AIAA2205

Section 3 Data, Model, Task

第三节 数据、模型、任务

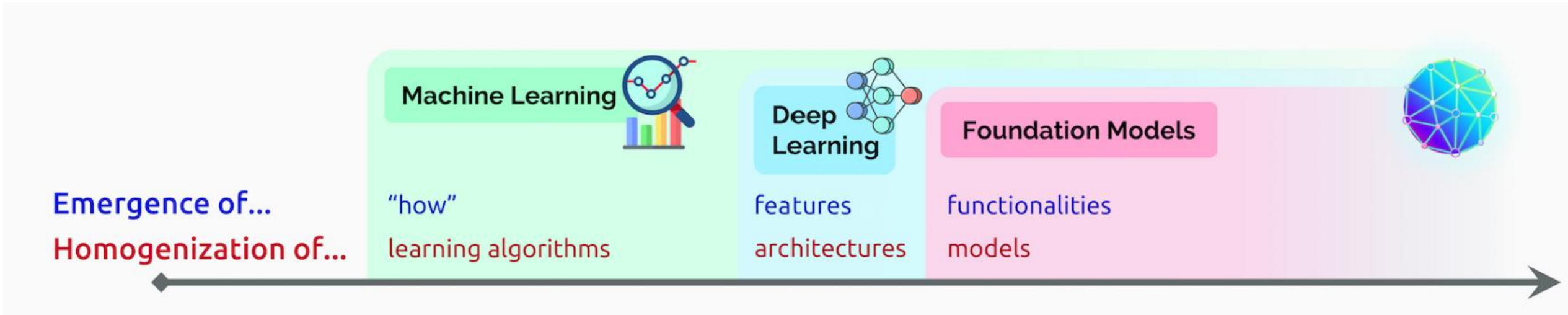
Part (1) The big picture

第一部分 人工智能的大图景

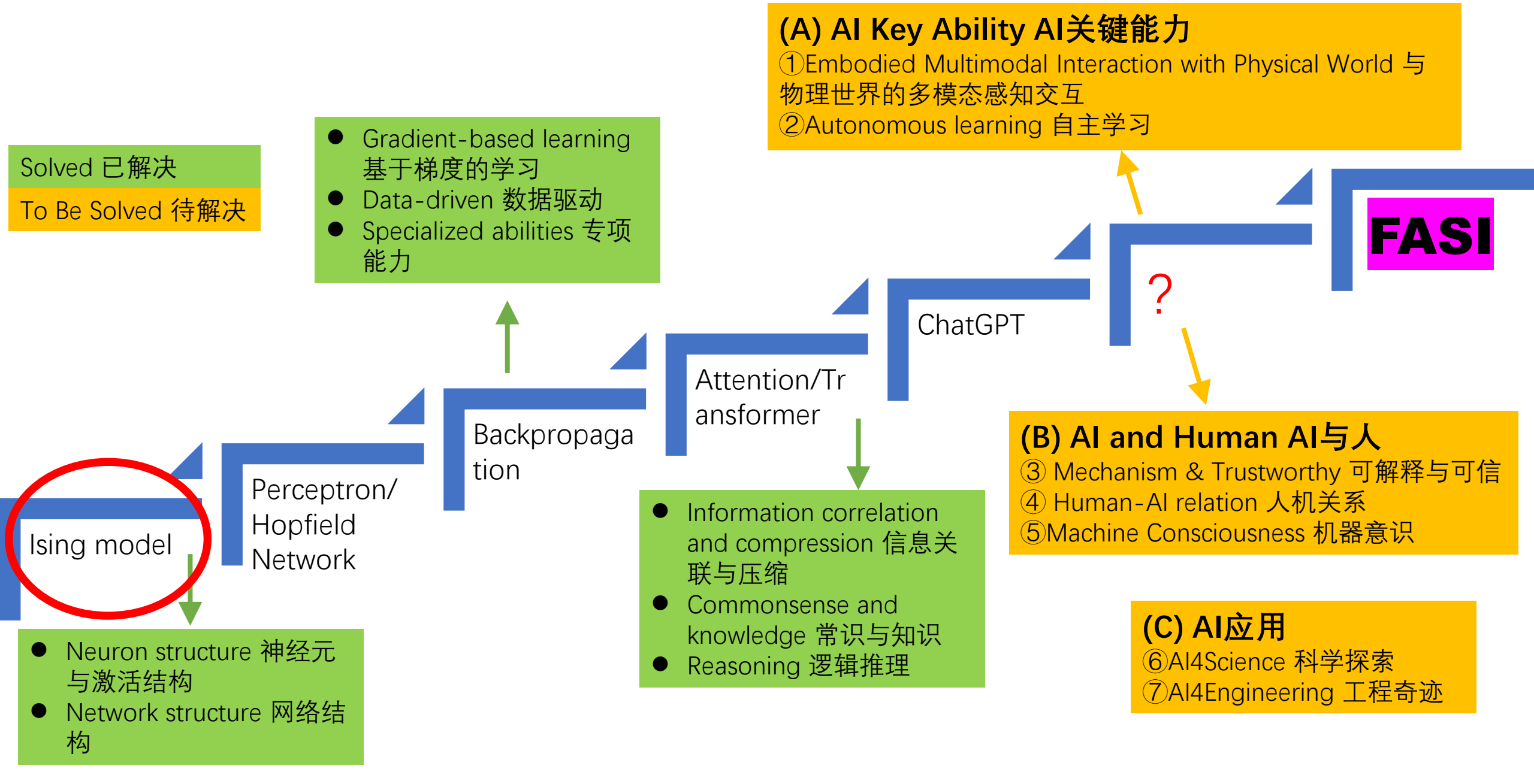
Yutao Yue 岳玉涛

AI technology

- Where are we?
- How did we get here?
- Where are we going?

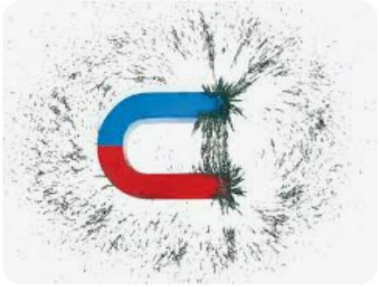

Machine Learning
Deep Learning
Foundation Models
Emergence of...
Homogenization of...
"how"
learning algorithms
features
architectures
functionalities
models

Pathway to Full Artificial Super Intelligence

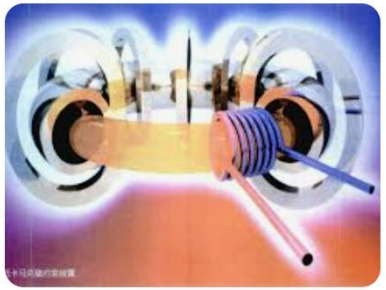


From magnets to LLM?

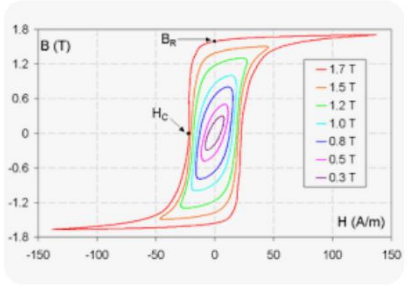
Common phenomena around us身边的常见现象



凤凰网
磁现象的三个大秘密__凤凰网



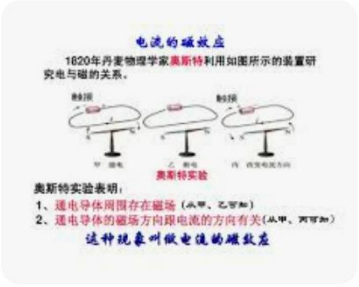
enjoyphysics.cn
第三章1 磁现象和磁场



Wikipedia
磁滞现象- 维基百科，自由的百科全书

www.51dz.com
一、简单的磁现象

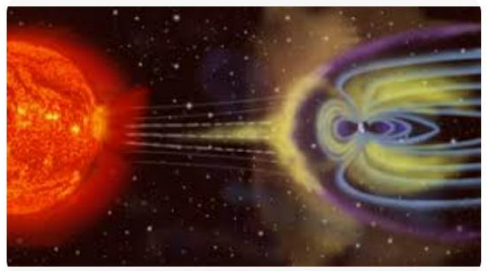
www.wuxiangrensheng....
磁现象磁场



搜狗百科
电流磁效应(物理现象)_搜狗百科

宁波众诚磁业有限公司
磁的奥秘：从何而来？_宁波...

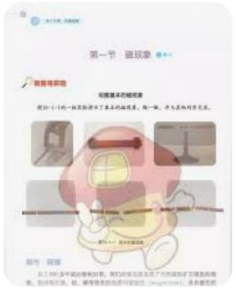
第一PPT
磁现象磁场》电与...



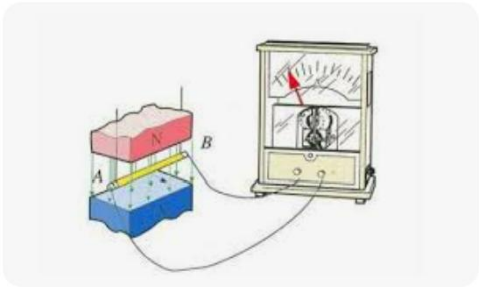
Wikipedia
磁暴- 维基百科，自由的百科全书

enjoyphysics.cn
第十一章第一节磁现象磁感线

第一PPT
磁现象磁场》电与...



好多电子课本网
第一节磁现象(Pag...



电子发烧友
什么是电磁感应现象_电磁感应现象的应用介...

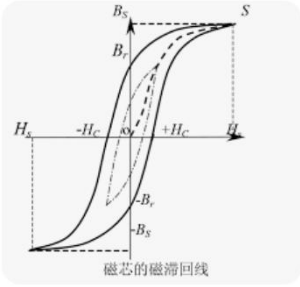
Wikiwand
磁- Wikiwand

enjoyphysics.cn
第三章1 磁现象和磁场

昆西空压机
永磁电机失磁原因都在这里了-昆西

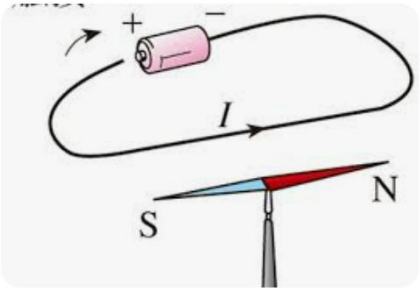
www.51dz.com
一、简单的磁现象

好多电子课本网
一磁现象(Page28) ...



大大通
磁学基础和偏磁饱和现象- ...

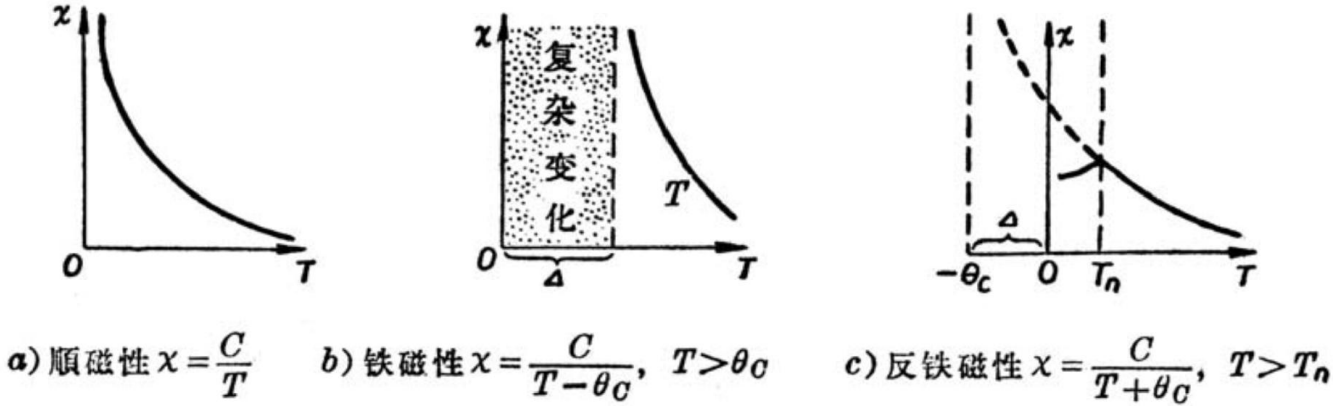
233网校
初中物理教师资格证面试真题...



抖音百科
电生磁- 抖音百科

Different types of magnetism 不同类型的磁性

- Diamagnetism抗磁性
  - 金、银、铜
- Paramagnetism顺磁性
  - 铝、钠
- Ferromagnetism铁磁性
  - 铁、镍、钴
- Antiferromagnetism反铁磁性
  - 铬、锰
- Ferrimagnetism亚铁磁性
  - 磁铁矿（Fe3O4）
- Superparamagnetism超顺磁性



a) 顺磁性 $\chi=\frac{C}{T}$　b) 铁磁性 $\chi=\frac{C}{T-\theta_C}$, $T>\theta_C$　c) 反铁磁性 $\chi=\frac{C}{T+\theta_C}$, $T>T_n$

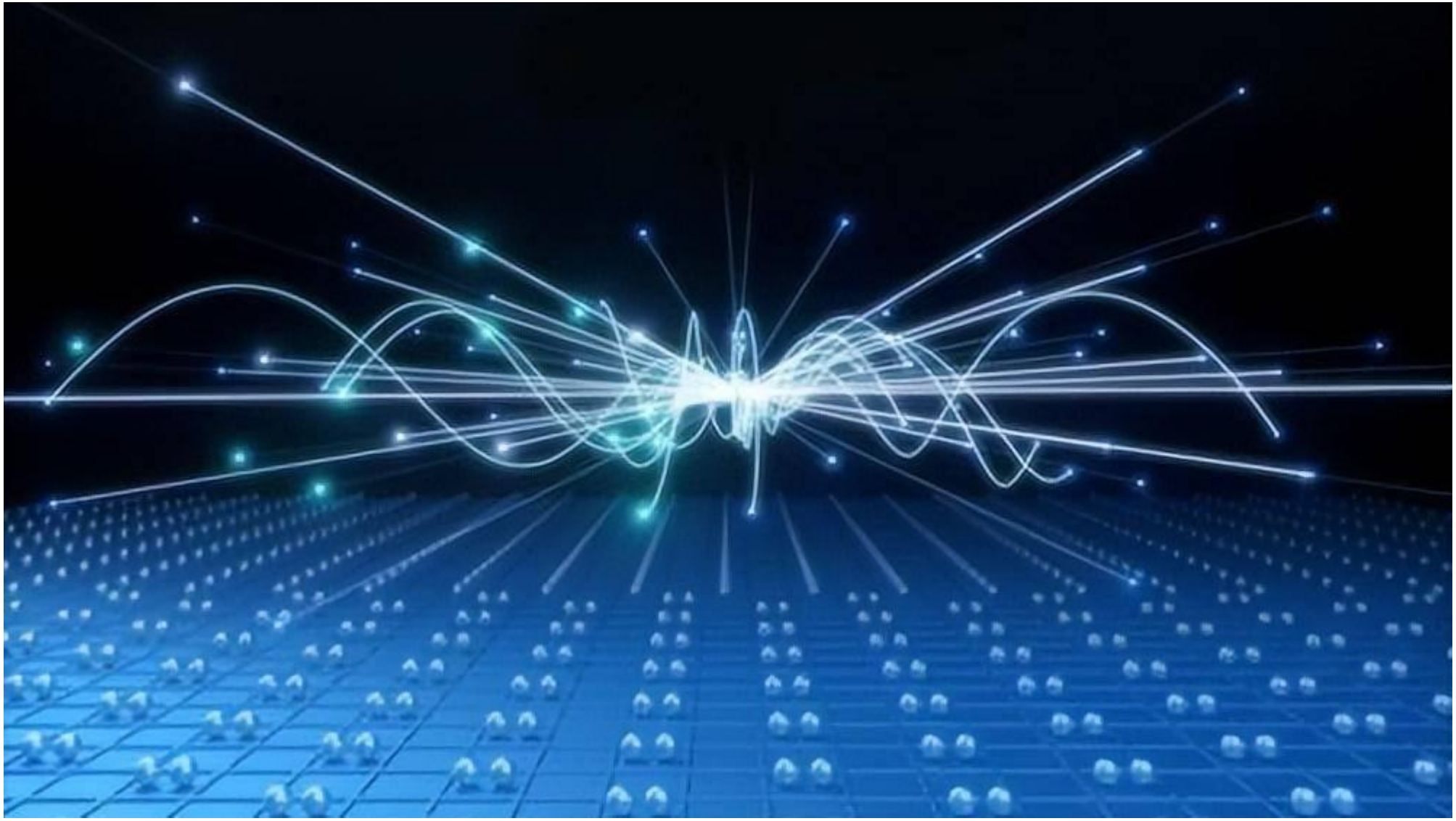
Magnetism is closely related to “temperature”
磁特性与“温度”密切相关

Please note the keyword:
请记住关键词：

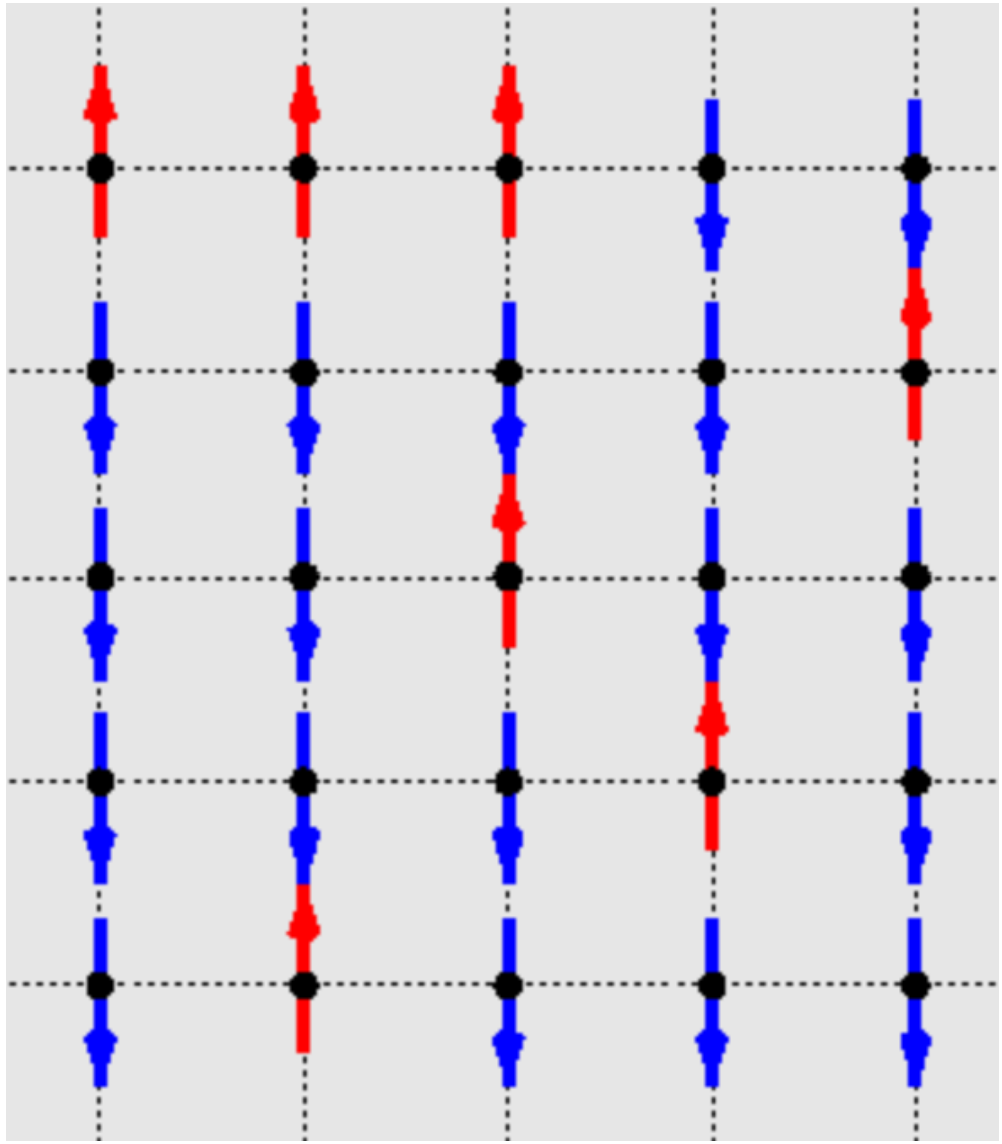
Temperature
温度

Figure from: 《Computational Physics》(Zejun Ding)

