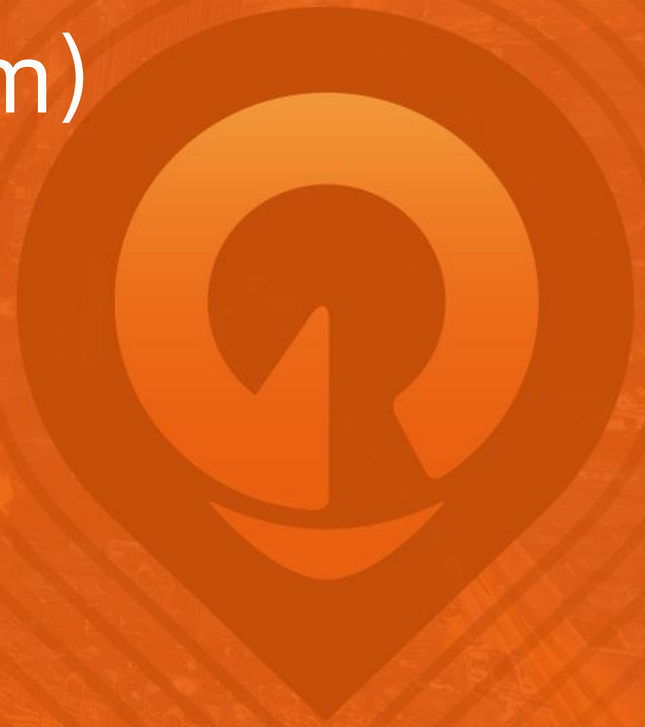
AI 算法进阶（Advanced AI System）



@Tyler

6. 连接主义：CNN 与 RNN

目录

为何需要表示学习?

极客时间 训练营

- **1. 原始数据的困境：脆弱且冗余**
- **原始表示 (Raw Representation):**
  - **高维、冗余：** 如图像中百万级的像素，信息密度低。
  - **高度敏感：** 对平移、光照、遮挡等微小扰动极其敏感。
- **直接学习的后果：**
  - 模型脆弱，泛化能力受限。
  - 容易学到数据中非本质的、偶然的统计相关性。

- **人类理解时，不是一个一个像素看的。**

为何需要表示学习?

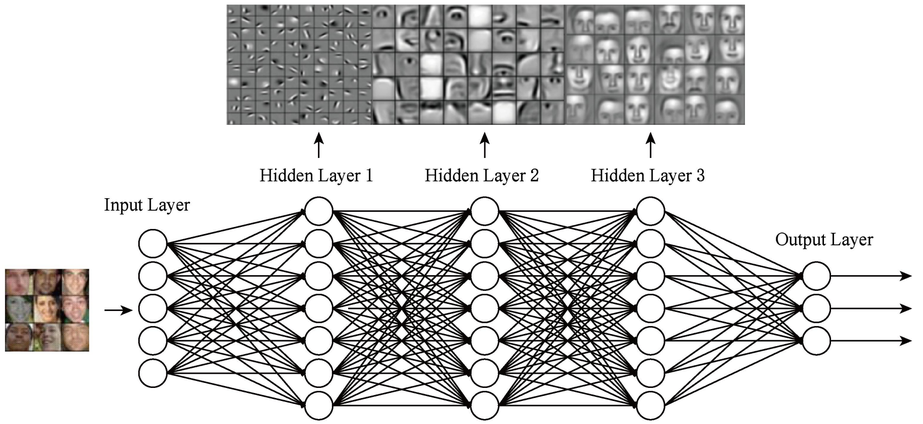
极客时间 训练营

- **2. 传统监督学习的局限：从“精确拟合”到“真正理解”**
- **传统目标：** 找到一个从输入到输出的精确**映射函数**。
- **潜在陷阱：** 当模型仅追求预测准确性时，可能会学到“**表面相关性**”。

- **3. 一个典型的失败案例：**
- **场景：** 训练集中，大部分“猫”的图片都带有特定水印。
- **错误学习：** 模型将**“水印”**错误地识别为**“猫”**的一个核心特征。
- **结论：** 这种基于“**记忆**”而非“**理解**”的模型，一旦数据分布变化（如遇到没有水印的猫），性能便会急剧下降。

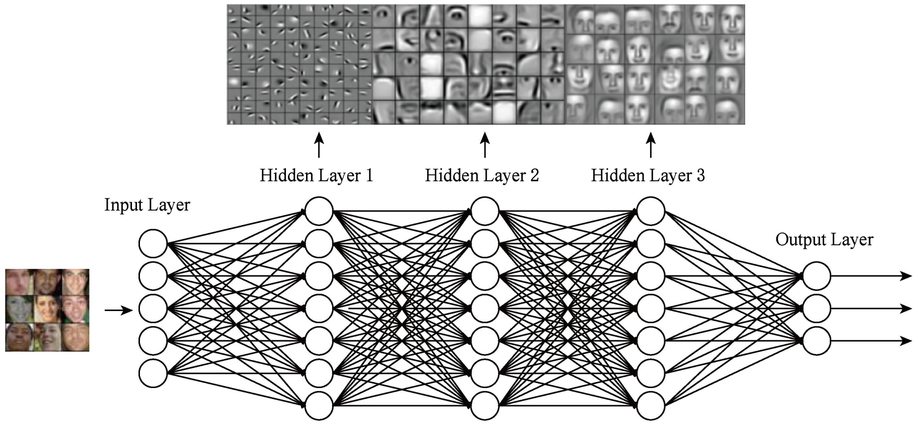
层次化的抽象过程

- **1. 什么是有效的表示？—— 层次化 (Hierarchical)**
- **核心思想：** 对复杂事物的理解是一个由简到繁、逐层抽象的**泛化过程**。
- **类比人类视觉认知：**
  - **低层表示 (Low-level):** 捕捉基础、局部的感知特征。
    - *示例：* 边缘、角点、颜色、纹理。
  - **中层表示 (Mid-level):** 组合低层特征，形成更复杂的结构基元。
    - *示例：* 物体的部件（眼睛、车轮）、形状轮廓。
  - **高层表示 (High-level):** 聚合中层结构，形成具有明确语义的抽象概念。
    - *示例：* “人脸”、“汽车”、“猫”。



