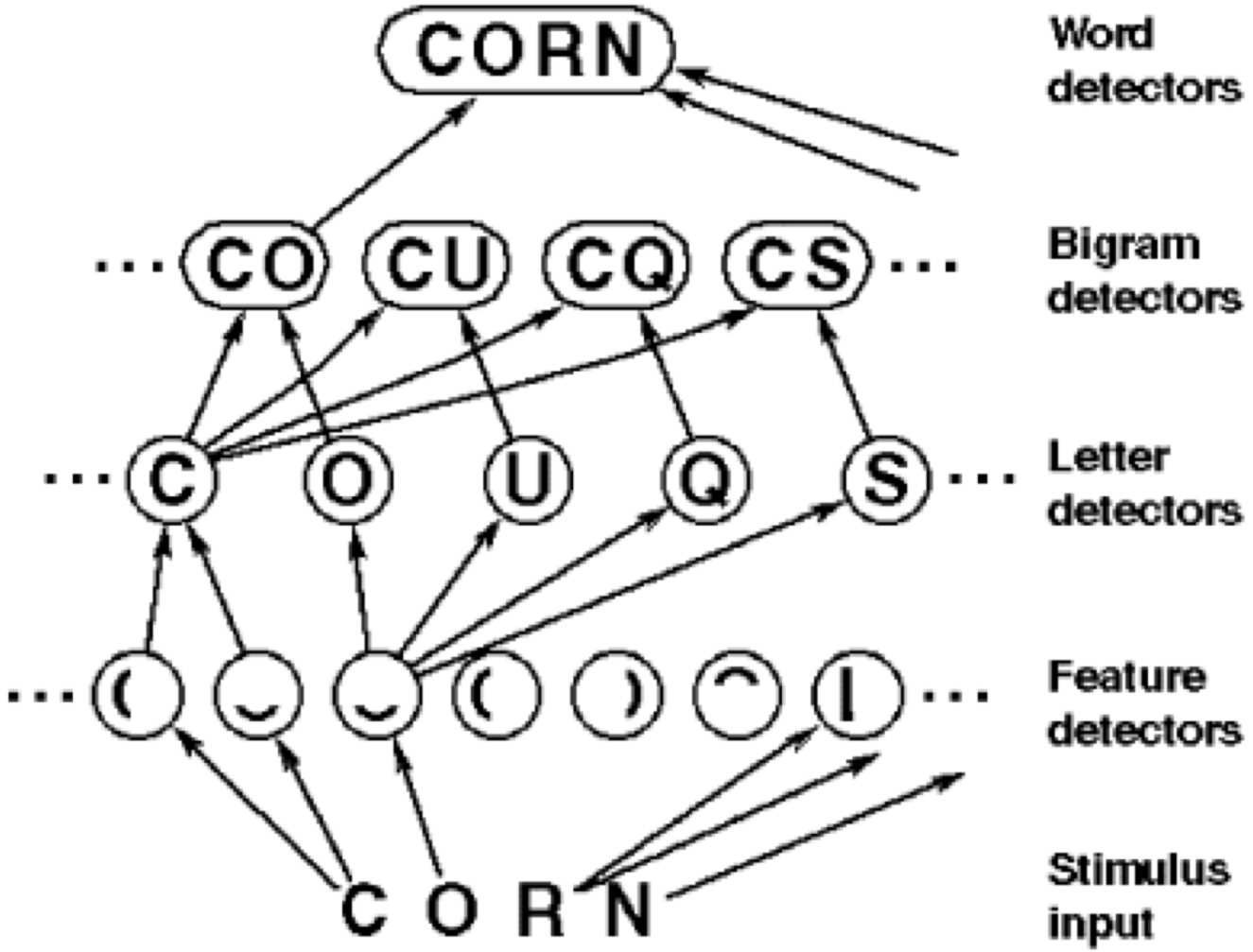
Wiesel and Hubel 特征检测理论



David H. Hubel

Torsten N. Wiesel

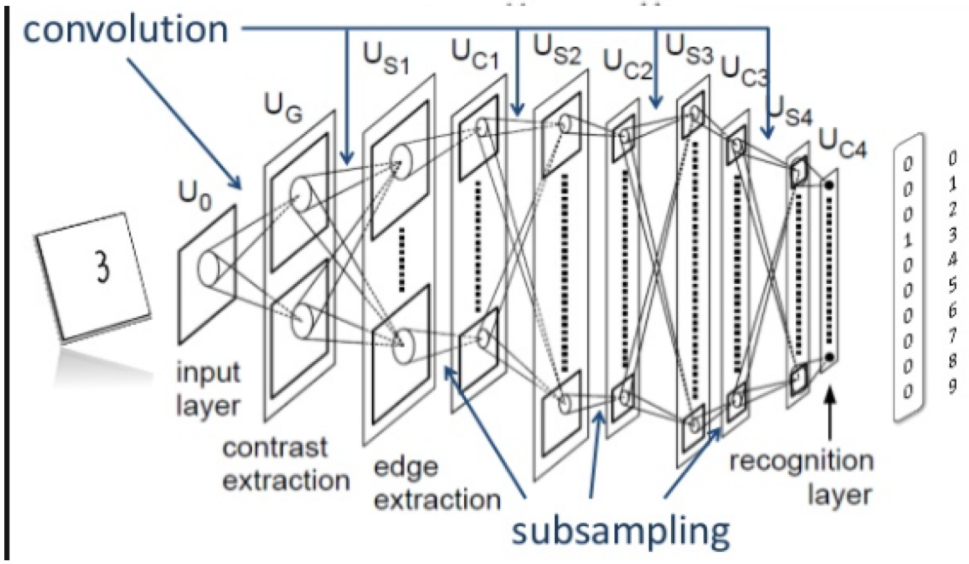

view-tuned cells
"complex composite" cells (C2)
"composite feature" cells (S2)
complex cells (C1)
simple cells (S1)
weighted sum
MAX


CORN
CO
CU
CQ
CS
C
O
U
Q
S
C O R N
Word detectors
Bigram detectors
Letter detectors
Feature detectors
Stimulus input

Neocognitron

神经认知机

Fukushima (1980). Hierarchical multilayered neural network

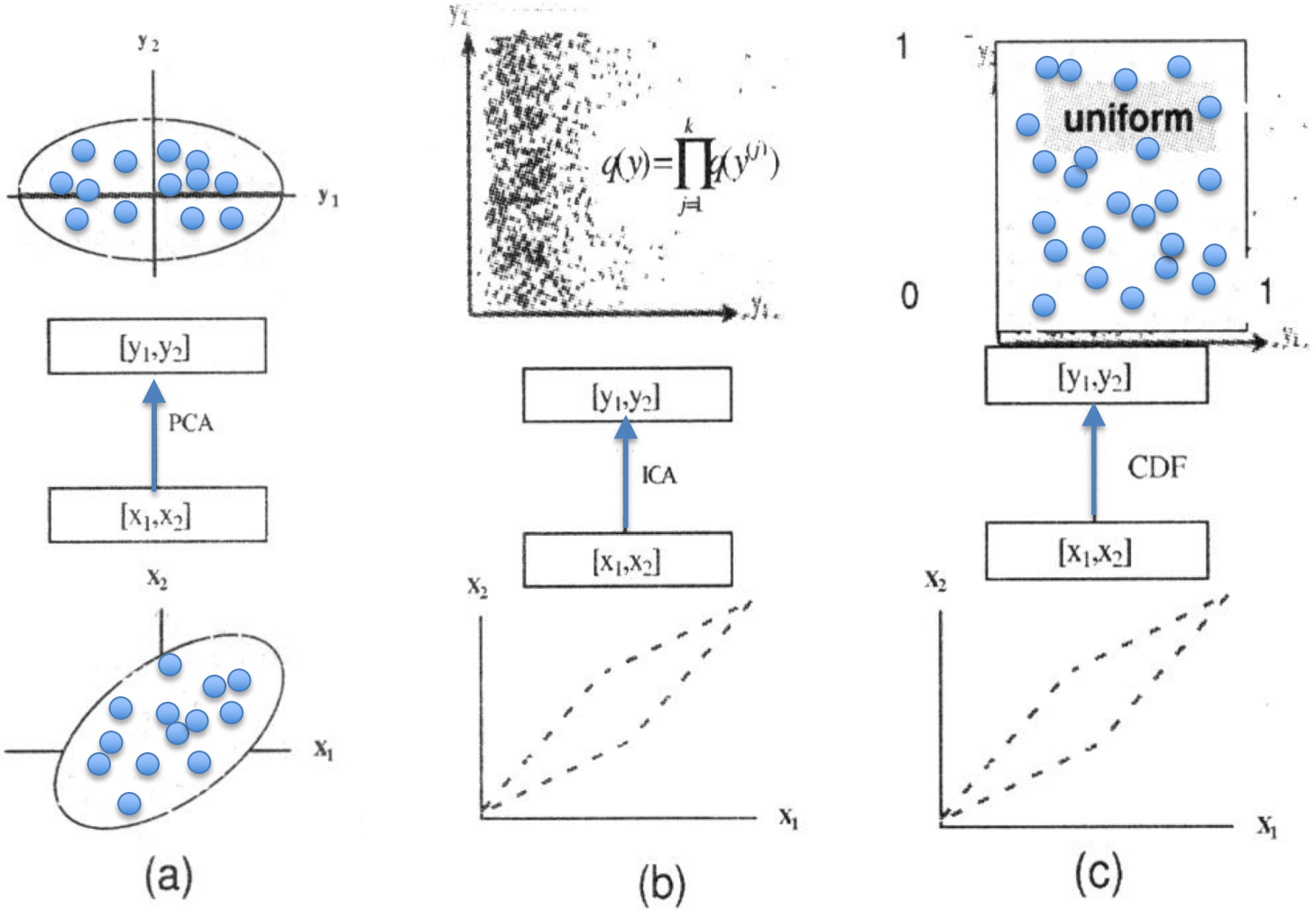


S-cells work as feature-extracting cells. They resemble simple cells of the primary visual cortex in their response.

