

# 深度学习

第 1 次课 绪论与机器学习基础回顾

---

课程编号 WJ64007N  
未来技术学院 江伟森  
F3b-212 (9月24日开始)

# 本节课的四个问题

---

01

## 深度学习是什么

它和人工智能、机器学习有什么关系

0-12 分钟

02

## 为什么在今天有效

数据、计算与算法如何共同推动发展

12-50 分钟

03

## 基本训练逻辑是什么

数据、模型、损失、优化与评估如何衔接

50-82 分钟

04

## 这门课怎么学

16 次课的内容结构与学习要求

82-90 分钟

# 课程基本信息

---

3

学分

48

总学时

32

讲授学时

16

讨论学时

---

**授课对象**

研究生

**授课语言**

中文

**考核方式**

考勤、论文复现，作业与期末考试，比例见下一页

**授课时间**

每周四下午，5-7节（其中5-6节讲授，第7节讨论）

**课程主页**

<https://www.wayson.tech/deeplearning>

# 课程评分原则

10%

考勤

随机考勤 2-4 次

10%

论文复现报告

1次

30%

作业

共 3 次作业

50%

期末考试

(开闭卷结合, 一页纸开卷)

综合考查课程内容

总评成绩 = 考勤成绩 + 论文复现成绩 + 作业成绩 + 期末考试成绩

# 考勤与补签规则

## 正常考勤

- 1 课堂向在场同学发放不记名唯一 QR code
- 2 扫码后填写本人姓名并提交
- 3 每个 QR code 只能提交一次

## 补签办法

### 回答 1 次课堂问题

获得 1 个回答问题 QR code

**1个回答问题码 = 抵消 1 次缺勤**

# 16 次课的内容地图

## 第一部分 深度学习基础

### 第 1-4 次课

绪论与机器学习回顾  
前馈网络、反向传播与自动微分  
优化、正则化、归一化与框架实践

## 第二部分 主流模型架构

### 第 5-10 次课

卷积神经网络及视觉任务  
注意力、Transformer 与大语言模型  
生成模型、扩散模型与流匹配

## 第三部分 学习范式

### 第 11-16 次课

自监督与迁移学习  
多任务、元学习与联邦学习  
持续学习与课程总结

# 课程学习目标

---

## 理解

掌握神经网络、训练方法和主流架构的基本原理

## 分析

比较不同学习范式的问题设定、优势和适用条件

## 实现

使用 PyTorch 完成模型训练、调试与评估

## 研究

阅读前沿论文，并将方法与自己的科研问题连接

# 学习方式与课程要求

---

- 课前快速浏览讲义
- 课堂关注问题设定、关键假设和方法之间的联系
- 课后跑实验学习课堂内容
- 阅读论文时区分贡献、证据、局限与可复现性

## 建议工具

Python  
PyTorch  
VS Code  
LLMs

**工具服务于问题，实验记录决定结果是否可信**

# 先修知识自检

---