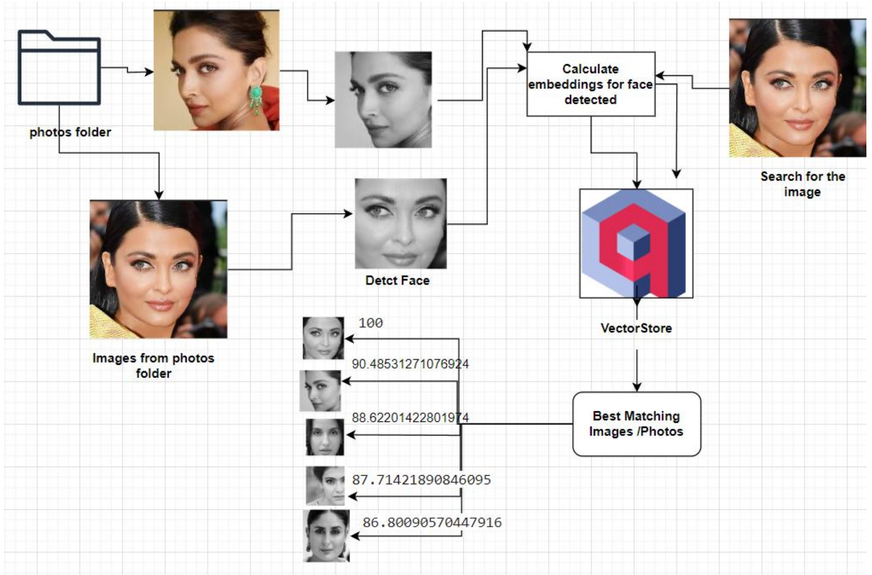
层次化的抽象过程

- **2. 深度神经网络 (DNN) 对该机制的模拟**
- **DNN 的每一层 ≈ 认知的一个抽象层次：**
  - **底层网络：** 提取基本的边缘和轮廓。
  - **中层网络：** 组合形状和纹理。
  - **高层网络：** 形成语义概念和类别。
- **核心机制：“层层提取 → 层层组合 → 层层抽象”**
- **带来的价值：** 模型不仅能完成单个预测任务，更有可能将在一个任务中学到的表示，**迁移 (Transfer)** 到其他相关任务中。例如，右图的中间层可以用于人脸识别，性别识别，年龄识别，颜值识别等等。



构建表示空间与新的学习范式

- **1. 使用深度学习的真正力量是：构建高效的“表示空间”(Representation Space)**
- **功能：** 深度神经网络 (DNN) 不仅是特征提取器，更是一个**知识的组织与构建系统**。
- **过程：** 它将高维、冗余的原始输入，投影到一个**更低维、结构化的流形**上。
- **特性：** 在这个空间里：
  - **语义相似** 的数据点在几何上相互 **靠近**。
  - **语义不同** 的数据点则相互 **远离**。
- **价值：** 这个“表示空间”本身，就是模型“理解”世界的形式，它既是**信息压缩**的结果，也是**知识迁移**的基础。



构建表示空间与新的学习范式

- **2. 学习目标的演进：从映射到知识**

学习阶段	核心问题	实现方式
监督学习(Supervised Learning)	如何通过标签最小化预测误差？	从数据中学习输入与输出的映射关系。
深度学习(Deep Learning)	如何用结构化方式提升建模能力？	多层神经网络，端到端可微优化。
表示学习(Representation Learning)	如何学到可迁移的结构化表示？	构建压缩有效、可泛化的中间表示。

- **最终结论：** 表示学习是深度学习从**“压缩记忆”**走向**“组织知识”**的关键一步，是支撑现代AI（如预训练大模型）进行**迁移与泛化**的底层逻辑。

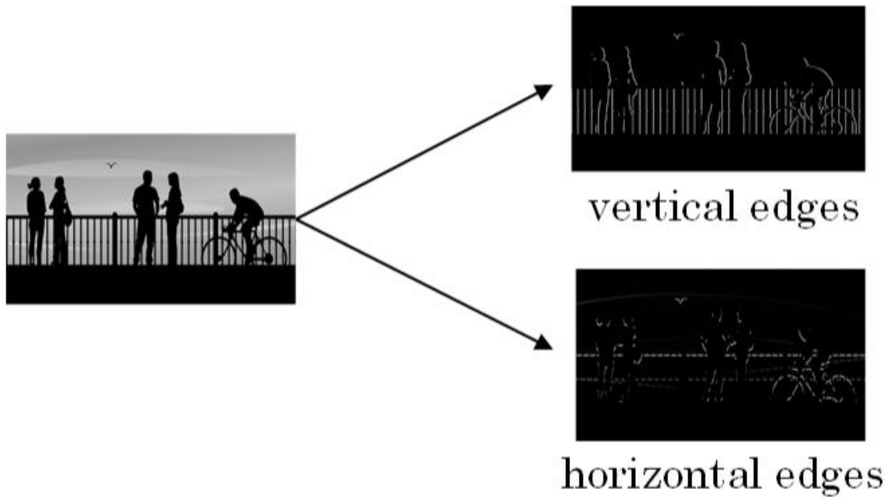
卷积神经网络 (CNN)：为图像表示而生的架构

- **CNN 的核心目标：**一种为解决**图像**这类高度结构化数据的表示学习问题而设计的**专用架构**。
- **目标：** 利用图像固有的**结构先验**，自动学习一个从原始像素到抽象语义的、**层次化的、可泛化的表示空间**。