C-cells, which resembles complex cells in the visual cortex, are inserted in the network to allow for positional errors in the features of the stimulus. The input connections of C-cells, which come from S-cells of the preceding layer, are fixed and invariable. Each C-cell receives excitatory input connections from a group of S-cells that extract the same feature, but from slightly different positions. The C-cell responds if at least one of these S-cells yield an output.

Transformation structures

$$p(y) = \int p(y|x)p(x)\mu(dx) \quad q(y) \Longrightarrow q(y) = \prod_{j=1}^{k} q(y^{(j)}) \quad \text{by} \quad \min \int p(y) \ln \frac{p(y)}{q(y|\theta)} dy$$



最佳抽象（信息之最佳保持）：最大传递、最小互信息、最小冗余

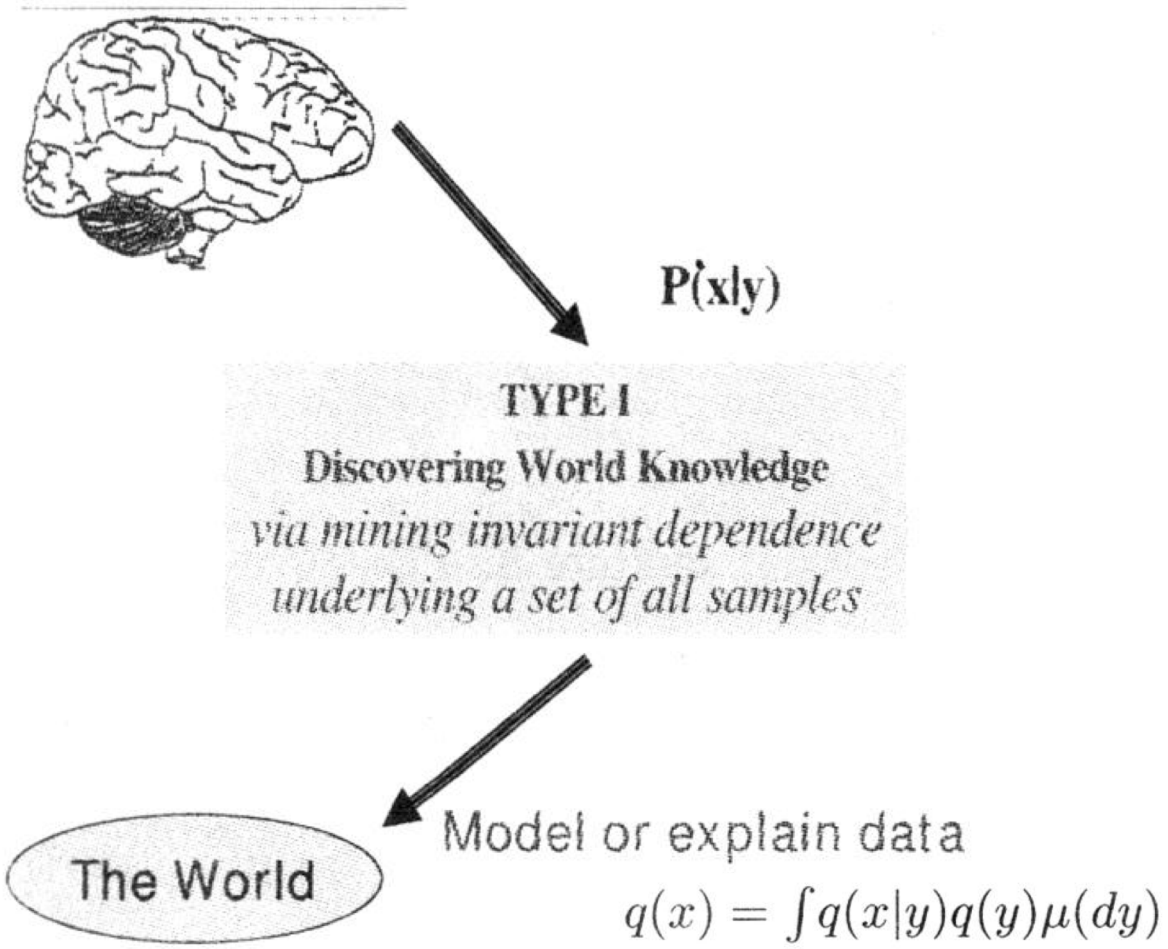
Outline

- **Fundamentals and challenges of learning**
  - Two types of intelligent abilities and learning
  - Three levels of inverse problems

- Bi-directional learning
  - Inbound learning theory
  - **Outbound learning theory**
  - Bi-directional architectures

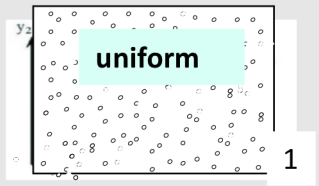
- Early bi-directional deep learning

Outbound architecture

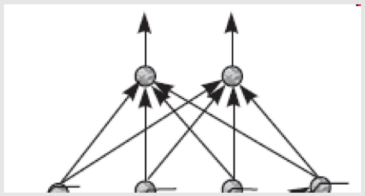


It describes dependence among variables $x^{(1)}, \ldots, x^{(d)}$ by attempting to reconstruct observations of x from certain inner factors y via $q(x|y)$ in an appropriate parametric form.

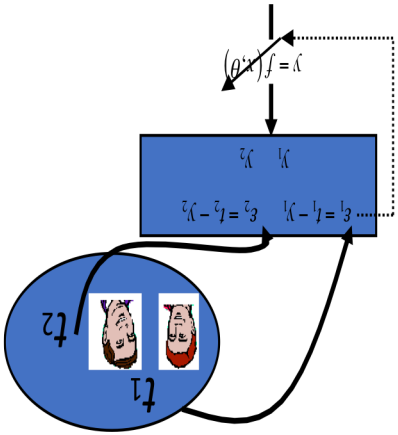
Best reconstruction 最佳重建

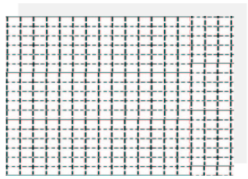


信息之最佳保持



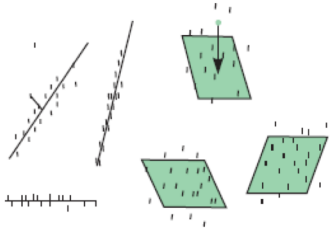
$$q(X \mid \Theta) = \frac{dF(X)}{dX}$$





Modeling Perspective (Best Data Matching)