## 线性代数

向量、矩阵乘法、范数与特征值

## 概率统计

随机变量、期望、条件概率与最大似然

## 机器学习

监督学习、损失函数、过拟合与评估

## Python

数组运算、函数、类与基本调试

# 从问题出发理解深度学习

---

机器学习让计算机从数据中调整行为  
深度学习进一步学习多层表示



# 人工智能、机器学习与深度学习

## 人工智能

设计或实现智能行为的技术集合

## 机器学习

从样本中学习规律

## 深度学习

用深度网络模型学习表示和预测

# 从规则编程到数据驱动学习



## 传统规则系统

人写规则，系统按规则执行

## 机器学习系统

人定义任务和训练过程，模型从样本中调整参数

**核心问题：训练数据是否代表真实使用场景**

# 表示学习减少了手工特征设计



低层

中层

高层

表示是否有用，由任务和数据共同决定

同一个训练目标  
可以同时调整

特征提取  
中间表示  
最终预测

# “深”意味着多次非线性变换



每一层完成一次可学习的变换，并通过非线性激活扩大可表达的函数范围

更深不等于更好：结构、数据、优化和计算预算必须匹配

# 深度学习任务的共同结构

## 感知

图像、语音、传感  
器信号

## 理解

分类、检测、分割  
、问答

## 生成

文本、图像、音频  
与视频

## 决策

推荐、控制与科学  
发现

**任务不同，但都需要把输入映射为可评价的输出**

# 视觉任务一：图像分类



输入

一张图像

输出

类别概率

训练信号

图像与标签

**分类只是起点：真实系统还要处理长尾类别、噪声标签与分布变化**

## 视觉任务二：检测与分割



### 分类

图中有什么

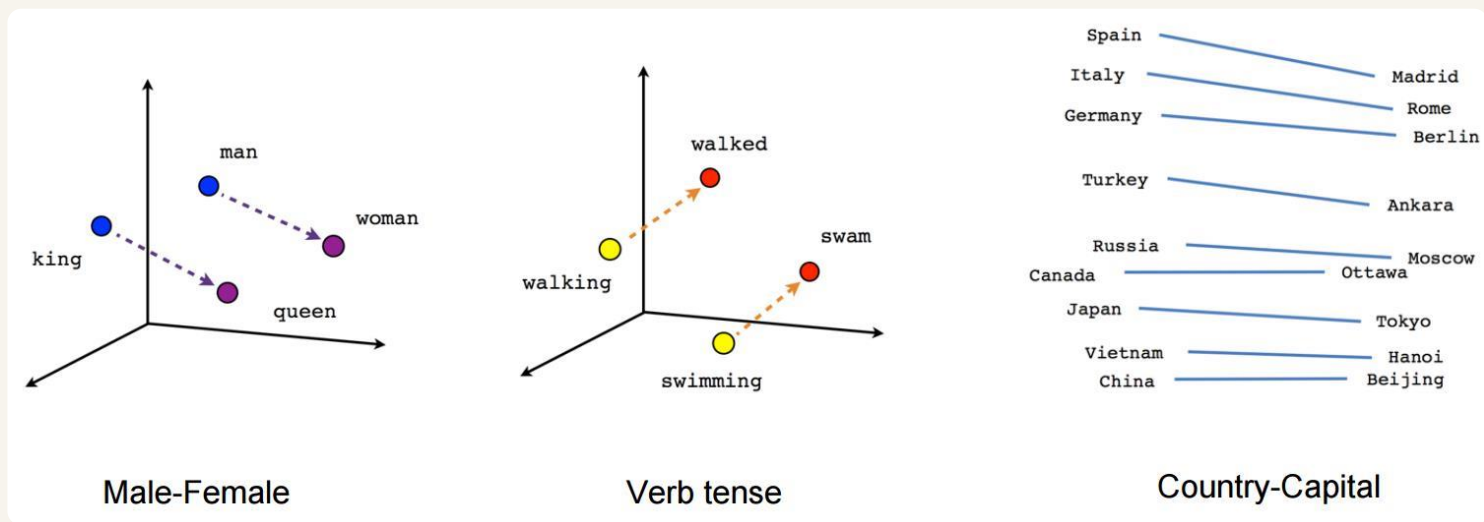
### 检测

对象在哪里

### 分割

每个像素属于谁

# 语言任务：把符号映射到连续表示

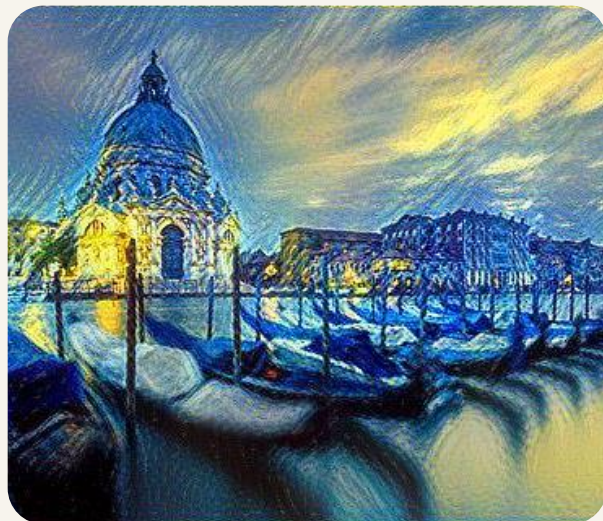


## 连续向量可以表达

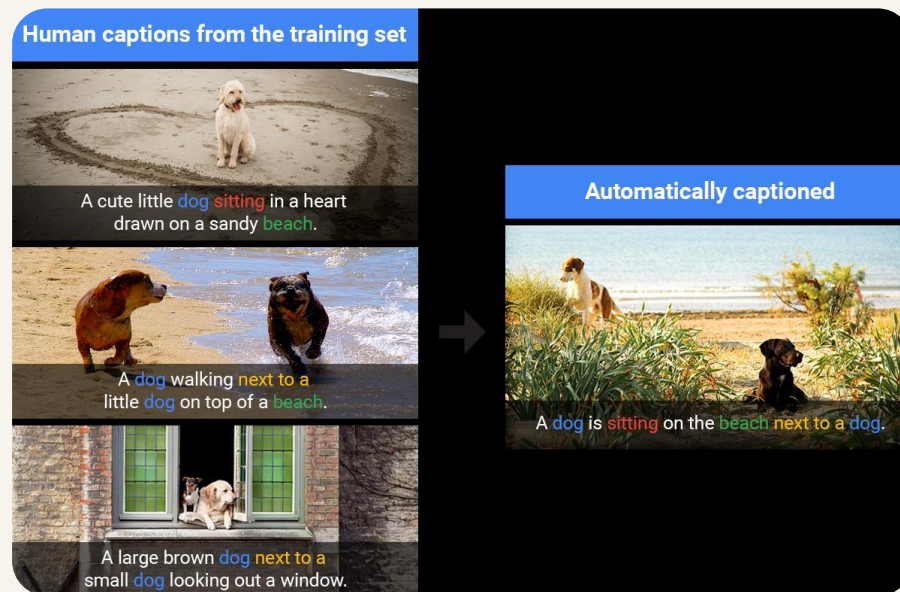
- 语义相似性
- 句法关系
- 上下文依赖
- 跨语言对应

现代语言模型表示会随上下文变化

# 生成与多模态任务



内容保持，视觉风格改变



图像与语言在同一任务中连接

生成模型学习数据分布，并据此产生新样本或条件输出

# 能力边界与常见风险

## 分布偏移

训练环境与部署环境不同

## 虚假相关

模型利用了数据中的捷径

## 不确定性

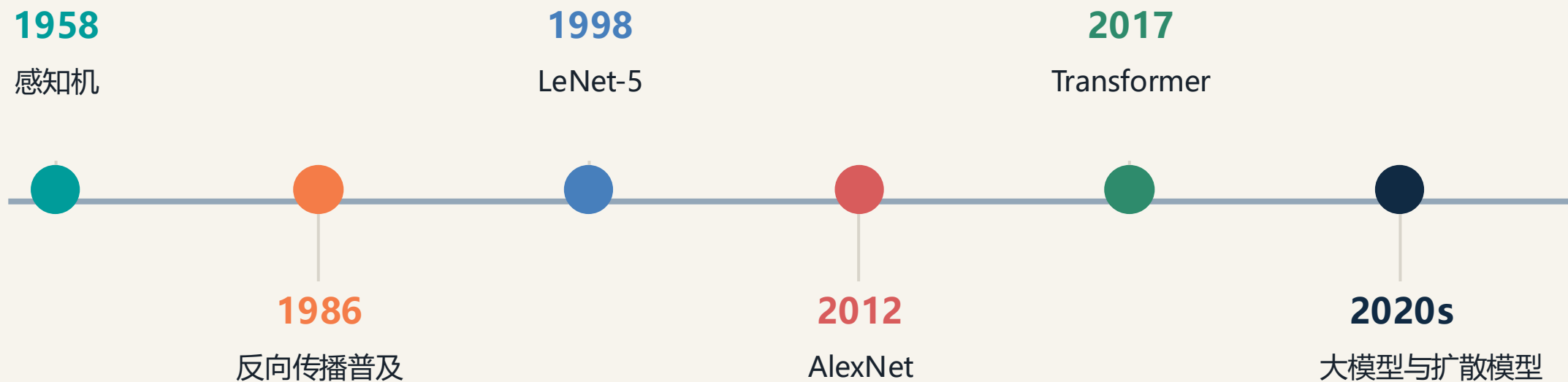
高置信度也可能预测错误

## 资源与责任

计算成本、隐私、公平与安全

研究问题通常从“模型在哪些条件下会失效”开始

# 深度学习的发展脉络



历史不是单线进步：理论、数据与计算条件共同决定方法能否落地

# 为什么是现在

## 数据

更大规模的数据集  
更低的采集与存储成本

## 计算

GPU 与专用加速器  