- 物理世界的特点：**物质空间是连续的，实体本身也是连续的**。因此**边缘**是最重要的视觉特征。

卷积神经网络 (CNN)：为图像表示而生的架构

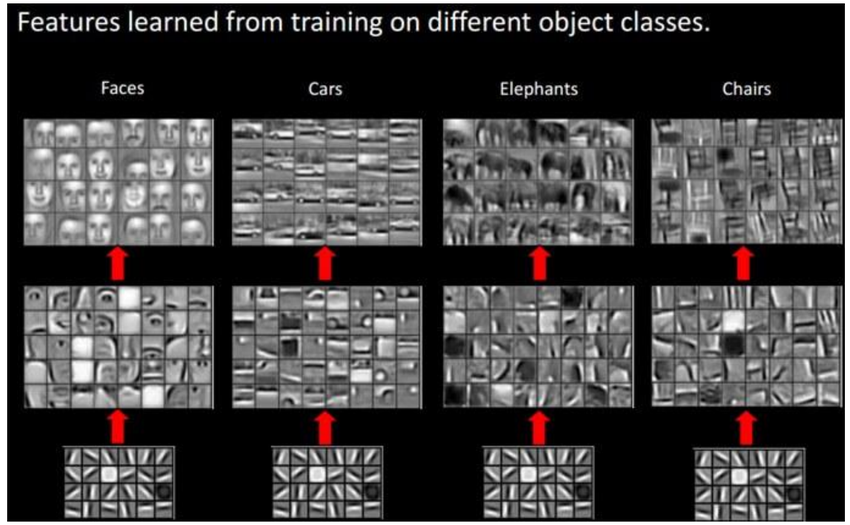
- **设计基石：图像的“局部相关性”(Locality)**
- **图像特性：** 相邻像素间高度相关，共同构成有意义的局部模式（如边缘、角点、纹理）。
- **识别逻辑：** 对物体的理解，依赖于对其**局部模式的组合**，而非对全局像素位置的记忆。
- **CNN 策略：** 将学习过程聚焦于识别这些局部模式，再通过堆叠组合成全局概念。



CNN 的工作机制：构建逐层抽象的表示体系

随着网络层数的增加，模型学习到的特征从具体到抽象，从局部到整体。

- **底层卷积层 (Low-level Layers):**
  - **功能：** 检测图像中最基本的视觉元素。
  - **示例：** 边缘、线条、方向变化、颜色块。
  - **感受野 (Receptive Field):** 小
- **中层卷积层 (Mid-level Layers):**
  - **功能：** 组合低级模式，构成更复杂的局部结构。
  - **示例：** 角点、形状、纹理块、物体部件（如“眼睛”、“车轮”）。
  - **感受野:** 中
- **高层卷积层 (High-level Layers):**
  - **功能：** 聚合中层信息，提取出具有**明确语义**的区域或整体特征。
  - **示例：** “动物面部”、“汽车轮廓”等完整概念。
  - **感受野:** 大



CNN 的工作机制：构建逐层抽象的表示体系

- **从像素到语义的路径**

- 原始像素 ➡ 边缘/纹理 ➡ 物体部件 ➡ 高级语义概念

- **关键机制：** 随着网络加深，模型的**感受野**（Receptive Field）不断扩大，使得网络能够综合越来越大的图像区域信息，最终形成适用于分类、检测等复杂任务的**高层语义表示**。

残差网络 (ResNet)：突破深度学习的瓶颈

极客时间 训练营

- **1. 深度带来的悖论：更深，为何反而更差?**
- **理论上：** 更深的网络拥有更大的函数空间，应具备更强的表示能力，效果应该更好或至少不差。
- **实践中：** 在CNN研究中发现，简单地堆叠网络层数，当深度达到一定程度后，模型性能会急剧下降。

- **2. 深度网络面临核心瓶颈**
- **性能退化 (Degradation):**
  - **现象：** 一个更深的模型（如56层）在**训练集和测试集**上的表现，反而劣于一个相对较浅的模型（如20层）。

残差网络 (ResNet)：突破深度学习的瓶颈

极客时间 训练营

- **梯度消失 (Vanishing Gradients):**
  - **原因：** 在反向传播过程中，梯度通过链式法则逐层传递。当网络过深，**多个小于1的梯度值连乘**，会导致传递至浅层网络的梯度信号极其微弱，近乎于零。

$$\frac{\partial L}{\partial W_{old}} = \frac{\partial L}{\partial y_{new}} \cdot \frac{\partial y_{new}}{\partial y_{old}} \cdot \ldots \cdot \frac{\partial y_{start+1}}{\partial y_{start}} \cdot \frac{\partial y_{start}}{\partial W_{old}} \to 0$$

  - **后果：** 浅层网络的参数几乎无法得到有效更新，导致训练停滞。

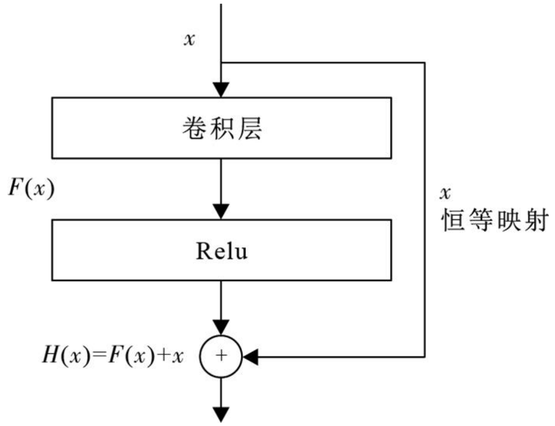
- **核心挑战：** 如何构建一种更稳健的**信息与梯度传播机制**，从而有效利用深度带来的表示优势，让网络“想深就能深”？

ResNet 的核心创新：残差连接

- **传统网络 (Plain Network):** 每一层的目标是学习一个从输入 x 到输出 H(x) 的复杂映射。在网络很深的时候，让每一层都学习到一个完美的恒等映射（即 H(x) = x）都非常困难，因为这需要将无数的参数（权重和偏置）精确地调整到特定的值。

- **残差网络 (ResNet):** 将学习目标改变了。它不再直接学习 H(x)，而是学习一个“差值”或“修正量”，即 **残差 F(x) = H(x) – x**。这样，原始的映射就变成了：