$G(X)$

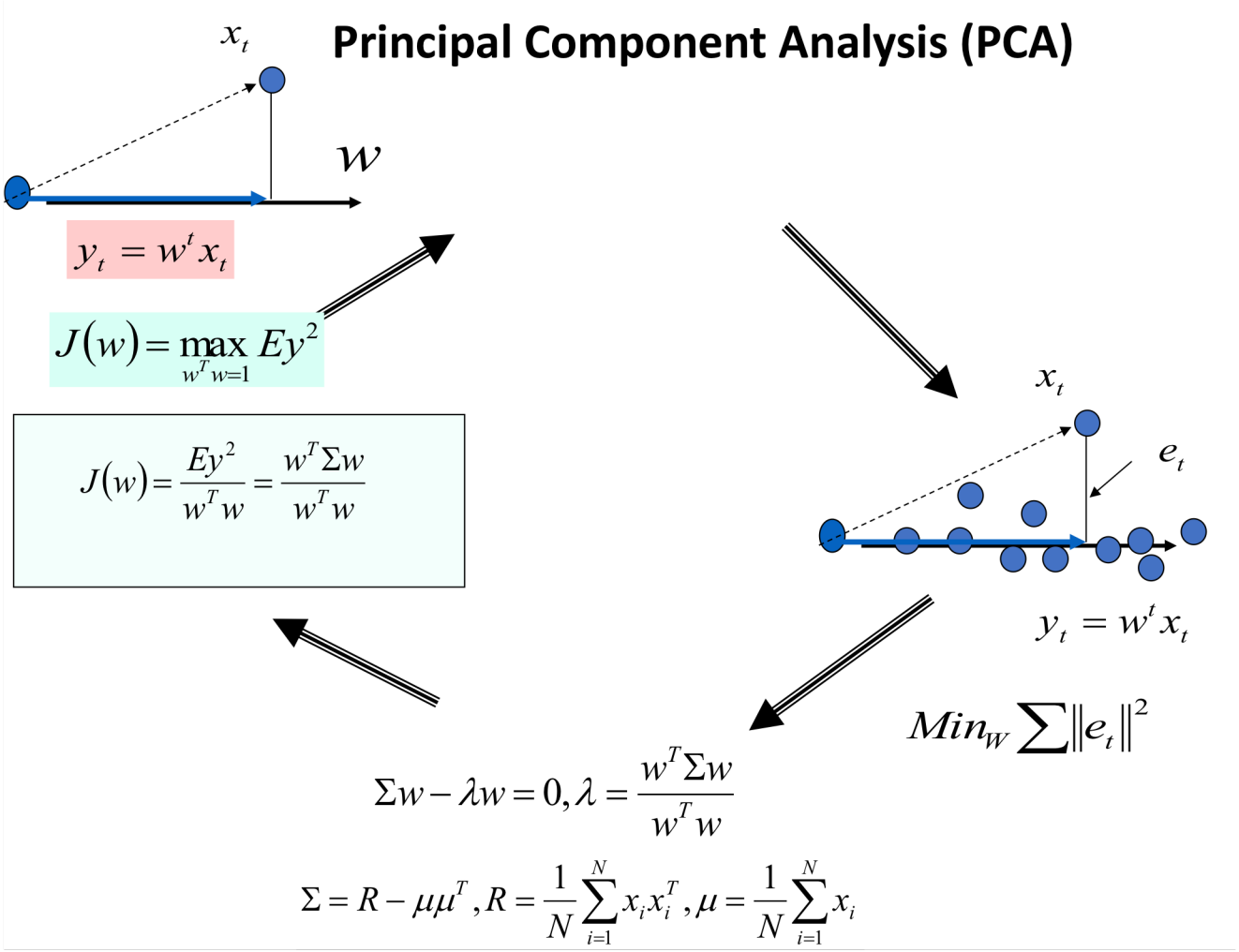


样本之最佳表述

$$q(X_N \mid \Theta)$$

Best Reconstruction （BR）

PCA reconstruction

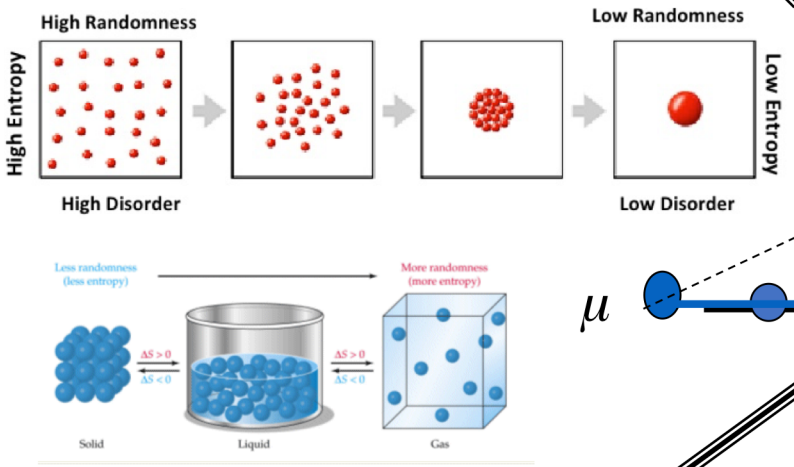


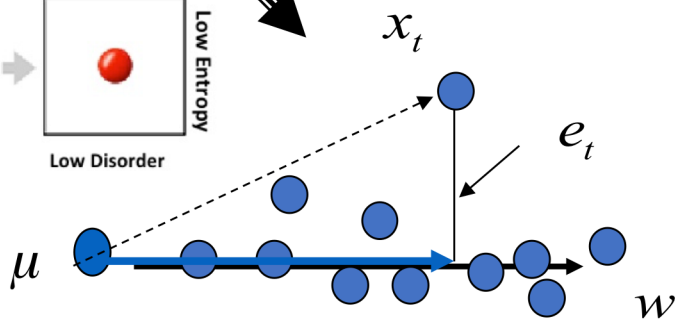
BA theory : maximum versus minimum information transfer

best inverse

$$\max_{w^T w=1} Ey^2 \qquad \min_{w^T w=1} Ey^2$$

$$y_t = w^t (x_t - \mu)$$





Entropy

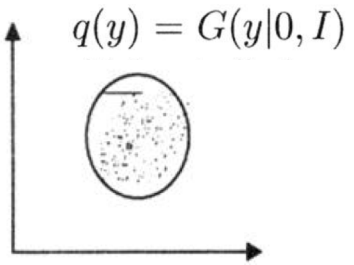
$$-\int p(x)\ln p(x)\,dx$$

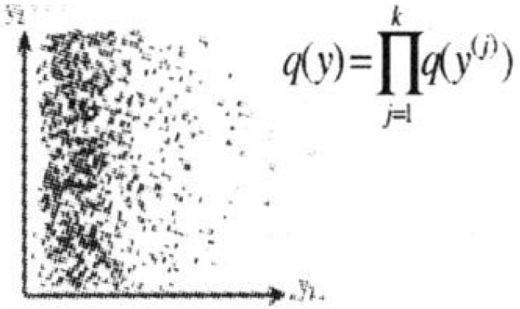
$$-\int G(\mathbf{x}\mid\mu,\sigma^2)\ln G(\mathbf{x}\mid\mu,\sigma^2)\,d\mathbf{x} = 0.5\ln(2\pi e) + 0.5\sigma^2$$