适合并行矩阵运算

## 算法与软件

更稳定的优化方法  
与开源框架

**三项条件相互放大：更大的模型需要更多数据和计算，也需要更可靠的软件**

# 课程实验环境

---

Python

数据处理、实验脚本与可视化

PyTorch

张量运算、自动微分与模型训练

Colab

边运行边记录，便于复现实验

**第一周目标：能创建张量、运行前向计算并查看梯度**

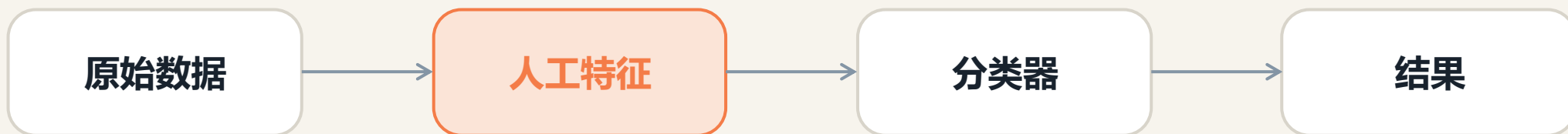
# 一次可复现的实验流程



**复现失败通常不是模型问题，而是环境、数据划分或配置没有被记录**

# 从特征工程到端到端学习

## 传统机器学习流程



## 端到端深度学习流程



端到端学习减少了人工特征设计，但增加了数据、计算与验证要求

# 机器学习基础回顾

---

**数据 模型 损失 优化 评估**

五个要素组成一个可训练、可验证的学习系统

# 监督学习的问题设定

## 训练数据

$D = \{x_i, y_i\}$   
 $x_i$ : 输入  
 $y_i$ : 目标输出

模型  $f_{\theta}(x)$

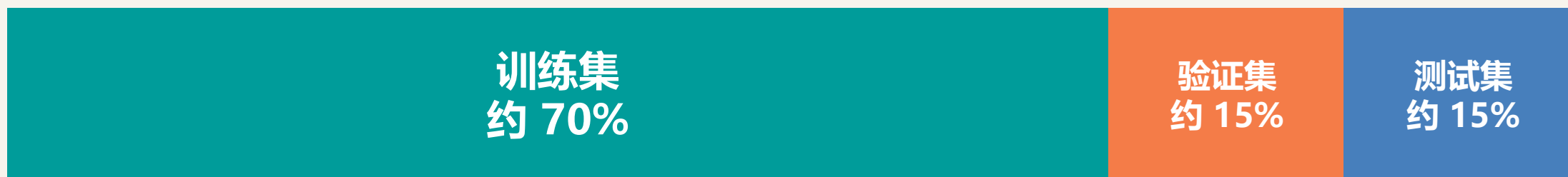
## 学习目标

找到参数  $\theta$   
使模型在未见样本上  
也能给出有用预测

监督学习关心泛化，不只关心记住训练集

# 训练集、验证集与测试集

## 一个常见的示例划分



训练集

学习模型参数

验证集

选择模型与超参数

测试集

最终估计泛化性能

**比例不是规则，关键是避免信息泄漏并匹配真实部署条件**

# 数学工具回顾

---

**标量 向量 矩阵 张量**

这些对象构成深度学习计算的基本语言

# 从标量到张量

**标量**

0 阶

$$x = 2$$

一个数

**向量**

1 阶

$$x = [x_1, x_2, x_3]$$

一组有序特征

**矩阵**

2 阶

$$X = \mathbb{R}^{m \times n}$$

m 个样本, 每个 n 维

**张量**

3 阶及以上

$$X = \mathbb{R}^{B \times C \times H \times W}$$

批次图像

阶数描述轴的数量, shape 给出每条轴的长度

# 向量的基本运算

$$\mathbf{x} = [1, 2, 3]$$

$$\mathbf{y} = [4, 0, -1]$$

逐元素相加

$$\mathbf{x} + \mathbf{y} = [5, 2, 2]$$

标量乘法

$$2\mathbf{x} = [2, 4, 6]$$

**形状必须兼容：不能直接把 3 维向量与 4 维向量相加**

# 点积衡量方向的一致程度

$$\mathbf{x} \cdot \mathbf{y} = \sum_i x_i y_i$$

$$[1, 2, 3] \cdot [4, 0, -1] = 1 \times 4 + 2 \times 0 + 3 \times (-1) = 1$$

