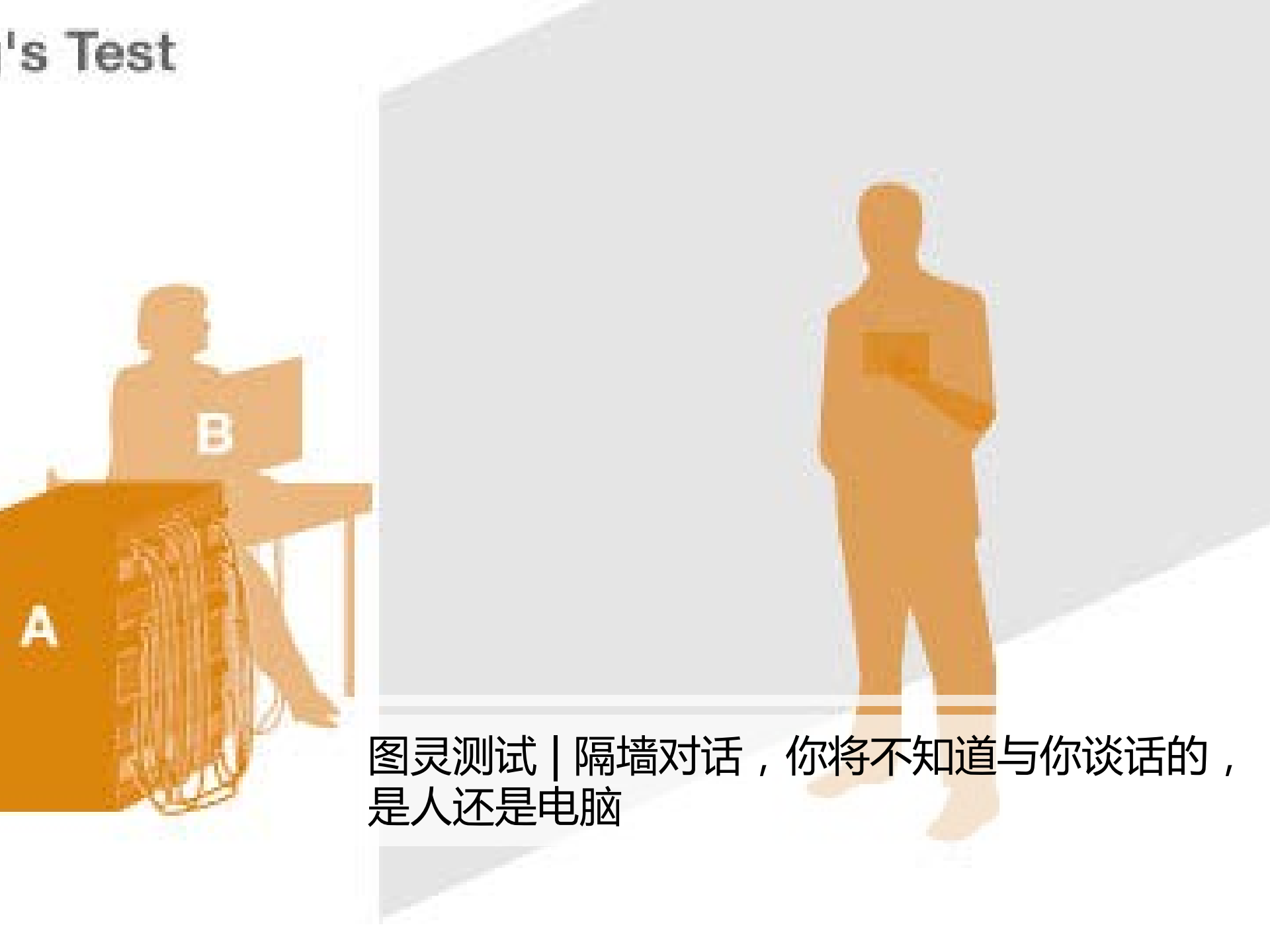


# Deep learning and ELM

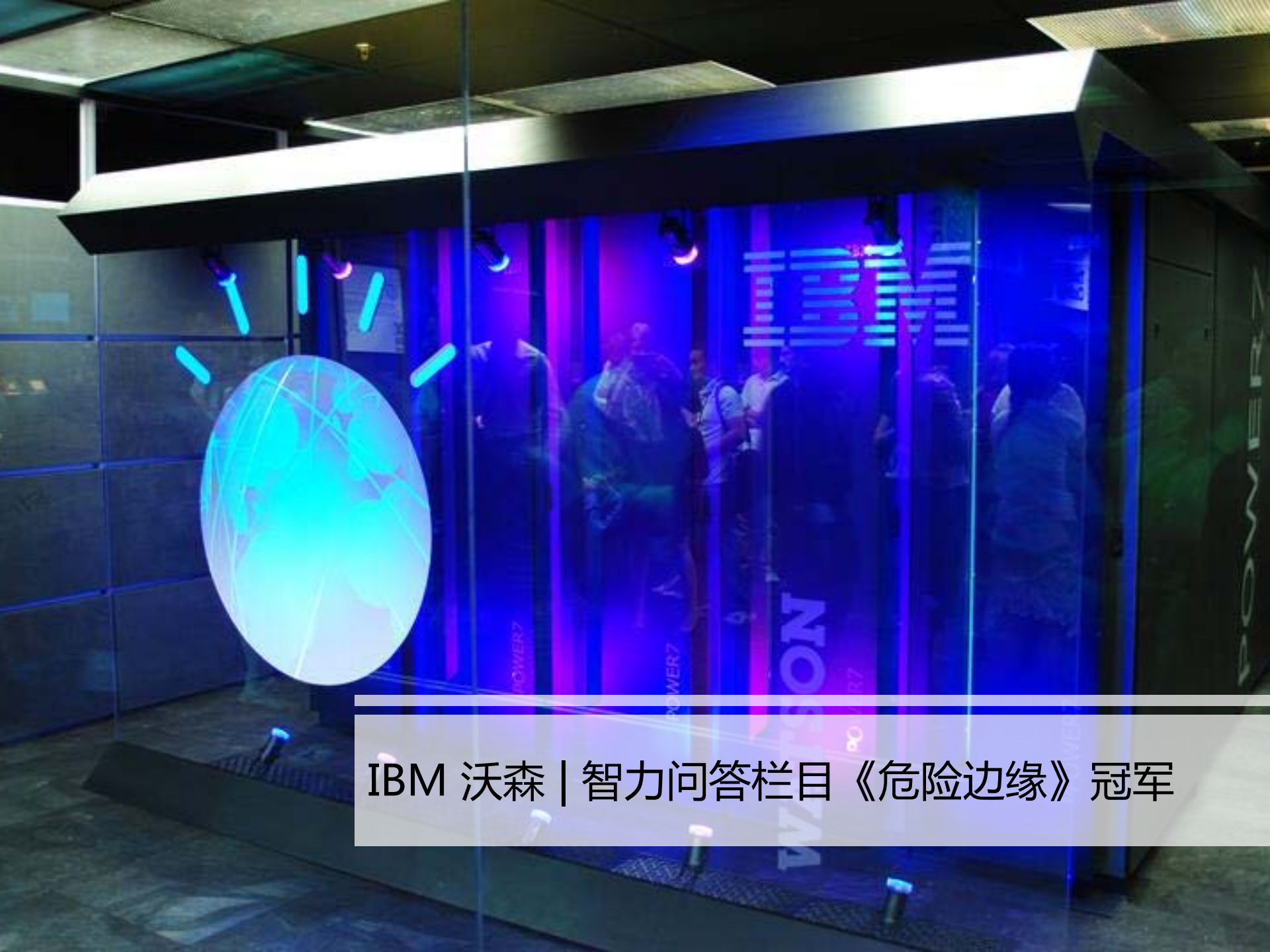
大头雨山

# 's Test

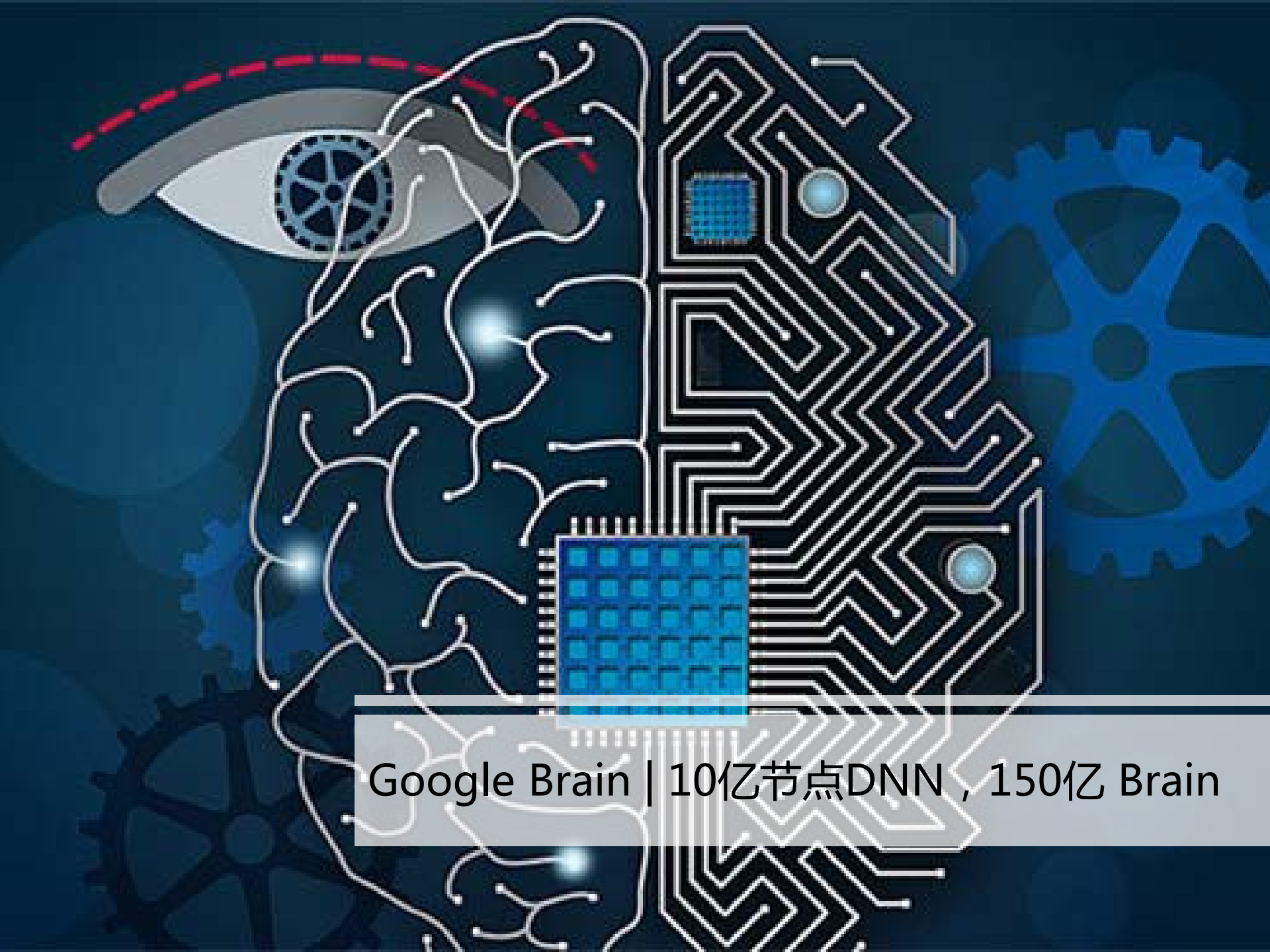


图灵测试 | 隔墙对话，你将不知道与你谈话的，  
是人还是电脑






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




And we hope in a few years that we'll be able to break down  
language barriers between people.  
I personally believe this is going to lead to a better world.  
I hope you enjoy the rest of the presentations today.