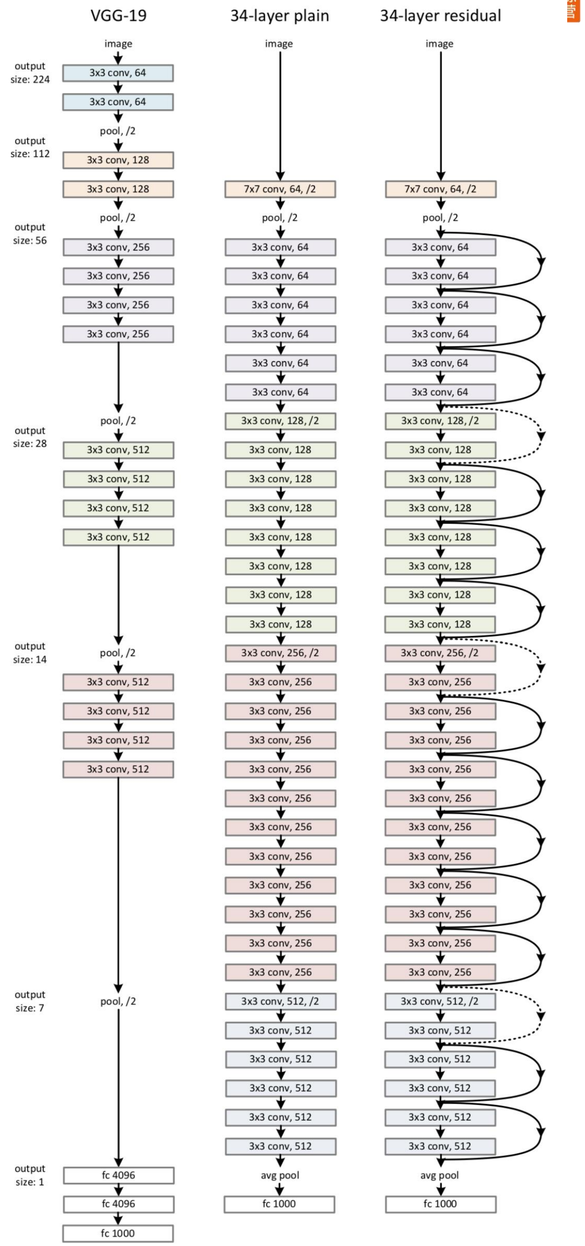
$$H(x)=F(x)+x$$



- **梯度包含一个 +1 项：**这个通道确保了**即使残差块自身的梯度 ∂ F/∂ x 很小**，信息（梯度）也能够畅通无阻地在层与层之间传递，从而极大地缓解了深度网络中的梯度消失问题，使得训练数百甚至上千层的网络成为可能。

ResNet 的架构实现与深远影响

- 实际的 ResNet 架构由两种核心的**残差块 (Residual Block)** 作为基本构件，通过堆叠这些构件来构建极深的网络。
- **恒等块 (Identity Block):**
  - **用途：** 当输入与输出的维度**完全相同**时。
  - **捷径路径：** 直接的恒等连接，将输入 x 原封不动地加到输出上。
- **投影块 (Projection Block):**
  - **用途：** 当主路径的卷积操作改变了输入的空间维度（如步长为2）或通道数时。
  - **捷径路径：** 通过一个 **1x1 卷积**对输入 x 进行线性投影，使其维度与主路径的输出匹配，然后再相加。
- **成果：** 基于此模块化设计，诞生了 ResNet-18、ResNet-34、ResNet-50、ResNet-101 等一系列经典架构，其深度从几十层到上百层不等。



ResNet 的架构实现与深远影响

极客时间 训练营

- **从表示学习视角：**
  - ResNet 提供了一种**稳定、增量式**地构建高阶语义表示的通用机制。
  - 每个残差块都在已有表示的基础上进行**精炼和补充**，而非完全重构。形成一种**“恒等映射保留旧知识，残差学习探索新知识”**的高效学习模式。

- **深远影响：**
  - 残差连接的理念已远超计算机视觉，成为现代深度学习不可或缺的**基础设施**。
  - 它被广泛应用于自然语言处理（如 **Transformer**）、语音识别等众多领域，是深度学习架构设计的一次**范式革命**。
  - 同时，它也成为了第一代**预训练视觉模型的基石**。

目录

从“温室”到“荒野”：AI模型的泛化挑战

极客时间 训练营

- **理想的假设：温室中的 AI 训练（I.I.D. 世界）**

- **核心前提：训练与测试数据服从相同分布**
- 在传统监督学习中，我们假设训练数据和测试数据来自同一分布（**i.i.d.：独立同分布**），这是模型泛化能力成立的基础。

- 这就像在一个受控的温室里，所有植物（数据）都遵循着已知的、统一的生长规律。

从“温室”到“荒野”：AI模型的泛化挑战

极客时间 训练营

- **理想破灭：I.I.D. 假设在真实世界频繁失效**

- 现实世界并非温室，而是复杂多变的“**AI 荒野**”——模型必须面对前所未见的环境与分布。
- **分布变化（Domain Shift）**→ *训练在晴天，应用在雨雪夜*
  自动驾驶模型需适应多种天气与路况条件。
- **任务变化（Task Shift）**→ *从猫狗识别转向野生动物分类*
  原用于宠物图片识别的模型，在野外监测任务中需处理全新物种、不同拍摄角度与标签体系。
- **风格变化（Style Shift）**→ *高清图训练，模糊图部署*
  图像识别系统需从网络高分辨率照片迁移到监控视频中的低质、非标准图像。

从“温室”到“荒野”：AI模型的泛化挑战

极客时间 训练营

- **在熟悉环境中表现优异：**
- 多数模型在训练数据分布内（i.i.d.）表现亮眼，准确率高、收敛快。
- **一旦分布外（OOD），性能急剧下滑：**
- 面对**新任务、新分布、新场景**，模型不再可靠
- 原因在于其“知识”仅限于训练时见过的数据类型与标签体系

- **核心问题：**
- **如何让模型走出“数据舒适区”？**
  如何让它不再只是“背熟答案”，而是真正具备**泛化到未知任务的能力**？
- 我们需要一座桥梁，将模型从**通用知识的积累**引向**专业任务的适配**——
  这正是**迁移学习**与**预训练范式**所要解决的核心问题。

破局之道：迁移学习（Transfer Learning）

极客时间 训练营

- **不是具体算法，而是一种通用策略：**
- 迁移学习（Transfer Learning）**指的是：**
  将模型在一个大规模、通用任务（源任务）**上学到的知识，**
  迁移并适配到一个新的、具体但相关的任务（目标任务）上。

- **比喻启发：通才到专家的成长路径**
- 在“温室”中训练模型解决标准任务，积累**通用知识与表征能力**
- 在“荒野”中遇到新任务，通过少量调整，快速转化为领域“专家”
- 就像先培养一个知识全面的“通才”，再派去特定环境中快速胜任新角色