$$\mathbf{x} \cdot \mathbf{y} > 0$$

方向较一致

$$\mathbf{x} \cdot \mathbf{y} = 0$$

相互正交

$$\mathbf{x} \cdot \mathbf{y} < 0$$

方向较相反

# 范数描述向量的大小

## L1 范数

$$\|x\|_1 = \sum_i |x_i|$$

更强调稀疏性

## L2 范数

$$\|x\|_2 = \sqrt{\sum_i x_i^2}$$

欧氏长度

## L $\infty$ 范数

$$\|x\|_\infty = \max_i |x_i|$$

最大分量

距离可以写成  $\|x-y\|$ ，不同范数对应不同的几何假设

# 矩阵乘向量完成一次线性变换

$$A \in \mathbb{R}^{m \times n}$$

$$x \in \mathbb{R}^n$$

$$y = Ax \in \mathbb{R}^m$$

每个输出分量都是  $A$  的一行与  $x$  的点积

$$A = [[1,2],[0,-1]], \quad x = [3,4] \Rightarrow Ax = [11,-4]$$

# 矩阵乘法与空间变换

---

$$A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}$$

$$AB \in \mathbb{R}^{m \times p}$$

内侧维度  $n$  必须一致

缩放

旋转

投影

特征混合

多层线性变换若没有非线性，仍等价于一次线性变换

# 几类常见特殊矩阵

---

单位矩阵  $I$

$$Ix = x$$

对角矩阵  $D$

各维度独立缩放

---

对称矩阵  $A = A^T$

二次型与协方差常见

正交矩阵  $Q^T Q = I$

保持长度和夹角

---

特殊结构能降低计算量，也能带来更明确的几何解释

# 张量的 shape 表达数据语义

---

表格数据

[batch, feature]

图像

[batch, channel, height, width]

文本

[batch, sequence, hidden]

视频

[batch, time, channel, height, width]

很多运行错误来自 shape 不符合预期，而不是公式本身

# 索引与切片

$X[1, 2]$

取一个元素

$X[1, :]$

取一整行

$X[:, 1]$

取一整列

$X[:, :3]$  表示保留所有样本，只取前 3 个特征

切片通常返回视图或共享存储，原地修改时要格外小心

# 广播规则：无需显式复制即可完成张量运算

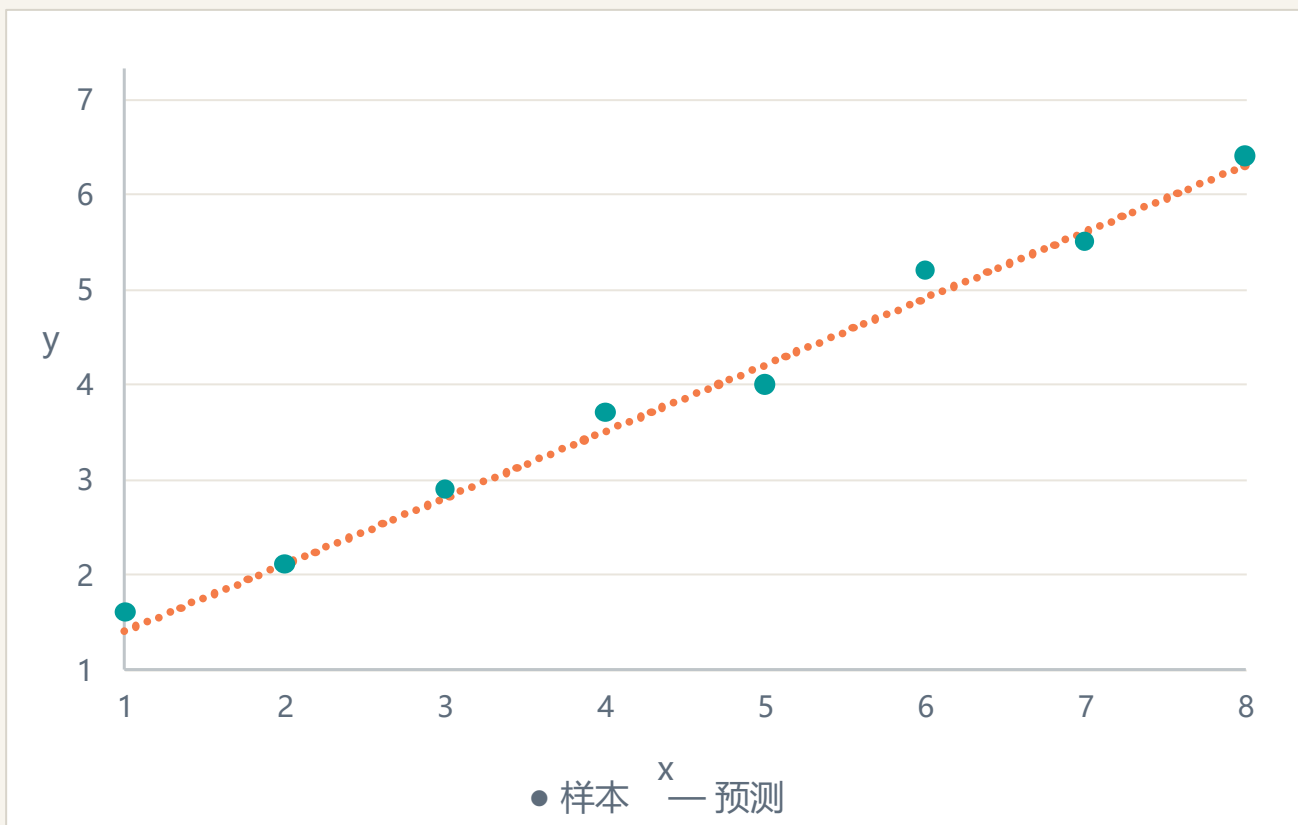
$$X \in \mathbb{R}^{B \times D} + b \in \mathbb{R}^D = Y \in \mathbb{R}^{B \times D}$$