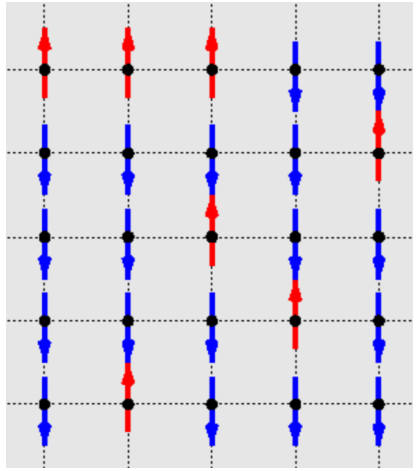
A pure quantum phenomenon



Simple theoretical model: the Ising Model



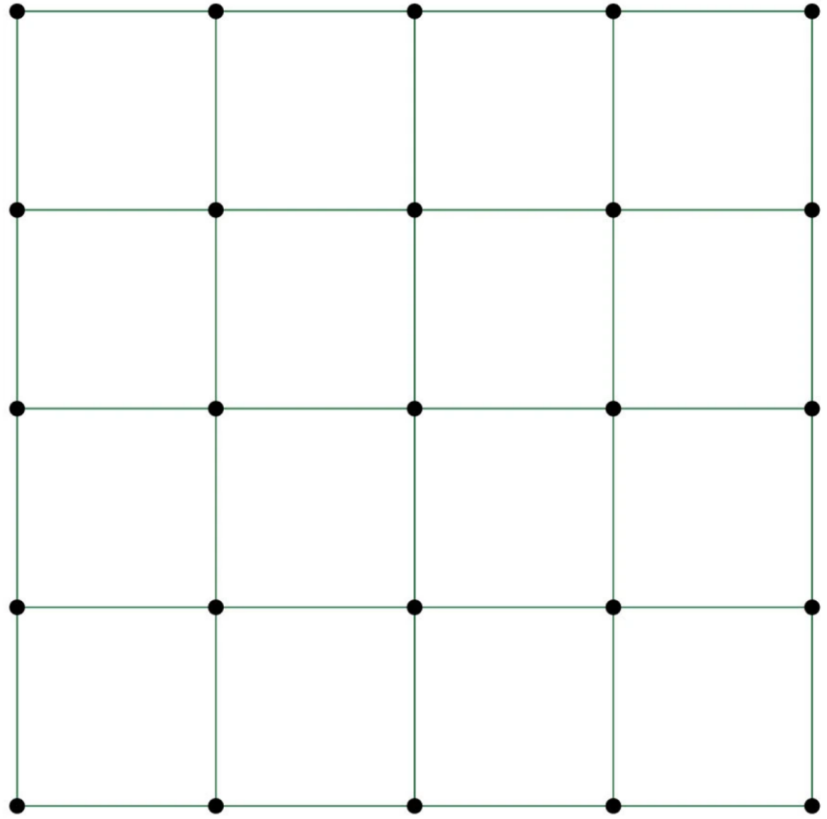
Basics of Ising model

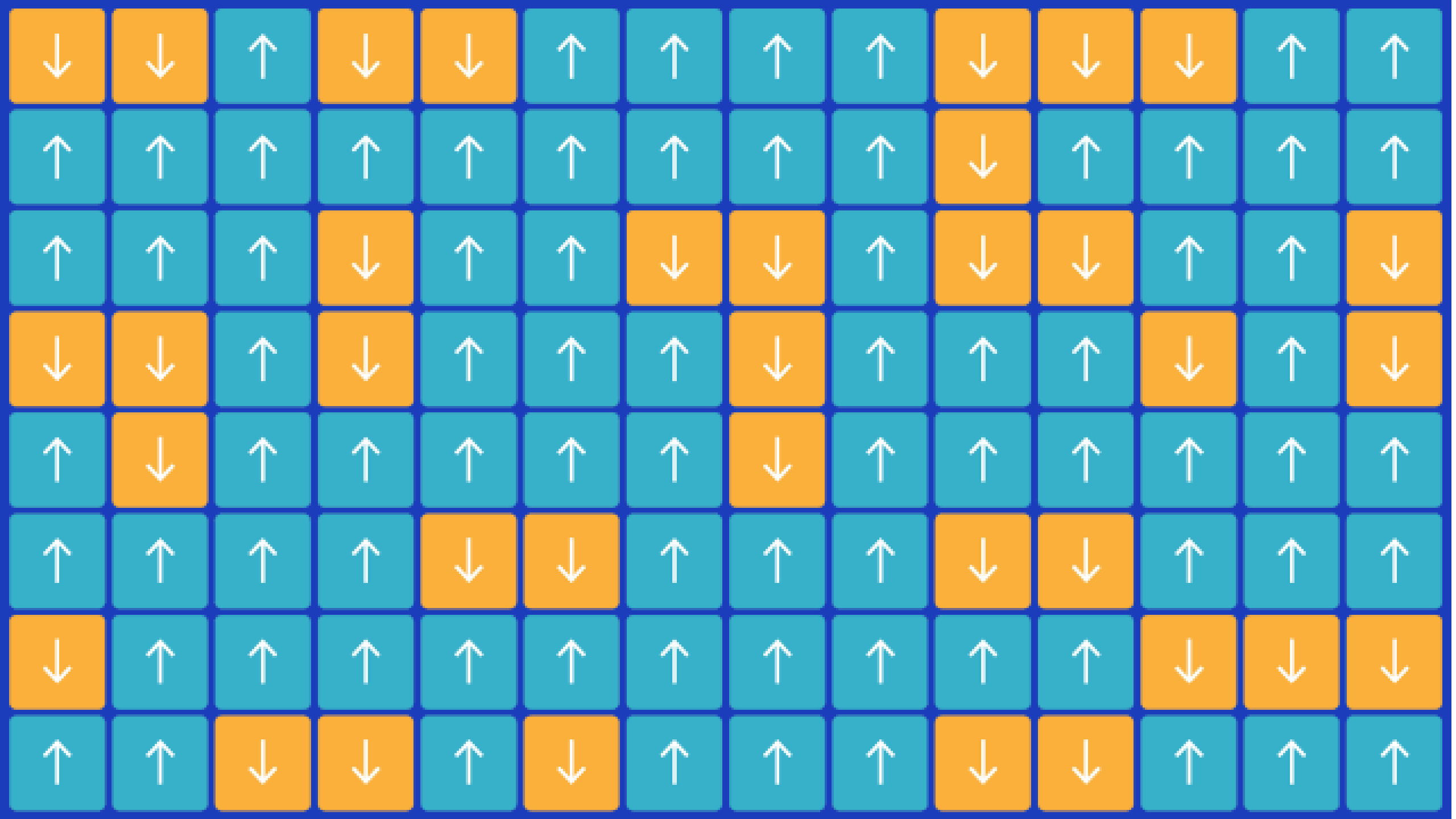


- **Spin Variables**:
- **Interactions**:
- **Energy Function (Hamiltonian)**:
- **Temperature Effect**:
- **External Field**:

$$H = -J \sum_{\langle i,j \rangle} S_i S_j - h \sum_i S_i$$

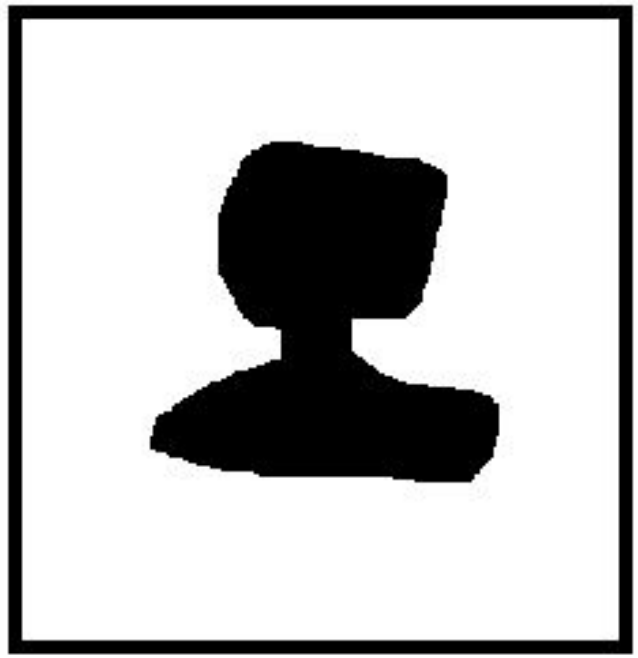
$$P(S) = \frac{e^{-H(S)/kT}}{Z}$$



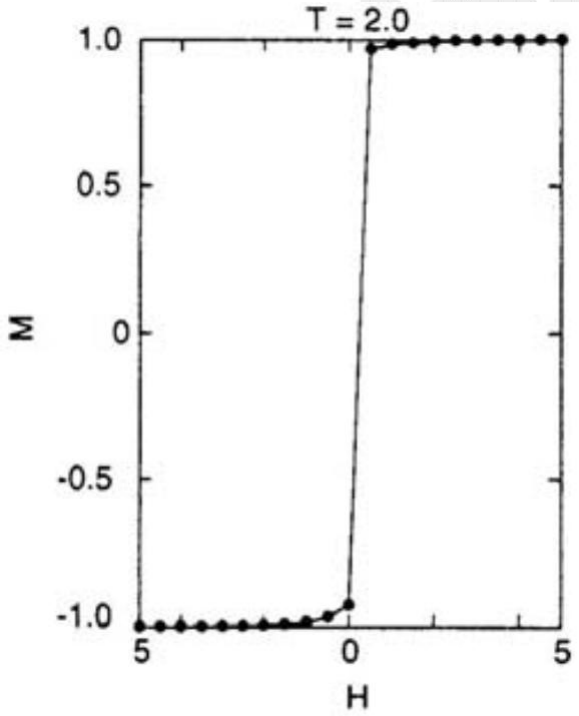


Monte Carlo Method

- Area computation
- Randomly drop dots
- Count number of dots in and out
- → area



Explaining magnetic phenomenon



ferromagnetic　　critical state　　paramagnetic

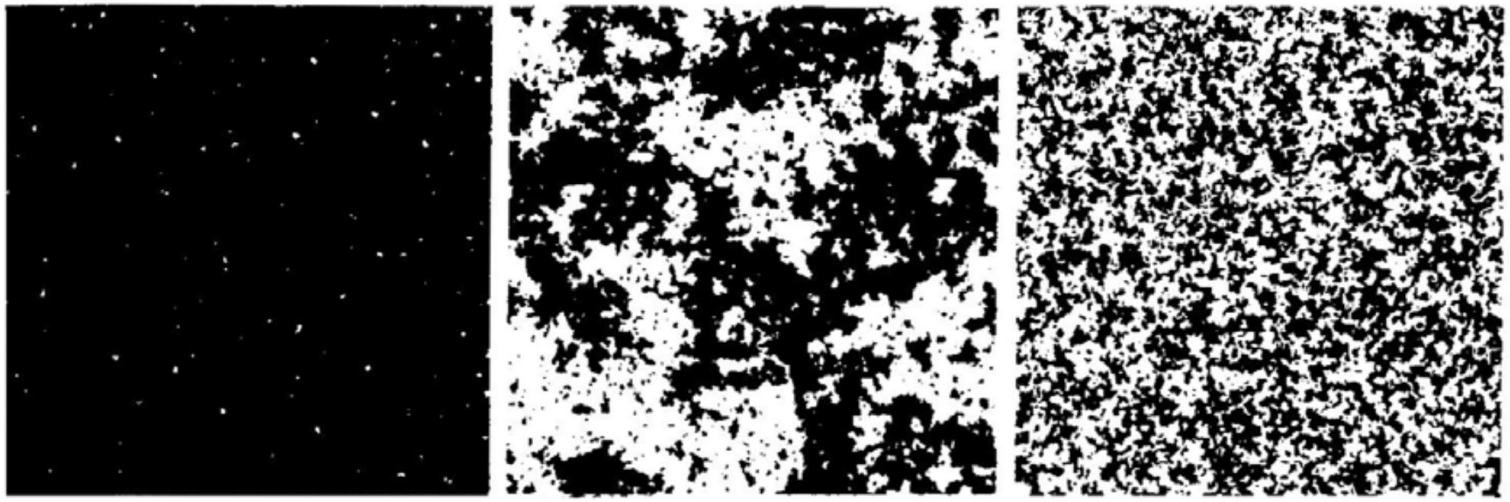


图2.3.1.7-3 模拟计算出的2维 Ising 点阵上的自旋构型：$T \ll T_c$(左)、$T \approx T_c$(中)、$T \gg T_c$(右)。

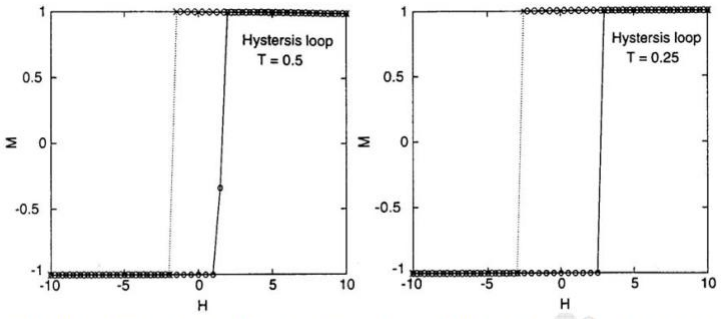
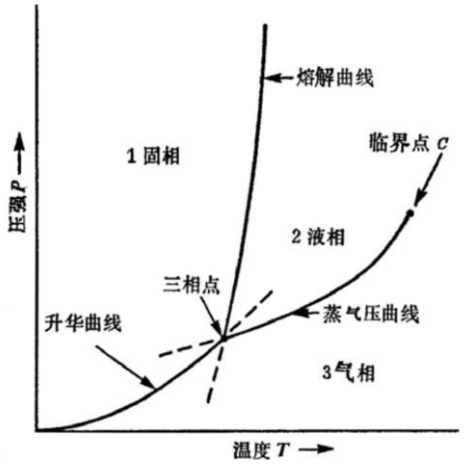


图2.3.1.8-4 温度在临界点之下时，2维10×10正方 Ising 点阵上的磁滞回线。实线是当磁场由负值开始增加时的曲线，虚线是当磁场由正值开始减小时的曲线。



Please note the keyword:
请记住关键词：
(discontinuity/singular)

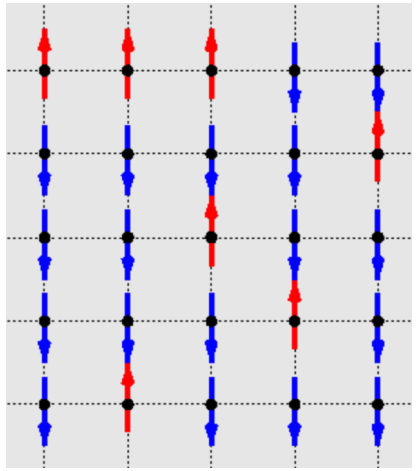
Phase shift
相变

Figure from: 《Computational Physics》(Zejun Ding)

Ising model
→ collective behavior, opinion dynamics, or influence in social networks.

- a village needs to hold a vote on an important issue, such as whether to support a new policy
- +1 represents support (e.g., casting a "yes" vote).
- -1 represents opposition (e.g., casting a "no" vote).
- Interactions:
  - Each villager's attitude is influenced not only by their own preferences (e.g., prior beliefs, information) but also by interactions with their neighbors (other villagers).
  - If most of their neighbors support the policy, the likelihood of the individual supporting the policy increases, and vice versa.
  - This "neighborhood influence" is described by the interaction strength J in the Ising model.
    - If J is positive, conformity among neighbors (whether supporting or opposing) is strengthened.
    - If J is negative, the opinions will counteract each other.
- Temperature Effect:
  - In the Ising model, temperature (T) represents the level of uncertainty or disorder in the system.
  - At high temperatures, individual behavior is more random and can be influenced by external factors (e.g., changing information or social events), leading to unstable voting outcomes.
  - At low temperatures, attitudes within the group tend to align, and one group (e.g., those in favor of the policy) may dominate, resulting in stable collective behavior.

Ising model is about…



Individual个体

Collective群体

System整体

Components部分

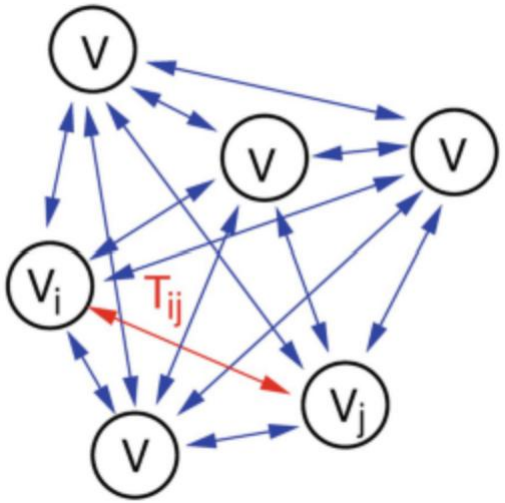
Order秩序

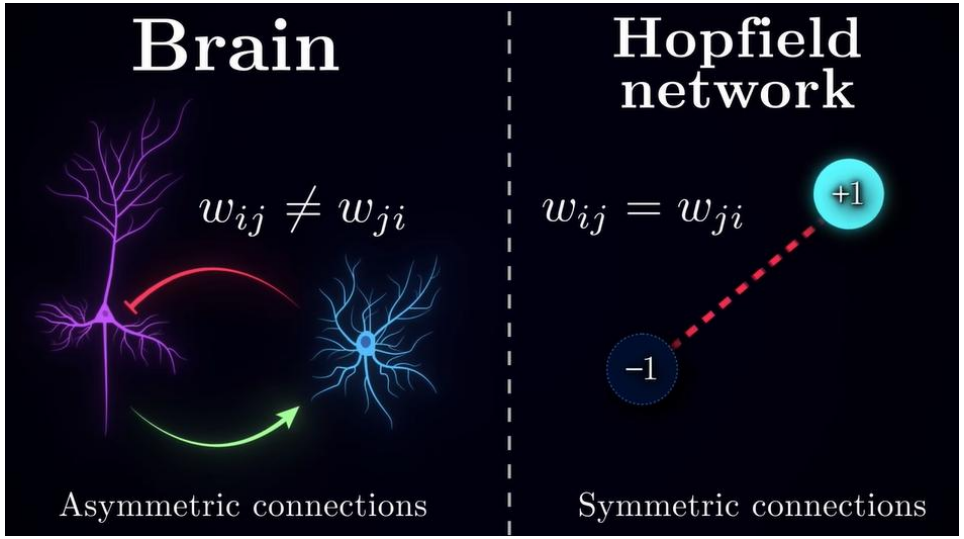
Disorder混沌

Ising model → Hopfield network

- **Neurons: +1, -1**
- **Interactions: w_{ij}**
  - **Not uniform**
  - **Trainable!**
  - Symmetric
- **Energy Function**