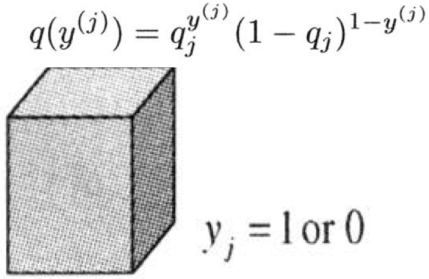
$$Min_W \sum \|e_t\|^2$$

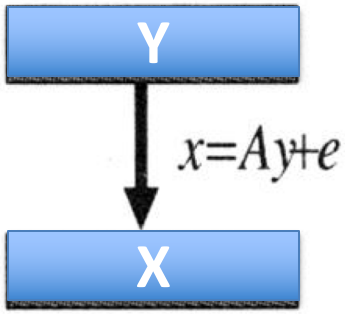
BR theory : minimum error fitting

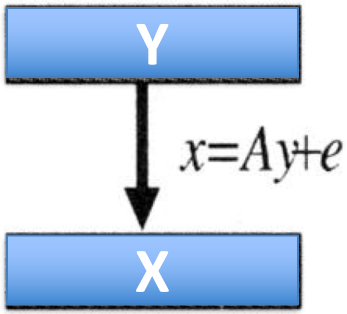
Linear latent structures

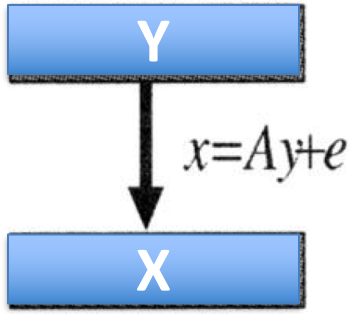


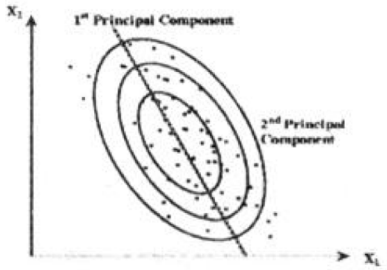


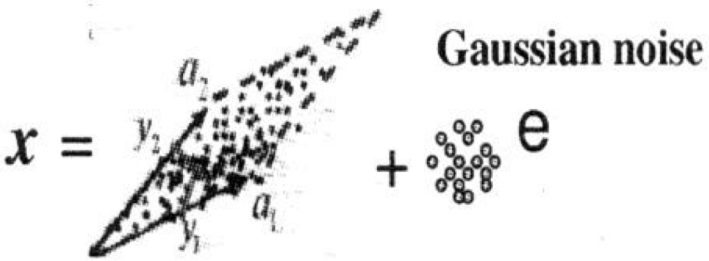


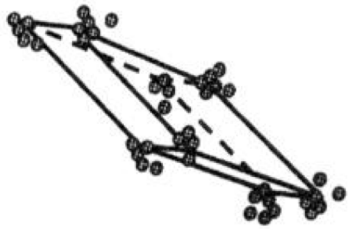




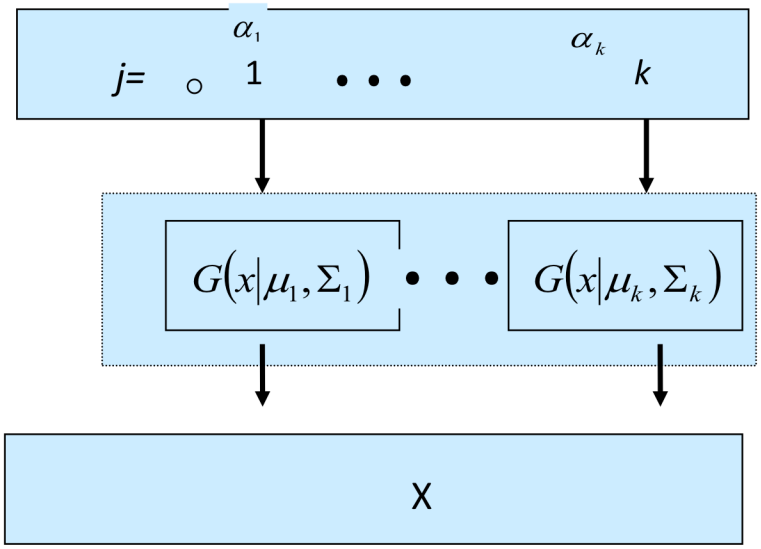




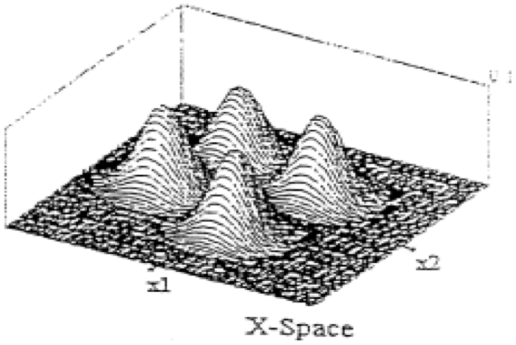


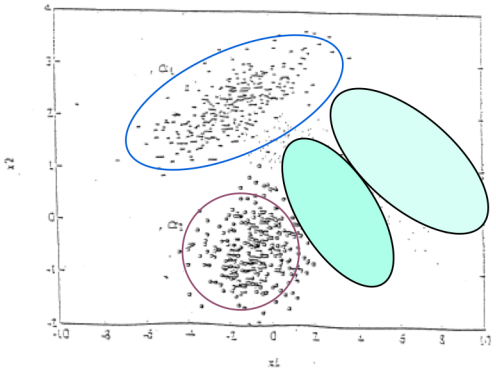


Mixture structures



$$\alpha_j \geq 0, \quad \sum_{j=1}^{K} \alpha_j = 1$$





$$q(x) = \sum_{r=1}^{k} \alpha_r G(x|\mu_r, \Sigma_r)$$

Maximum Likelihood

$$Max \sum_{t=1}^{N} \ln q(x_t)$$

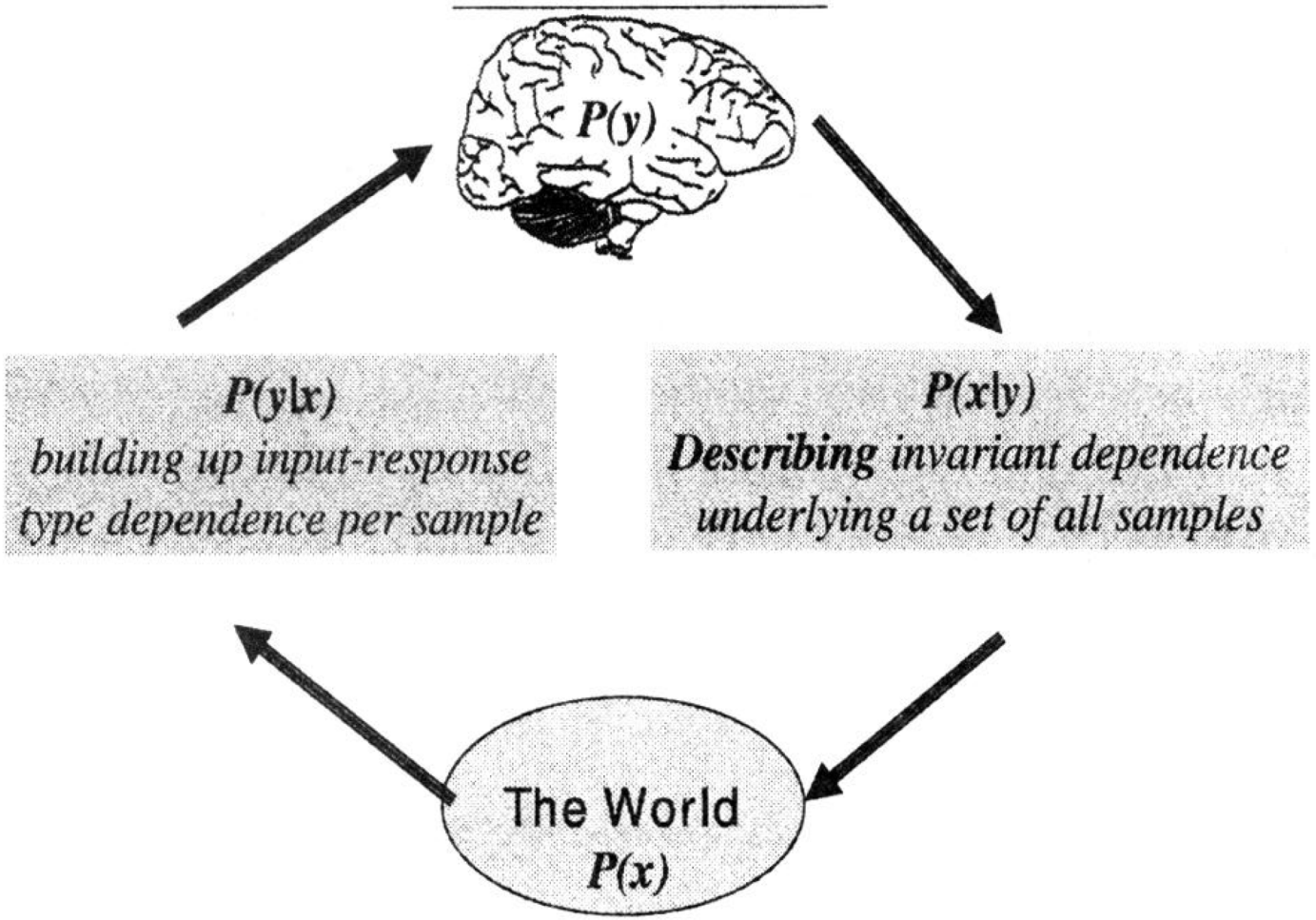
Outline

- **Fundamentals and challenges of learning**
  - Two types of intelligent abilities and learning
  - Three levels of inverse problems

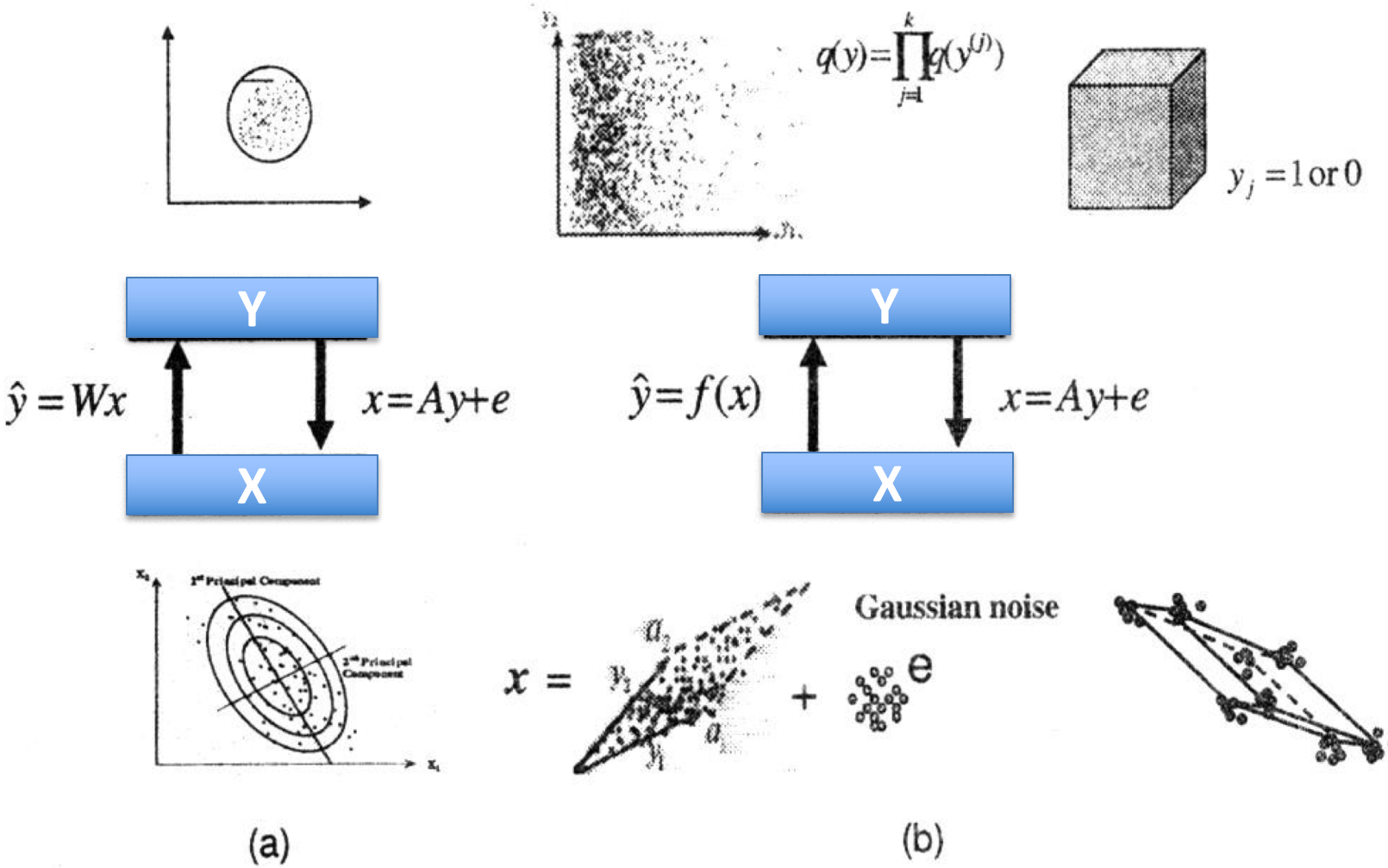
- Bi-directional learning
  - Inbound learning theory
  - Outbound learning theory
  - **Bi-directional architectures**

