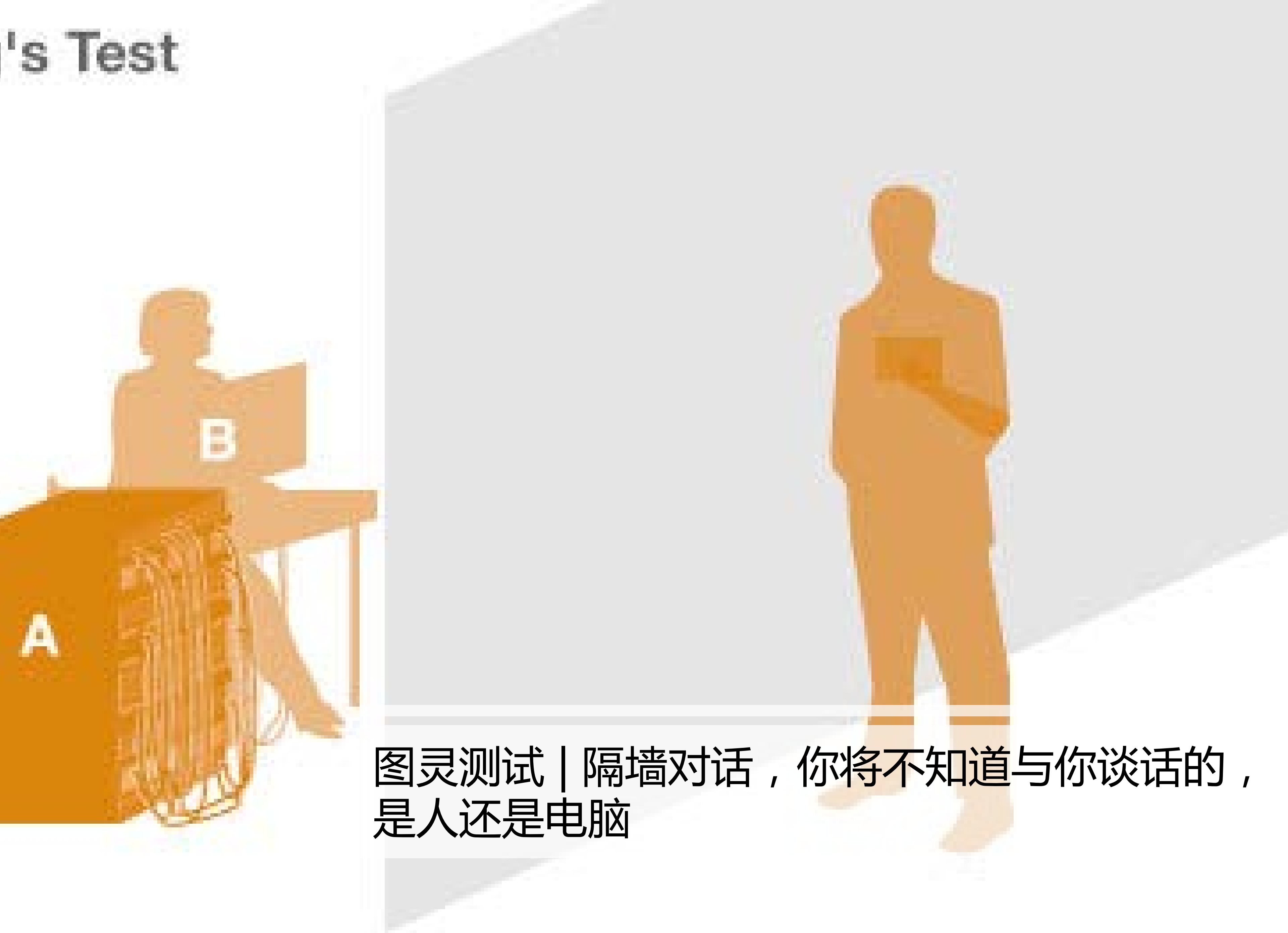
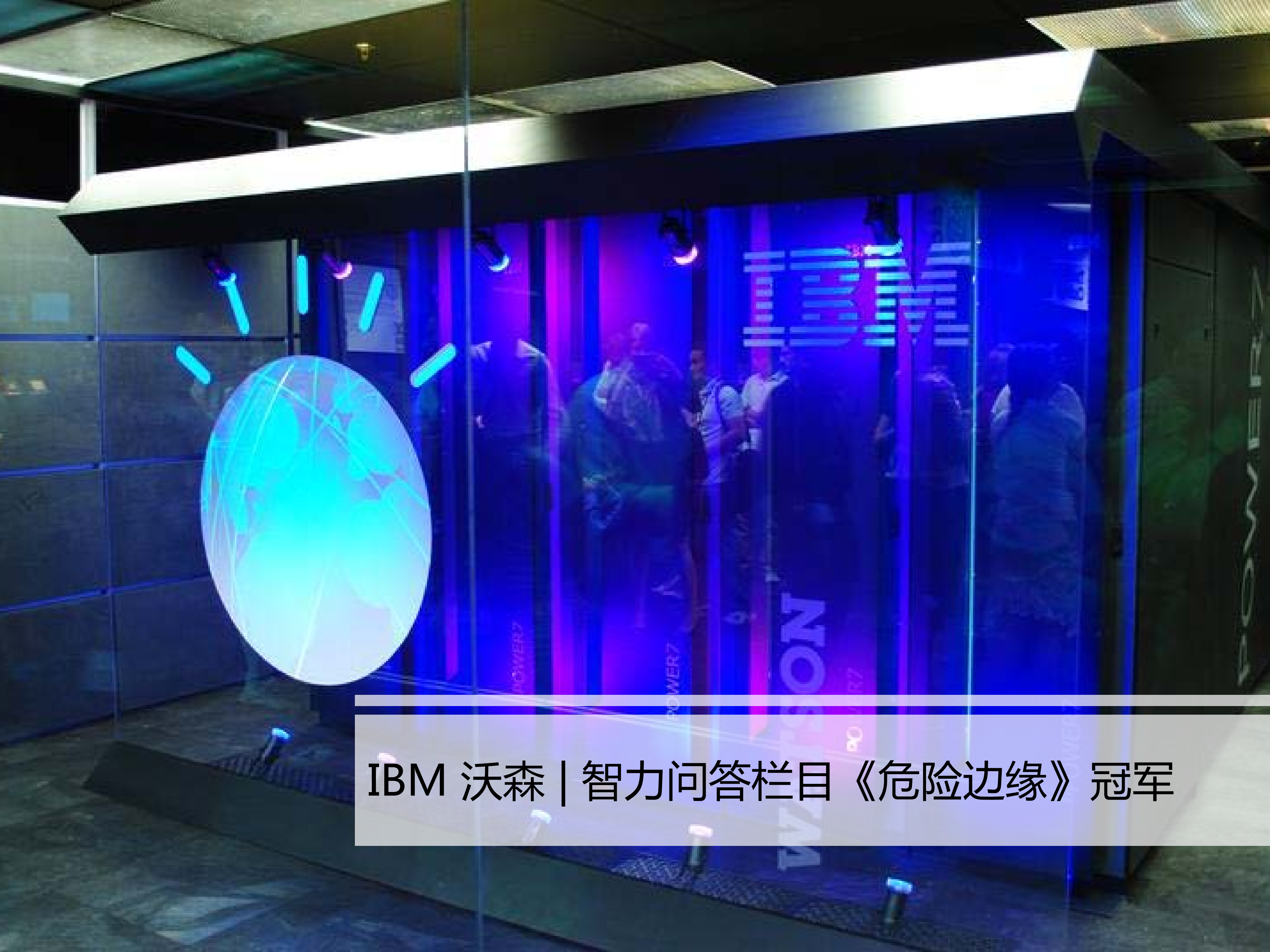
Deep learning and ELM