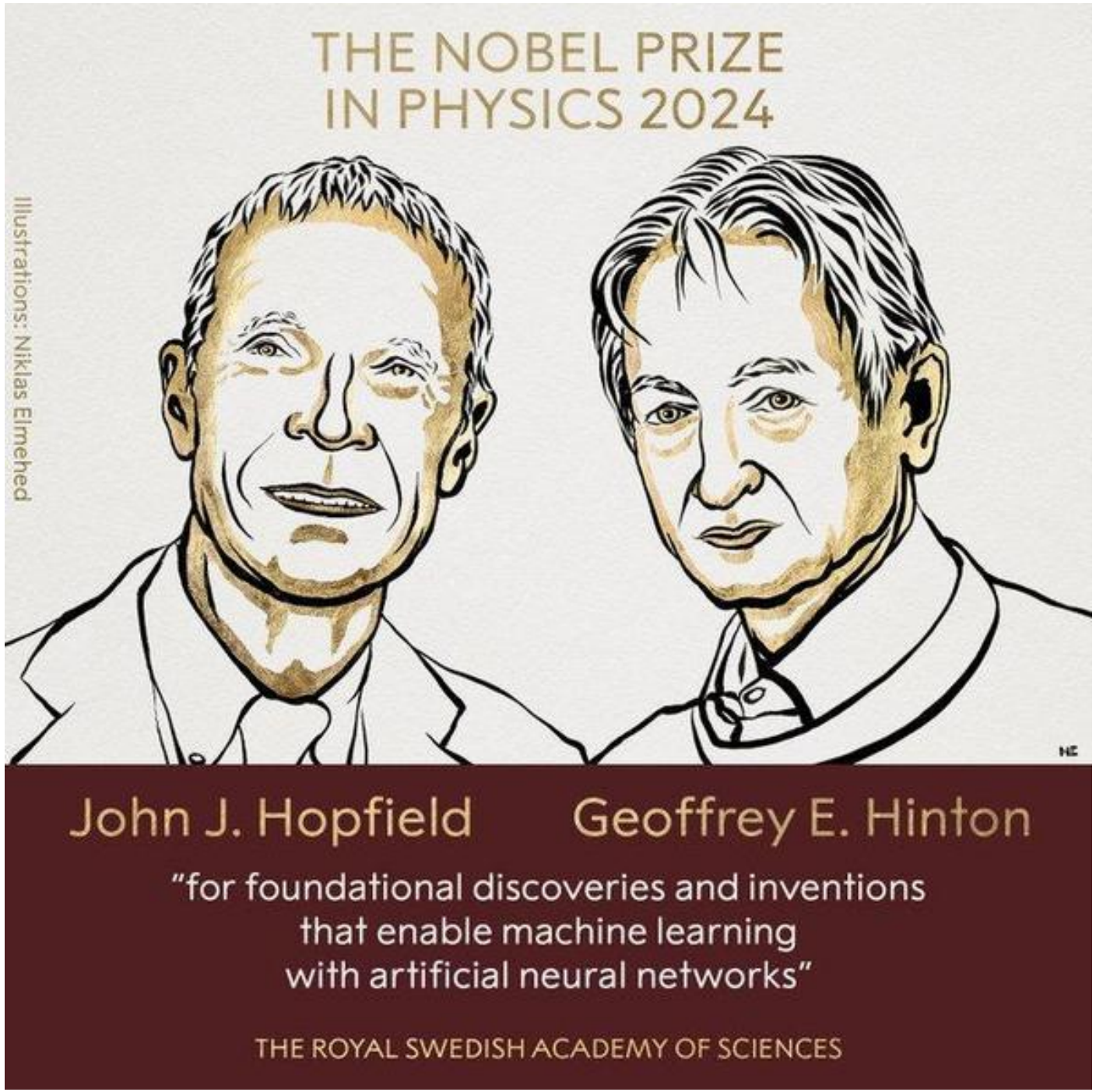
$$E = -\sum_{i<j} w_{ij} x_i x_j - \sum_i \theta_i x_i$$





- **Energy Minimization**:
  - The state update process aims to minimize the energy function, and the network will converge to a stable state that represents a stored pattern.

$$w_{ij} = \frac{1}{N} \sum_p x_i^{(p)} x_j^{(p)} \qquad \Delta w_{ij} = \eta \cdot s_i \cdot s_j \qquad x_i(t+1) = \text{sign}\left(\sum_j w_{ij} x_j(t) - \theta_i \right)$$

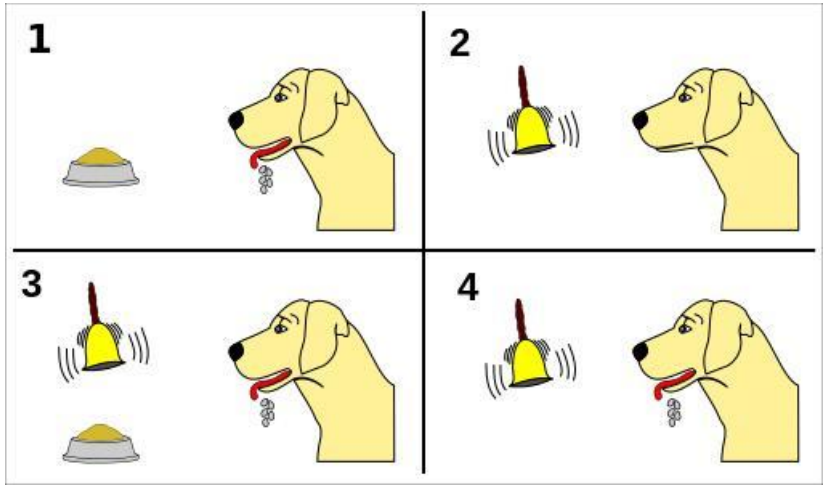

THE NOBEL PRIZE
IN PHYSICS 2024
Illustrations: Niklas Elmehed
John J. Hopfield
Geoffrey E. Hinton
"for foundational discoveries and inventions
that enable machine learning
with artificial neural networks"
THE ROYAL SWEDISH ACADEMY OF SCIENCES

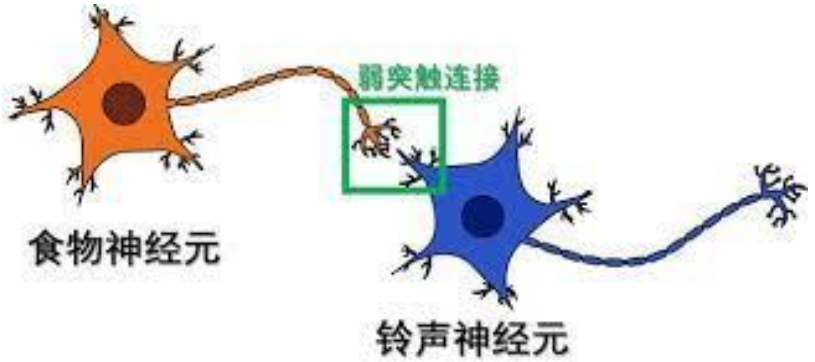
Applications of Hopfield Networks

- Pattern Recognition:
  - Hopfield networks can retrieve complete patterns from partial or noisy inputs. This makes them suitable for applications like image recognition and speech recognition.
- Memory Storage and Retrieval:
  - Hopfield networks are often used to simulate memory storage and retrieval in biological systems. When presented with a partial or noisy input pattern, the network can restore the full pattern based on its stored memories.
- Optimization Problems:
  - Hopfield networks can be used to solve combinatorial optimization problems, such as the traveling salesman problem, graph coloring, and other NP-complete problems. By constructing an appropriate energy function, the network can find the optimal solution in its lowest energy state.
- Adaptive Filtering:
  - Due to their ability to handle noise and partial information, Hopfield networks can be used as adaptive filters in applications like image denoising and image restoration.

Hebb's Rule 赫布法则

- **Fire together, wire together 同时激活、连接增强**
  - If two neurons are activated at the same time, the connection between them strengthens
- Biological relevance
  - Long-Term Potentiation (LTP)
    - a critical mechanism for learning and memory in the brain.
    - In the **hippocampus（海马体）**, a region crucial for learning and memory, LTP has been shown to occur when two neurons are activated together. When high-frequency stimulation is applied to one neuron, followed by stimulation of a second neuron, the synaptic strength between the two neurons is strengthened
  - Long-Term Depression (LTD)
    - LTD is typically induced by low-frequency stimulation, especially when two neurons' activities are not in synchrony. This weakening of the synaptic connection corresponds to Hebb's idea that if two neurons do not fire together, their connection will weaken.





Pavlov's conditioned reflex
巴甫洛夫条件反射

$$w_{ij} = \frac{1}{N} \sum_{p} x_i^{(p)} x_j^{(p)}$$

Two more messages

- Ising model: question → answer
  - Given: fixed energy function
  - what pattern will occur
- Hopfield network: answer → question
  - Given: certain pattern
  - What “energy function” (weight combination) is needed
  - “**learning**”
  - Error correction

- Same fundamental rule: energy minimization (temperature cooling) 能量最小化（降温）
  - The “learning and error correction mechanism that the nature bestowed to us” 大自然赋予的学习与纠错方法 (Yizhuang You)

假设：

- **网络规模**：有 $N = 3$ 个神经元。
- **存储模式**：存储一个模式 $\xi^\mu = [+1, -1, +1]$。

步骤：

1. **学习阶段**：

   根据Hebb规则计算权重：

   $$w_{ij} = \frac{1}{N}\xi_i^\mu \xi_j^\mu$$

   计算结果：

   - $w_{11} = 0$（自连接权重为0）
   - $w_{12} = \frac{1}{3}(+1) \times (-1) = -\frac{1}{3}$
   - $w_{13} = \frac{1}{3}(+1) \times (+1) = \frac{1}{3}$
   - $w_{21} = -\frac{1}{3}$
   - $w_{22} = 0$
   - $w_{23} = -\frac{1}{3}$
   - $w_{31} = \frac{1}{3}$
   - $w_{32} = -\frac{1}{3}$
   - $w_{33} = 0$
2. **输入受损模式**：

   输入模式 $s = [+1, +1, +1]$，与存储模式有一个位置不同。
3. **状态更新与错误校正**：
   - **更新神经元1**：

     $$h_1 = w_{12}s_2 + w_{13}s_3 = \left(-\frac{1}{3} \times +1\right) + \left(\frac{1}{3} \times +1\right) = 0$$

     s_1 仍为 $+1$。
   - **更新神经元2**：

     $$h_2 = w_{21}s_1 + w_{23}s_3 = \left(-\frac{1}{3} \times +1\right) + \left(-\frac{1}{3} \times +1\right) = -\frac{2}{3}$$

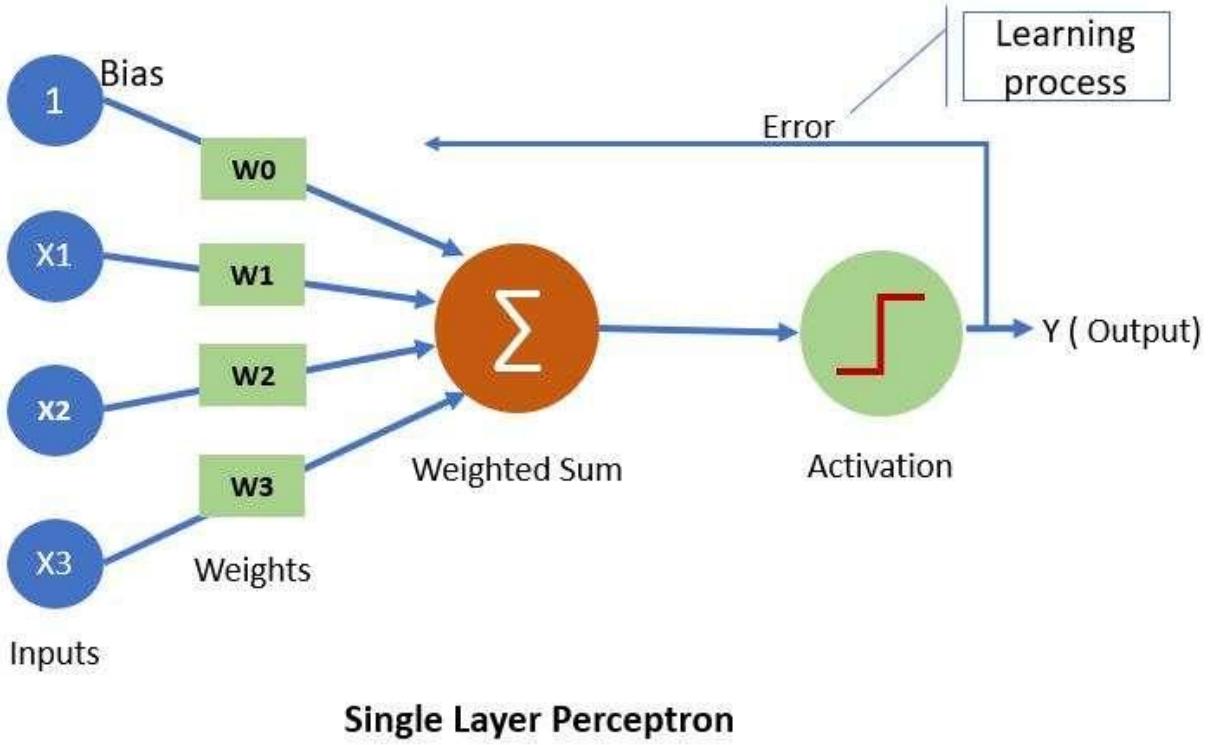
     $s_2 = \text{sgn}(h_2) = -1$。
   - **更新神经元3**：

     $$h_3 = w_{31}s_1 + w_{32}s_2 = \left(\frac{1}{3} \times +1\right) + \left(-\frac{1}{3} \times -1\right) = \frac{2}{3}$$

     s_3 仍为 $+1$。
   - 更新后，网络状态变为 $s = [+1, -1, +1]$，与存储的模式一致，实现了错误校正。

Perceptron感知机

- Structure
  - Inputs (x_1, x_2, ..., xn)
  - Weights (w_1, w_2, ..., wn)
  - Bias (b)
  - Activation Function
- Training:
  - Initialization
  - Prediction
  - Error Calculation
  - Weight Update

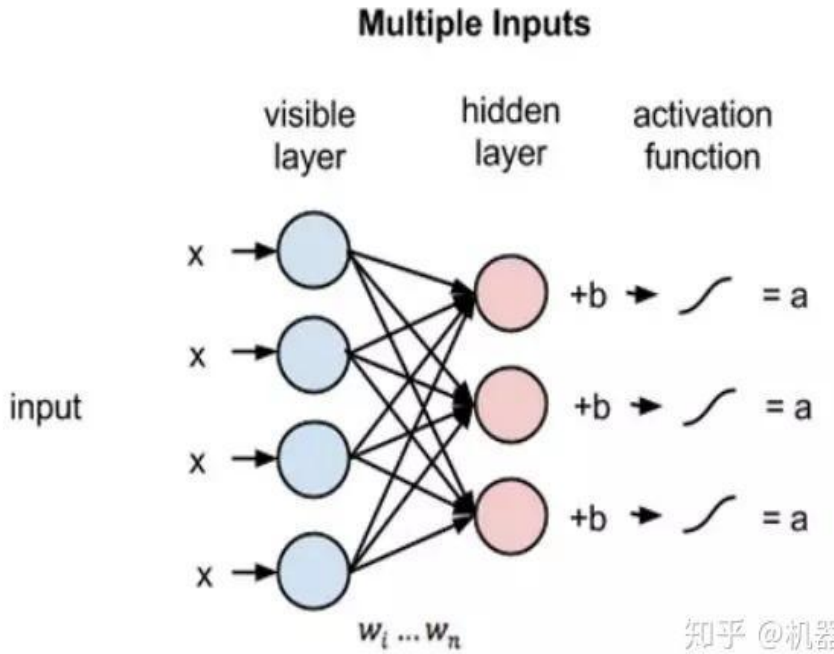


Single Layer Perceptron

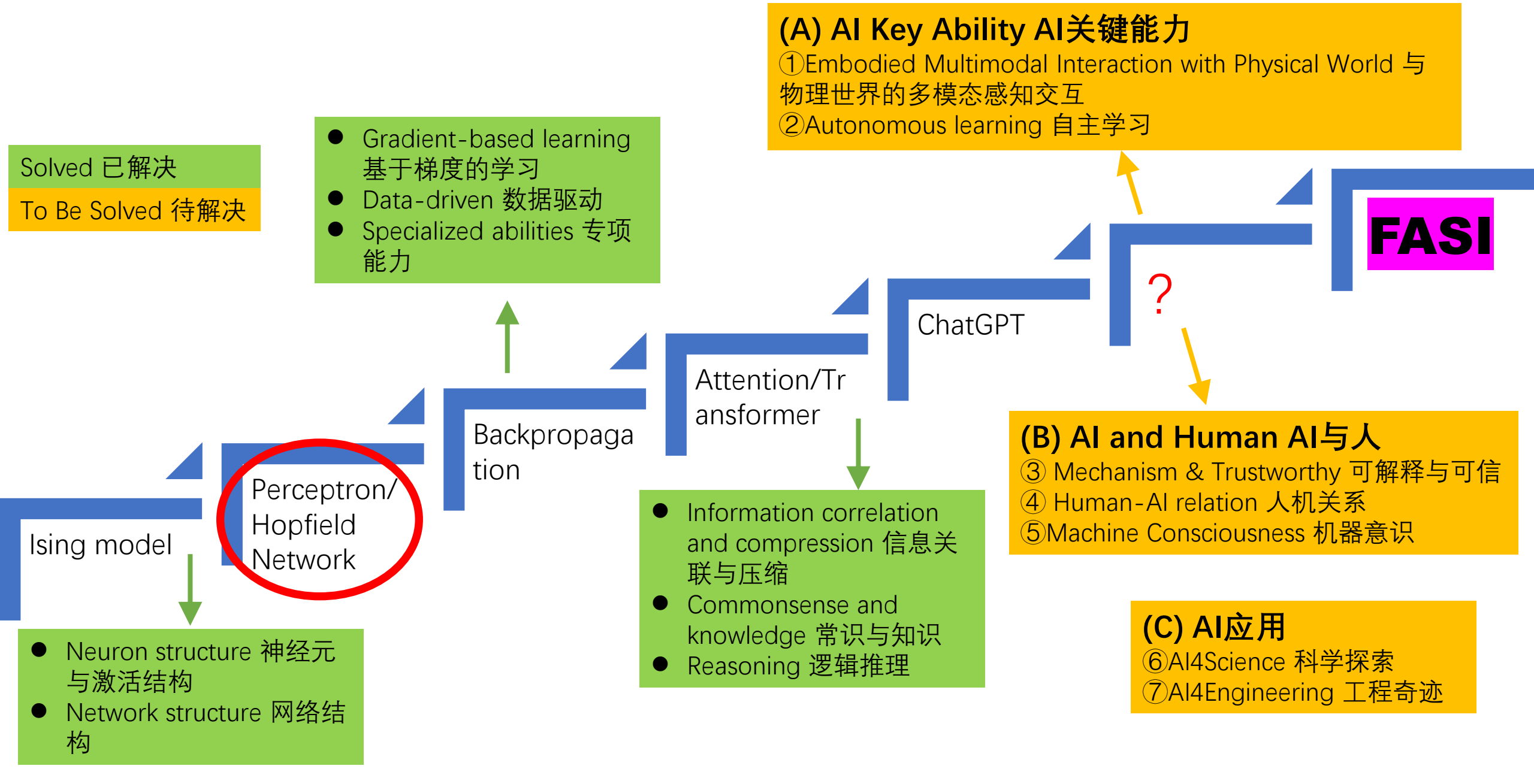
(Restricted) Boltzmann Machine （受限）玻尔兹曼机

- Structure
  - Nodes (Neurons)
  - Connection Weights
  - Energy Function
  - Boltzmann Distribution
- Node State Updates:
  - Each node's state is updated based on the following probability
  - σ is the Sigmoid function, which converts the weighted sum into an activation probability.
- Introducing probabilities
  - RBM 将神经元的状态从确定性变为随机性，即神经元的激活状态遵循某种概率分布。这意味着神经元以一定的概率被激活或抑制，而不是绝对地取决于输入。RBM transforms the state of neurons from deterministic to stochastic, meaning the activation state of neurons follows a certain probability distribution. This implies that neurons are activated or inhibited with a certain probability, rather than being absolutely determined by the input.
- 平衡态统计物理 → 非平衡态统计物理 From equilibrium statistical physics → non-equilibrium statistical physics:
  - 不需要达到热平衡，其实可以提供学习信号 It does not require reaching thermal equilibrium, and can actually provide learning signals.
  - 即使跟随不完全正确的信号，也可以优化至同样正确的解 Even by following partially correct signals, it can still optimize to the same correct solution.

$$P(s_i = 1) = \sigma\left(\sum_j w_{ij} s_j + b_i\right)$$



Pathway to Full Artificial Super Intelligence



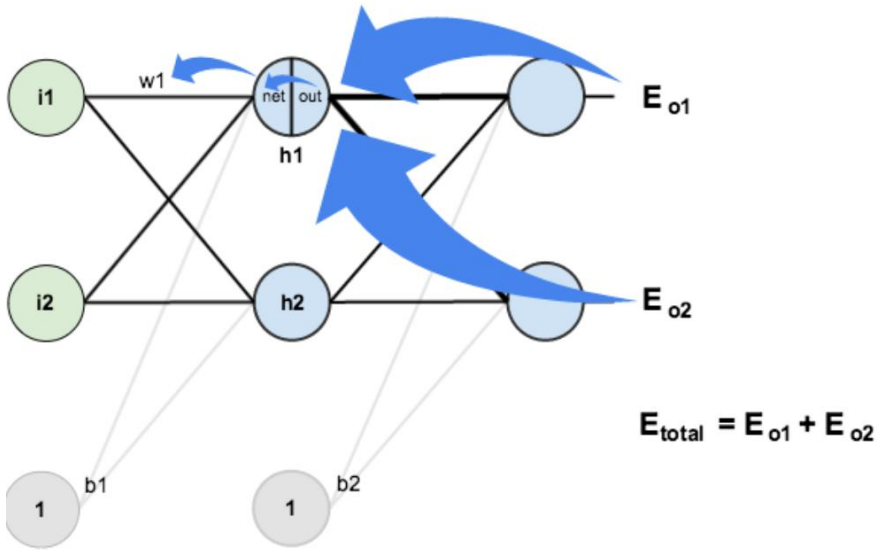
Backpropagation 反向传播

- More layers/DNN 多层/深度神经网络训练
- Loss minimization task → Gradient calculation
- Chain rule