- Early bi-directional deep learning

Bi-directional architectures



Bi-directional latent structures



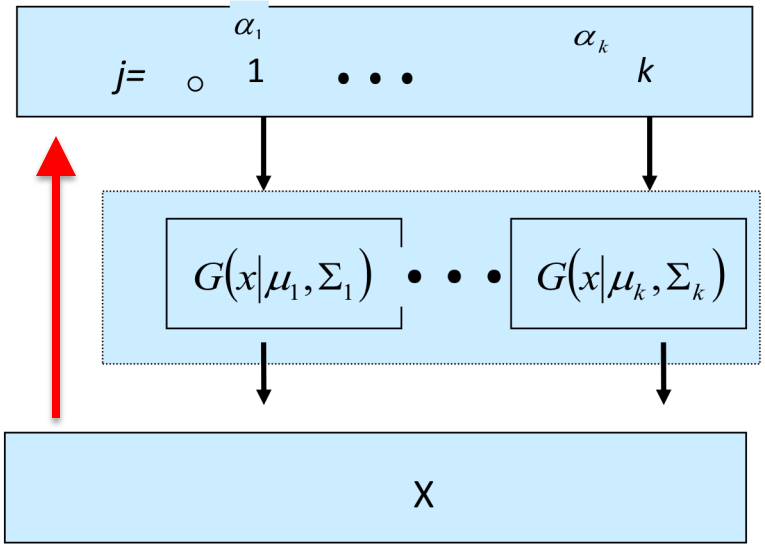
GMM in bi-directional architecture

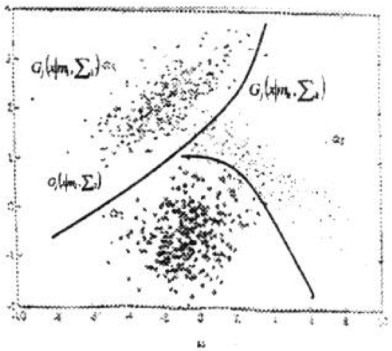
$$\alpha_j \geq 0, \quad \sum_{j=1}^{K} \alpha_j = 1$$

Bayesian Inversion

$$p(j \mid x_t) = \frac{\alpha_j G\left(x_t \middle| m_j, \Sigma_j\right)}{\sum_r \alpha_r G\left(x_t \middle| m_r, \Sigma_r\right)}$$

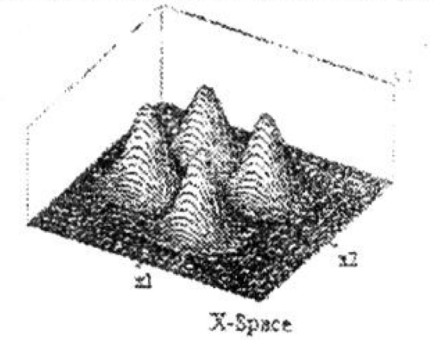
$$p_{\ell,t} = \begin{cases} p(\ell \mid x_t), & \text{no label} \\ \ell(x_t), & \text{labeled} \end{cases}$$





$$p(x) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - x_i)$$

$$q(x) = \sum_{r=1}^{K} \alpha_r G(x|\mu_r, \Sigma_r)$$



Manifold factor analysis

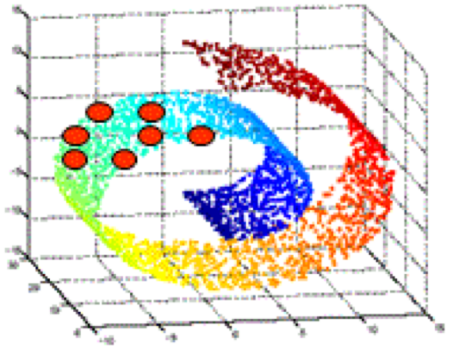
$$q(Y \mid \Lambda) = \frac{\exp\{-0.5Tr[\Lambda^{-1}YLY^{T}]\}}{Z(L,\Lambda)}$$

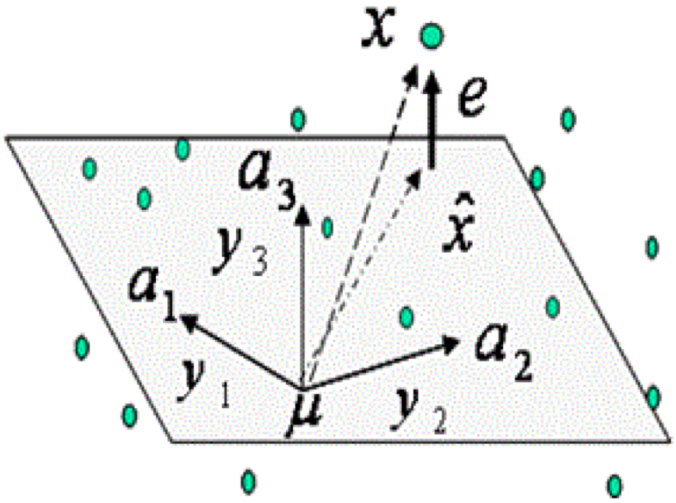
$q(Y \mid \theta_y) = q(Y \mid \Lambda)$

$Y = WX$

$X = AY + E$

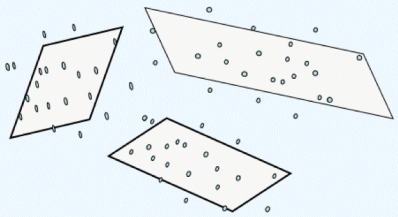
$\boldsymbol{X}$





Given Laplacian $\mathbf{L}$ of the nearest neighbor graph $\mathbf{G}$ of X

Local factor analysis (LFA)

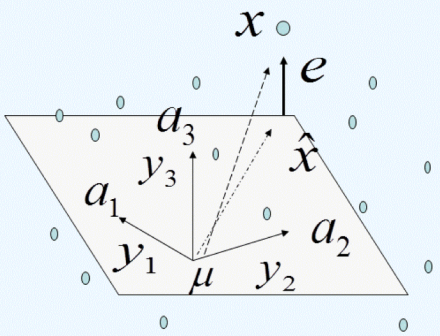


$$x = A_\ell y + \mu_\ell + e,$$

$$E[ey^T] = 0,$$

e from $G(e|0,\Sigma_\ell)$,

$$\ell = 1 \cdots k.$$



y from $G(y|0,\Lambda_\ell)$

(a)

$$q(\theta_\ell) = q(\alpha)q(A_\ell)q(\mu_\ell)q(\Lambda_\ell)q(\Sigma_\ell)$$

(1) Dirichlet distribution $q(\alpha) \propto \prod_\ell \alpha_\ell^{\gamma_\ell}$, $\gamma_\ell \geq -1$,
particularly, it becomes Jeffrey priori if $\gamma_\ell = -0.5$.