向量  $b$  会在批次维上自动扩展，每个样本加上同一个偏置

从最后一个维度向前比较：相等、其中一个为 1，或该维不存在

广播只描述运算逻辑，不一定真的复制内存

# 线性模型是最小的可学习系统

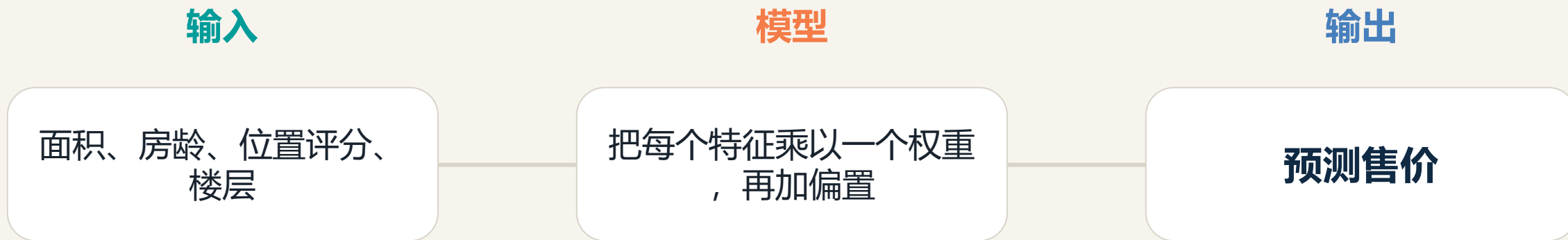


$$\hat{y} = w^T x$$

参数  $w$  决定方向和尺度  
参数  $b$  决定整体偏移

训练就是用数据选择参数

# 房价预测：从问题到模型



**建模前需要明确：预测对象、可用信息、评价标准和部署场景**

# 特征决定模型能看到什么

| 特征      | 单位   | 含义与风险      |
|---------|------|------------|
| 面积 x1   | 平方米  | 通常与价格正相关   |
| 房龄 x2   | 年    | 可能与价格负相关   |
| 位置评分 x3 | 0-10 | 把复杂区位压缩为数值 |
| 楼层 x4   | 层    | 关系可能不是简单线性 |