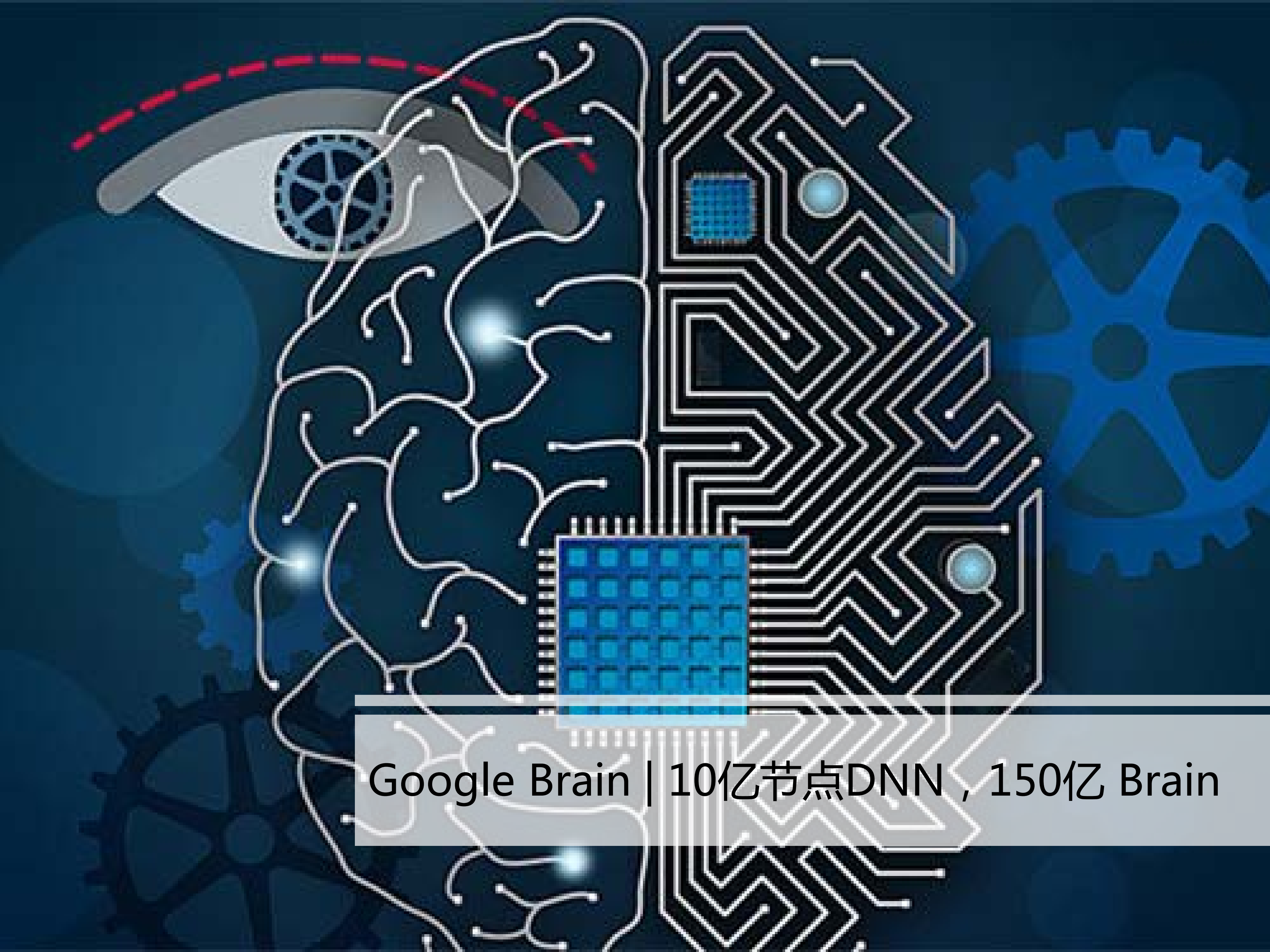
大头雨山



图灵测试 | 隔墙对话，你将不知道与你谈话的，是人还是电脑

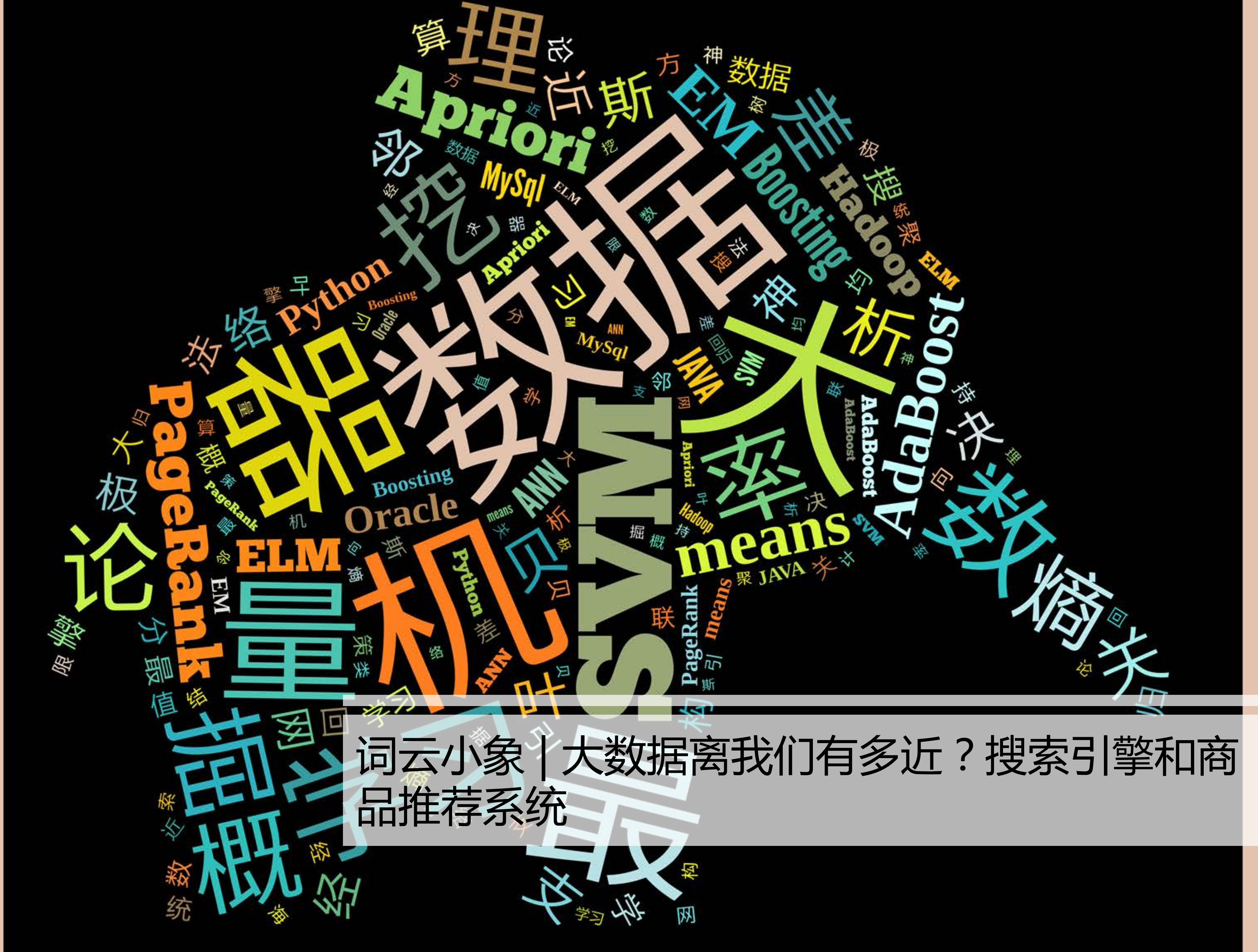


IBM 沃森 | 智力问答栏目《危险边缘》冠军

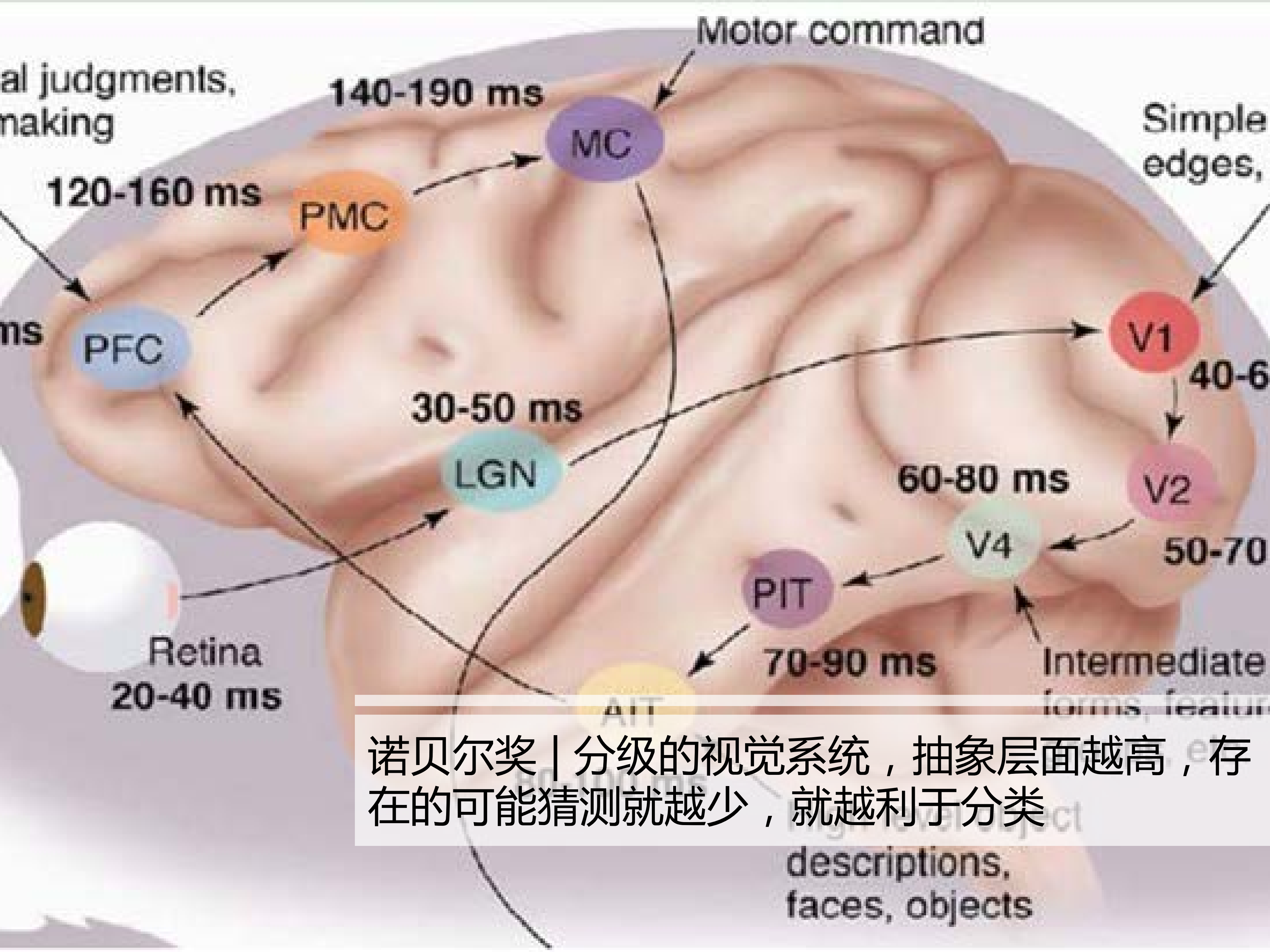

Google Brain | 10亿节点DNN，150亿 Brain

同声传译 | 语音识别、英中机器翻译，中文语音合成

Baidu IDL | 百度深度学习研究院
商品图像检索 | http://shitu.baidu.com/app



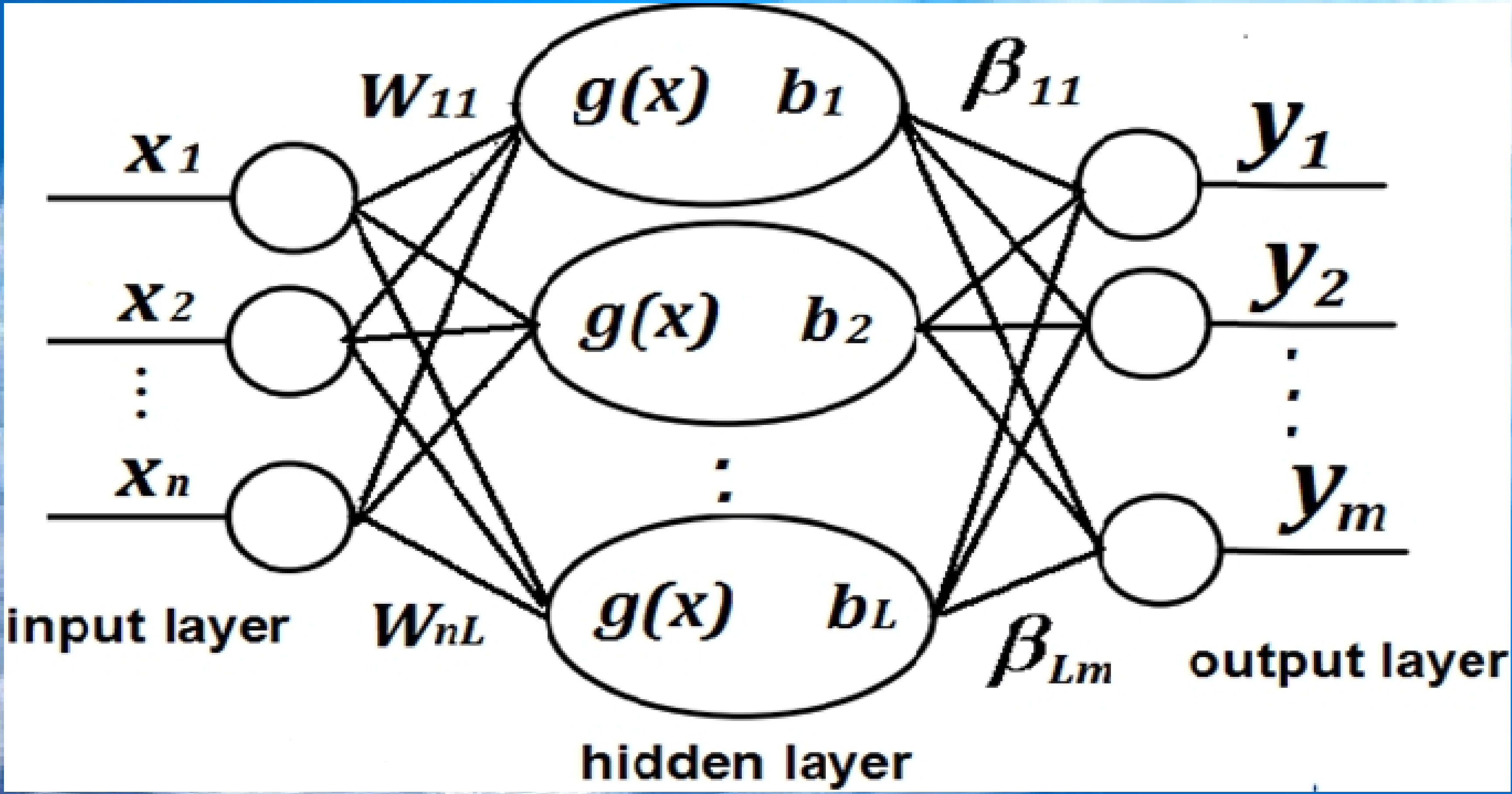
词云小象 | 大数据离我们有多近？搜索引擎和商品推荐系统



诺贝尔奖 | 分级的视觉系统，抽象层面越高，存在的可能猜测就越少，就越利于分类

深度学习 |多层次学习；原始数据不同抽象层度的表示，提高分类和预测的准确性

Artificial Neural Network



BP

$$E = \frac{1}{2}(d_i - y_i)^2$$

d_i ***desired output***

y_i ***NN output***

$$E = \frac{1}{2}[d_i - f(W_i^T X)]^2$$