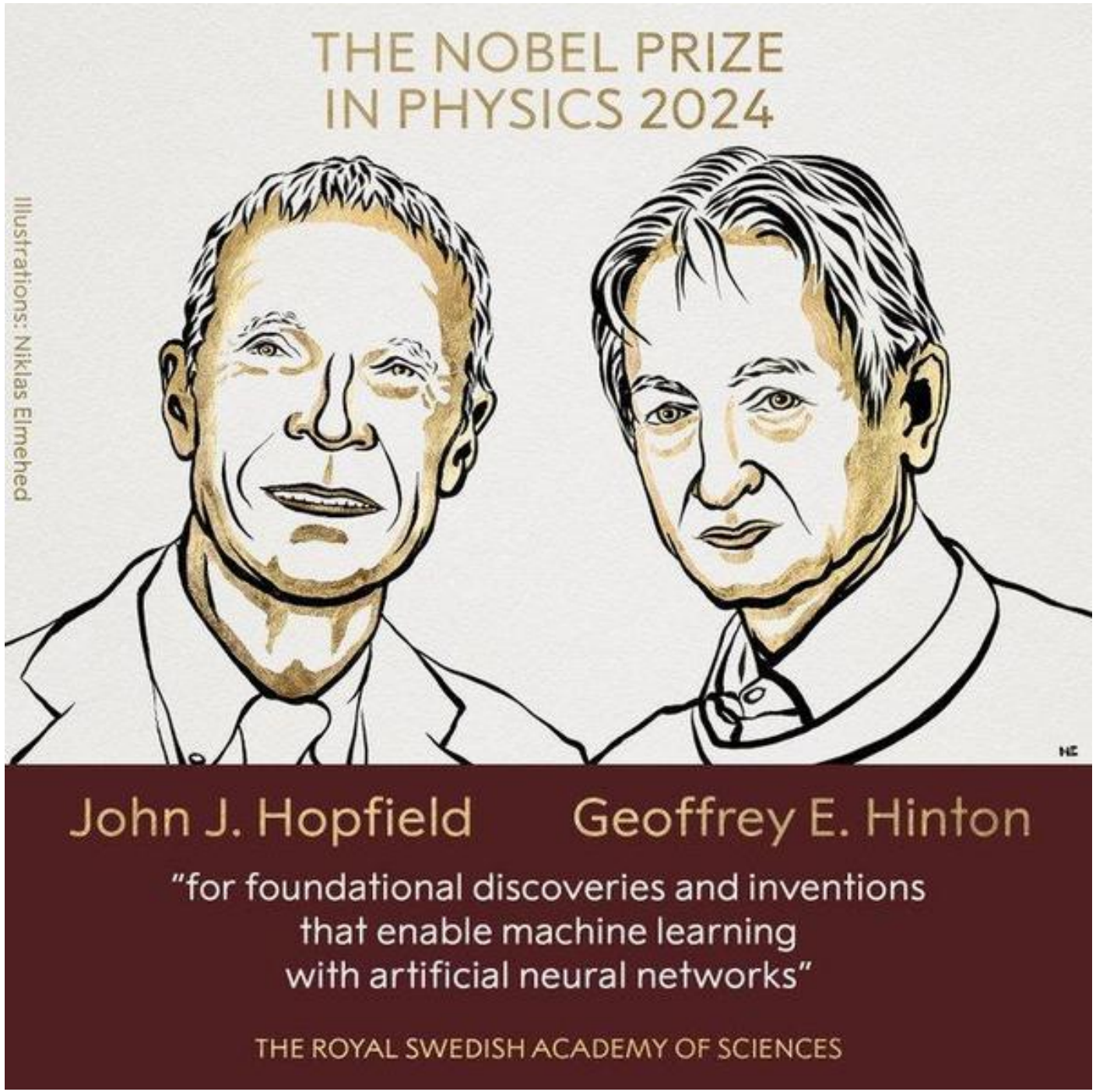
$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\downarrow$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



1. **Forward Propagation**:
   - Pass the input data from the input layer through each layer to the output layer, calculating the network's predicted output. For each layer, the output of the nodes can be expressed as:
     $$o^{(l)} = f(W^{(l)} \cdot o^{(l-1)} + b^{(l)})$$
     where:
     - $o^{(l)}$ represents the output of the l-th layer,
     - $W^{(l)}$ is the weight matrix of that layer,
     - $b^{(l)}$ is the bias vector,
     - f is the activation function (e.g., Sigmoid, ReLU).
2. **Calculate Error**:
   - For each training sample, compute the error between the network output and the target output. Common error metrics are **Mean Squared Error (MSE)** or **Cross-Entropy Error**.
     $$E = \frac{1}{2} \sum (y_{\text{pred}} - y_{\text{true}})^2$$
     where y_{pred} is the predicted output, and y_{true} is the actual label.
3. **Backpropagate the Error**:
   - Starting from the output layer, calculate the gradient for each neuron in each layer using the **chain rule**. For a loss function L and output $o^{(L)}$ of the last layer, the gradient of the weights $W^{(L)}$ at the last layer is:
     $$\frac{\partial L}{\partial W^{(L)}} = \delta^{(L)} \cdot o^{(L-1)^T}$$
     where $\delta^{(L)} = \frac{\partial L}{\partial o^{(L)}} \cdot f'(z^{(L)})$ is the error term for the L-th layer.
4. **Update Weights**:
   - Update weights using gradient descent. For each weight $w_{ij}^{(l)}$, the update rule is:
     $$w_{ij}^{(l)} = w_{ij}^{(l)} - \eta \cdot \frac{\partial L}{\partial w_{ij}^{(l)}}$$
     ↓
     where η is the learning rate, which controls the step size of each update.


THE NOBEL PRIZE
IN PHYSICS 2024
Illustrations: Niklas Elmehed
John J. Hopfield
Geoffrey E. Hinton
"for foundational discoveries and inventions
that enable machine learning
with artificial neural networks"
THE ROYAL SWEDISH ACADEMY OF SCIENCES

Face Recognition: Where is Zhang Zihan? | 人脸识别：张子翰在哪里?

山西吕梁失踪儿童，6岁

Missing Child, 6 years old

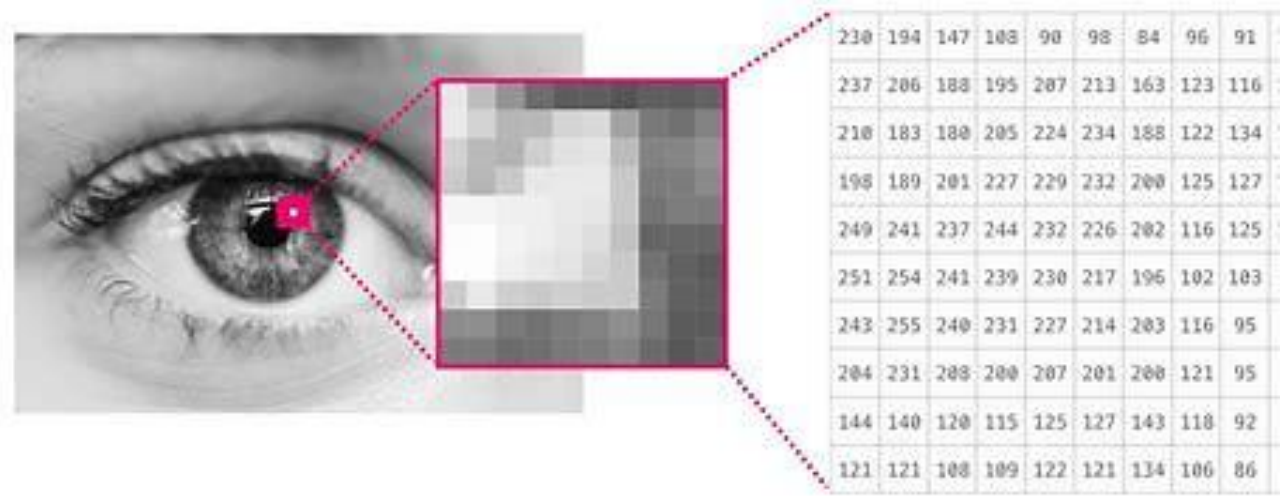
Massive face datasets,
how does the algorithm determine if it's Zhang Zihan?
海量人脸，算法如何判断是否张子翰？

输入：
人脸图片
Input: face imge

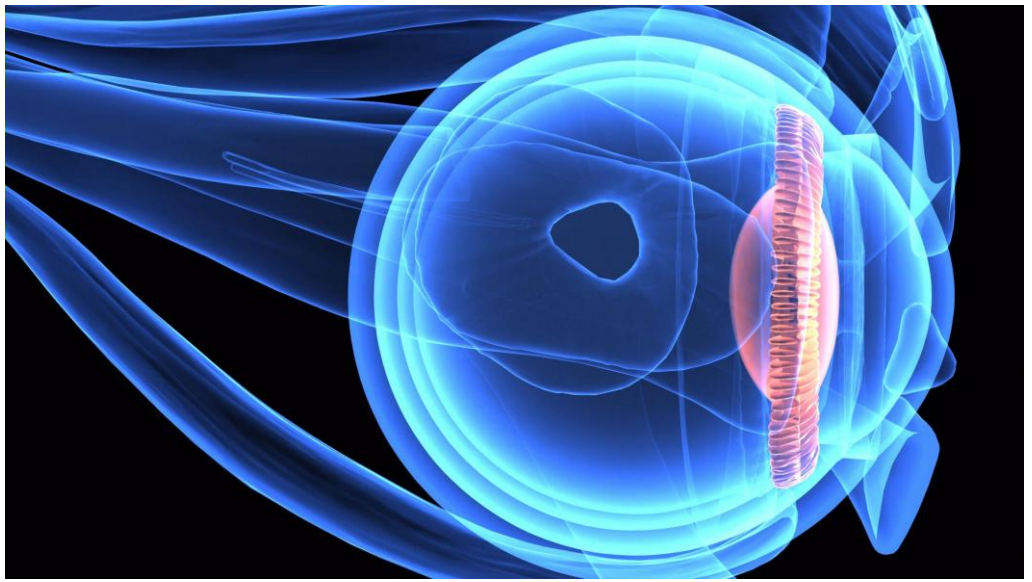
人脸识别算法
Face recognition algorithm

输出：
是否张子翰?
Output:
yes or no?
(1/0)
0: 99.2%
1: 0.8%

Face = array of numerical values | 人脸=很多数字



- Image: matrix of pixel data | 图片：很多像素点数据



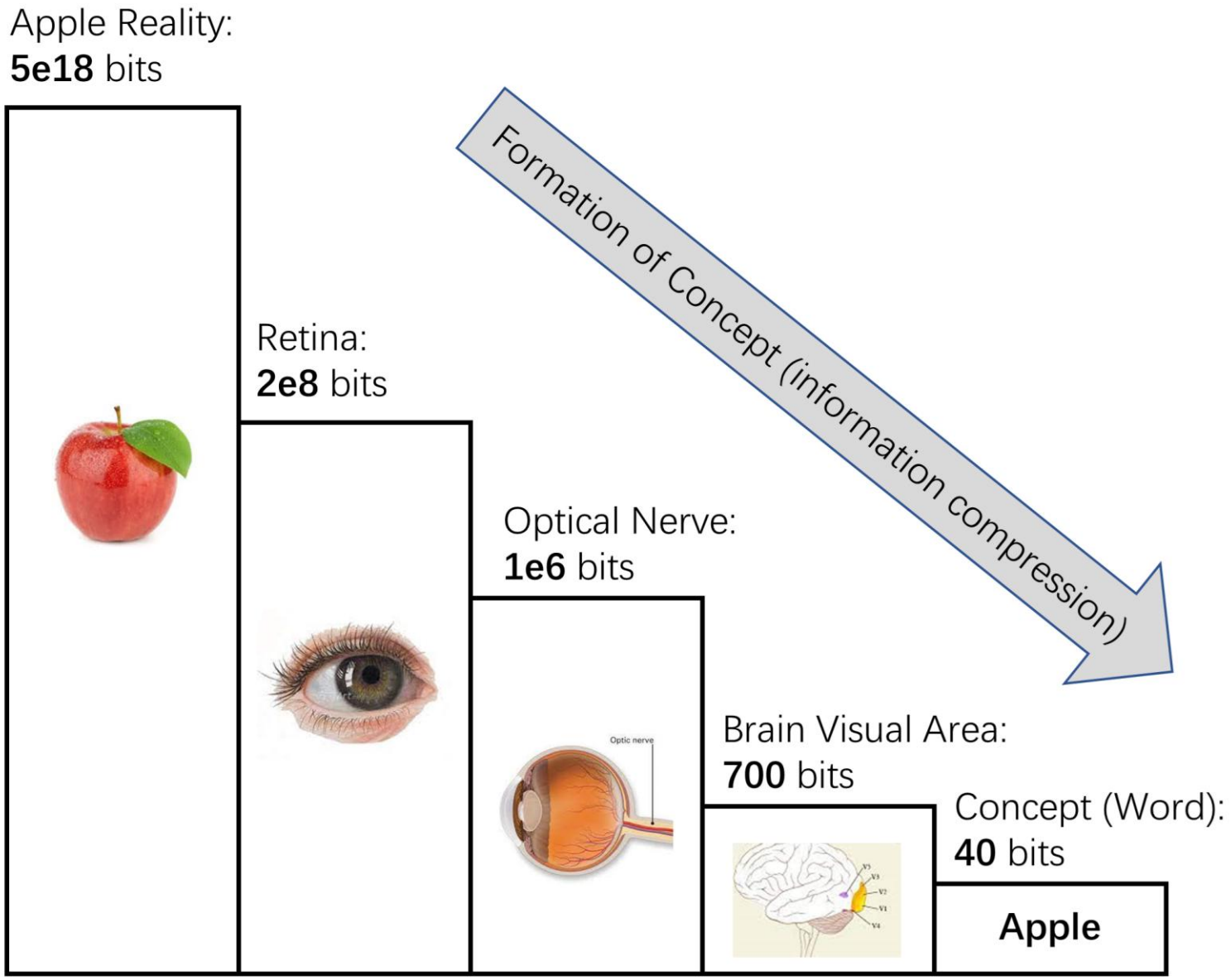
- Retinal imaging: photoelectric signals from photoreceptor cells | 视网膜成像：很多感光细胞的光电信号

Can we directly compare these values? | 我们可以直接比较这些数字吗？

- **同一个人=**
很多不同光线/角度/表情
Same person, many light conditions, angles, emotions

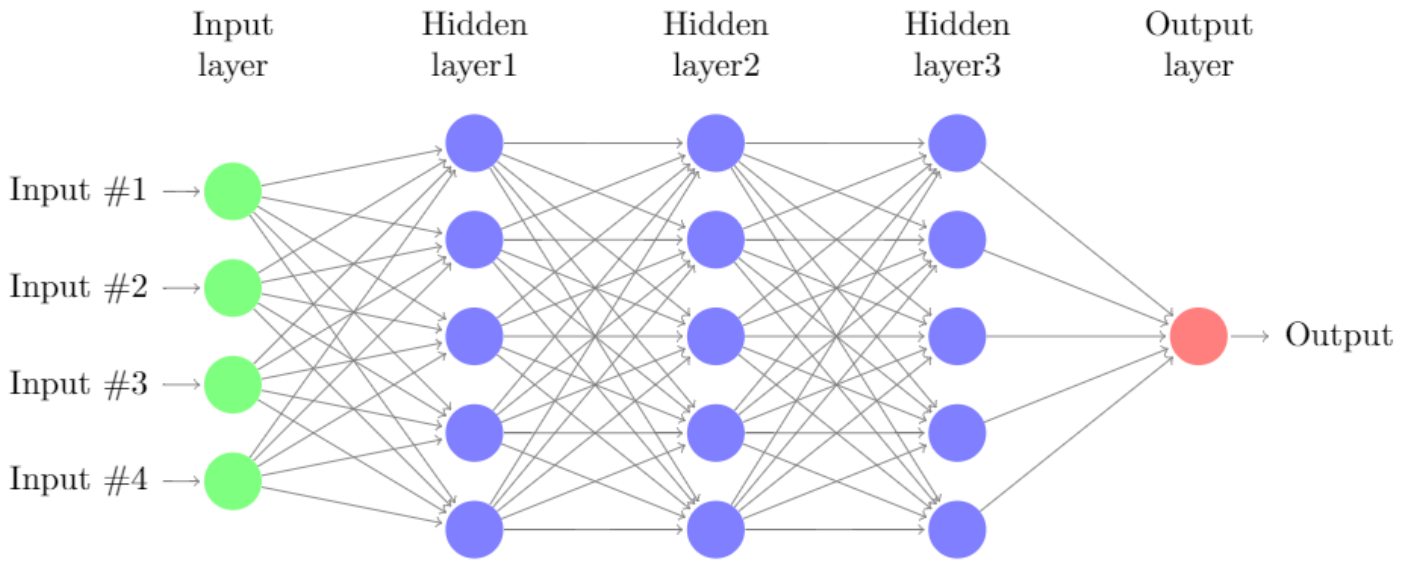
Humans compress information and identify **features**

人类压缩信息并识别“**特征**”

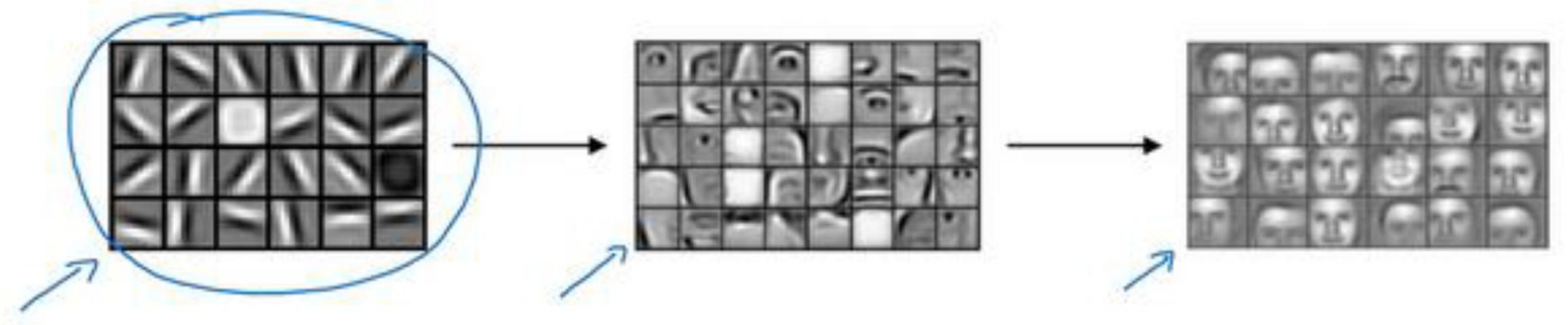


How do we extract hierarchical features at different granularities? 我们怎样提取不同层次的特征?