线性模型只会使用我们提供的表示，深度学习会进一步学习表示

# 仿射模型的逐项计算

$$\hat{y} = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

$x_i$

输入特征

$w_i$

特征权重

$b$

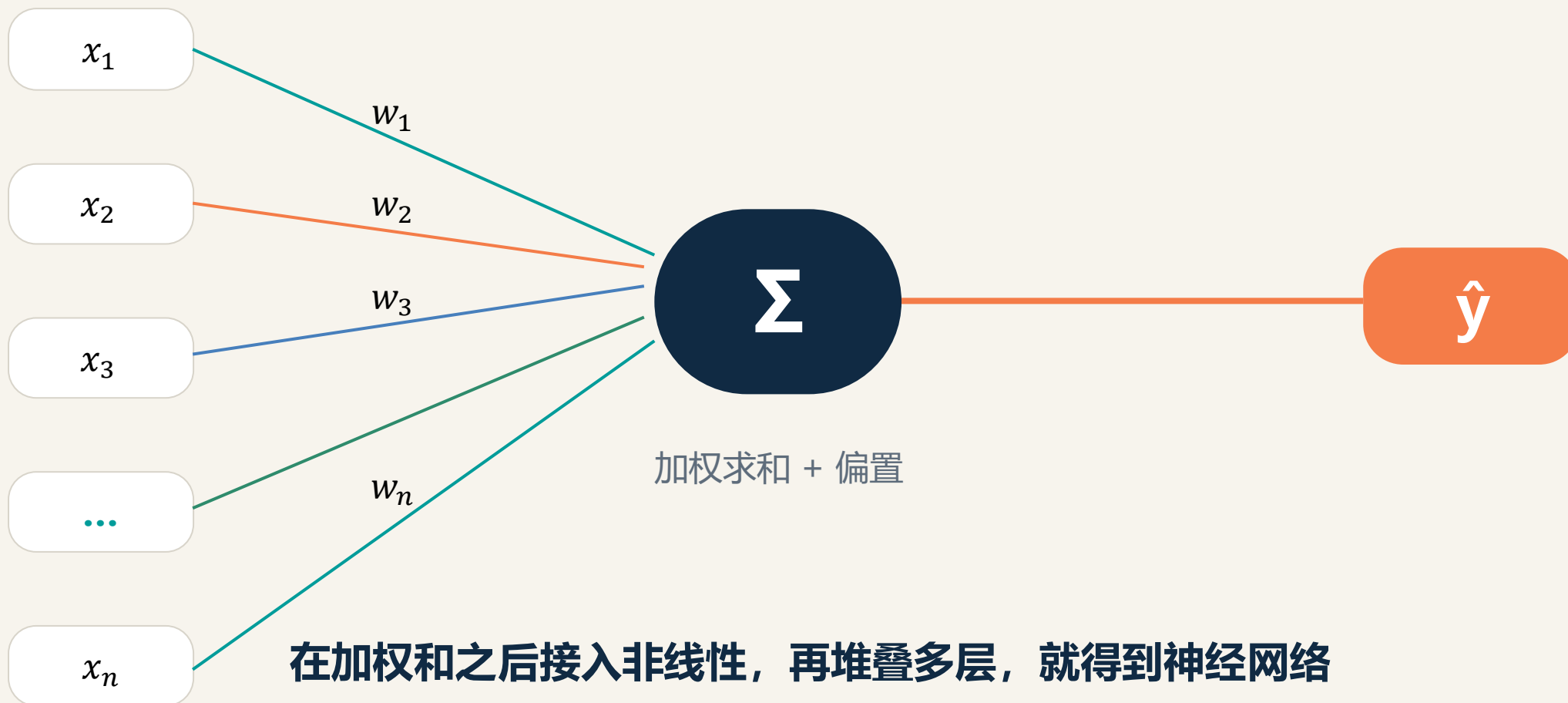
整体偏置

$\hat{y}$

模型预测

向量形式:  $\hat{y} = w^T x + b$

# 线性模型也是单层神经元



# 残差连接预测与真实值

真实值  $y$

————

预测值  $\hat{y}$

————

误差  $e = y - \hat{y}$

例：真实房价 300 万，模型预测 270 万，则残差为 30 万

训练需要把许多样本的残差汇总成一个目标函数

# 批量预测的矩阵形式

$$X \in \mathbb{R}^{(n \times d)}$$

n 个样本

$$w \in \mathbb{R}^d$$

d 个参数

$$\hat{y} = Xw + b$$

同时得到 n 个预测

$$MSE(w, b) = \frac{1}{n} \|y - Xw - b\|^2$$

向量化让 GPU 一次处理整批样本

# 闭式解与迭代优化

## 闭式解

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

- 线性最小二乘可直接求解
- 维度高时求逆代价大
- 矩阵可能不可逆或病态

## 迭代优化

从初始参数出发，反复降低损失

- 适用于大规模和非线性模型
- 可以使用小批次数据
- 需要学习率和停止条件

深度学习几乎都依赖迭代优化

# 损失函数把误差变成可优化的数值

单样本误差

$$\ell(y, \hat{y})$$

整套数据的平均损失

$$L(w) = \frac{1}{n} \sum_i \ell(y_i, \hat{y}_i)$$

训练目标

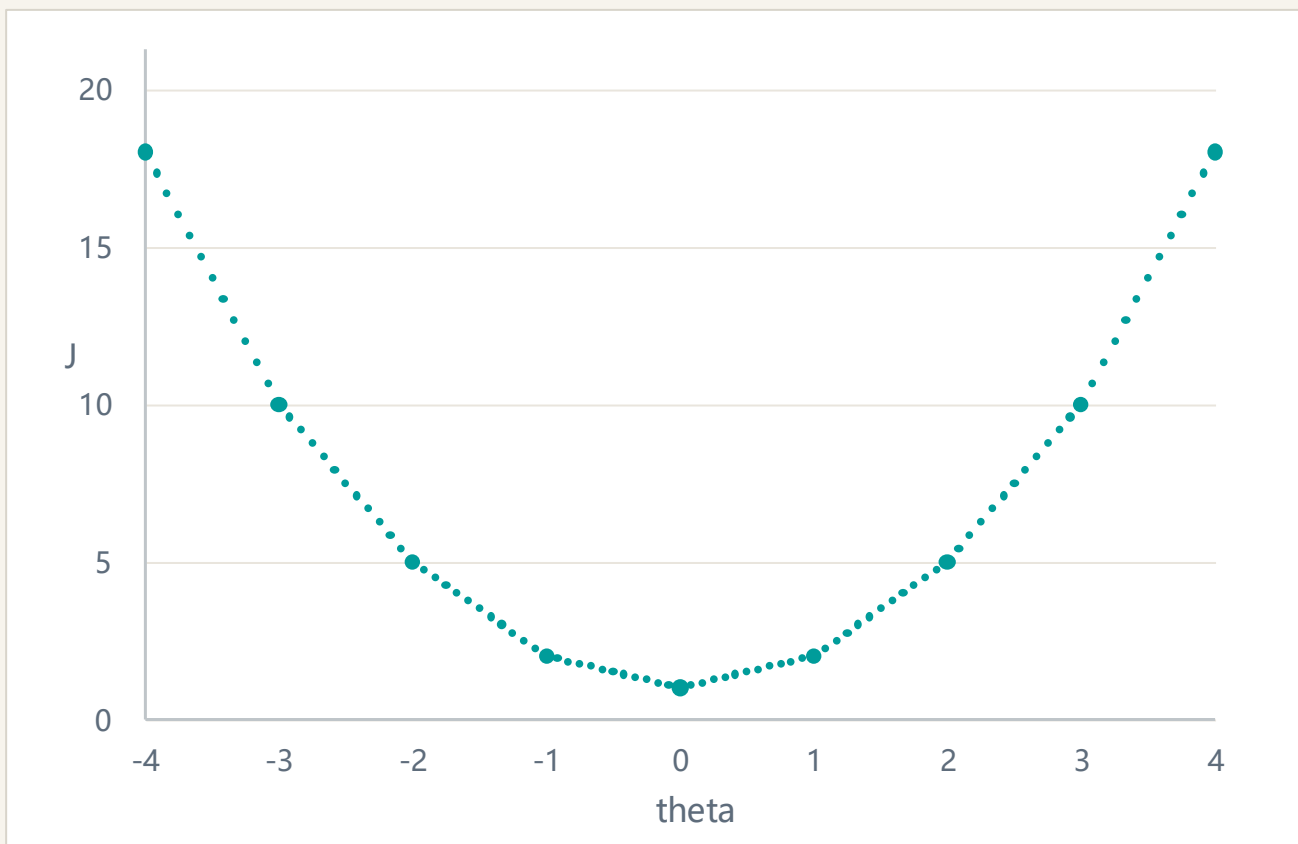
选择  $w$  使  $L(w)$  尽量小

回归常用均方误差

分类常用交叉熵

损失函数隐含了我们希望模型重视什么

# 梯度下降沿局部斜率更新参数



$$w_+ = w - \eta \nabla L(w)$$

$\eta$  太小

收敛慢

$\eta$  太大

震荡或发散

学习率控制每一步走多远

# 导数与梯度

---

一维函数

$$dJ/dw$$

当前点的局部斜率

多维函数

$$\nabla J(w)$$

每个参数偏导数组成的向量

更新方向

$$-\nabla J(w)$$

局部下降最快的方向