破局之道：迁移学习（Transfer Learning）

极客时间训练营

- **本质特征：知识的可迁移性**
- 不是从零开始学习，而是**复用已有认知结构**
- 减少对目标任务数据的依赖，提升泛化能力与样本效率

迁移学习的基石：表示学习

成功的迁移，源于成功的表示

本质理解：

迁移学习的价值，并不在于“复用一个模型”，
而在于复用其在大规模任务中学到的**高质量表示能力**（Representation Power）。

分工协作：表示学习 × 迁移学习

阶段	角色	功能
Step 1□	**表示学习（Representation Learning）**	构建“知识字典”：将复杂输入转化为可泛化的特征向量（Embedding）
Step 2□	**迁移学习（Transfer Learning）**	在下游任务中调用已有特征，少量数据即可高效适配

迁移学习的基石：表示学习

极客时间训练营

- **表示学习三要素：**
- **目标**：将原始数据（如图像像素）编码成结构化特征
- **方法**：在大规模数据（如 ImageNet）上进行自监督 / 有监督预训练
- **产物**：一个可广泛复用的**特征提取器（Encoder）**

迁移学习的基石：表示学习

- **ResNet + ImageNet → 通用视觉表示**
- **问题引出：**
- 在 ImageNet 上预训练的 ResNet，**到底学到了什么？**

- **答案超越分类本身：**
- ResNet 并不仅仅学会了“识别1000类物体”，
  它在这个过程中，被**迫使学会了一种更底层的能力**：
- 一种能将图像**编码为语义向量**的通用视觉表示系统。

迁移学习的基石：表示学习

极客时间 训练营

- **表示空间的意义：**
- 这个由 ResNet 编码出的向量空间，具有两个关键特性：
- **语义一致性**：相似图像 → 相近向量
- **任务通用性**：同一向量 → 可用于分类、检索、检测等多种任务
- 因此，ResNet 的输出不仅是“分类器的中间产物”，更是**所有下游任务的通用语言**。

迁移学习的基石：表示学习

极客时间 训练营

- **迁移学习如何接入：**
- 表示学习阶段：构建**高质量的语义空间**（如 ResNet 编码器）
- 迁移学习阶段：在此空间中，通过少量样本**“解码”出特定任务答案**
- 模型没有“换轨”，而是在**同一个表示空间中灵活适配多种任务**

核心产物：通用特征向量（Embedding）

极客时间 训练营

- **模块化迁移结构：适配千变万化的下游任务**
- 无论任务是分类、检测还是检索，迁移学习的工程范式**高度统一**——都可抽象为：
- **主干网络（Backbone） + 任务头（Task Head）**

核心产物：通用特征向量（Embedding）

极客时间 训练营

- **1 表示主干（Embedding Backbone）**
- 通常是一个预训练模型（如 ResNet），**去除最后的分类层**
- 作用：将原始图像 → **压缩为语义丰富的特征向量（Embedding）**
- 状态：常在下游任务中**保持冻结**，仅作特征提取器使用

- **2 任务头（Task-Specific Head）**
- 一个轻量模块（如 1~2 层全连接网络）
- 作用：将特征向量 → **映射到特定任务的输出空间**
- 状态：**唯一需要训练**的部分，负责快速适配任务目标

核心产物：通用特征向量（Embedding）

极客时间 训练营

- **核心目标：**
- 提取 ResNet 在 ImageNet 上预训练得到的**通用视觉表示（Embedding）**，作为迁移学习中最宝贵的“知识资产”。
- **实现路径（以 PyTorch 为例）：**

在 PyTorch 中，仅需三步即可完成特征提取：

```
import torchvision.models as models
import torch.nn as nn
# 加载预训练的 ResNet50
resnet = models.resnet50(pretrained=True)
# 移除最后的分类层（FC Layer），保留为特征提取器
backbone = nn.Sequential(*list(resnet.children())[:-1])
```

核心产物：通用特征向量（Embedding）

- 此时，backbone(input_tensor) 的输出为：

输出维度: [batch_size, 2048, 1, 1]

去除多余维度，得到最终的特征向量

embedding = backbone(img).squeeze() # → [batch_size, 2048]

- **关键理解：**
- 这个 [batch_size, 2048] 的向量，**不是分类结果**，而是：
- 一个浓缩了图像语义的高维表示
  可用于分类、检索、聚类、风格分析等多种任务
  是迁移学习的“共享语言”

核心产物：通用特征向量（Embedding）

- **迁移学习的工程三重利好：**
- **模块化复用：结构清晰、组合灵活**
- **主干网络**（如 ResNet）作为标准化的**特征提取引擎**
- 不同任务仅需接入各自的 **Task Head**，即可快速部署
- 支持多任务、多场景的一体化开发流程

核心产物：通用特征向量（Embedding）

- **训练高效：数据与算力需求大幅降低**
- 下游任务仅需在**低维表示空间**上训练一个轻量级头部网络（如 FC 层）
- 不依赖大规模标注数据，几分钟即可完成一次迁移学习微调
- 特别适用于资源受限环境（如边缘部署、小型样本）

核心产物：通用特征向量（Embedding）

极客时间 训练营

- **通用适配：统一的“知识接口”**
- ResNet 的特征向量（Embedding）作为**固定表示空间**，天然支持：
  - 图像分类（Classification）
  - 图像检索（Retrieval）
  - 对象检测（Detection）
  - 图像聚类（Clustering）
- 不同任务在**同一语义空间**中展开，迁移路径一致、泛化能力强

通用编码器的蜕变：ResNet的深度 × ImageNet的广度

- **不只是“识别1000类”，而是学会理解视觉世界**
- 在 ImageNet 这一开放多样的“视觉宇宙”中，
  ResNet 并未仅仅被训练成一个**任务特定的分类器**，
  而是被**逼迫进化**为一个具备**通用感知能力的编码器系统**。

- **结构深度 × 数据广度 → 表示进化**
- 深层网络结构：提供足够的表达能力捕捉多层次视觉规律
- 海量图像任务驱动：强制模型构建从**像素 → 边缘 → 形状 → 概念**的**多层抽象体系**
- 最终形成了一个“**从底层视觉信号到高阶语义表征**”的完整编码链条