- A multi-layer neural network用一个多“层”神经网络



Algorithm extracts and computes "features" | 算法提取并计算“特征”



第1层：细节特征
边缘、亮暗、方向……
Layer 1: details

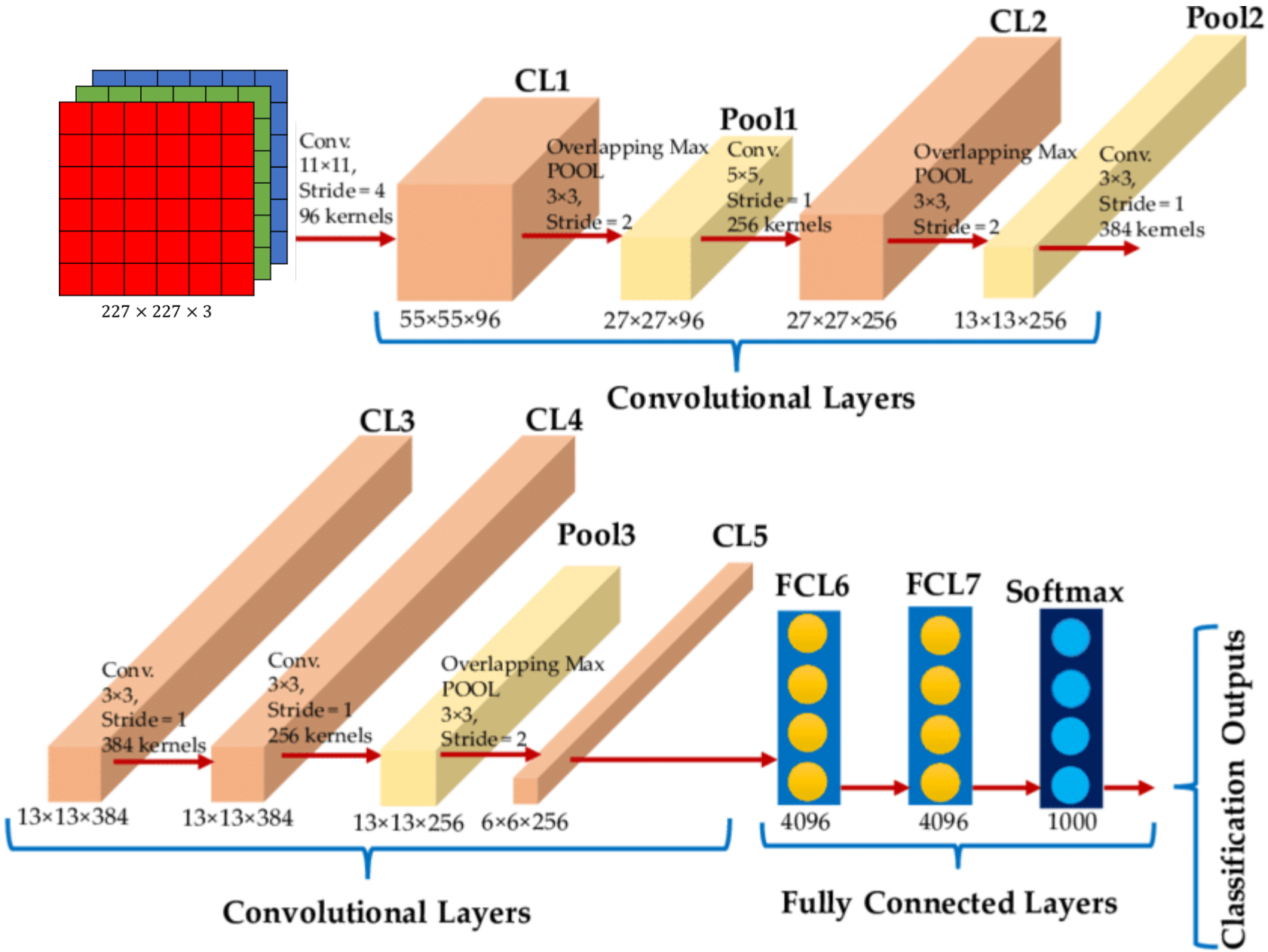
第2层：五官特征
眼、鼻、嘴……
Layer 2: eye nose mouth

第3层：全脸特征
脸型、五官位置……
Layer 3: face shape

不同“层”计算不同“层次”的特征
Difference network layer computes features at different granularity/aspect

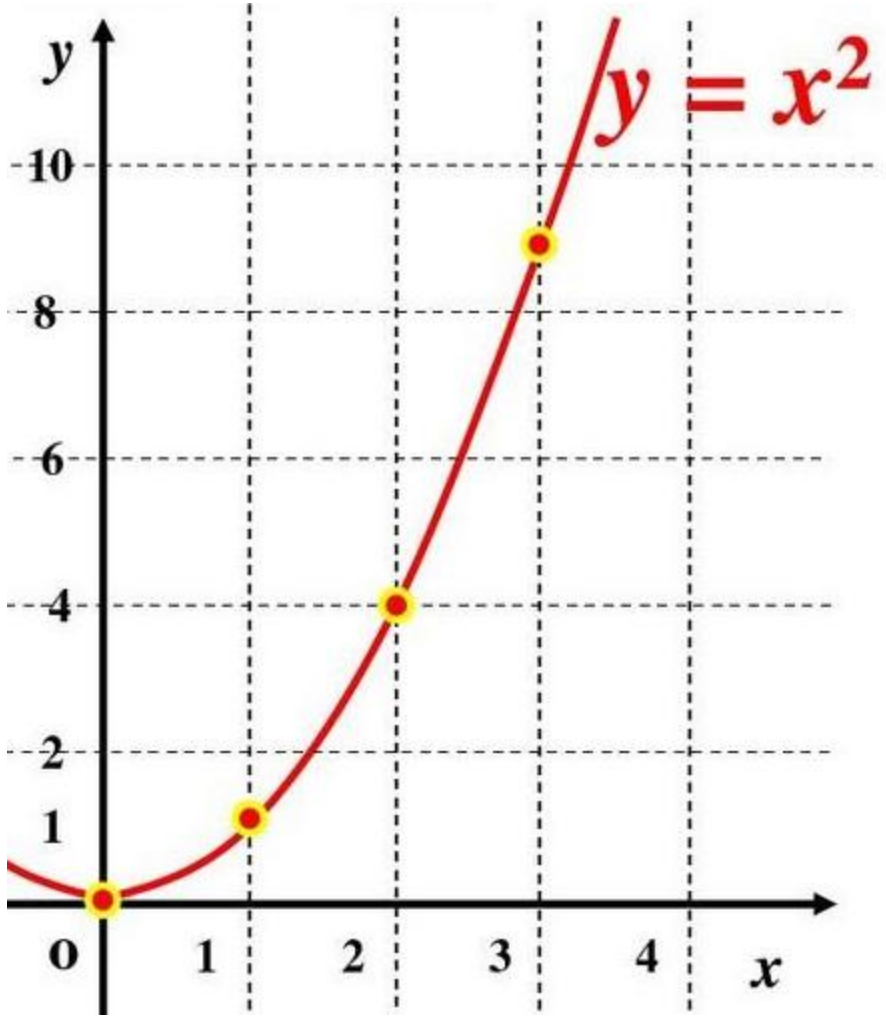
A typical real-world network: Alexnet

- 11层 11 layers
- 6000万个参数的函数 60 million parameters
- 80万个神经元（“小函数”） 0.8 million neurons (“little functions”)



What is a function:
input → output
mapping 什么是函数：输入→输出的映射规则

例：
输入2，给出4
input 2, output 4
输入3，给出9
input 3, output 9



What is a function with numerous parameters:
什么是含有很多“参数”的函数:

y=

1
x
$\frac{1}{2}\left(3x^2-1\right)$
$\frac{1}{2}\left(5x^3-3x\right)$
$\frac{1}{8}\left(35x^4-30x^2+3\right)$
$\frac{1}{8}\left(63x^5-70x^3+15x\right)$
$\frac{1}{16}\left(231x^6-315x^4+105x^2-5\right)$
$\frac{1}{16}\left(429x^7-693x^5+315x^3-35x\right)$
$\frac{1}{128}\left(6435x^8-12012x^6+6930x^4-1260x^2+35\right)$
$\frac{1}{128}\left(12155x^9-25740x^7+18018x^5-4620x^3+315x\right)$
$\frac{1}{256}\left(46189x^{10}-109395x^8+90090x^6-30030x^4+3465x^2-63\right)$

Inference function determines Zhang Zihan identification: 推断函数给出是否张子翰的判定：

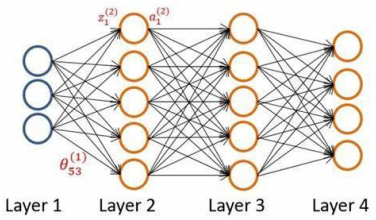
输入：
input

人脸识别算法
（推断函数）
Face recognition algorithm
Inference function

输出：
不是张子翰
Output:
Not ZZH
(1/**0**)
0: 99.2%
1: 0.8%

Mathematical framework: inference function

How do we obtain the inference function? 如何获取推断函数?



Input输入

Model模型
(推断函数/CNN卷积神经网络)

Output输出

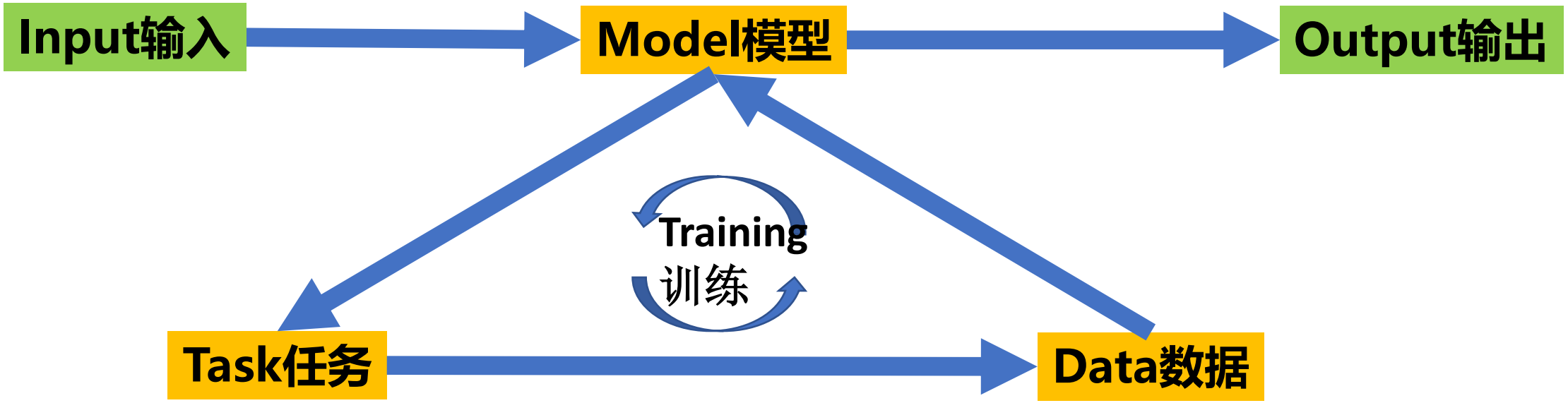
Training
训练

Task任务
(Loss Function 损失函数)

Data数据
(Supervision监督)

Data, Model, Task

Deep Learning 深度学习模型:
- CNN卷积神经网络
- RNN循环神经网络
- LSTM长短时记忆网络
- ……

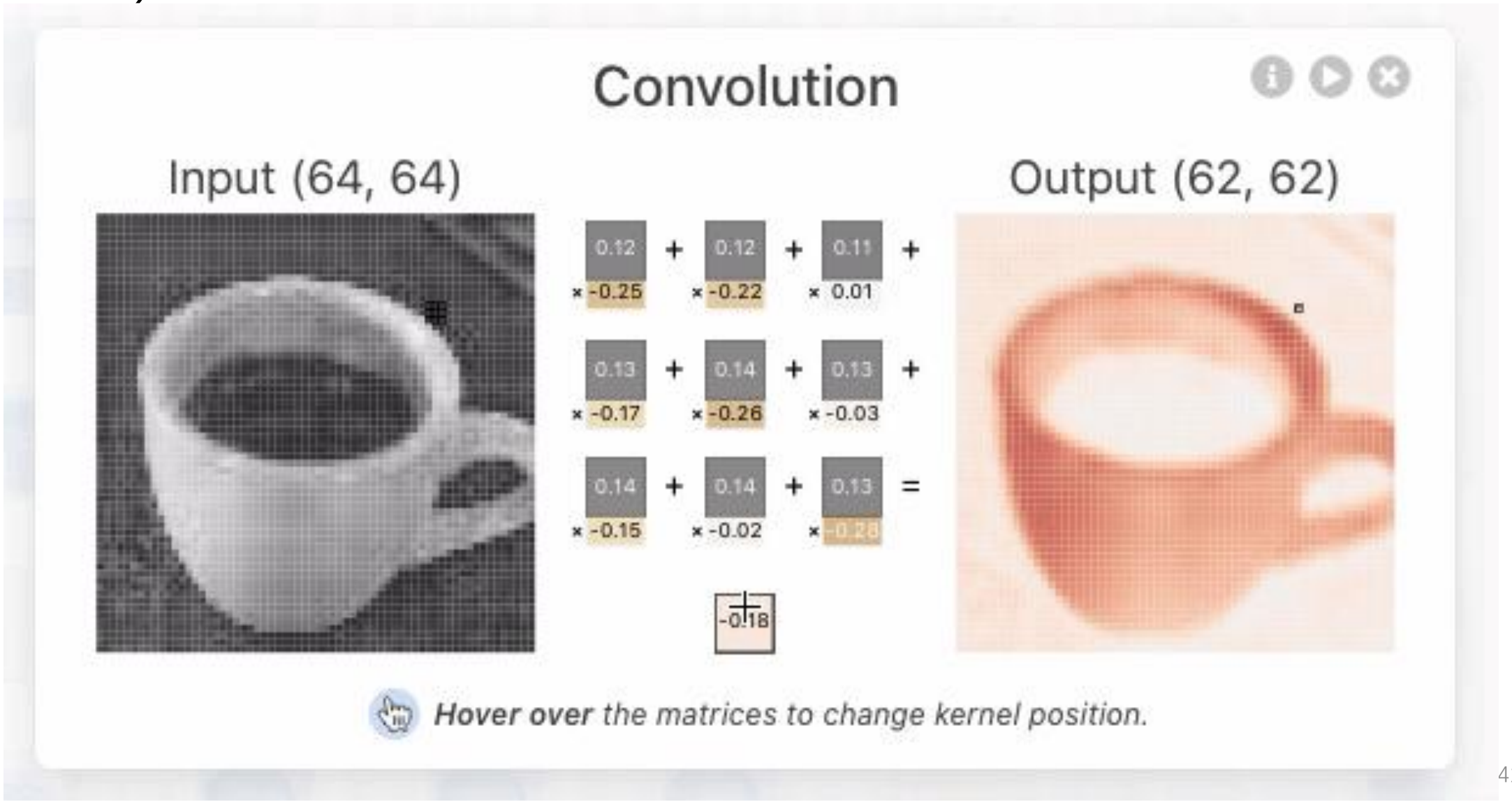


Specific Task 专项任务:
- face recognition 人脸识别
- Speech recognition 语音识别
- object detection 目标识别
- Machine translation 机器翻译
- ……

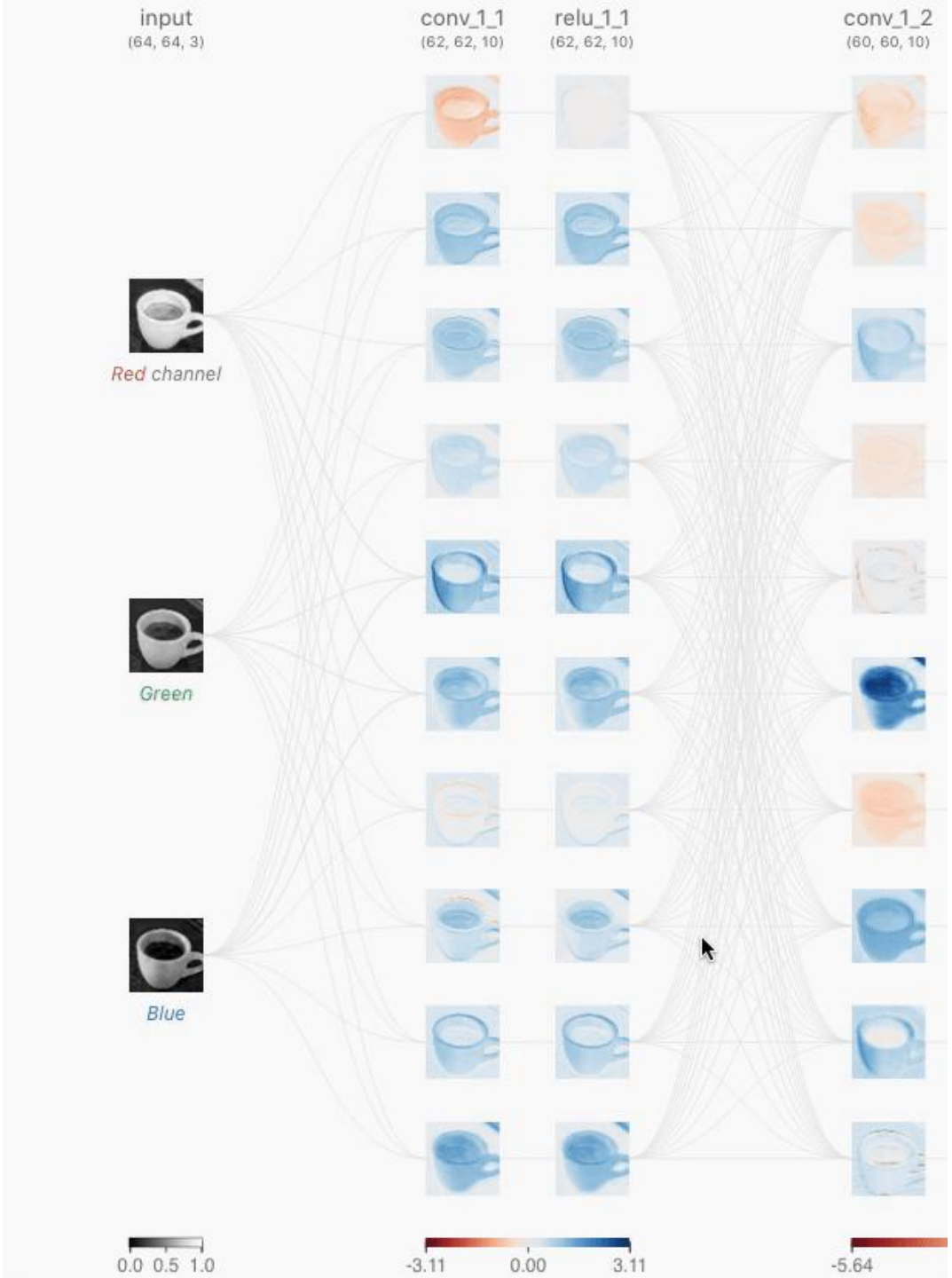
Annotated data标注数据:
- Pictures 图片
- Audio stream 语音
- NL text 自然语言文本
- ……