**梯度只提供当前位置的局部信息，需要反复计算和更新**

# 一次梯度更新的数值例子

当前参数

$$w = 4$$

梯度

$$dJ/dw = 6$$

学习率

$$\eta = 0.1$$

新参数

$$w = 3.4$$

$$w_+ = 4 - 0.1 \times 6 = 3.4$$

下一步要在  $w=3.4$  处重新计算梯度

# 三种梯度估计方式

---

## 批量梯度下降

每次使用全部训练样本

方向稳定，单次成本高

---

## 随机梯度下降

每次使用 1 个样本

更新频繁，噪声较大

---

## 小批量 SGD (也可以叫随机梯度下降)

每次使用一小批样本

计算效率和噪声之间折中

---

**深度学习默认采用小批量训练**

# 批大小影响速度、显存与噪声

---

## 较小 batch

- 显存占用较低
- 梯度噪声较大
- 单步快但并行度有限
- 每个 epoch 更新次数多

## 较大 batch

- 更充分利用并行计算
- 梯度估计更稳定
- 需要更多显存
- 学习率常需同步调整

**batch size 是计算资源与优化行为共同决定的超参数**

# 分类模型输出各类别的概率



概率需要校准，最高概率也不等于一定正确

# 回归与分类解决不同问题

---

## 回归

### 输出连续数值

房价、温度、剩余寿命

常见损失：MSE、MAE

## 分类

### 输出离散类别

猫狗识别、故障类型、文本主题

常见损失：交叉熵

**同一个输入表示可以连接不同的输出层与损失函数**

# 分类器先输出分数，再决定类别

---

$$\text{logits} = Wx + b$$

猫: 2.1

狗: 1.0

其他: -0.5

**argmax 选择最大分数，因此预测类别为 “猫”**

# One-hot 编码表示类别标签

---

类别集合

[猫, 狗, 其他]

真实类别: 狗

$k = 1$

One-hot

$y = [0, 1, 0]$

**只有真实类别位置为 1，其余位置为 0**

实际代码通常直接保存类别索引，不需要显式构造 one-hot 向量

## 交叉熵只关注真实类别的概率

$$L = - \sum_k y_k \log p_k$$

若真实类别是狗,  $y = [0, 1, 0]$

$$L = -\log p_{\text{狗}}$$

$p_{\text{狗}}$  越接近 1, 损失越小; 越接近 0, 惩罚越大

# 分类中为什么通常不用平方损失

---

## 平方损失

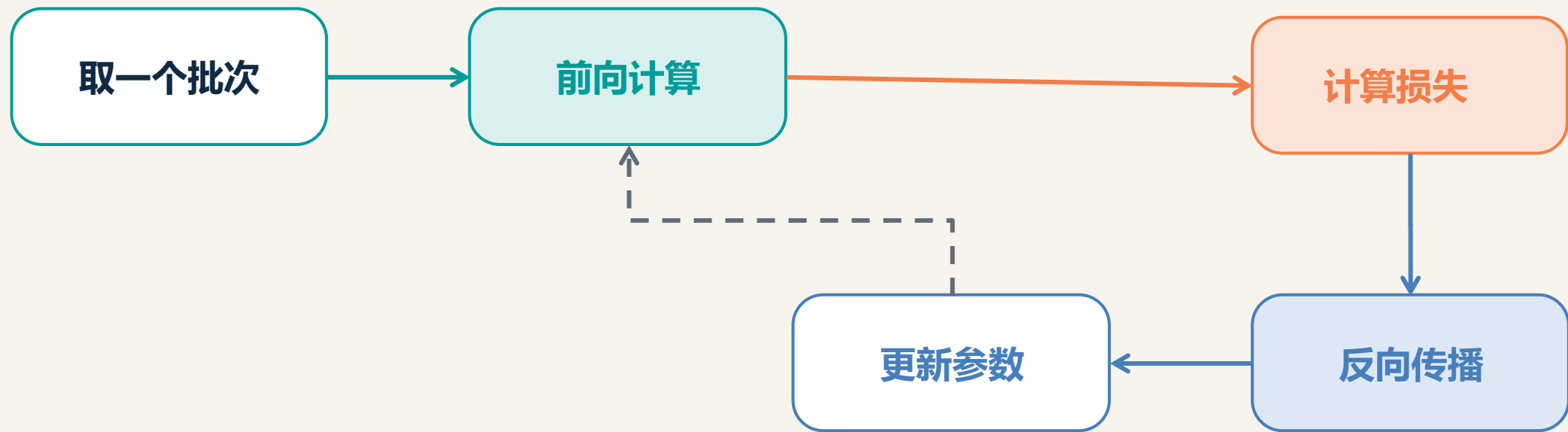
- 把类别概率当作普通连续数值
- 与 softmax 组合时梯度可能较弱
- 概率解释不够自然