(2) Jeffreys prior

$$q(\Lambda_\ell) \propto |\Lambda_\ell|^{-0.5}, \quad q(\Sigma_\ell) \propto |\Sigma_\ell|^{-0.5}.$$

(3) Gaussian - Jeffreys joint prior

For $A_\ell = [a_\ell^{(ij)}]$,

$$q(A_\ell, \{\eta_{\ell,ij}^A\}) \propto \prod_{ij} \{G(a_\ell^{(ij)}|0, \eta_{\ell,ij}^A) / \sqrt{\eta_{\ell,ij}^A}\}.$$

For $\mu_\ell = [\mu_\ell^{(1)}, \cdots, \mu_\ell^{(d)}]^T$,

$$q(\mu_\ell, \{\eta_{\ell,i}^\mu\}) \propto \prod_i \{G(\mu_\ell^{(i)}|0, \eta_{\ell,i}^\mu) / \sqrt{\eta_{\ell,i}^\mu}\}.$$

(b)

LFA/MFA = FA + GMM

Learning LFA

$$p(y,\ell \mid x) = p(y \mid x,\ell)p(\ell \mid x)$$

$$p(y \mid x_t,\ell) = G(y \mid W_\ell(x_t-\mu_\ell), \Pi_\ell^{y\,-1}+\rho_\ell)$$

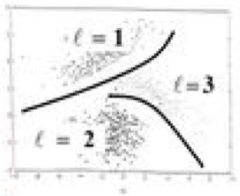
$$y_{t,\ell} = \operatorname{argmin}_y \varepsilon(y,x_t,\theta_\ell) = \Pi_\ell^{y\,-1}A_\ell^T\Sigma_\ell^{-1}(x_t-\mu_\ell),$$

$$q(\ell \mid x,\theta_\ell) = \frac{\exp[-\varepsilon_t(\theta_\ell) - \tfrac{1}{2}\ln\{|\Pi_\ell^y|/(2\pi)^{m_\ell}\}]}{\sum_j \exp[-\varepsilon_t(\theta_j) - \tfrac{1}{2}\ln\{|\Pi_j^y|/(2\pi)^{m_j}\}]}$$

$$p(\ell \mid x_t,\theta_\ell) = \begin{cases} q(\ell \mid x_t,\theta_\ell), & \ell \in C_\kappa(x_t), \\ 0, & \ell \notin C_\kappa(x_t). \end{cases}$$

$C_\kappa(x_t) = \{\ell$: for the first κ largest values of $L(x_t,\theta_\ell)$ by Eq.(6)$\}$.

$p_{\ell,t} = p(\ell \mid x_t,\theta_\ell^{old})[1+\Delta_{\ell,t}(\theta_\ell^{old})]$ by Eq.(7).



$$q(y,\ell) = q(y \mid \ell)q(\ell)$$

$$q(\ell) = \alpha_\ell = e^{c_\ell}/\sum_j e^{c_\ell}$$

$$q(y \mid \ell) = G(y \mid 0, \Lambda_\ell)$$

$$q(x \mid y,\ell) = G(x \mid A_\ell y + \mu_\ell, \Sigma_\ell)$$

$$\varepsilon(y,x_t,\theta_\ell) = -\ln[\alpha_\ell G(y \mid 0,\Lambda_\ell)G(x \mid A_\ell y+\mu_\ell,\Sigma_\ell)],$$

$$\Pi_\ell^y = A_\ell^T\Sigma_\ell^{-1}A_\ell + \Lambda_\ell^{-1},$$

$$\varepsilon_t(\theta_\ell) = \varepsilon(y_{t,\ell},x_t,\theta_\ell),$$

$$\delta_{t,\ell} = y_{t,\ell} - W_\ell(x_t-\mu_\ell).$$

$$\theta_\ell = \{A_\ell,\mu_\ell,\Lambda_\ell,\Sigma_\ell,W_\ell,c_\ell\},$$

$$\theta_\ell^{new} - \theta_\ell^{old} \propto \sum_{t\in T_t} G_t(\theta_\ell^{old}).$$

$$G_t(\theta_\ell) = p_{\ell,t}\nabla_{\theta_\ell}\varepsilon_t(\theta_\ell) + p(\ell \mid x_t,\theta_\ell^{old})\nabla_{\theta_\ell} r_t(\theta_\ell),$$

$$r_t(\theta_\ell) = \tfrac{1}{2}\delta_{t,\ell}^T\Pi_\ell^{y\,-1}\delta_{t,\ell} + \tfrac{1}{2}h^2\mathrm{Tr}[\Sigma_\ell^{-1}] - \ln q(\theta_\ell).$$

Outline

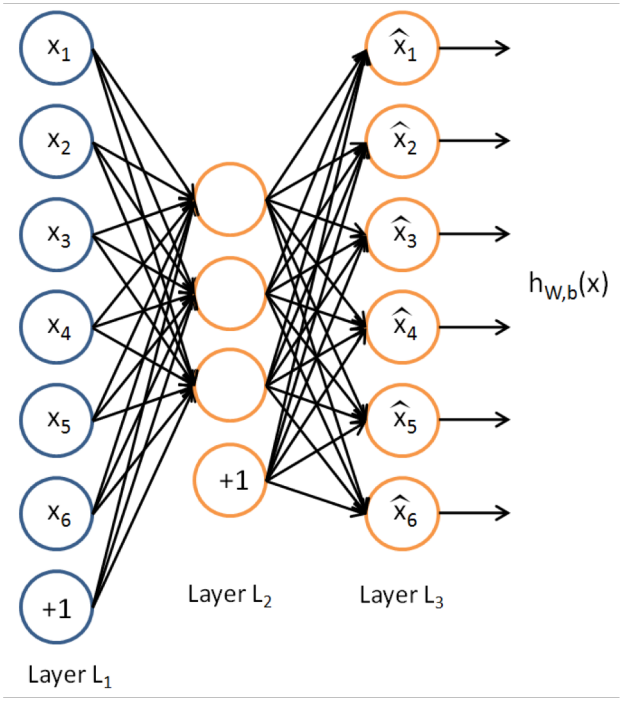
- **Fundamentals and challenges of learning**
  - Two types of intelligent abilities and learning
  - Three levels of inverse problems

- Bi-directional learning
  - Inbound learning theory
  - Outbound learning theory
  - Bi-directional architectures

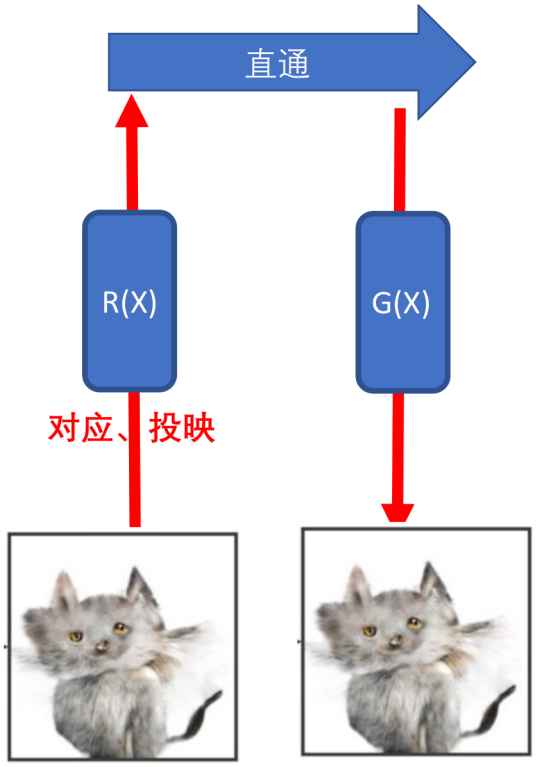
- **Early bi-directional deep learning**

Autoencoder (AE)

- An **autoencoder** neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs.



The autoencoder tries to learn a function $h_{W,b}(x) \approx x$. In other words, it is trying to learn an approximation to the identity function, so as to output $\hat{x}$ that is similar to x. The identity function seems a particularly trivial function to be trying to learn; but by placing constraints on the network, such as by limiting the number of hidden units, we can discover interesting structure about the data



General architecture

Sparse autoencoder

Imposing sparsity on hidden units

Informally, we will think of a neuron as being "active" (or as "firing") if its output value is close to 1, or as being "inactive" if its output value is close to 0. We would like to constrain the neurons to be inactive most of the time.[3]

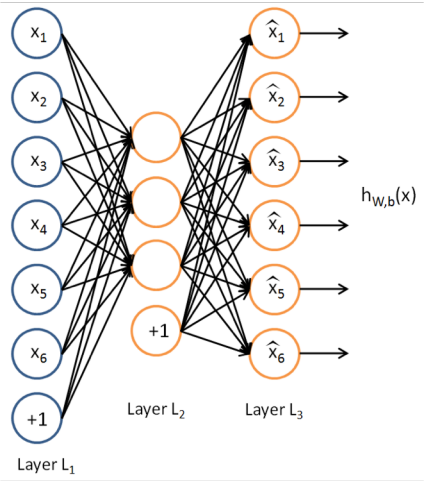
Recall that $a_j^{(2)}$ denotes the activation of hidden unit j in the autoencoder. However, this notation doesn't make explicit what was the input x that led to that activation. Thus, we will write $a_j^{(2)}(x)$ to denote the activation of this hidden unit when the network is given a specific input x. Further, let

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

be the average activation of hidden unit j (averaged over the training set). We would like to (approximately) enforce the constraint

$$\hat{\rho}_j = \rho,$$

where ρ is a **sparsity parameter**, typically a small value close to zero (say $\rho = 0.05$). In other words, we would like the average activation of each hidden neuron j to be close to 0.05 (say). To satisfy this constraint, the hidden unit's activations must mostly be near 0.