$$\frac{\partial E}{\partial w_{ij}} = -(d_i - y_i) f'(W_i^T X) x_j$$

j=1,2,…,n

$$\Delta W_i = -\eta \frac{\partial E}{\partial w_{ij}}$$

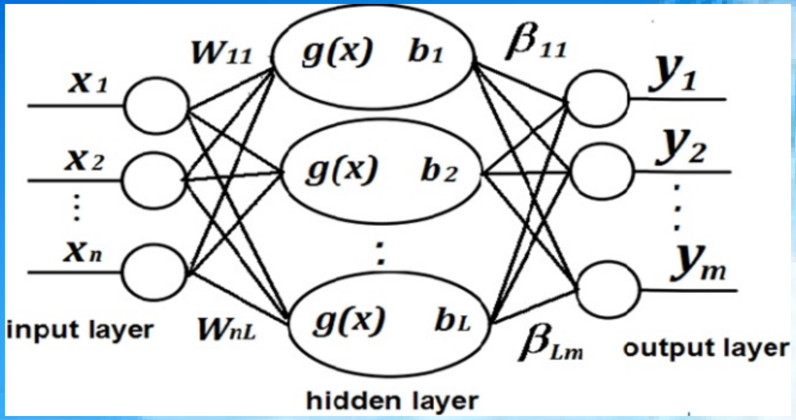
$$\Delta W_i = \eta(d_i - y_i) f'(net_i) X$$

$$\Delta w_{ij} = \eta(d_i - y_i) f'(net_i) x_j$$

BP

- slow gradient-based learning algorithms
- all the parameters of the networks are tuned iteratively
- local minima, improper learning rate and overfitting
- only work for differentiable activation functions

ELM



For Training Set: $(x_j, y_j), j = 1, 2, 3 \cdots N$

activation functions: $g(x) = \text{sigmoid}(x) = \dfrac{1}{1+e^{-x}}$

ELM Model is as follows

$$\sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i g_i(w_i \bullet x_j + b_i) = y_j, \; j = 1, 2, 3, \cdots N$$

input layer weight: w_i output layer weight: β_i

Number of the hidden layer nodes: L

$$\sum_{i=1}^{L}\beta_i g_i(x_j)=\sum_{i=1}^{L}\beta_i g_i(w_i \bullet x_j + b_i) = y_j,\ j=1,2,3,\cdots N$$

$$H\beta = Y$$

where $H(w_1, w_2, \cdots, w_L; b_1, b_2, \cdots, b_L; x_1, x_2, \cdots, x_N) =$

$$\begin{vmatrix} g(w_1x_1+b_1) & g(w_2x_1+b_2) & \cdots & g(w_Lx_1+b_L) \\ g(w_1x_2+b_1) & g(w_2x_2+b_2) & \cdots & g(w_Lx_2+b_L) \\ \vdots & \vdots & \ddots & \vdots \\ g(w_1x_N+b_1) & g(w_2x_N+b_2) & \cdots & g(w_Lx_N+b_L) \end{vmatrix}_{N\times L}$$