数年后，我们希望能够打破人们之间的语言障

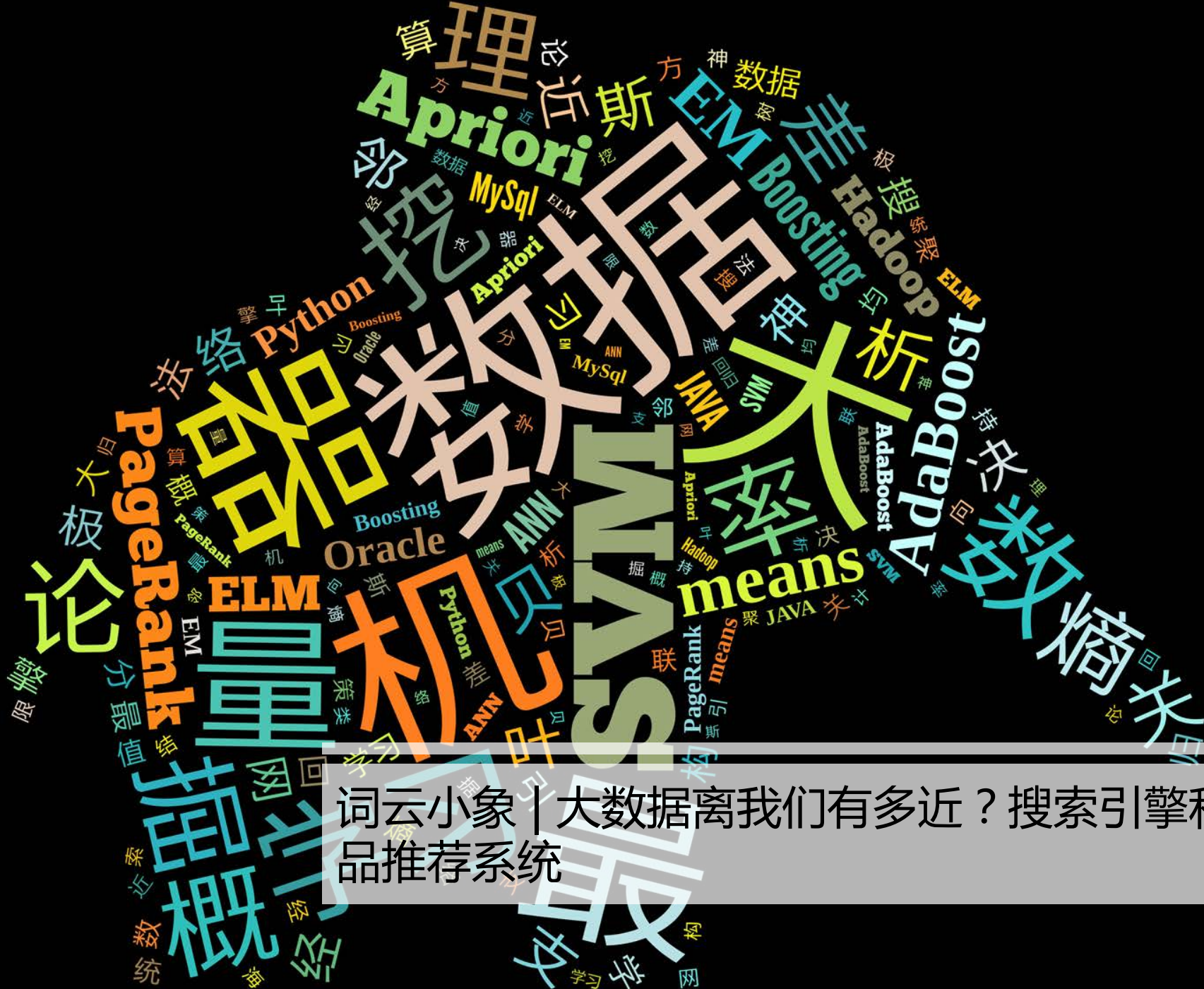
这我个人认为这将导致更好的世界。  
希望今天你喜欢其他的讲座。

同声传译 | 语音识别、英中机器翻译，中文语音  
合成

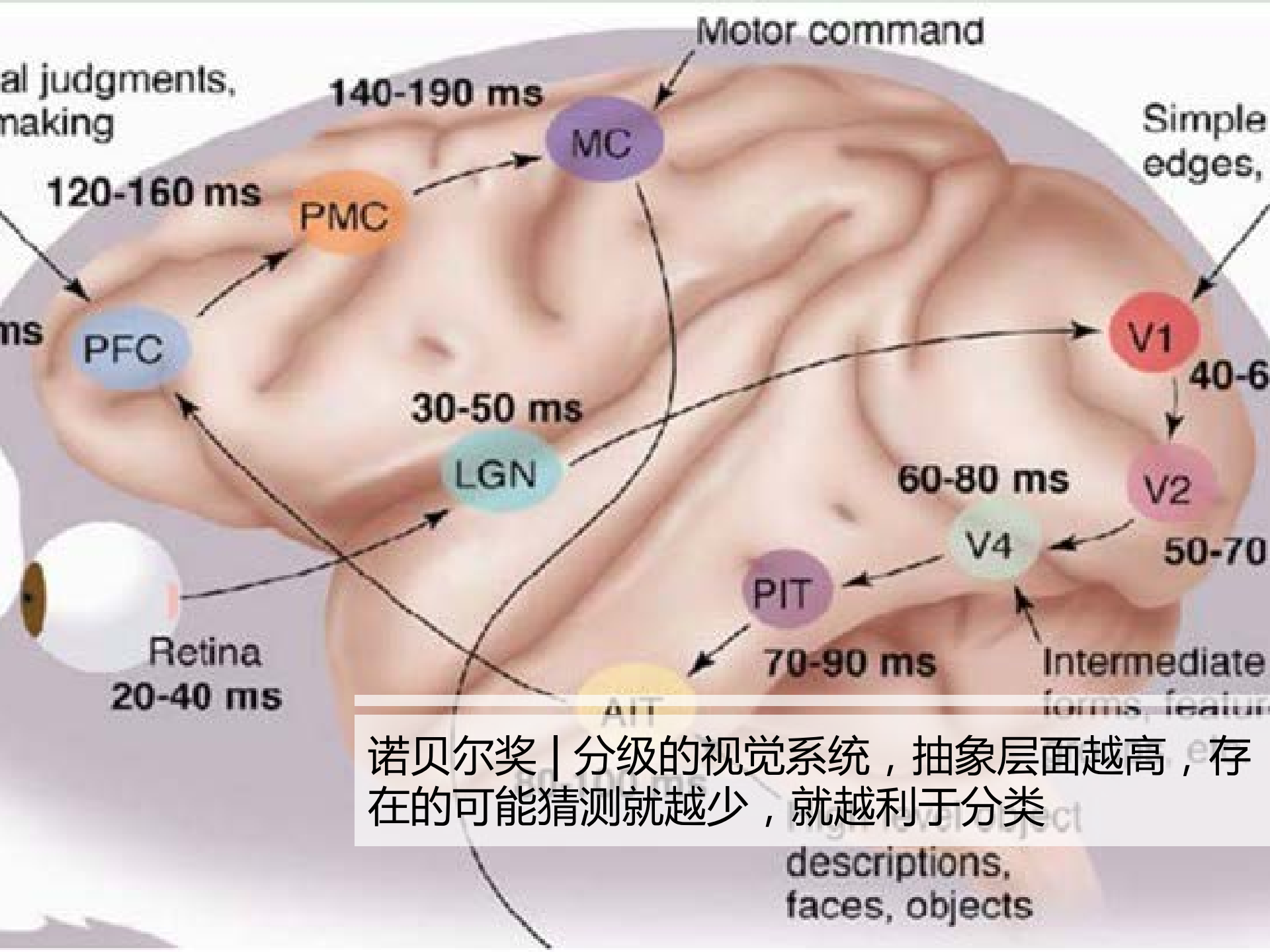


Baidu IDL | 百度深度学习研究院  
商品图像检索 | <http://shitu.baidu.com/app>





词云小象 | 大数据离我们有多近？搜索引擎和商品推荐系统



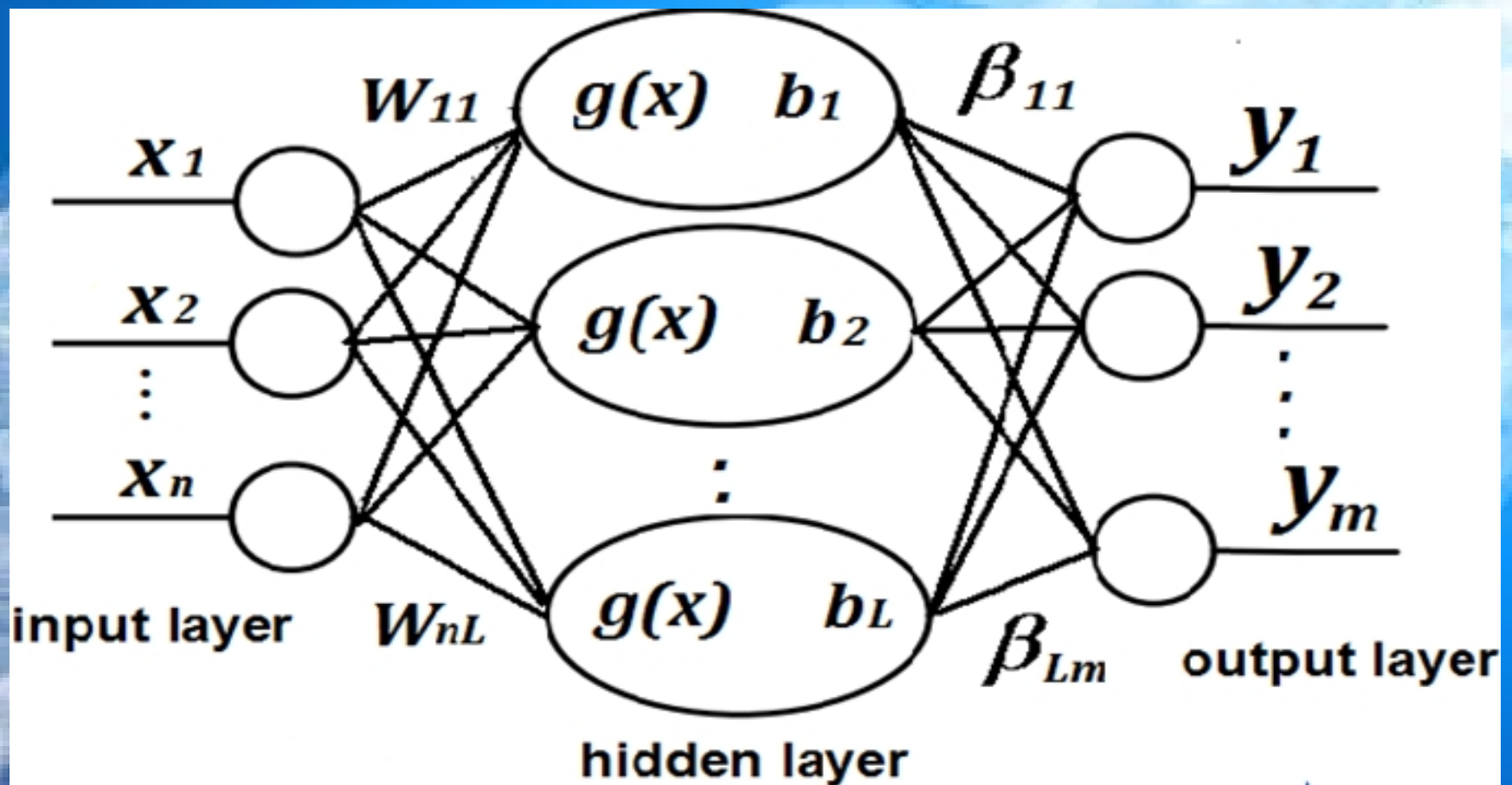
诺贝尔奖 | 分级的视觉系统，抽象层面越高，存在的可能猜测就越少，就越利于分类



A stylized blue brain is the central focus, rendered with a translucent, wireframe-like texture. It is surrounded by numerous bright blue light rays that radiate outwards, creating a sense of dynamic energy and connectivity. The background is a deep blue, filled with faint, vertical columns of binary code (0s and 1s), which adds to the technological and digital theme of the image.