From Andrew Ng

Penalize the objective with sparsity constraints

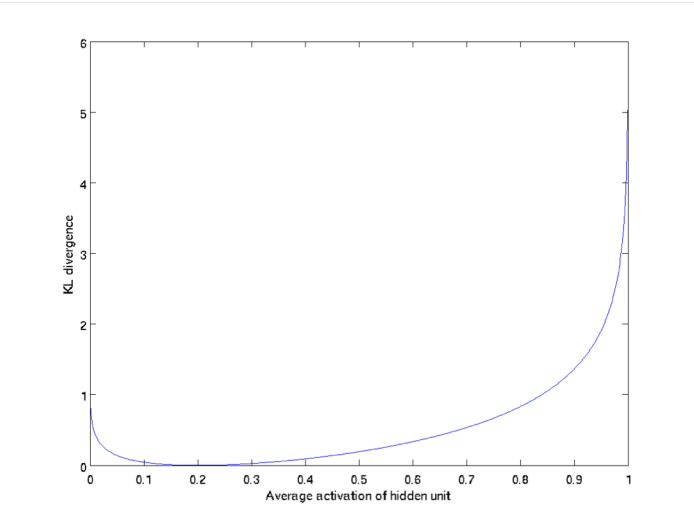
Using KL divergence to represent the sparsity constraint:

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

Average activation

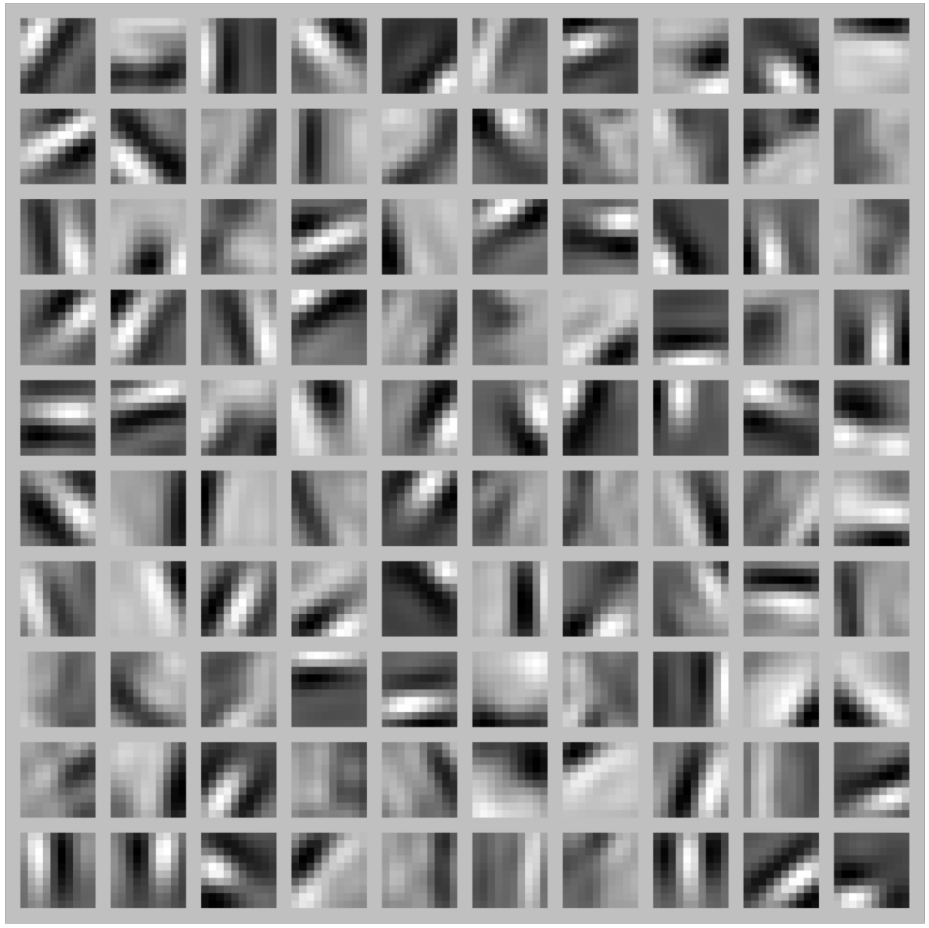
$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} \left[a_j^{(2)}(x^{(i)}) \right]$$

This penalty function has the property that $\mathrm{KL}(\rho||\hat{\rho}_j) = 0$ if $\hat{\rho}_j = \rho$, and otherwise it increases monotonically as $\hat{\rho}_j$ diverges from ρ. For example, in the figure below, we have set $\rho = 0.2$, and plotted $\mathrm{KL}(\rho||\hat{\rho}_j)$ for a range of values of $\hat{\rho}_j$:



Visualize the weights: input->hidden

Input is image

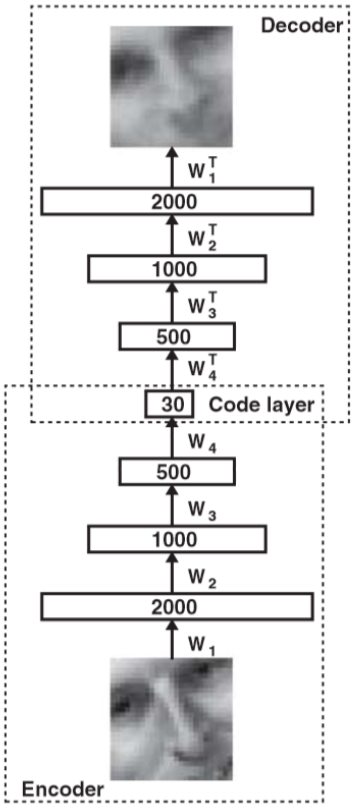


$$x_j = \frac{W_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{100}(W_{ij}^{(1)})^2}}.$$

Each square in the figure above shows the (norm bounded) input image x that maximally actives one of 100 hidden units. We see that the different hidden units have learned to detect edges at different positions and orientations in the image.

早期双向深度学习：auto-encoder

Stacked RBMs
Hinton, et al (2006)

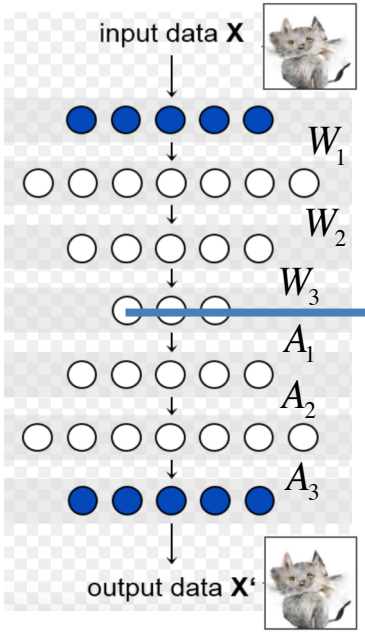


强制了参数对称性

$W_j = A_{ji}^T$

Early auto-associative networks

$W_j \neq A_{ji}^T$

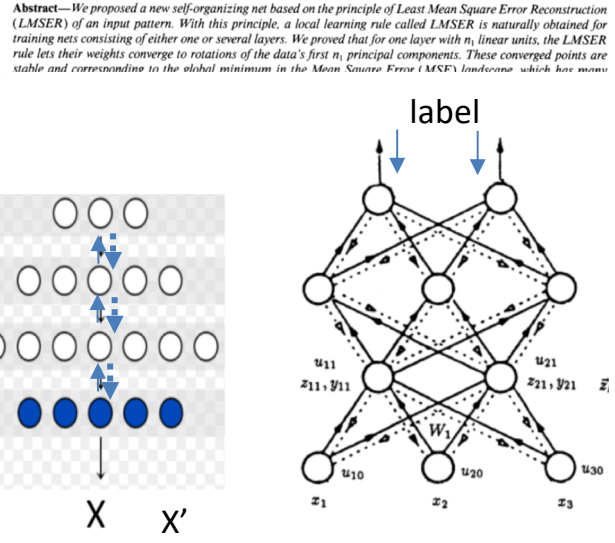


Bourlard, H., Kamp, Y.
Auto-association by multilayer perceptrons … .
Biological cybernetics 59(4-5), 291-294 (1988)

Xu, *Proc. IJCNN 91,* Singapore, pp. 2362-2373
Neural Networks, vol. 6, pp. 627-648, 1993