$$\beta = \left[\beta_1^T, \beta_2^T, \cdots, \beta_L^T\right]_{M\times L}^T$$

$$Y = \left[y_1^T, y_2^T, \cdots, y_N^T\right]_{M\times N}^T$$

ELM

ELM Model: $H\beta = Y$

$H(w_1, w_2, \cdots, w_L; b_1, b_2, \cdots, b_L; x_1, x_2, \cdots, x_N)$

If w and b *are given* ***randomly***,

the output weights can be ***analytically*** determined,namely

$$\beta = H^{-1}Y \Rightarrow \beta = H^{+}Y$$

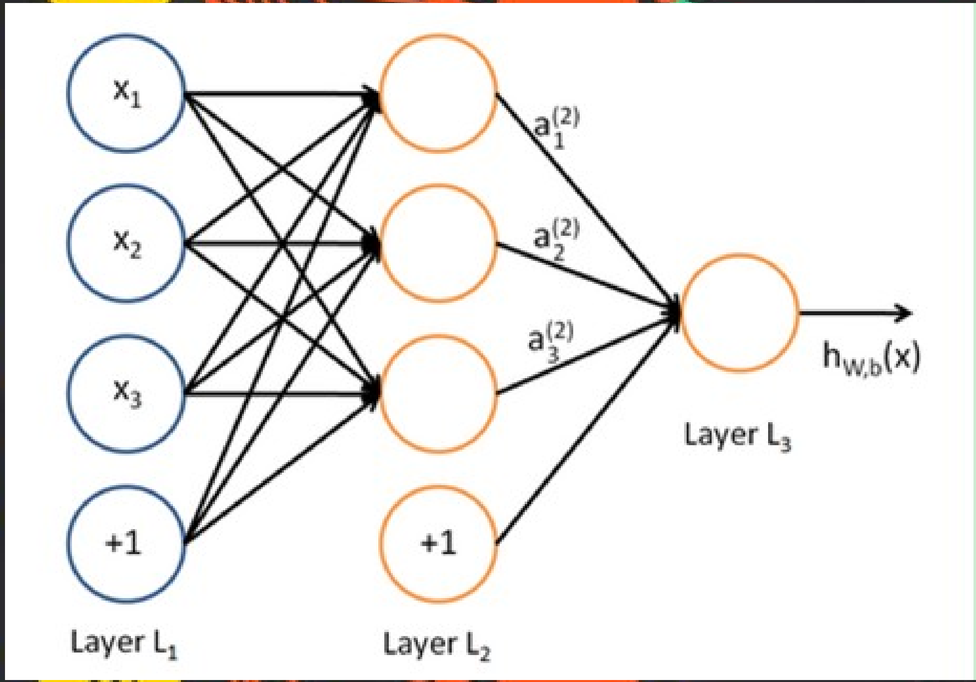
The only one artificial setting is number of the hidden layer nodes , *L*

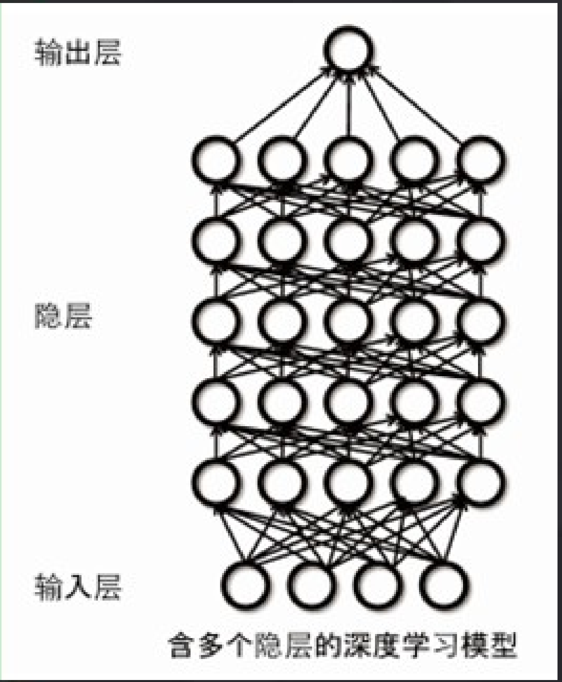
ELM

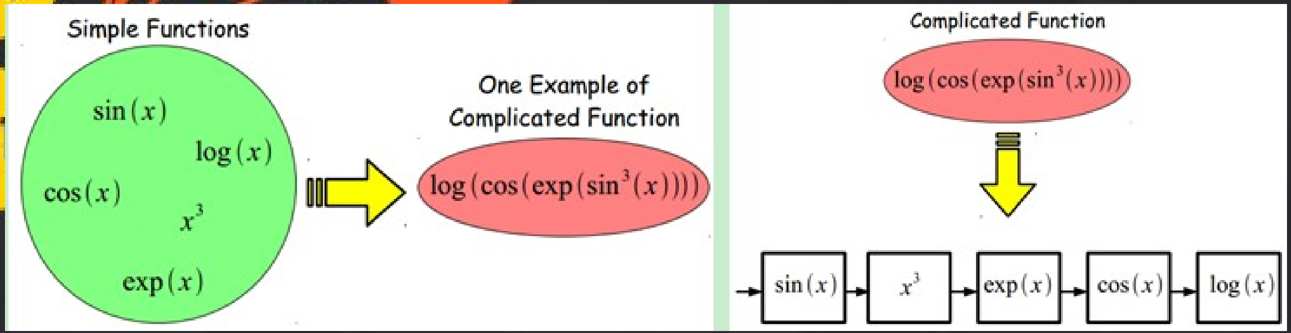
- ELM advantages
  - Batch training, extremely fast learning speed
  - better generalization performance
  - adopt the simplest method to overcome local minima, improper learning rate and overfitting
  - work for differentiable and nondifferentiable activation functions
  - mathematical foundation
- ELM disadvantages
  - The number of the hidden layer nodes is artificially given.
  - *H is generally a non-square matrix*