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*

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

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

$j=1,2,\dots,n$

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

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

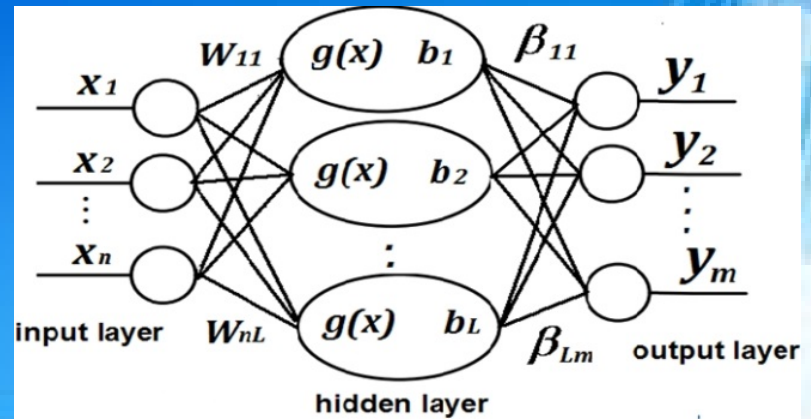
$$\Delta w_{ij} = \eta (d_i - y_i) f'(net_i) x_j$$



# 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:  $(x_j, y_j), j = 1, 2, 3 \dots N$

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

ELM Model is as follows

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

input layer weight:  $w_i$       output layer weight:  $\beta_i$

Number of the hidden layer nodes:  $L$

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



$$H\beta = Y$$

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

$$\begin{bmatrix} g(w_1 x_1 + b_1) & g(w_2 x_1 + b_2) & \cdots & g(w_L x_1 + b_L) \\ g(w_1 x_2 + b_1) & g(w_2 x_2 + b_2) & \cdots & g(w_L x_2 + b_L) \\ \vdots & \vdots & \ddots & \vdots \\ g(w_1 x_N + b_1) & g(w_2 x_N + b_2) & \cdots & g(w_L x_N + b_L) \end{bmatrix}_{N \times L}$$

$$\beta = [\beta_1^T, \beta_2^T, \dots, \beta_L^T]^T_{M \times L}$$

$$Y = [y_1^T, y_2^T, \dots, y_N^T]^T_{M \times N}$$



# ELM

ELM Model:  $H\beta = Y$

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

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

the output weights can be analytically determined, namely

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

★ 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



Input

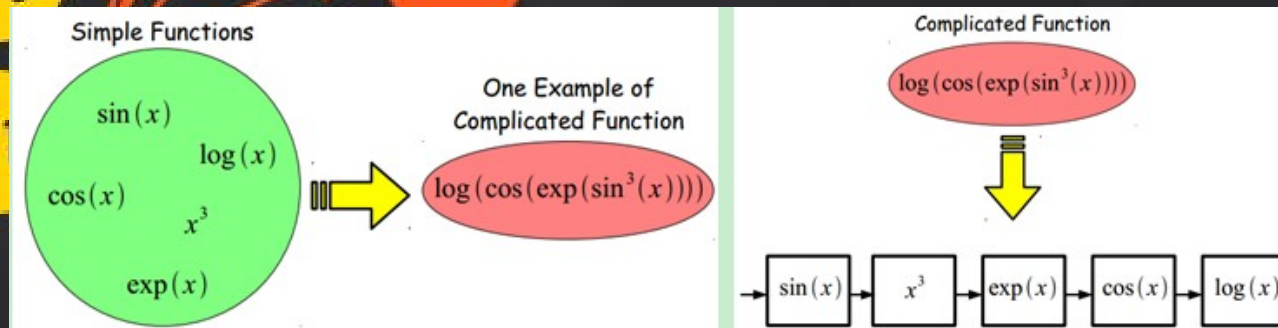
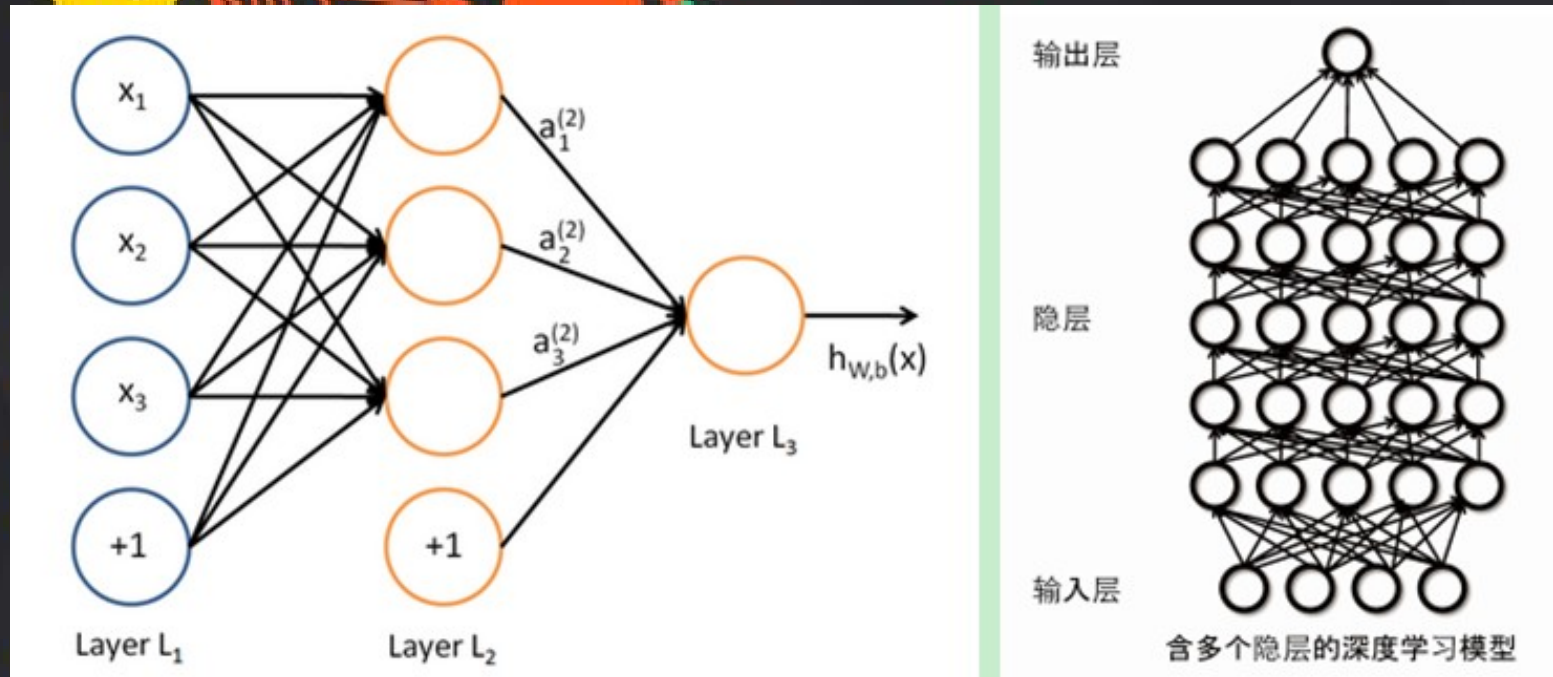
# STACKED AUTOENCODERS