Tricks例1：卷积Convolution

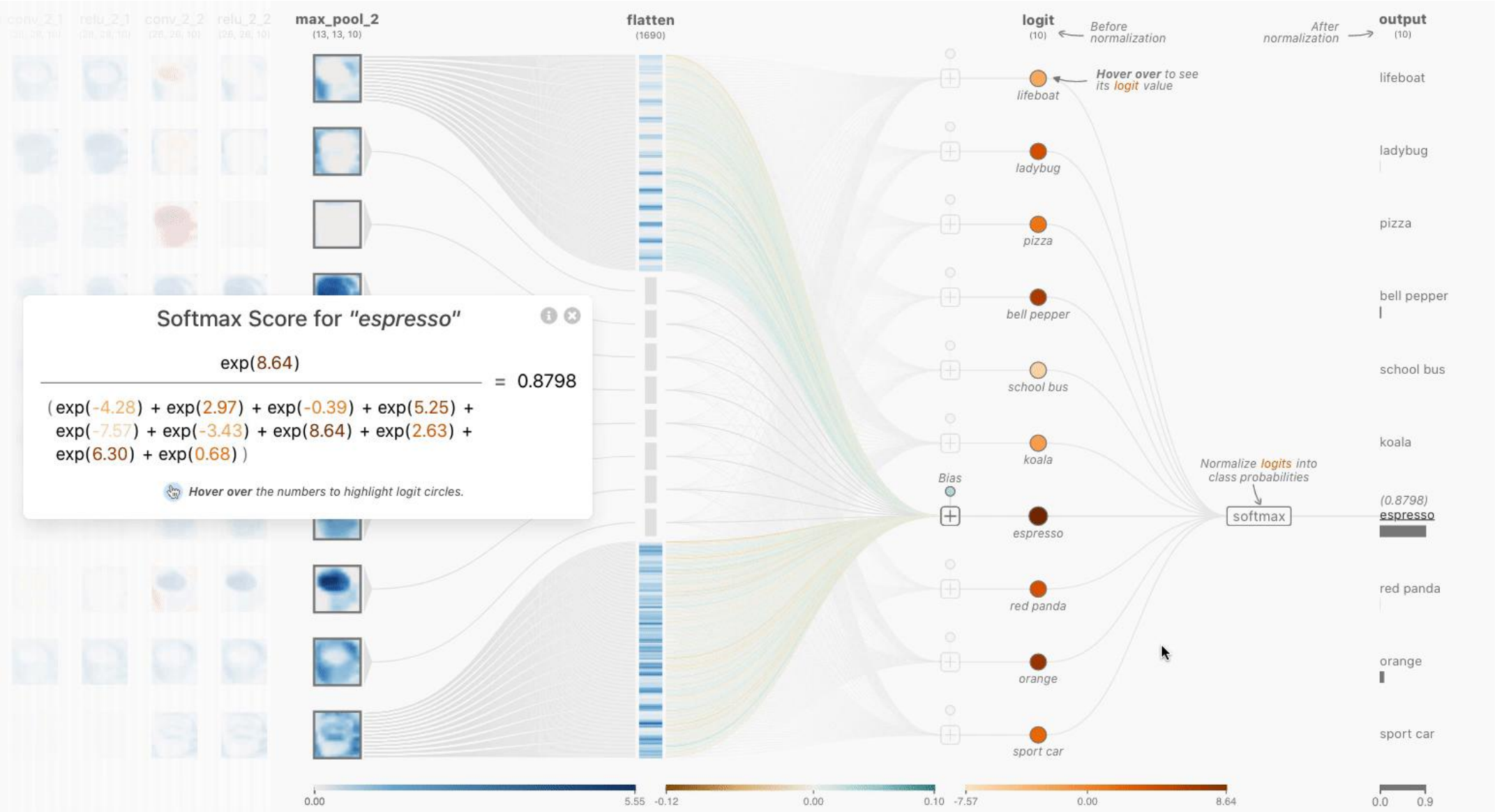
（用“卷积核”抓取“特征”extract “features” using “convolution kernels”）



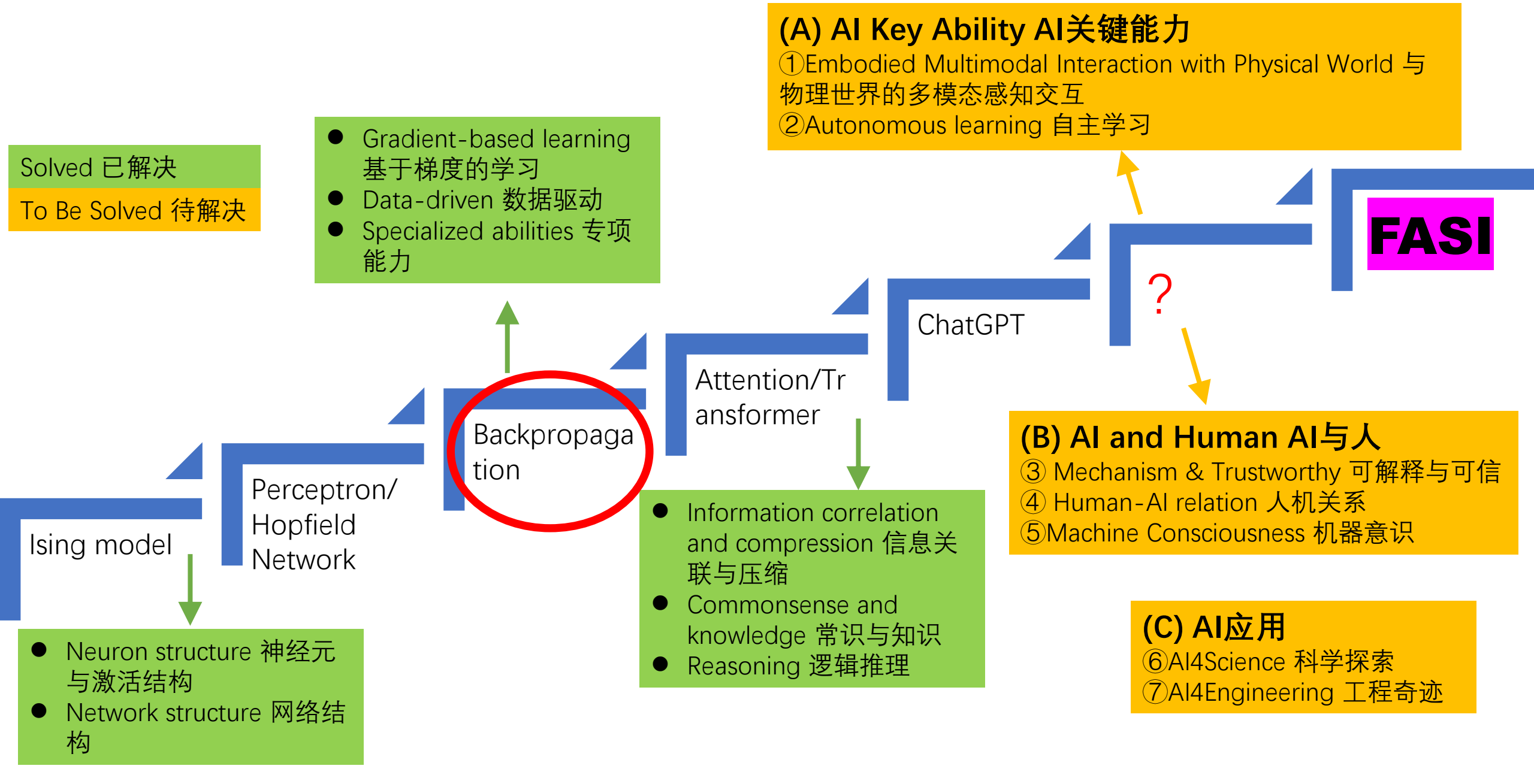
Tricks例2：multiple layers（不同层-不同卷积核-不同特征 different layer → different kernel → different features）



Trick例3：特征吻合度 → 分类概率
Feature Matching Degree → Classification Probability



Pathway to Full Artificial Super Intelligence



Attention & Transformer 注意力与Transformer

- Human's attention
  - Top down
  - Bottom up
- Correlation among components
  - Global Attention
  - Local Attention
  - Self-Attention

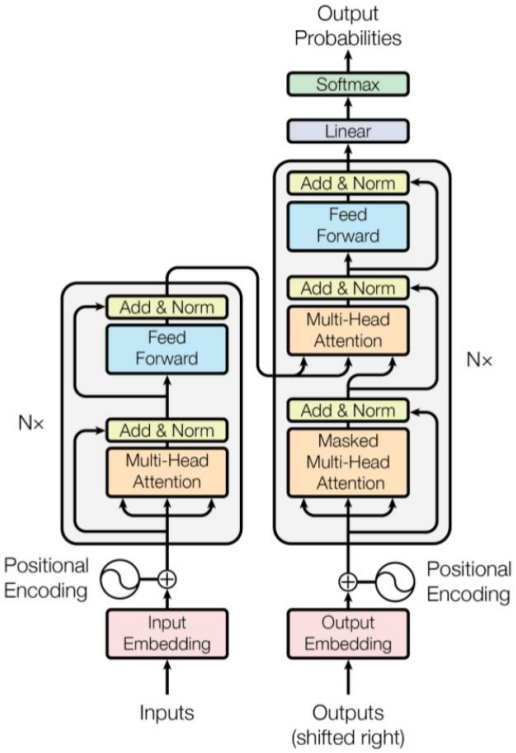
1. **Calculate Attention Weights**:
   - For each decoding step, the model calculates the relevance of each position in the input sequence to the current decoding position. This is typically done using a **similarity scoring function**, such as **dot product** or a **learnable feed-forward neural network**.
   - For example, for an input vector h_i and the current decoding vector s_t, the attentic weight $a_{t,i}$ is calculated as:

$$a_{t,i} = \frac{\exp(\text{score}(s_t, h_i))}{\sum_j \exp(\text{score}(s_t, h_j))}$$

   where $\text{score}(s_t, h_i)$ measures the relevance between the input and the output.
2. **Compute Weighted Context Vector**:
   - Apply the attention weights to the input vectors to obtain a weighted context vector, representing the input information the decoder focuses on at the current step:

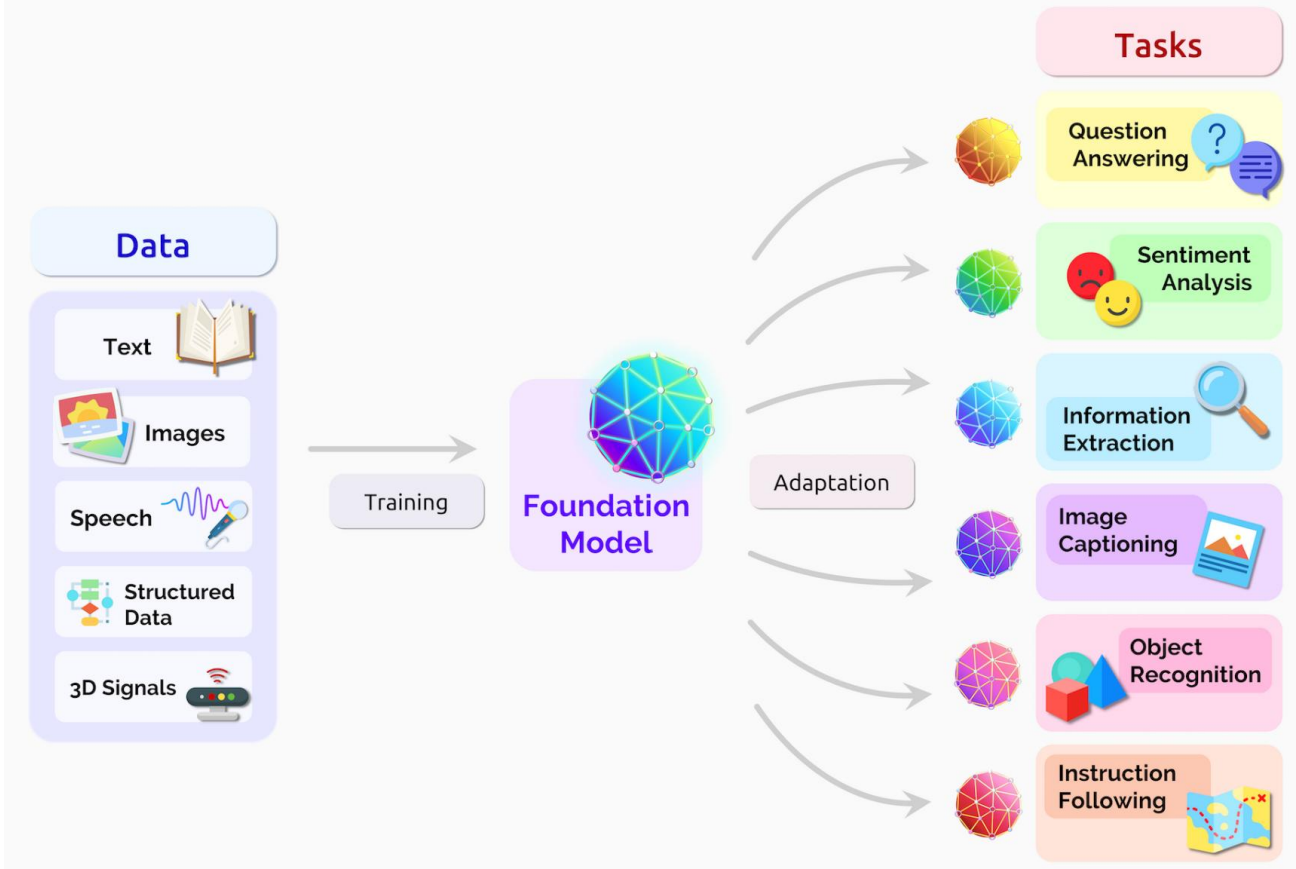
$$c_t = \sum_i a_{t,i} \cdot h_i$$

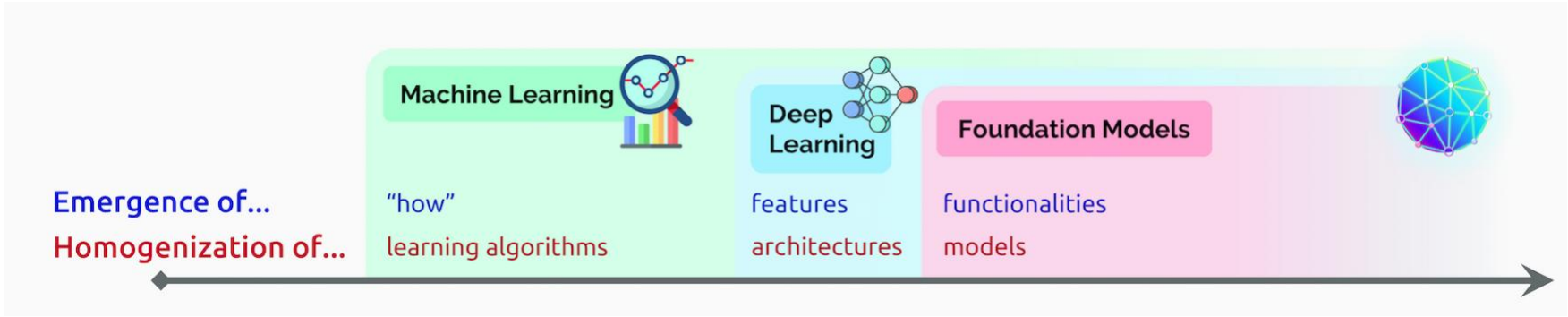
3. **Generate Output**:
   - Combine the context vector c_t with the current decoding state s_t to generate the new output. This allows the model to dynamically adjust its output based on the context vector.



Foundation model 基座模型

- Large-scale pre-training: Training on massive datasets to capture complex patterns and structures in the data. 大规模预训练：在庞大的数据集上进行训练，捕获数据中的复杂模式和结构。
- Self-supervised learning: Learning feature representations by predicting parts of the input data without manual annotation. 自监督学习：无需人工标注，通过预测输入数据的一部分来学习特征表示。
- Strong generalization: Pre-trained models can be fine-tuned for various specific tasks, reducing the need for large-scale labeled data. 通用性强：预训练的模型可以通过微调，适用于多种特定任务，减少了对大规模标注数据的需求。
- Large number of parameters: Typically ranging from hundreds of millions to hundreds of billions of parameters, enhancing the model's expressive power. 参数量大：通常拥有数亿到数千亿的参数，提升了模型的表达能力。

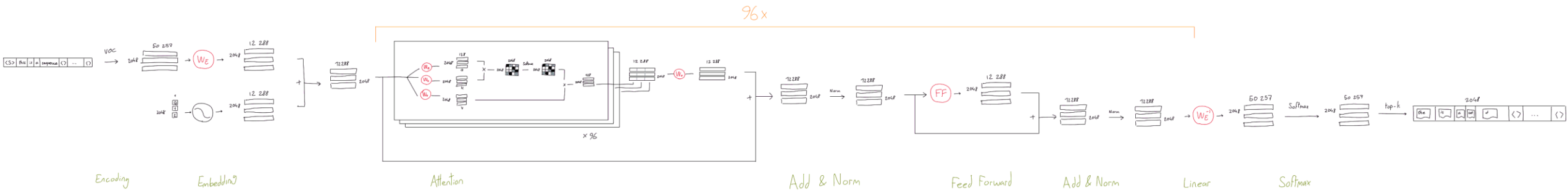




Bommasani, Rishi, et al. "On the Opportunities and Risks of Foundation Models." *Cornell University - arXiv,Cornell University - arXiv*, Aug. 2021.

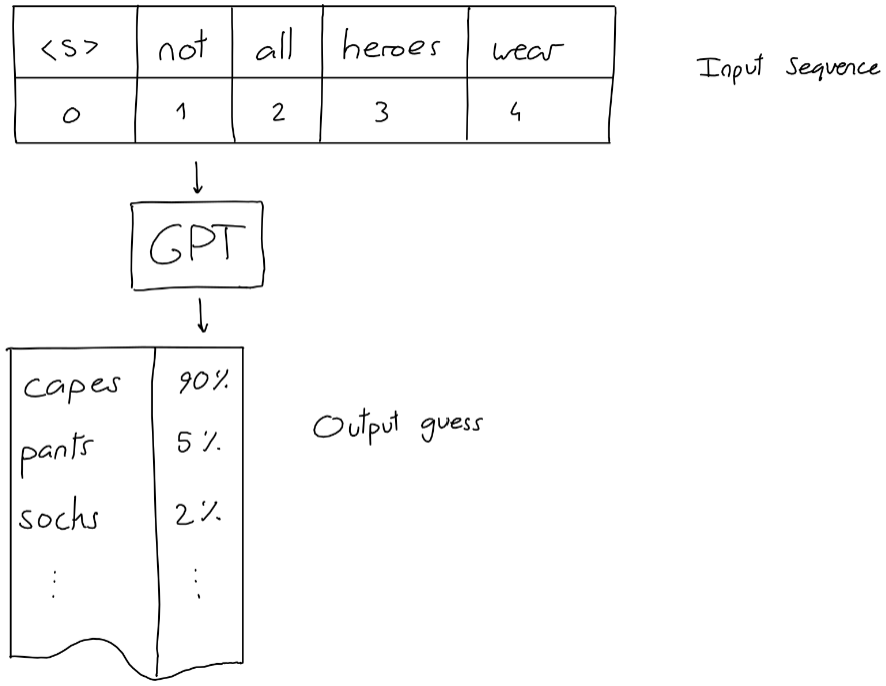
ChatGPT

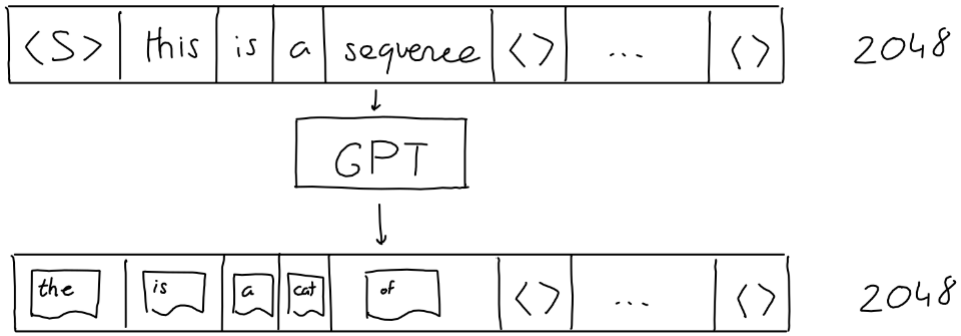
- Next token prediction (NTP), Autoregressive
- Decoder only
- Code training
- RLHF
- Large language model