Least Mean Square Error Reconstruction Principle for Self-Organizing Neural-Nets

多层LMSER自组织学习



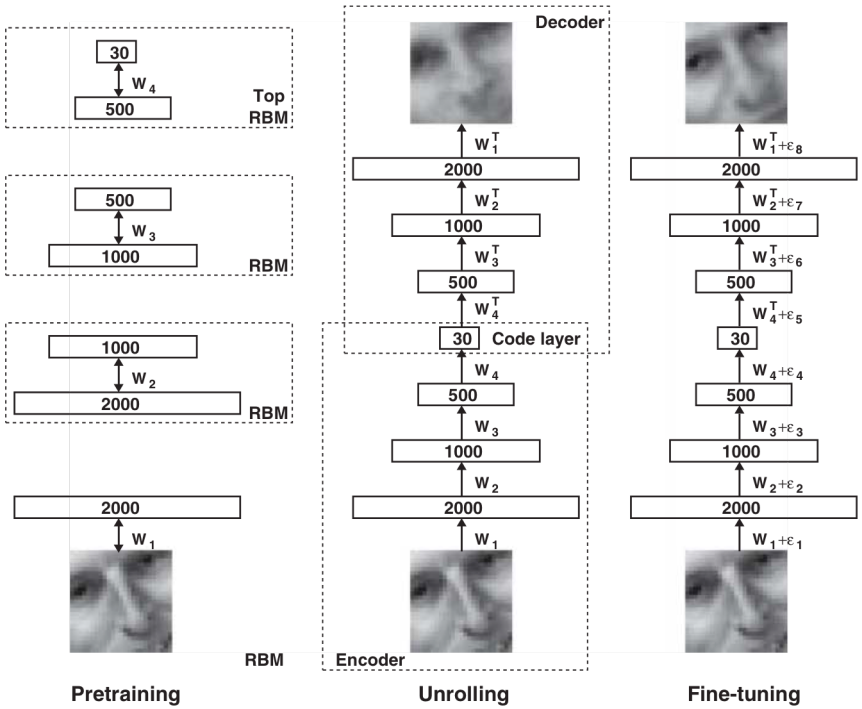
强制了上下两半部的参数对称性
和神经元强度的对称性 $W_j = A_{ji}^T$

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

28 JULY 2006 VOL 313 **SCIENCE** www.sciencemag.org



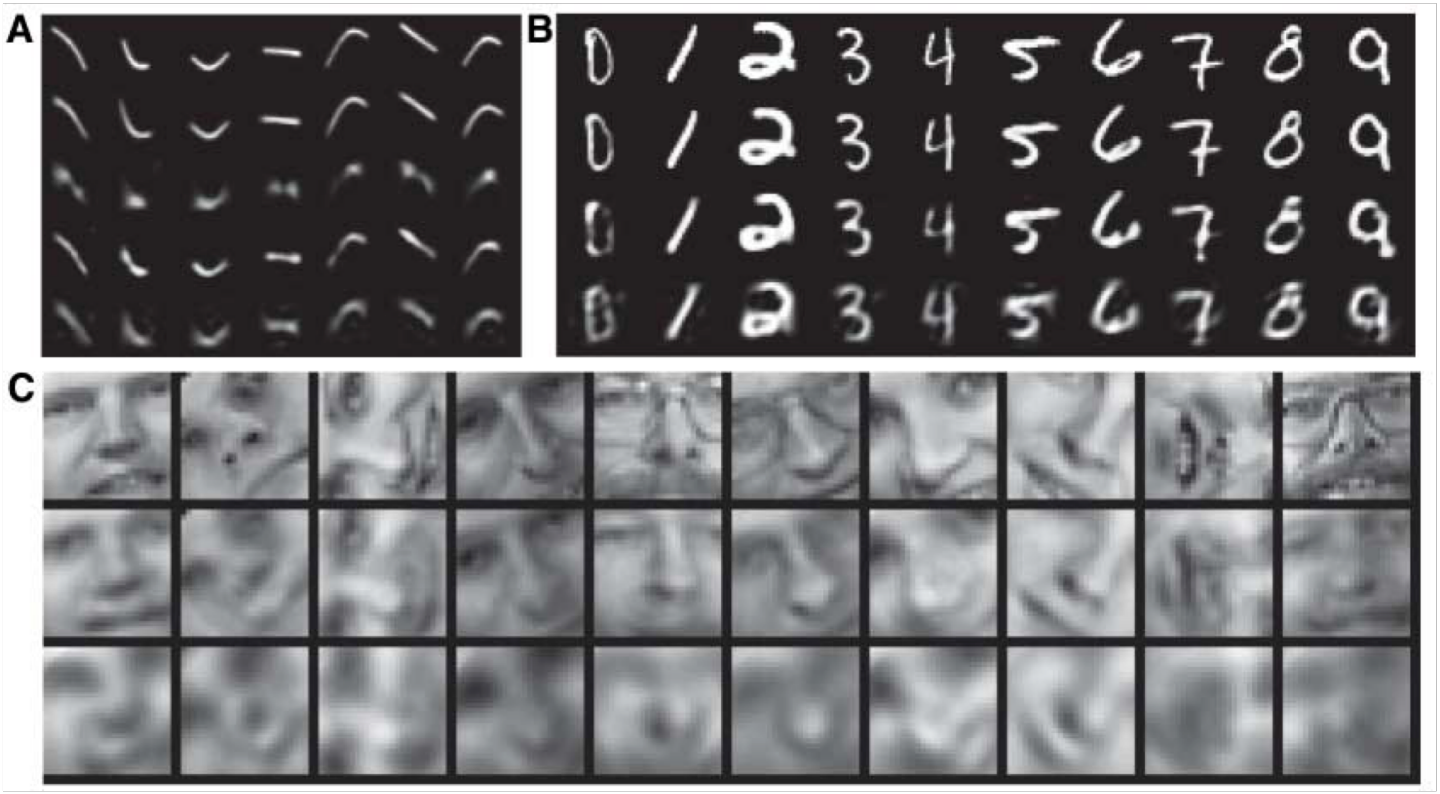
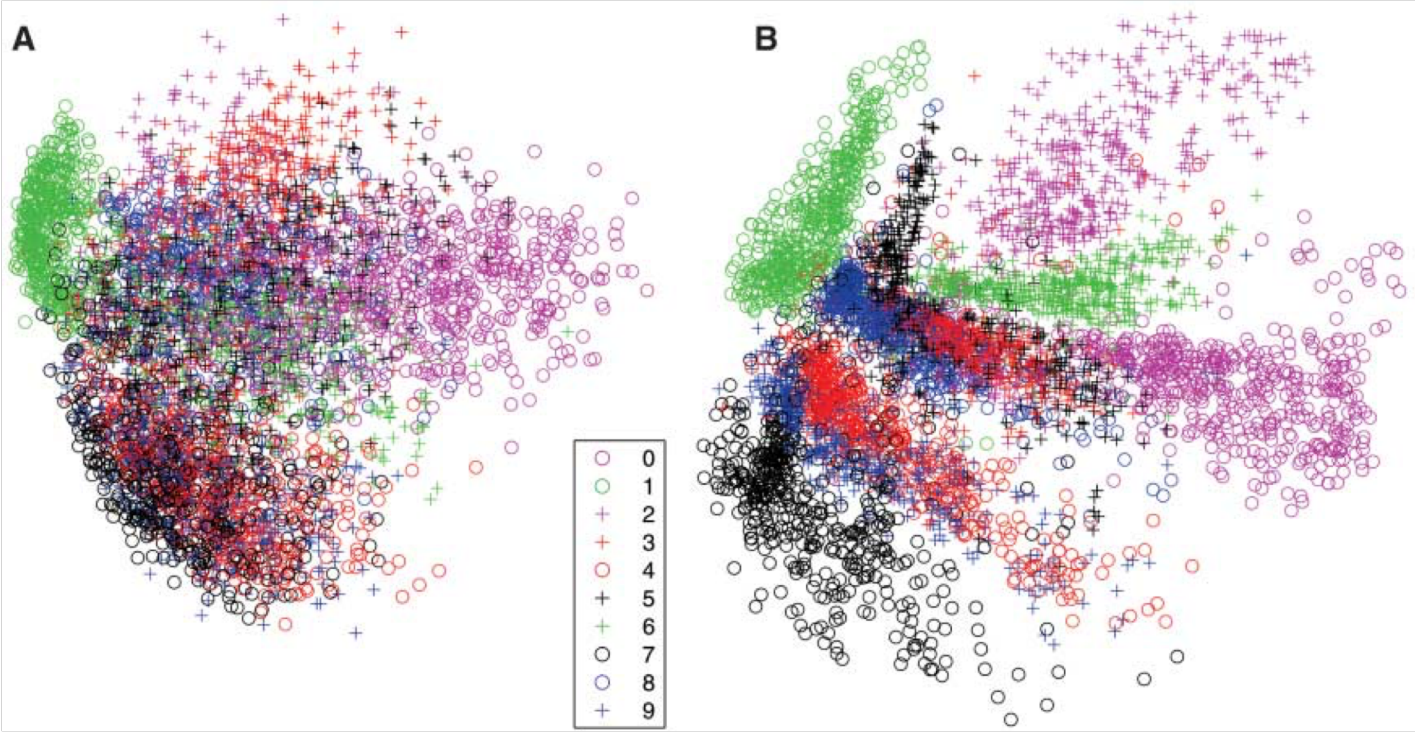


Fig. 2. (**A**) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by "logistic PCA" (*8*) using six components; reconstructions by logistic PCA and standard PCA using 18 components. The average squared error 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.

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*).



Thank you!