## Deep Learning

Hidden I

Hidden II

Outputs



Input

# STACKED AUTOENCODERS

## Deep Learning

Hidden I

Hidden II

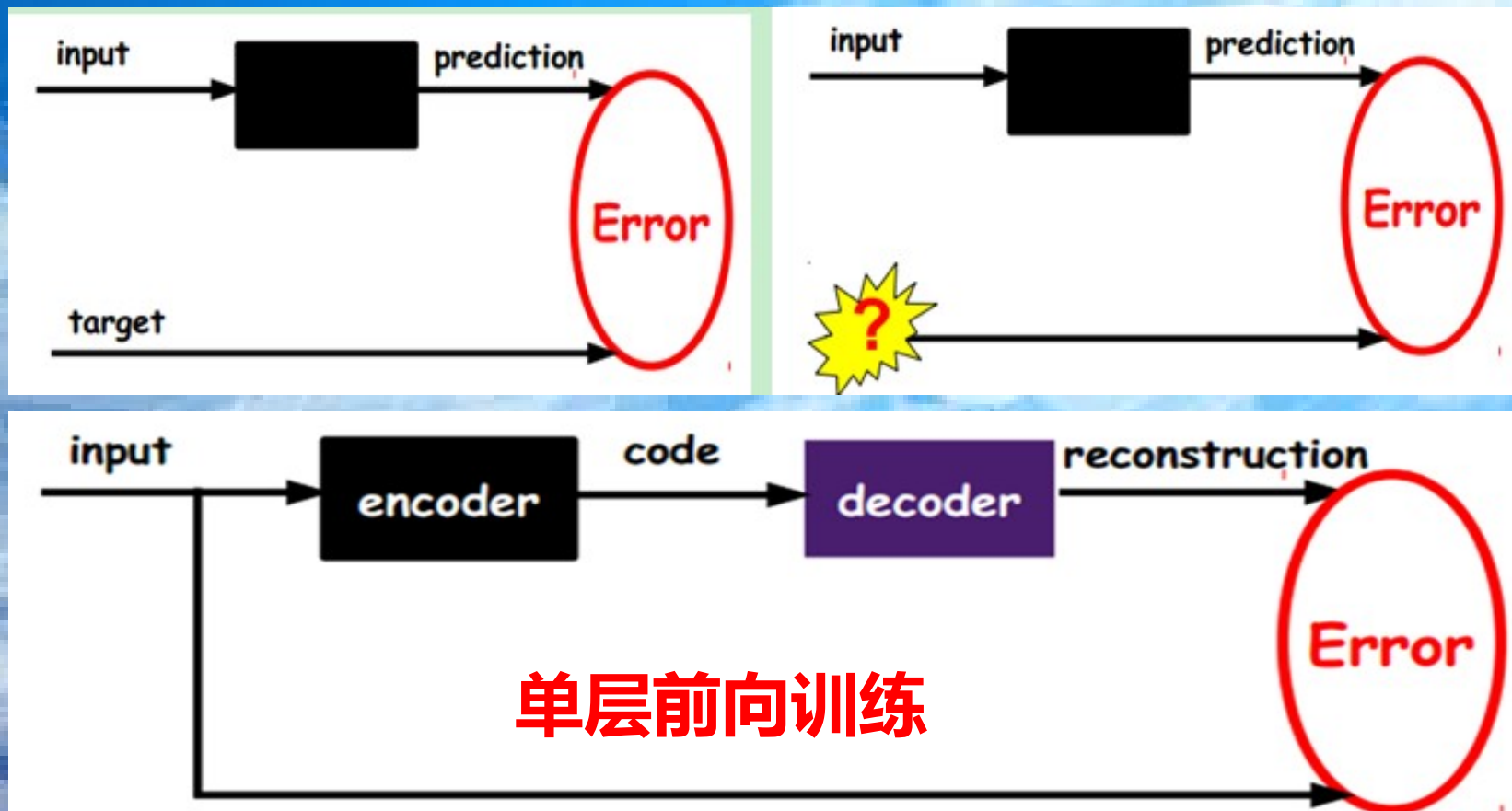
Outputs

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

逐层前向训练

去掉

input

prediction

一步反馈训练

label

更多其他结构的  
深度学习网络

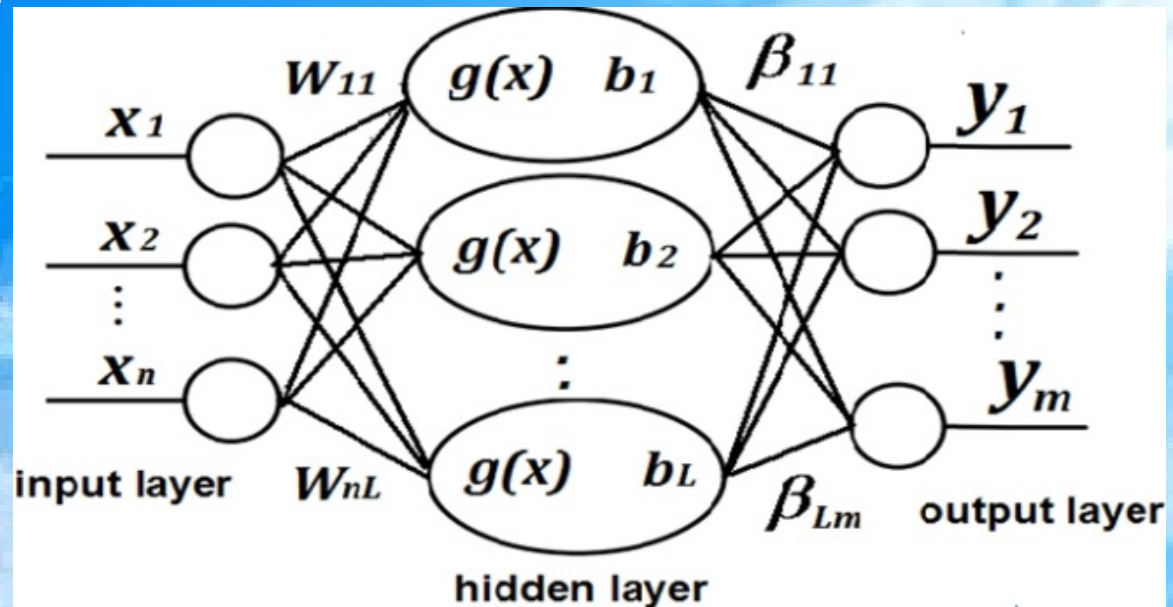
input

prediction

逐步反馈训练

label

# ELM-AE



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