Geoffrey Hinton | 深度学习：多隐层 ，逐层初始化，无监督学习

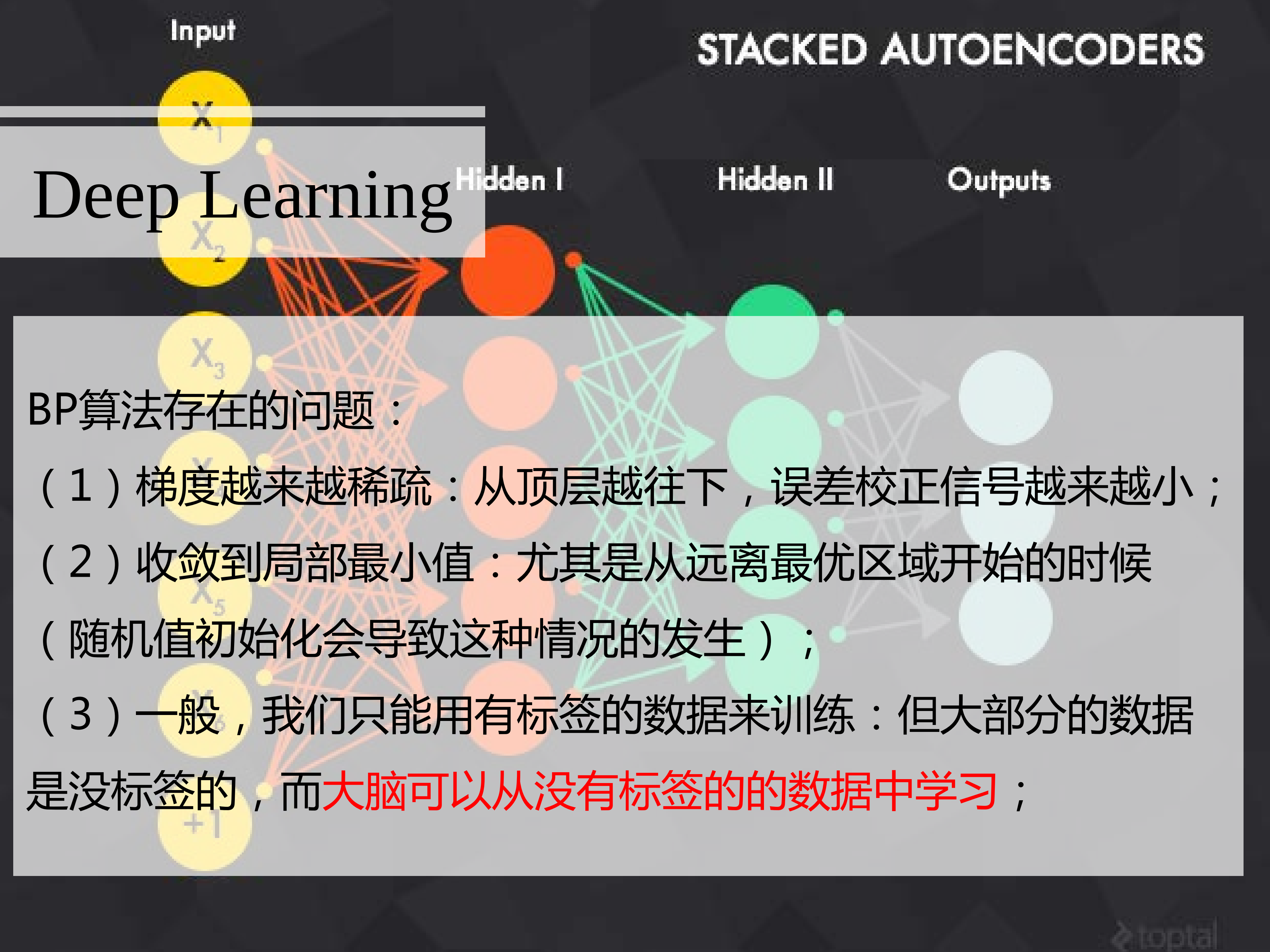
Deep Learning







Deep Learning



BP算法存在的问题：

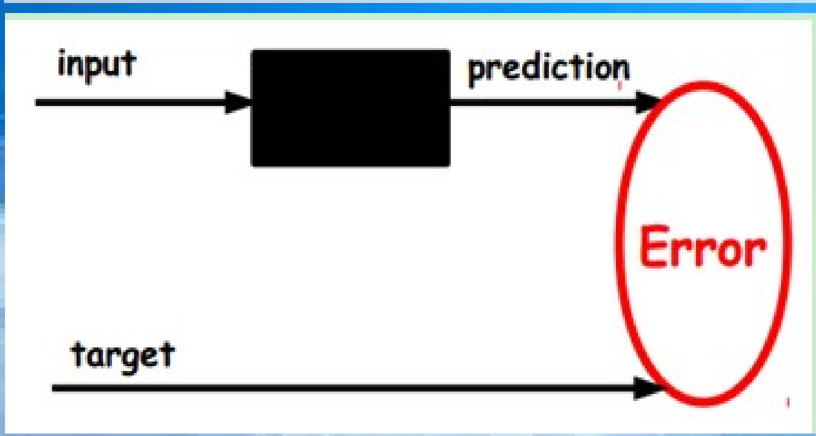
（1）梯度越来越稀疏：从顶层越往下，误差校正信号越来越小；

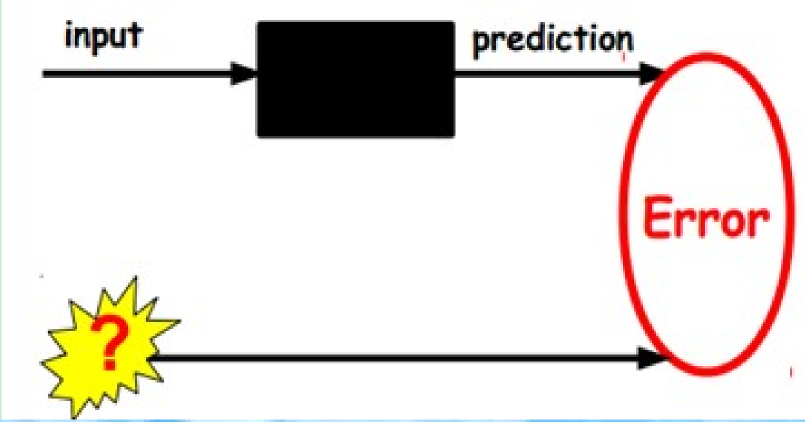
（2）收敛到局部最小值：尤其是从远离最优区域开始的时候（随机值初始化会导致这种情况的发生）；

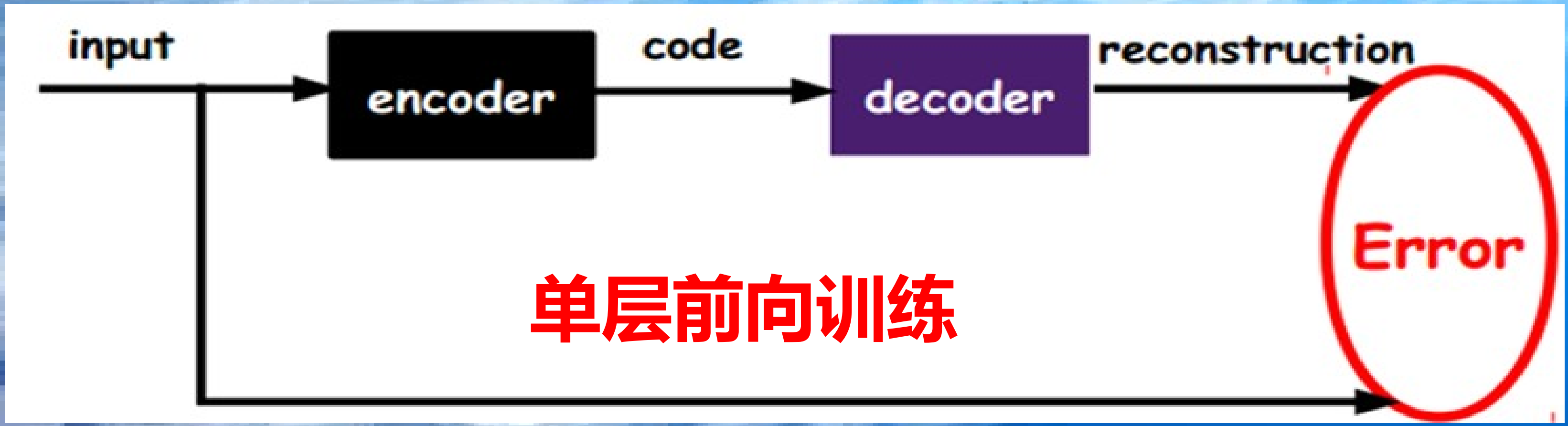
（3）一般，我们只能用有标签的数据来训练：但大部分的数据是没标签的，而大脑可以从没有标签的的数据中学习；