通用编码器的蜕变：ResNet的深度 × ImageNet的广度

- **通用视觉编码器（Universal Visual Encoder）**
- 这个由 ResNet 主干学到的视觉编码器，具备：
- **语义抽象能力**：可识别相似形状、风格、语境中的共同结构
- **任务无关性**：编码结果可被任意任务头解读与利用
- **高迁移性**：在新任务中“换头不换身”，即插即用

- **迁移学习的知识基石**
- **ResNet Embedding** 并非偶然产物，而是**深度结构 + 表示任务 + 大规模训练**共同催化的结果
  它是迁移学习一切下游任务得以成功的**通用知识载体**

目录

从图像到文字：Embedding 表示的另一种挑战

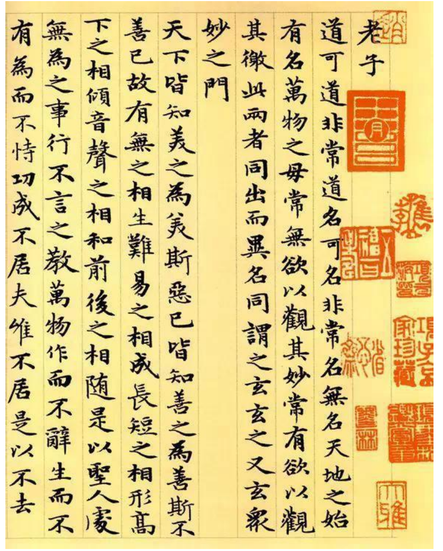
- 在图像任务中，我们通常通过卷积神经网络（CNN），将一张二维像素图压缩成一个结构化的表示空间。例如，ResNet 可将图像编码为一个固定维度的 **Embedding 向量**，该向量凝聚了从边缘到物体语义的多层抽象信息。

- 然而，当我们转向语言任务时，问题本质发生了转变：
- **我们不再处理像素阵列，而是需要理解由词语组成的线性序列。**

为什么 CNN 不适合语言建模?

- 在视觉任务中，CNN 的成功建立在一个关键的结构性先验之上：**空间局部性（spatial locality）**：图像中**相邻像素**之间存在强烈的结构关联；局部特征（如边缘、纹理）可以通过卷积核的滑动操作被逐步提取并组合成全局语义。
- 然而，当我们转向语言任务时，这一结构先验不再成立：文本是一维序列，词语之间的**关系由顺序决定**，而非空间邻近；
- “猫在追狗”与“狗在追猫”的词汇排列可能相似，但**语义方向完全相反**；决定含义的，不是词之间的物理邻接，而是它们在句法与语义结构中的位置和依赖关系。

- 因此：**CNN 的滑动窗口机制无法感知词序的差异，也无法捕捉语言中的上下文依赖；**它缺乏一种**时间敏感的“记忆机制”**，来处理语言中“当前词”与“前文语境”之间的动态关系，这正是我们需要引入循环神经网络（RNN）的根本原因。

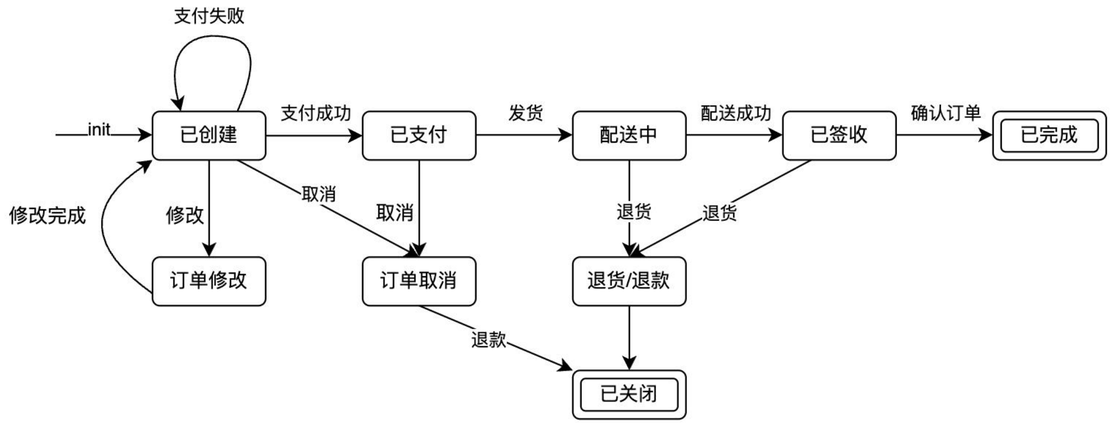


语言的本质：时序结构与状态传递

- 语言是一类**典型的时序数据（Sequential Data）**，其核心特征在于：
- 每个词语的含义**依赖于其在序列中的位置**以及**前后上下文**；
- 句子的语义通常**不能通过孤立地理解单词来获得**，而是需要通过对词序列的**逐步解析与语义累积**来建构。
- 这带来一个关键结论：
- **语言理解的本质，是一个动态状态建模（Finite-State Modeling）过程。**