# ELM-AE

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

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

$$X = Y$$

$$H \beta = X$$

$$\beta^T \beta = I$$



$$H = X \beta^T$$

**PCA**

$$TP^T = X$$

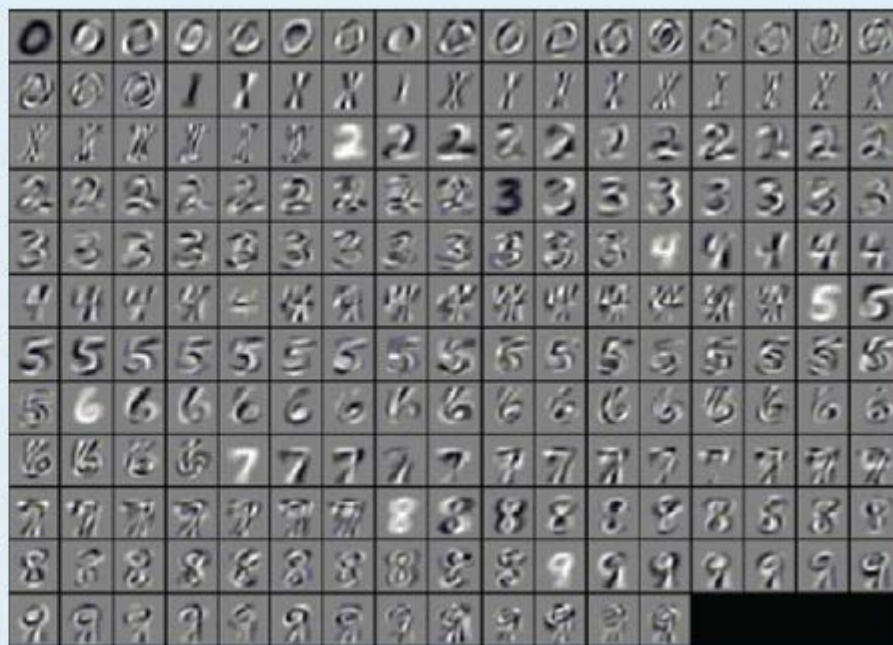
# Deep Learning

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

**PCA**



(a)



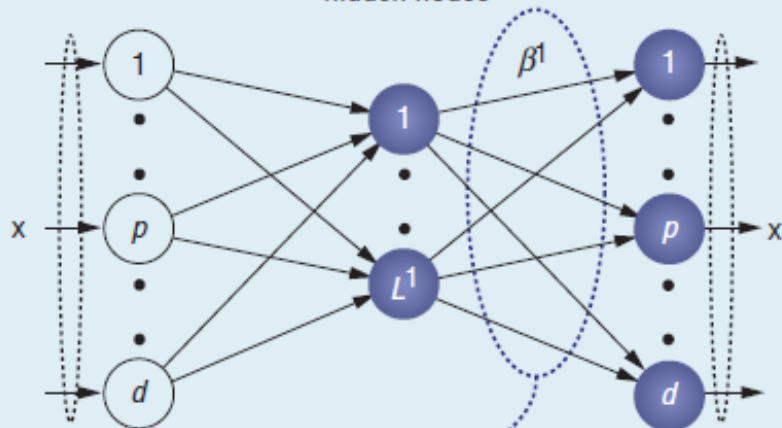
(b)