Auto Encoder

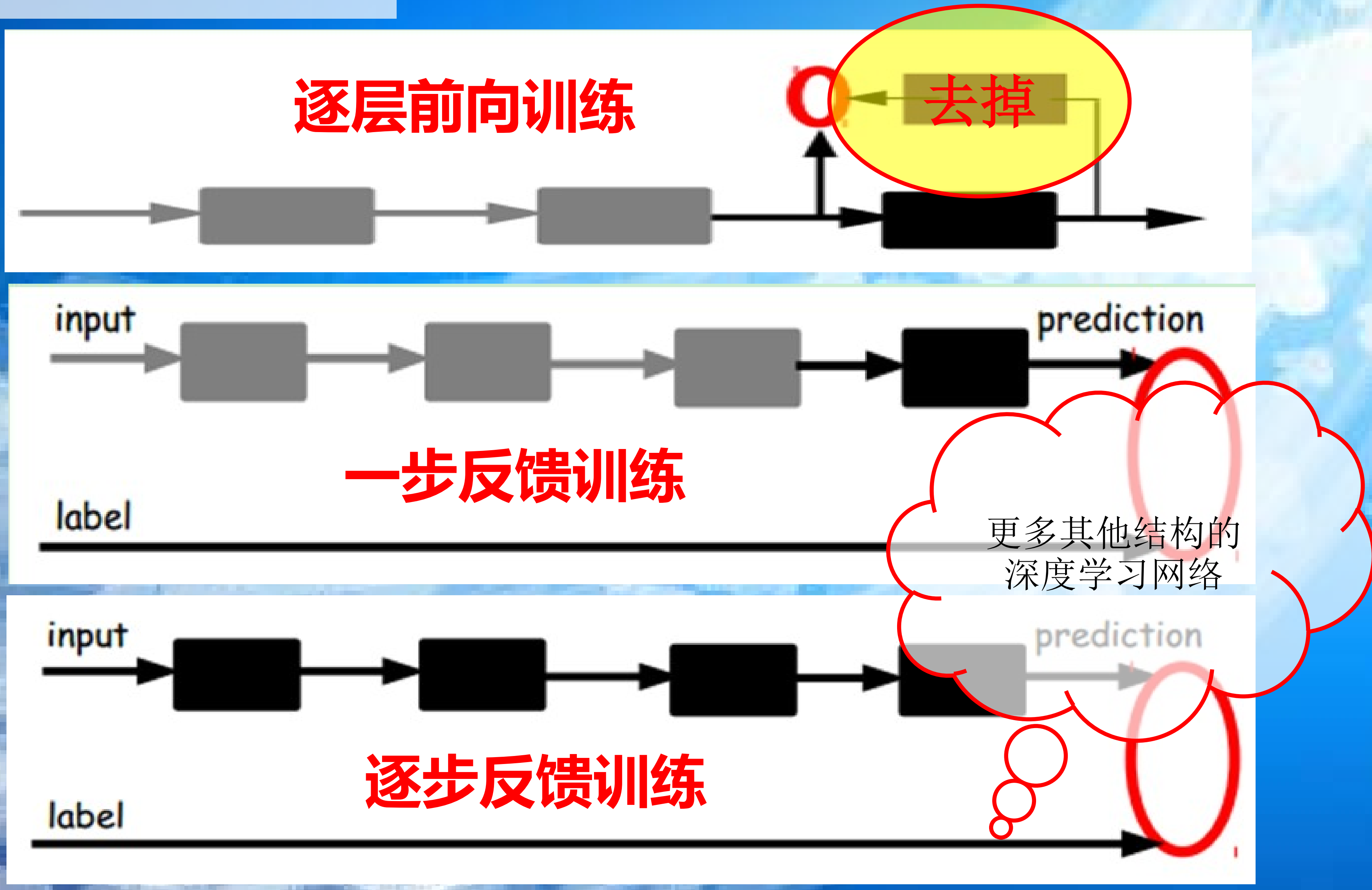
自动编码器就是一种尽可能复现输入信号的神经网络.



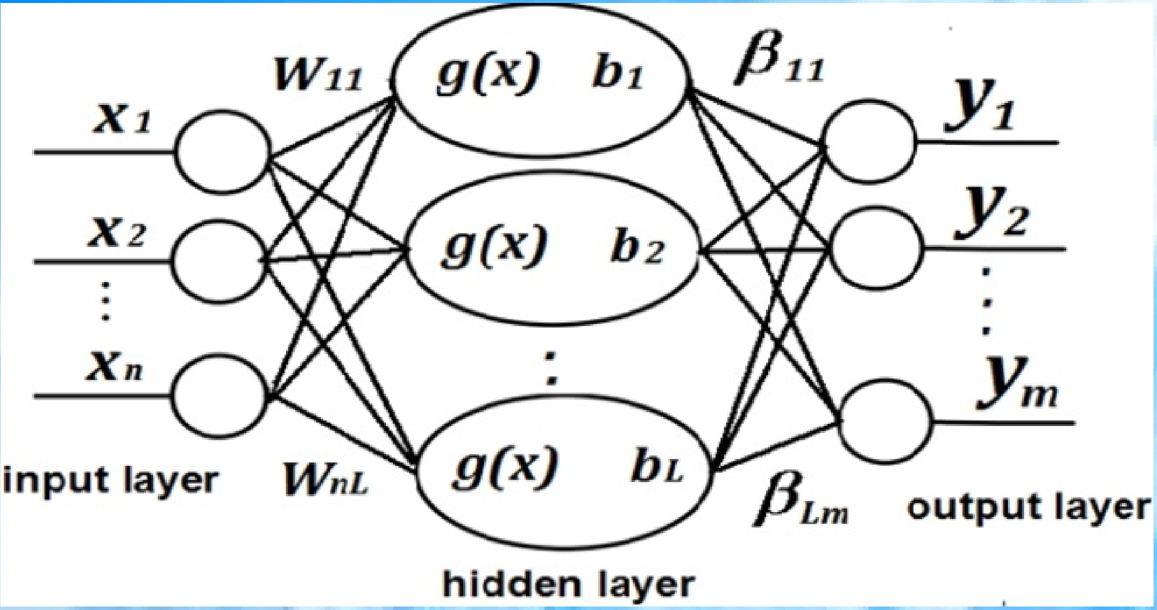




Auto Encoder



ELM-AE



- target output is the same as input x： $x = y$
- the hidden node parameters are made orthogonal after being randomly generated： $w_i^T w_i = \mathrm{I}, b^T b = 1$