Ref: https://dugas.ch/artificial_curiosity/GPT_architecture.html

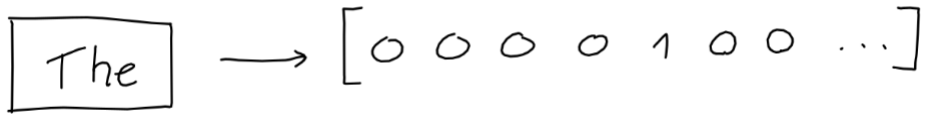
Input and output

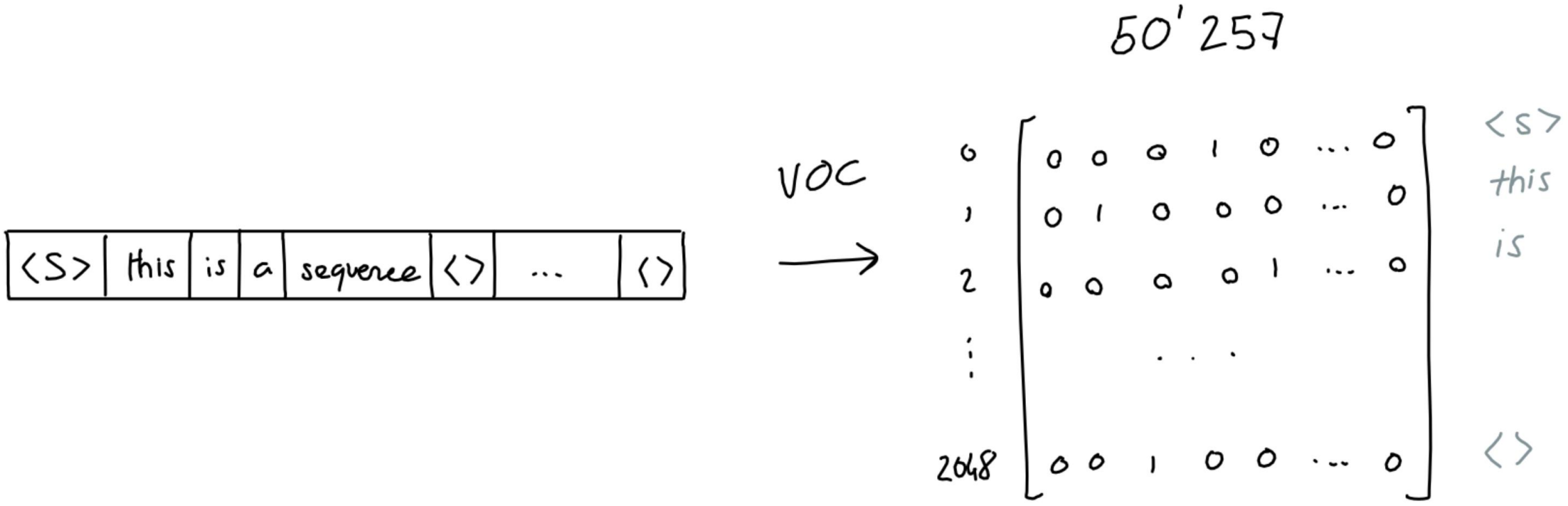




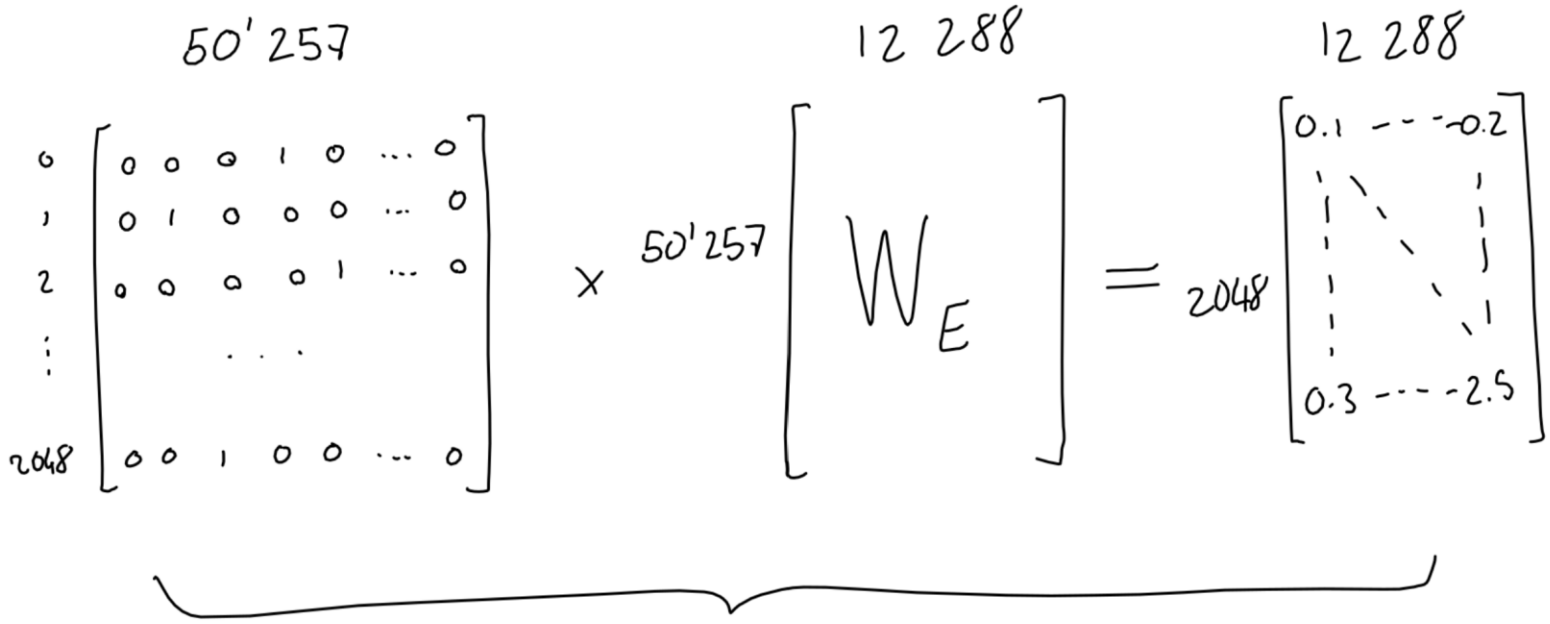
Not all heroes wear capes -> *but*
Not all heroes wear capes but -> *all*
Not all heroes wear capes but all -> *villans*
Not all heroes wear capes but all villans -> *do*

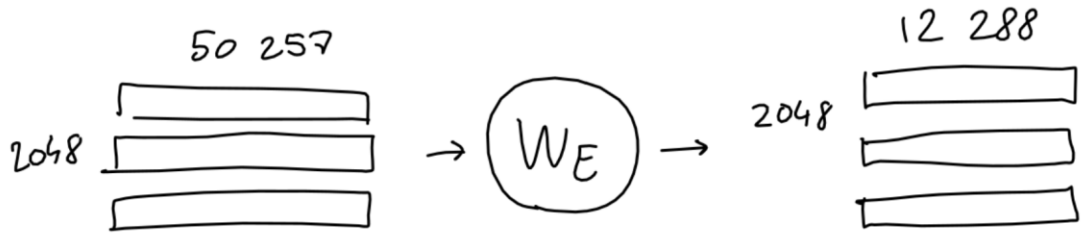
Embedding


The
0 0 0 0 1 0 0 ...

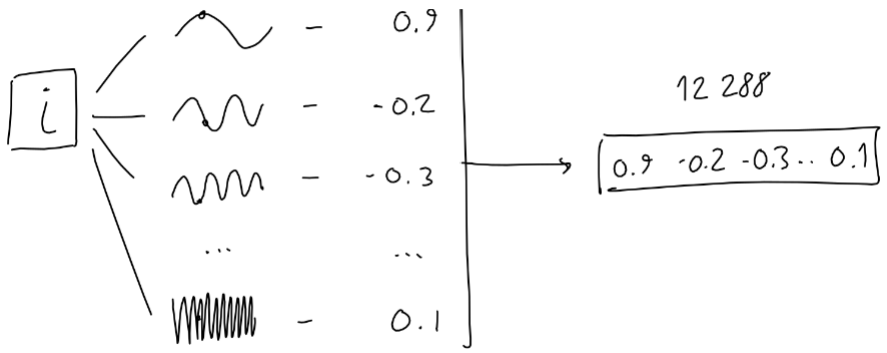

50'257
<S>
this
is
a
sequence
<>
...
<>
VOC
0
1
2
2048
0 0 0 1 0 ... 0
0 1 0 0 0 ... 0
0 0 0 0 1 ... 0
. . .
0 0 1 0 0 ... 0
<S>
this
is
<>

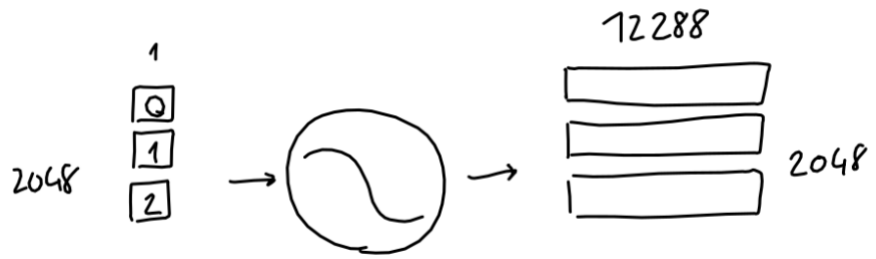
Embedding projection



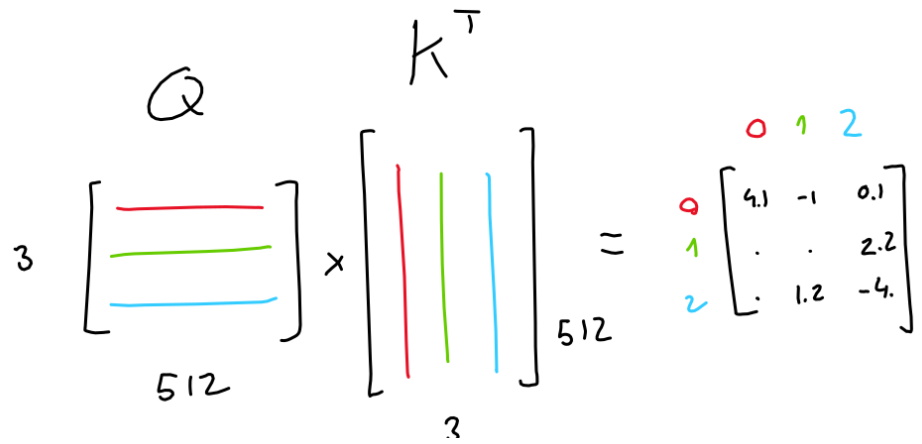


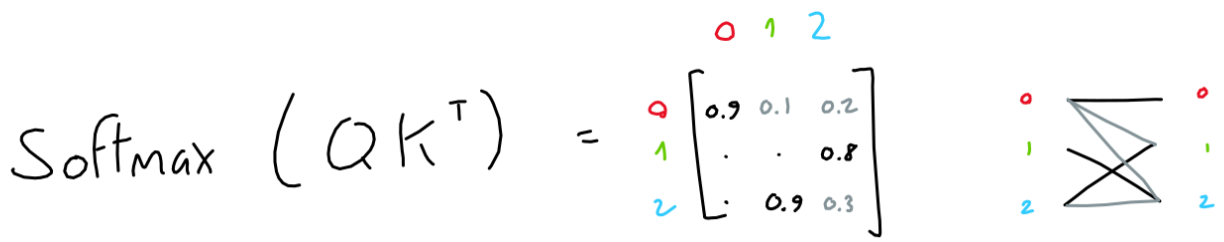
Position embedding

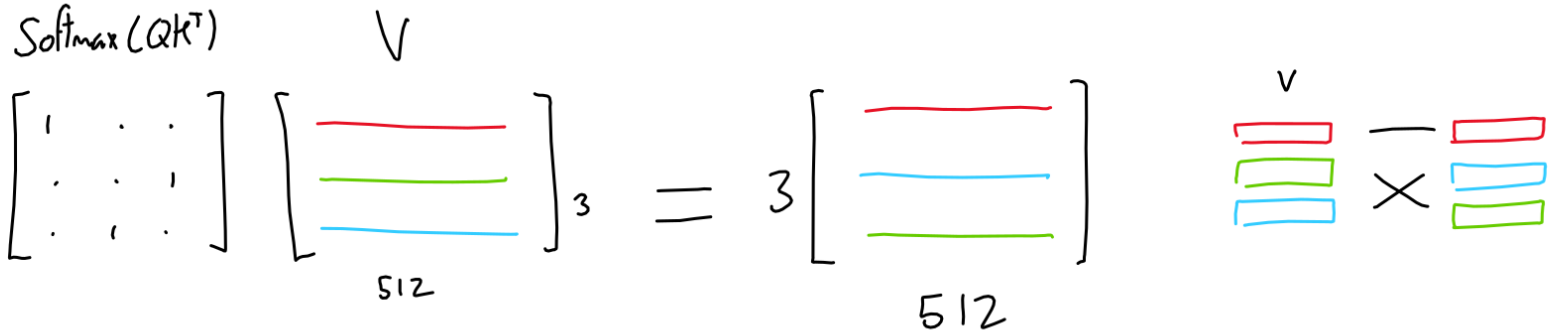




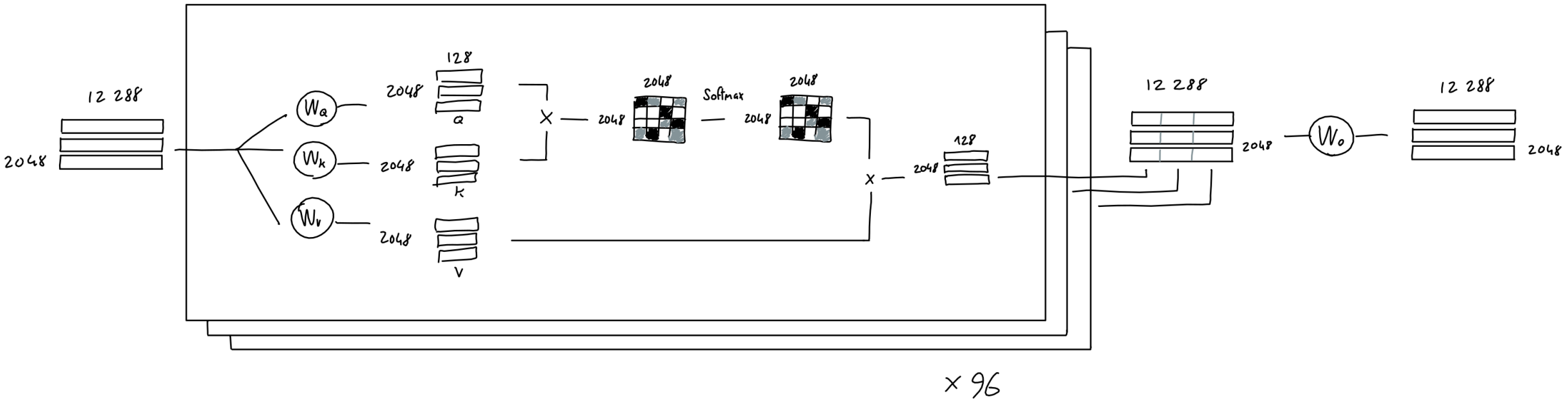
Attention calculation



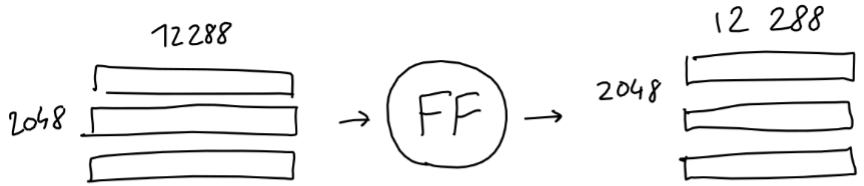


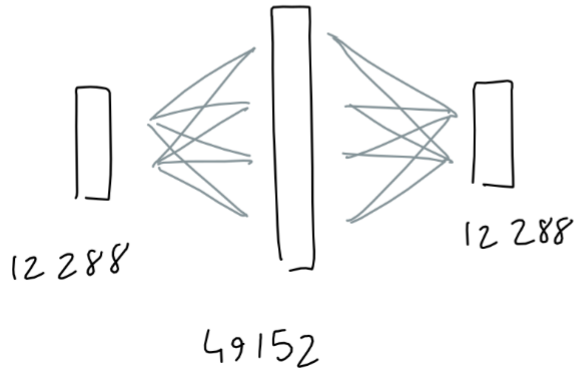


Multi-head attention

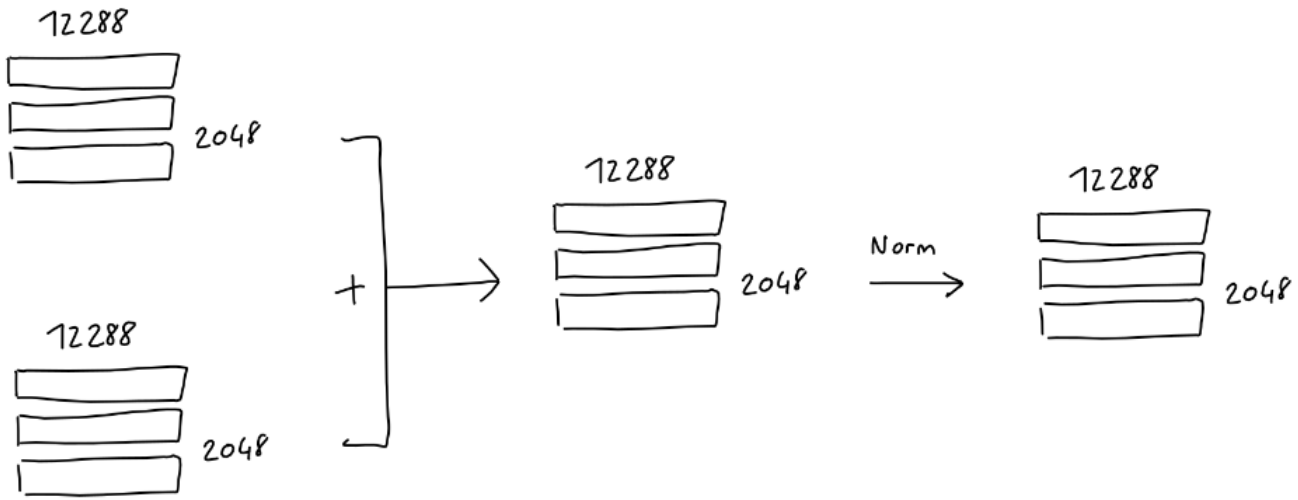


Feed forward

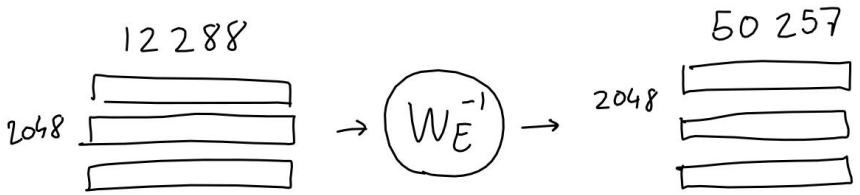




Add & Norm



decoding



“Temperature” in LLM
大模型中的“温度”

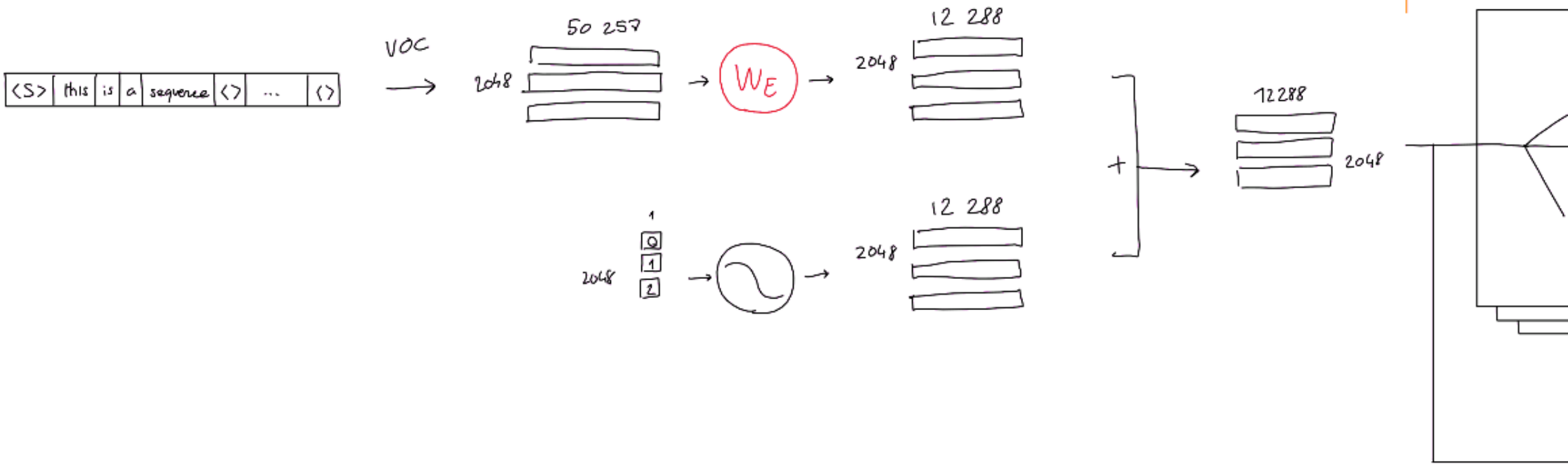
- control the randomness and diversity
- Probe for a wider solution space

Suppose we give the model a prompt like, “The weather is nice today, perfect for...” with different temperature settings:

- **Temperature=0**: The model might generate “The weather is nice today, perfect for a walk.” It chose the most likely word, resulting in conservative content.
- **Temperature=0.7**: The model might generate “The weather is nice today, perfect for a hike or a picnic.” This is somewhat more varied than temperature 0.
- **Temperature=1.5**: The model might generate “The weather is nice today, perfect for bungee jumping, swimming, or even a hot air balloon adventure!” This has much greater diversity but is also more random.

$$p_i^{\text{new}} = \frac{\exp\left(\frac{\log p_i}{T}\right)}{\sum_j \exp\left(\frac{\log p_j}{T}\right)}$$

Full architecture - 1

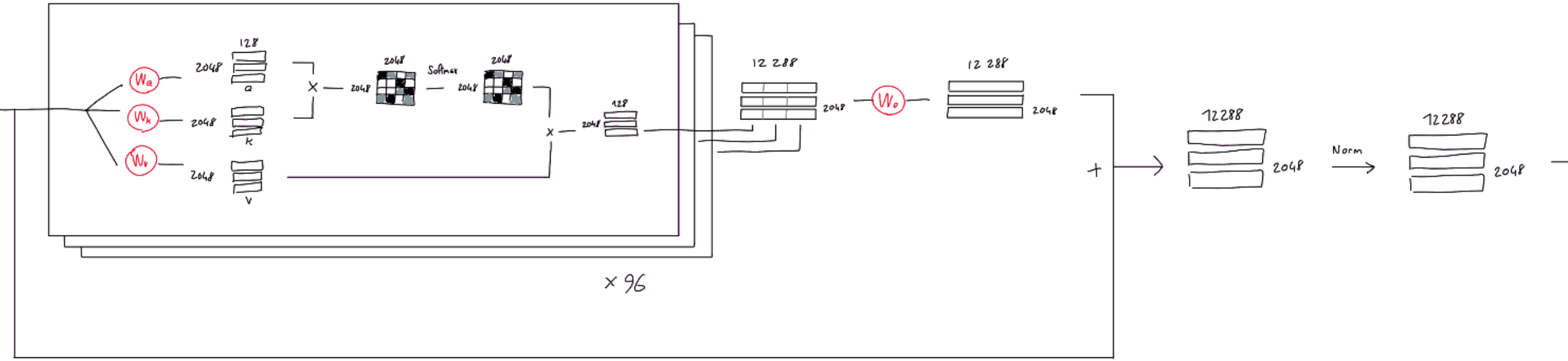


Encoding

Embedding

Full architecture - 2

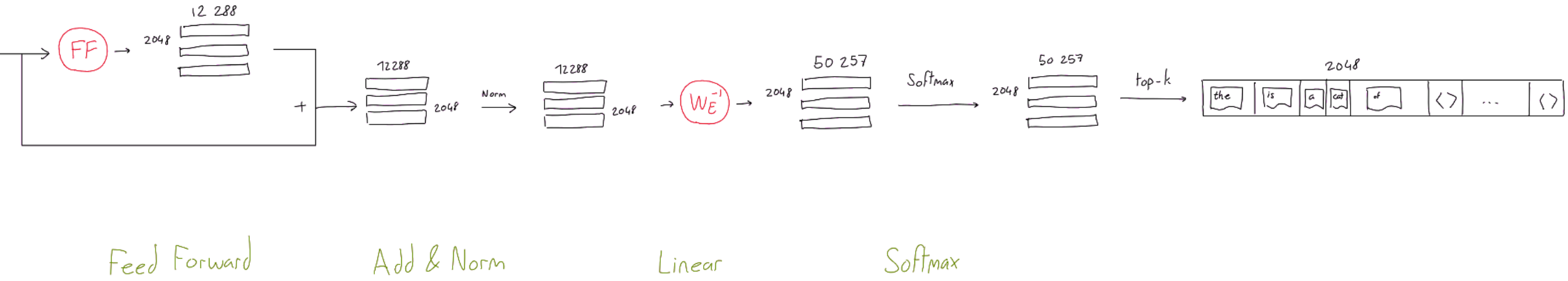
96 x



Attention

Add & Norm

Full architecture - 3



Feed Forward

Add & Norm

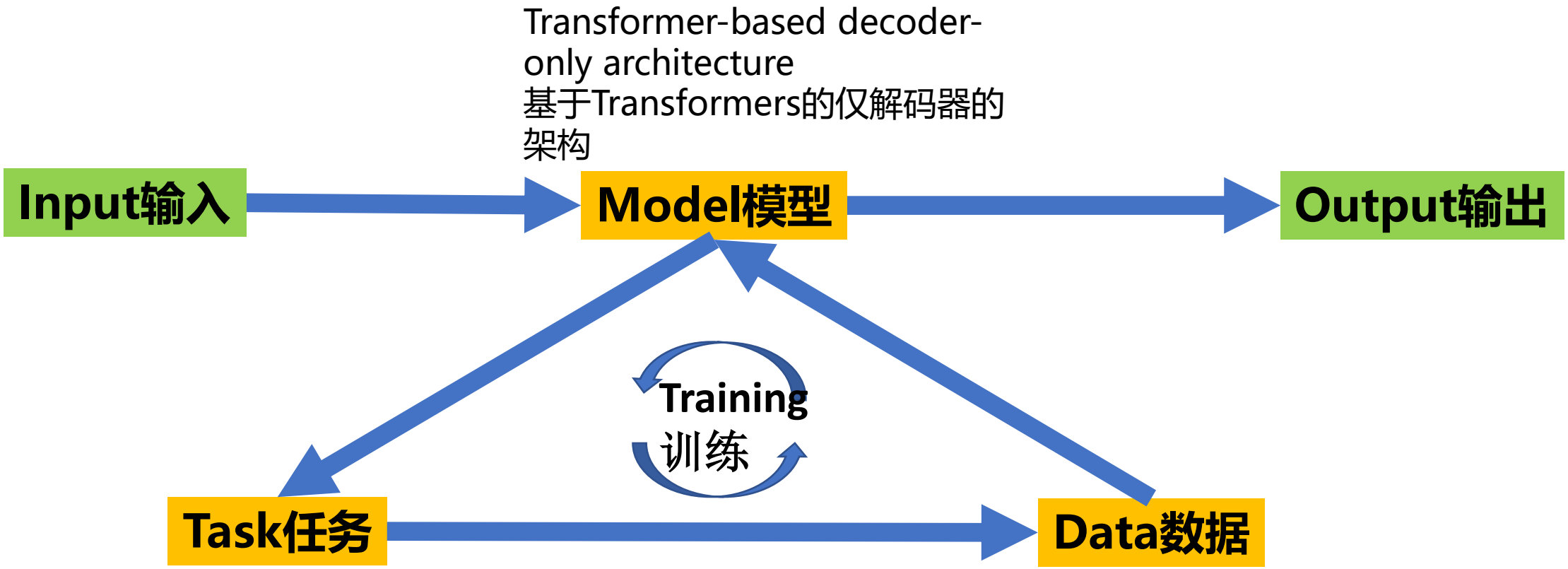
Linear

Softmax

Typical LLM Training Pipeline 大模型训练流程

- **1. Data Preparation 数据**
  - - X0TB raw text corpus (Common Crawl, GitHub, books) 原始文本语料
  - - Multilingual filtering & deduplication 多语言过滤与去重
  - - Quality ranking & toxicity removal 质量分级与毒性内容清除
- **2. Pre-training 预训练**
  - - Transformer architecture (decoder-only GPT-style) 仅解码器架构
  - - 300B+ tokens training corpus 超3000亿token训练
  - - 3D parallelism (tensor/pipeline/data) 三维并行策略
  - - Loss patterns monitoring (perplexity, grad norms) 损失模式监控
- **3. Supervised Fine-Tuning (SFT) 监督微调**
  - - 50k-100k human demonstrations 人工标注指令数据
  - - Instruction-response pairs construction 指令-回答对构建
  - - Multi-turn dialogue formatting 多轮对话格式优化
  - - Catastrophic forgetting prevention 灾难性遗忘预防
- **4. Reward Modeling & RLHF 奖励建模与强化学习**
  - - Preference data collection (10k+ comparisons) 偏好数据收集
  - - Reward model training (Bradley-Terry model) 奖励模型训练
  - - PPO optimization with KL penalty 带KL约束的PPO优化
  - - Safety layer integration 安全层集成

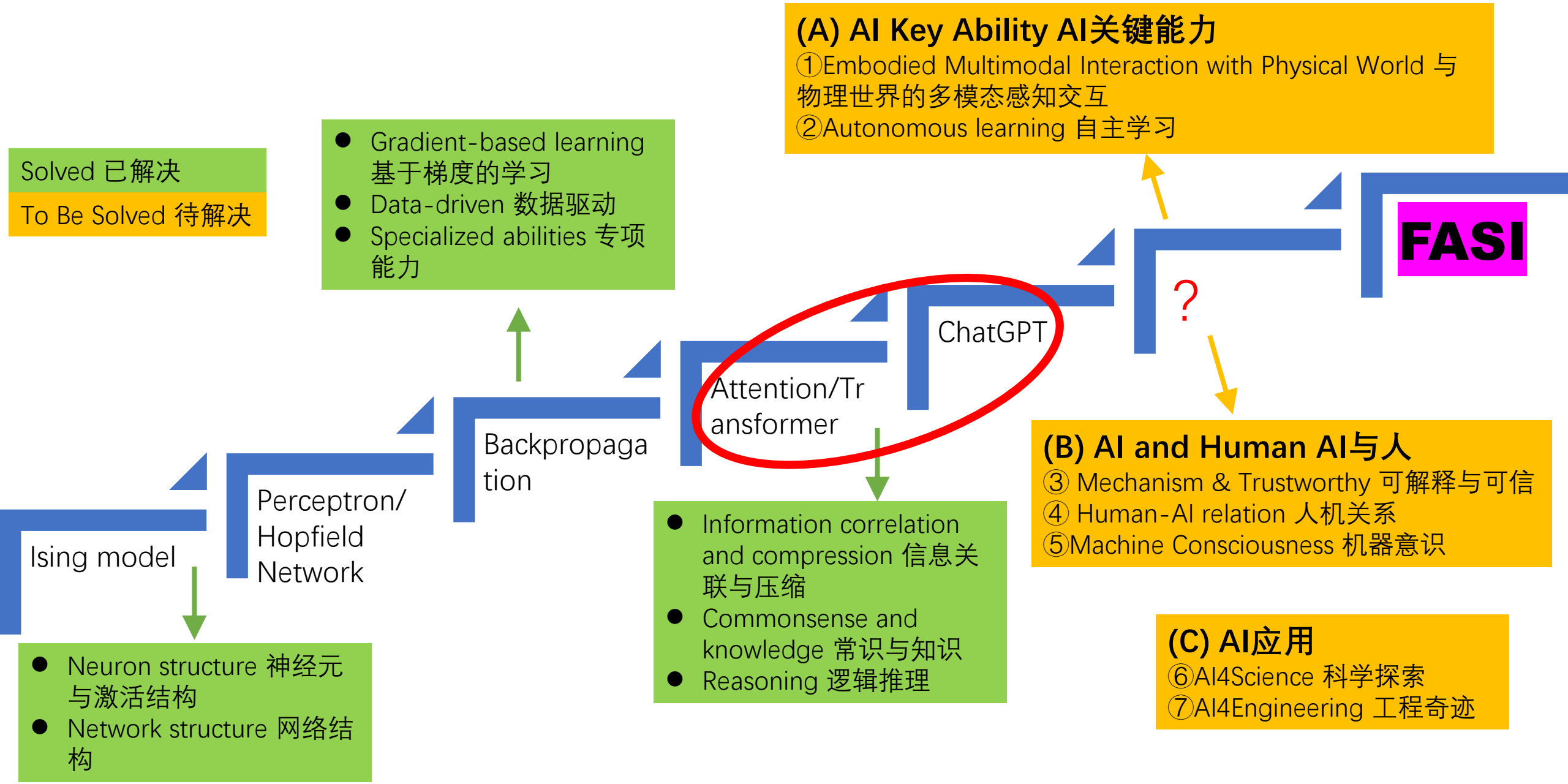
Data, Model, Task



- Pretraining预训练: Next Token Prediction (NTP, 预测下一个词)
- SFT 监督微调：Q&A ability 拟合人类回答问题和对话能力
- RW+RLHF 强化学习：Favor human preference 贴近人类偏好给答案打的分

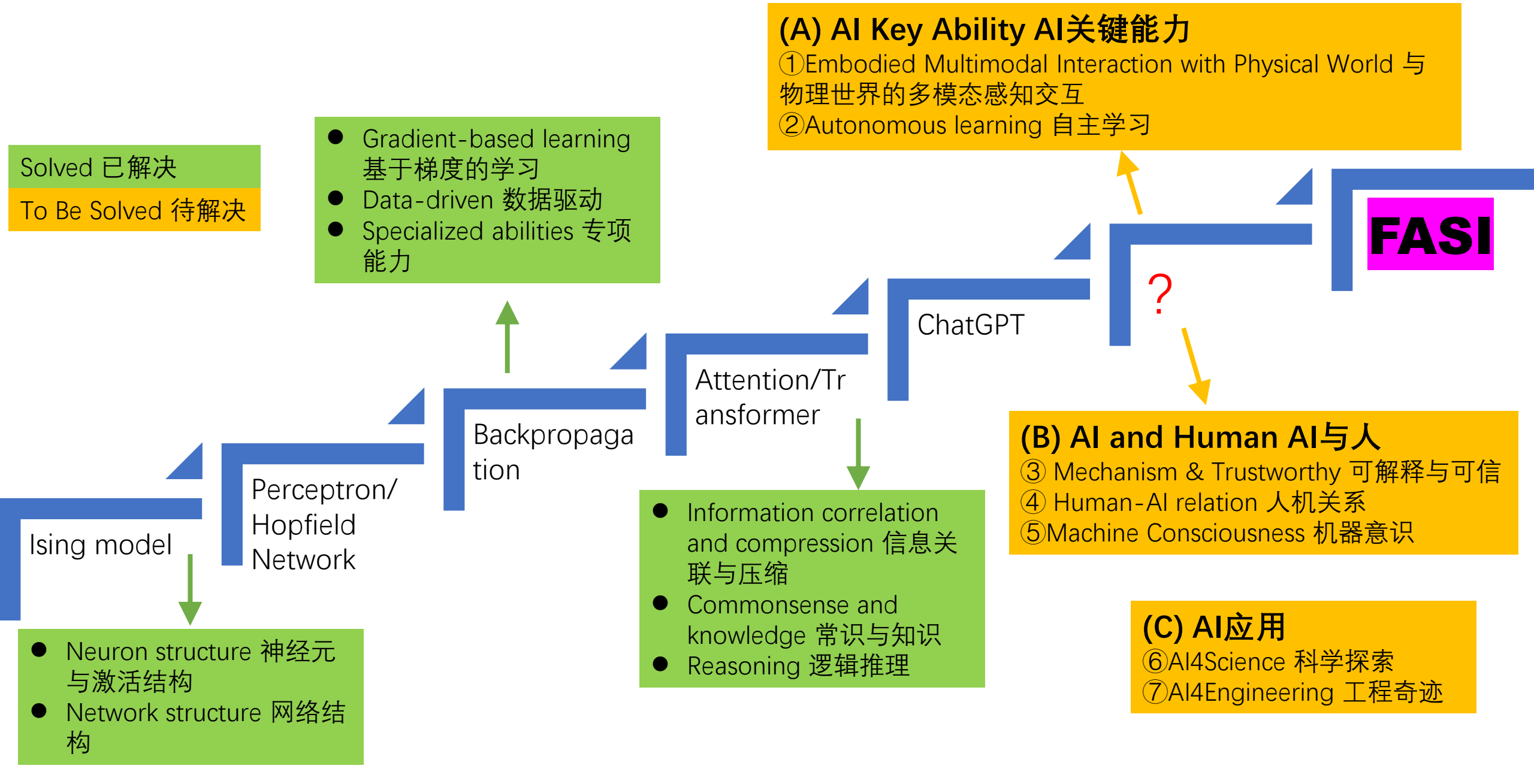
- Huge amount of text 海量文本
- Some annotated Q&A 一些问答对
- Ordered different answers to same question人类偏好打分排序的对同一问题的不同回答

Pathway to Full Artificial Super Intelligence




ChatGPT (GPT3.5)
GPT-4
GPT-4o
o1 / o1 pro
o3
(GPT-5)
?

Pathway to Full Artificial Super Intelligence



How do we “construct” artificial intelligence?

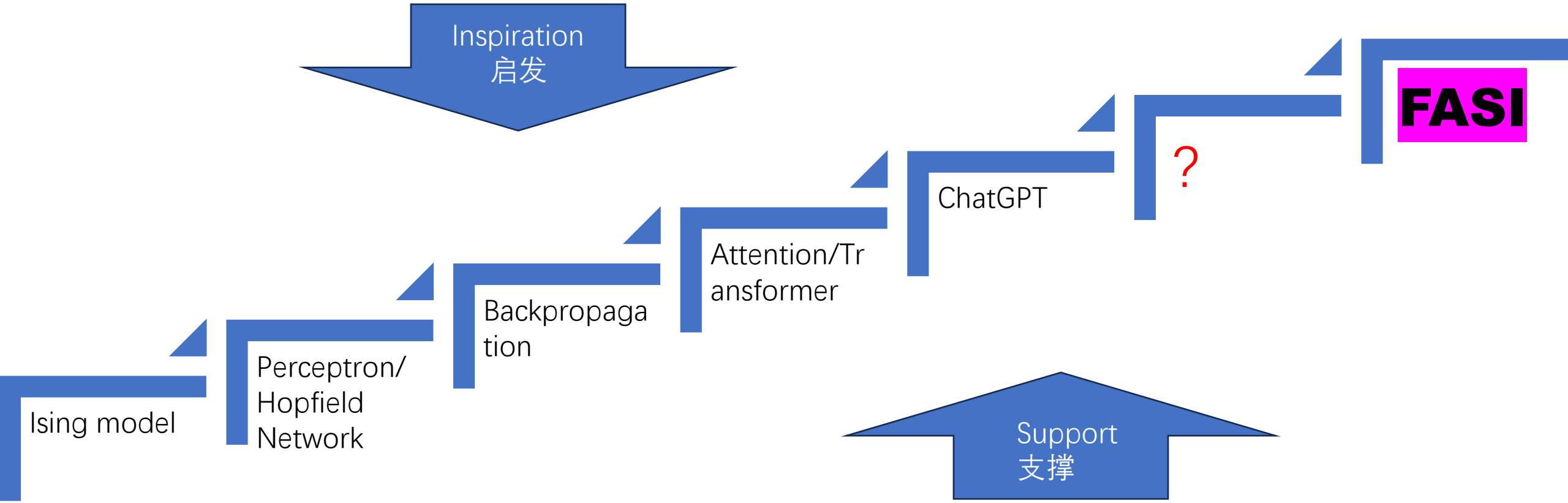
- Connectionism
  - Intelligence is in the connections of neurons
  - Typical:
    - CNN based cat-noncat classifier
    - Multi-layer neuron network
    - Supervised training
      - Back-propagation updating weights
    - Input-calculate-output
    - Multi-step feature transformation
      - On multiple granularities
- Symbolism
  - Intelligence is in the symbol-based Factorings
  - Typical:
    - MYCIN医疗诊断系统
    - Knowledge (based on symbol/language)
      - Rules
      - Factoring
      - 如果病人有发热，而且血液培养结果显示有革兰阴性菌，那么病人可能有败血症sepsis。
- Behaviorism
  - Intelligence is adjusting behavior to achieve goal by interacting with environment
  - Typical:
    - Q-learning (reinforcement learning)
    - State-Action-Reward
    - Q-value
      - Expected reward for a certain action on a certain state
    - Policy
      - epsilon-greedy

①Physics 物理学

②Neuroscience 神经科学

③CS/Informatics 计算机/信息论

④ Many other disciplines … 许多其他学科



①Linear Algebra 线性代数

②Probability and Statistics 概率与统计

③Calculus微积分

Thank you!

Email电邮：yutaoyue@hkust-gz.edu.cn

HKUST-GZ香港科技大学（广州）
（Wechat Subscription公众号）

Yutao Yue岳玉涛
（Personal Wechat个人微信）

Pillars.Creation 创世之柱（Left左: C8HD, Right右: Hubble.Space.Telescope哈勃） | Target: d~5700Ly/D~5Ly(pillar)/VM=+6.4 | Telescope: D=203mm/F=1439mm/T~336min | Time.Loc.: 20230528/CN-SZ-KJY@Bortle9 | Photo by: 馒头星系Matt