$784 \times 20 \times 784$



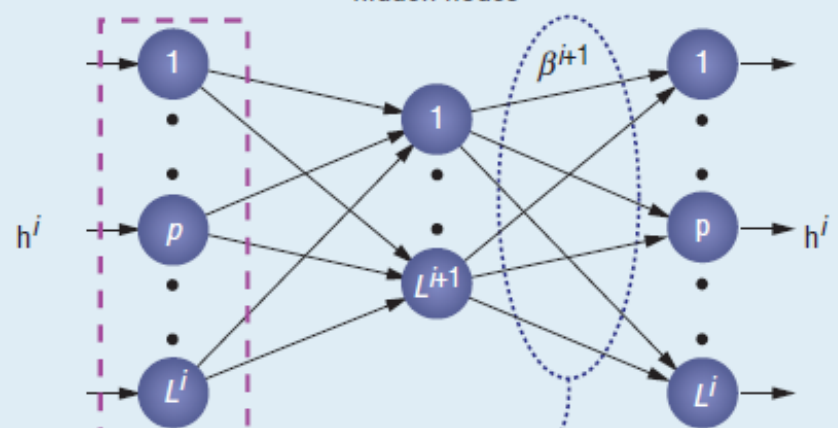
# Deep Learning

Orthogonal random  
hidden nodes

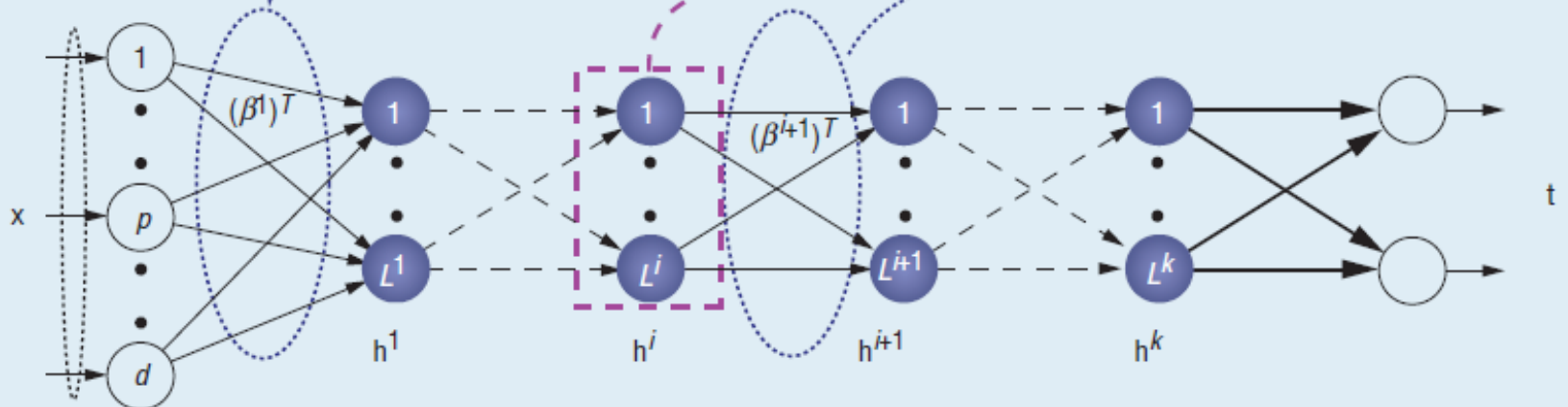


(a)

Orthogonal random  
hidden nodes



(b)




(c)

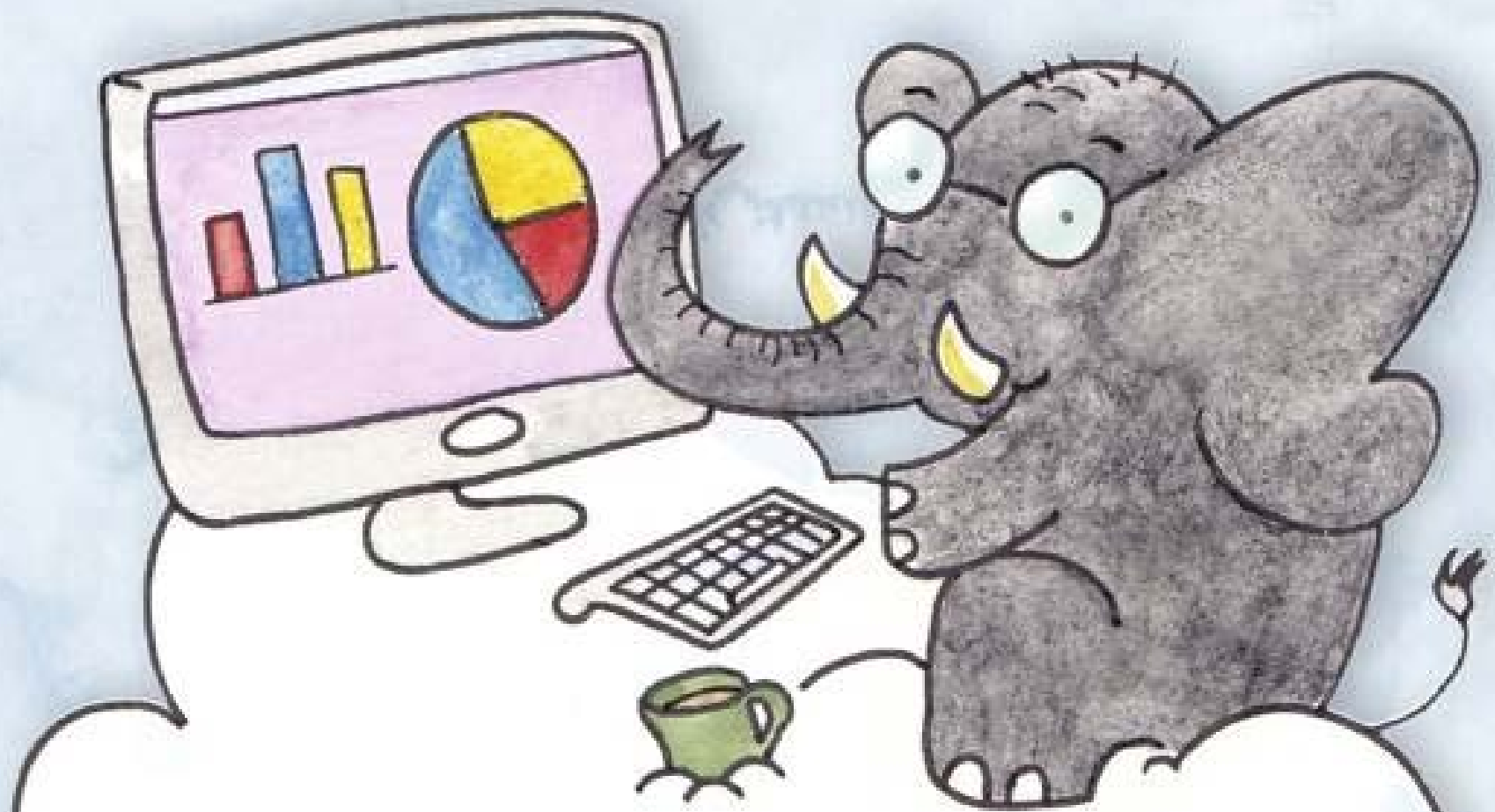
# Deep Learning

**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 ( $\pm 0.04$ )	444.655 s
Extreme learning machine (ELM random features)	97.39 ( $\pm 0.1$ )	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 denoising 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



# Intel Hadoop Manager

安装、部署、配置、监控、告警和访问控制

Sqoop  
关系数据ETL工具

Mahout  
数据挖掘

Pig  
数据流处理语言

Hive  
数据仓库

MapReduce  
分布式计算框架

Flume  
日志收集工具

HBase  
实时、分布式、高维数据库

HDFS  
分布式文件系统

Zookeeper  
分布式协作服务

Hadoop 架构 | 好大的一个工程！

新春大吉

福

福

恭賀新禧

福

福