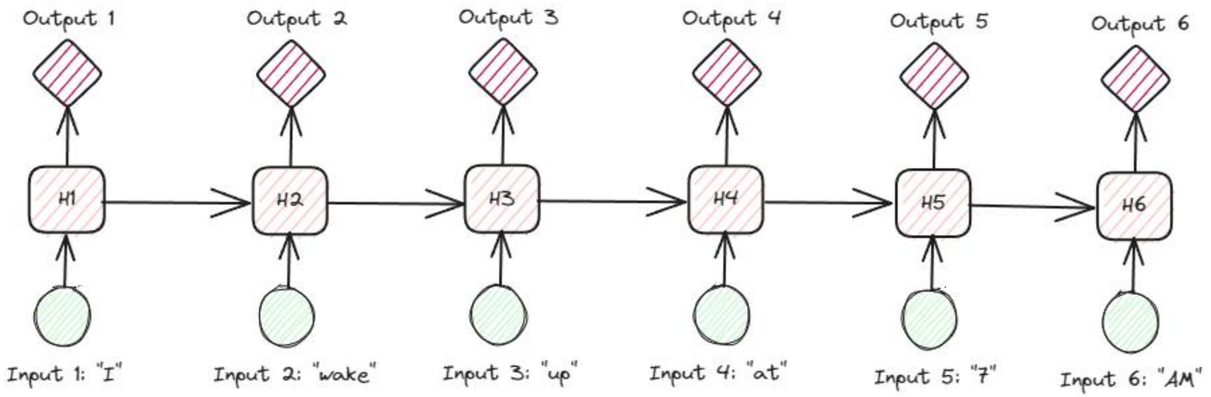
语言的本质：时序结构与状态传递

- 在这个过程中：
- 模型每接收一个新词，都应基于已有的语境更新其**内部状态**；
- 这个状态应“记住”前文的关键信息，为理解当前词和预测下一个词提供条件；
- 如同一个有限状态机（Finite-State Machine），模型需要具备**对历史信息的记忆与更新机制**，以构建完整语义。
- 这类需求，正是传统神经网络（如全连接网络）难以胜任，而循环神经网络（RNN）应运而生。



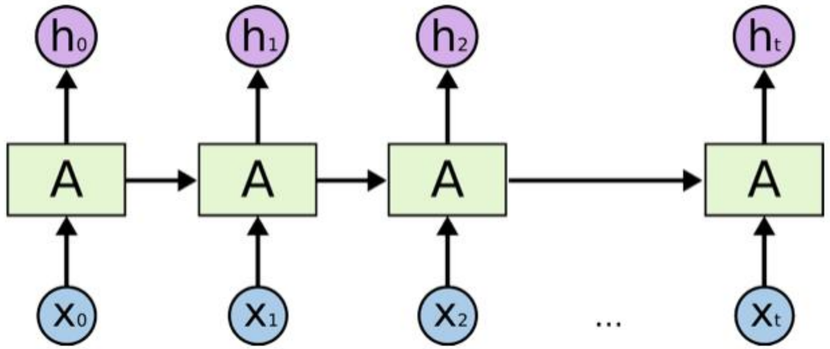
表示学习的关键挑战：需要“记忆”的能力

- 构建语言表示的核心挑战在于：
- **模型不仅要理解当前词语，还必须保留并更新对先前语境的理解。**
- 这与人类阅读句子的方式高度相似：
- 当我们读到“如果明天下雨，……”时，并不能立即确定整句话的意义；
- 但我们会**主动保留**“如果”和“明天”这类语义线索，在心理中维持一个**语境记忆**，等待后续内容补全语义链条。



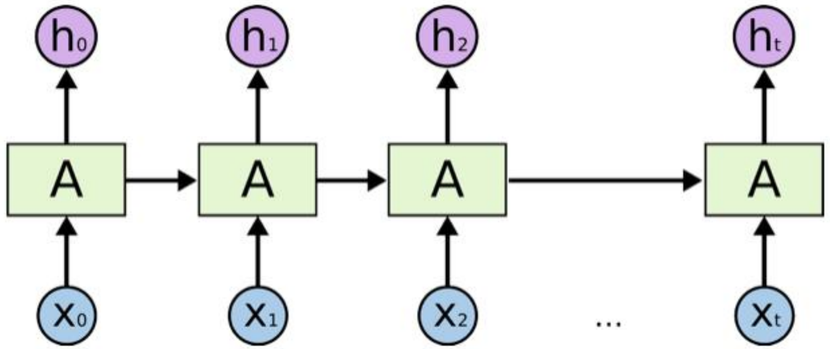
循环神经网络（RNN）：为记忆而生的语言建模架构

- RNN 的设计初衷，是为了解决语言任务中的两大关键问题：
- **顺序建模（Sequential Modeling）与记忆表达（Memory Representation）**
- 其基本机制可以概括为三步：
- 每一步输入一个词向量；
- 同时维护一个“隐藏状态”（Hidden State），作为上下文的“压缩记忆”；
- 当前输出依赖于**当前词**与**过去状态**的组合。



RNN 的局限性：记忆容量有限

- 尽管 RNN 引入了“隐藏状态”作为语义记忆载体，理论上具备对序列历史的建模能力，但在实际应用中，它的**有效记忆能力却极为有限**。
- 在较长序列中：
- 前部词语的影响必须**跨越多个时间步**才能作用到当前；
- 导致早期输入的监督信号被削弱甚至丢失，模型“遗忘”了前文关键信息。



RNN 的局限性：记忆容量有限

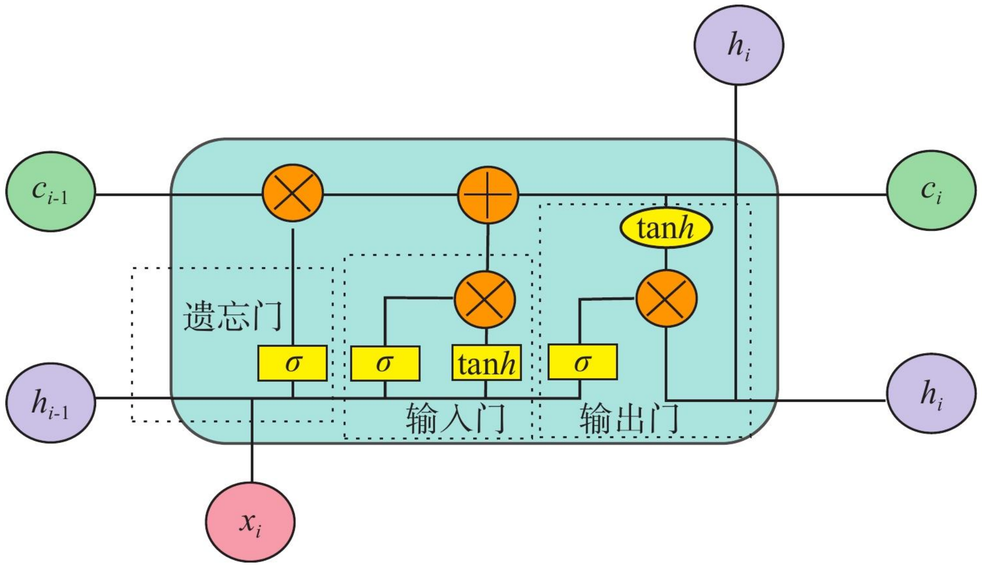
极客时间 训练营

- 要想构建具备**长期记忆能力**的语言模型，RNN 还需结构性的改进 ——
  这正是门控机制（如 LSTM）的研究动因。

- 为突破 RNN 的“记忆瓶颈”，研究者提出了**长短时记忆网络（Long Short-Term Memory, LSTM）**，
  其设计目标是：
- **在不丢失短期灵活性的前提下，引入稳定可靠的长期记忆通道，作为一个外挂记忆单元。**