ELM-AE

$$\beta = H^{-1}Y$$

$$\beta = \left(\frac{1}{C} + H^T H\right)^{-1} H^T Y$$

$$X = Y$$

$$H\beta = X$$

$$\beta^T \beta = I \Rightarrow H = X\beta^T$$

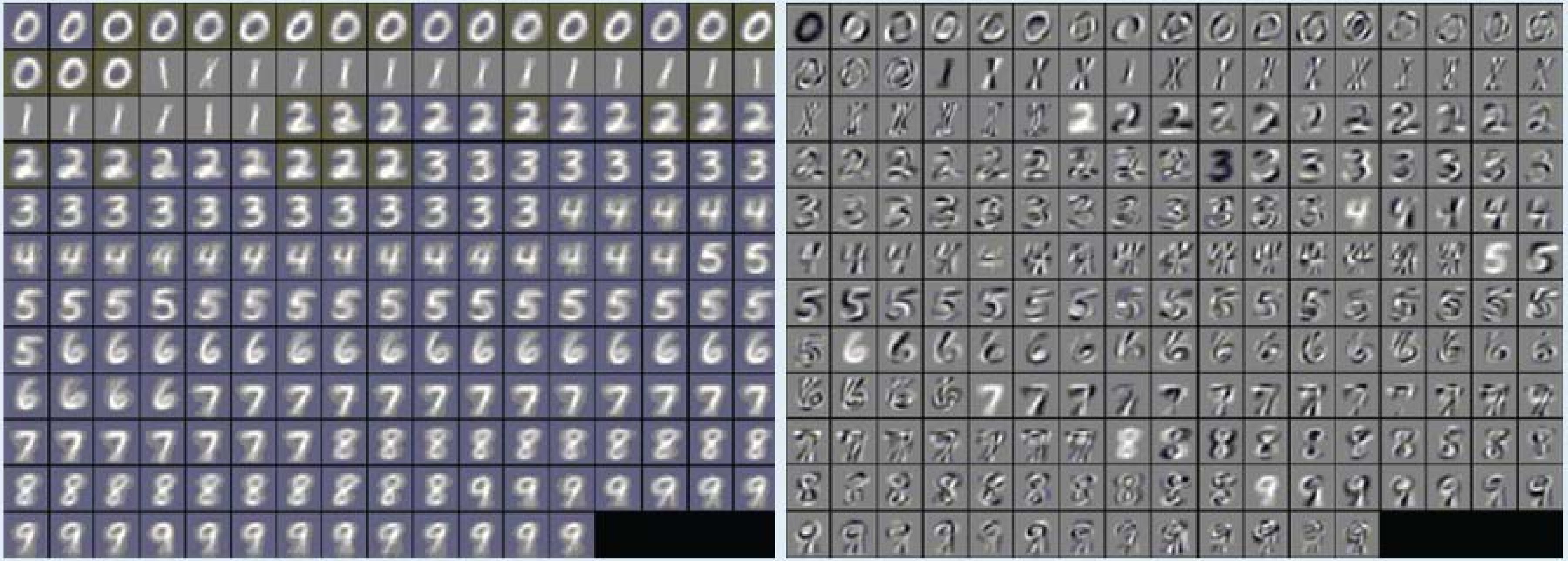
PCA

$$TP^T = X$$

Deep Learning

$$\beta = H^{-1}Y$$

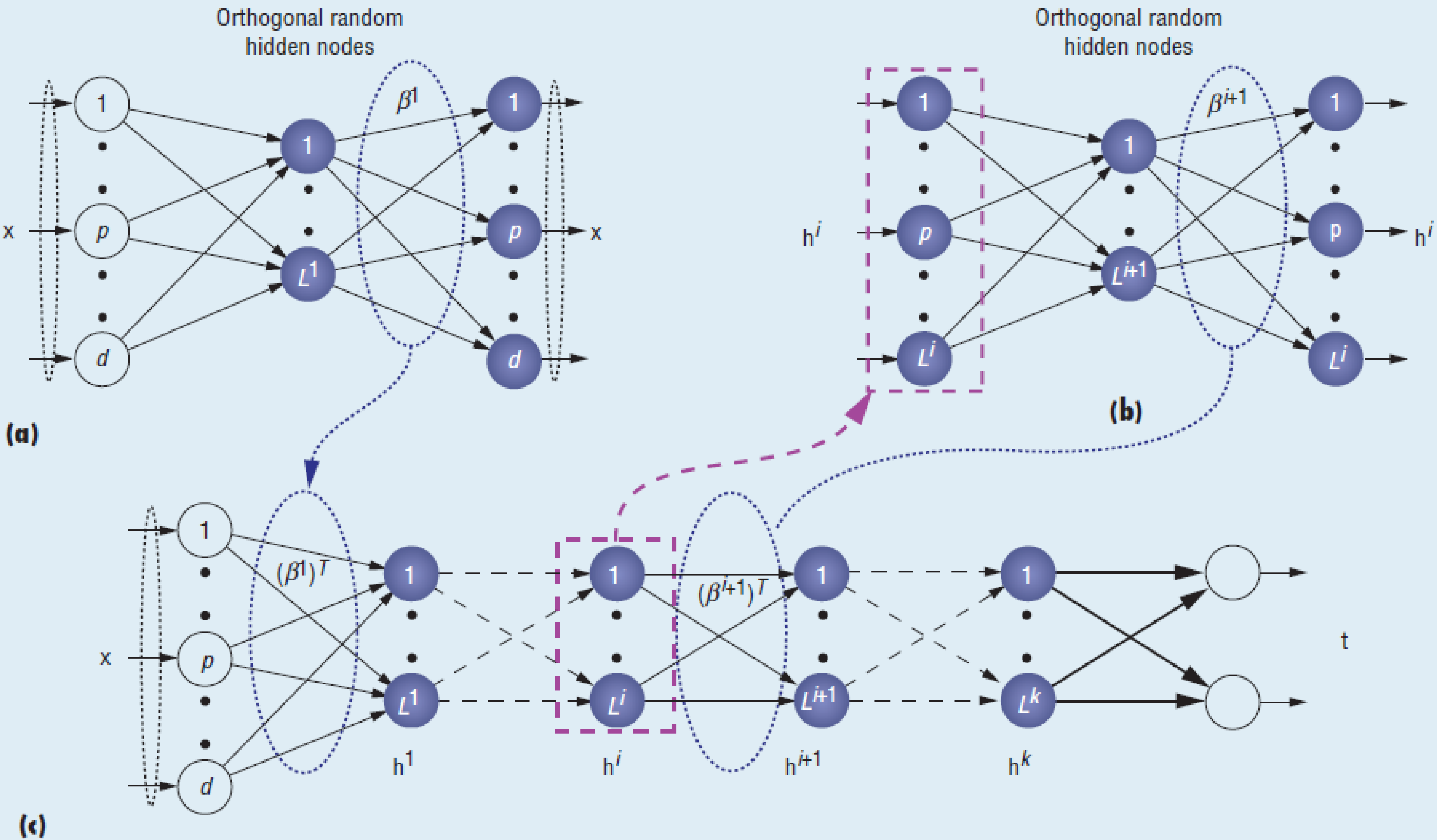
PCA



(a) (b)

$784 \times 20 \times 784$

Deep Learning



Deep Learning

Table 1. Performance comparison of ML-ELM with state-of-the-art deep networks.

Algorithms	Testing accuracy % (standard deviation %)	Training time
Multi-layer extreme learning machine (ML-ELM)	99.03 (±0.04)	444.655 s
Extreme learning machine (ELM random features)	97.39 (±0.1)	545.95 s
ELM (ELM Gaussian kernel); run on a faster machine	98.75	790.96 s
Deep belief network (DBN)	98.87	20,580 s
Deep Boltzmann machine (DBM)	99.05	68,246 s
Stacked auto-encoder (SAE)	98.6	–
Stacked eenoising auto-encoder (SDAE)	98.72	–

这么简单？| 深度学习，没有这么简单
大而复杂的模型，训练量大，收敛，并行计算

Hadoop | 分而治之，让大象跳舞

Hadoop 架构 | 好大的一个工程！


新春大吉
恭賀新禧
福
福