## 交叉熵

- 来自分类分布的最大似然
- 直接提高真实类别的对数概率
- 与 softmax 组合计算稳定

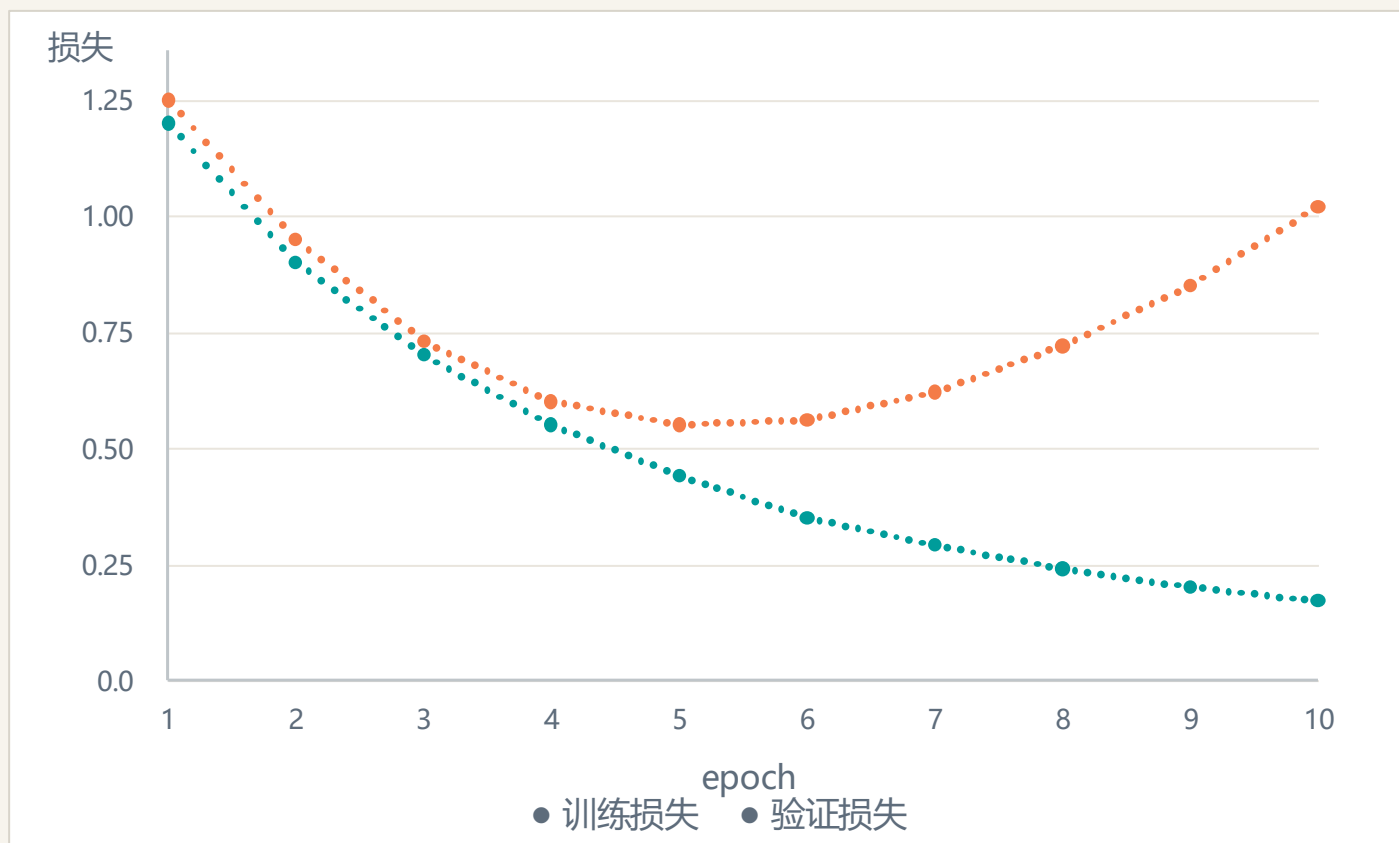
**选择损失函数等于选择模型应当如何看待错误**

# 一次训练迭代的计算图



重复多个批次和多个 epoch，直到达到停止条件

# 过拟合表现为训练与验证性能分离



欠拟合

训练集也表现较差

合适容量

验证性能最好

过拟合

训练继续变好，验证变差

正则化、数据增强和早停都在改善泛化

# 评估指标必须与任务目标一致

|       |                      |             |
|-------|----------------------|-------------|
| 回归    | MAE / MSE            | 误差尺度与异常值敏感性 |
| 分类    | 准确率 / 精确率 / 召回率      | 类别不平衡与错误代价  |
| 排序与检索 | Recall@K / NDCG /MRR | 前几个结果是否最重要  |
| 生成    | 自动指标 + 人工评估          | 质量、事实性与多样性  |

一个数字不能替代误差分析

## 例题：完成一次线性回归前向计算

给定  $x=[2,3]$ ,  $w=[0.5,-1]$ ,  $b=4$ ,  $y=1$

1

预测

$$\hat{y} = w^T x + b = 0.5 \times 2 - 1 \times 3 + 4 = 2$$

2

残差

$$e = y - \hat{y} = 1 - 2 = -1$$

3

平方损失

$$L = (y - \hat{y})^2 = 1$$

思考：怎样调整  $w$  或  $b$  才能让预测从 2 靠近 1？

## 例题：从 logits 到交叉熵

---

**logits = [2, 1, 0], 真实类别为第 1 类**

**指数**  $[e^2, e^1, e^0] \approx [7.39, 2.72, 1.00]$

**概率**  $p \approx [0.665, 0.245, 0.090]$

**损失**  $L = -\log(0.245) \approx 1.41$

**模型把最高概率给了第 0 类，因此对真实类别的损失较大**

# 三道快速检查题

---

1

为什么深度网络需要非线性激活？

2

验证集与测试集的职责有什么不同？

3

训练损失下降，验证损失上升说明什么？

独立思考 3 分钟

# 本节课的三条主线

---

## 深度学习

用多层神经网络从数据中学习表示和预测

## 训练逻辑

数据、模型、损失、优化与评估构成闭环

## 研究视角

能力与失效条件同样重要，泛化是核心目标

**下一次课：前馈神经网络、反向传播与自动微分**