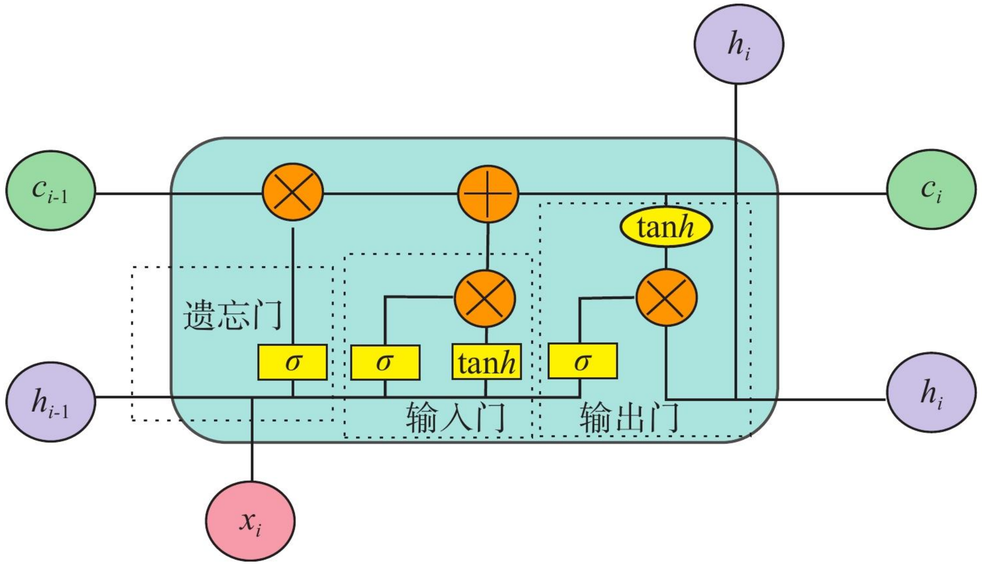
LSTM 的革新：长时记忆与门控机制

- **核心机制：引入显式的“长期记忆单元”**
- LSTM 在原始 RNN 的基础上，新增了一个关键结构：
- **细胞状态（Cell State）**：
  一条贯穿整个序列的外挂“**信息白板**”，专用于长期信息的存储与传递。
- 为了避免信息“泛滥”或“遗忘”，LSTM 通过三组**门控机制**，精细调控记忆的写入、保留与输出：



LSTM 的革新：长时记忆与门控机制

- **三大门控结构：信息流的守门员**
- **遗忘门（Forget Gate）**
  - 控制从旧记忆中保留多少
  - 类似橡皮擦：哪些内容该擦除？哪些该保留？
- **输入门（Input Gate）**
  - 决定当前输入中有多少新信息写入记忆
  - 类似笔：哪些新内容值得记下？
- **输出门（Output Gate）**
  - 控制当前时刻暴露多少内部记忆给外部使用
  - 类似讲稿：白板上的内容，当前需要说出哪些？



LSTM 的革新：长时记忆与门控机制

极客时间 训练营

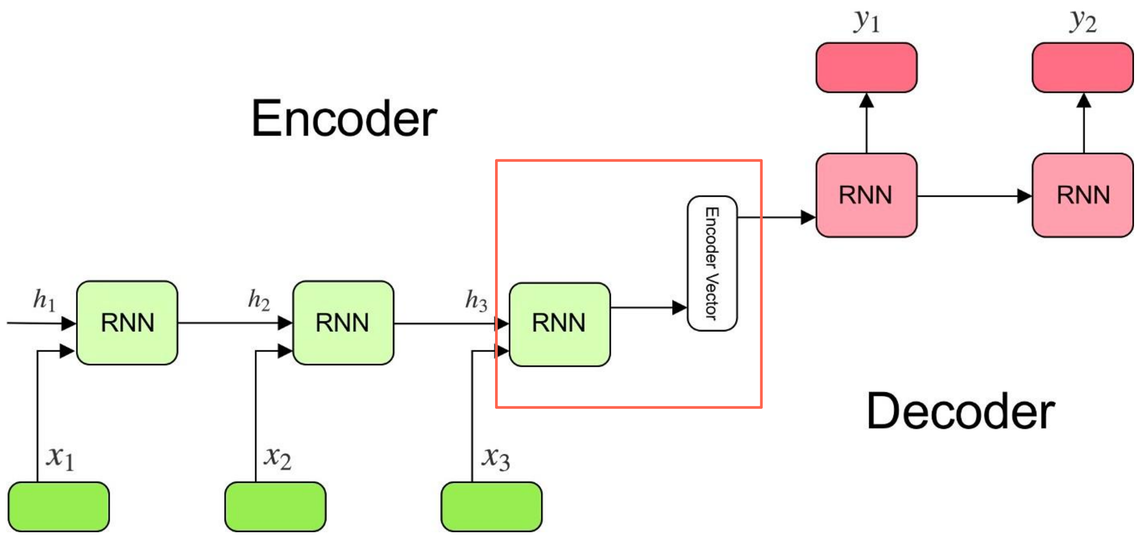
- 引入 RNN（尤其是 LSTM）之后，神经网络终于具备了处理时序数据的关键能力：
- **能够将一个序列的上下文信息“记住”在模型的内部状态中，并在后续预测中加以利用。**

- 换句话说，我们不再仅仅处理孤立的输入词，而是拥有了**一种动态演化的时序语义表示机制**，能够随着序列推进不断吸收新信息、**保留关键记忆**，并在整个句子的理解过程中起作用。

目录

语言表示的高成本：监督学习的局限

- 在前文中，我们已经学习了如何使用 **RNN / LSTM** 来处理时序数据。
  其中一个重要应用是：
- **将序列最后一个时间步的隐藏状态作为整段文本的语义表示（Embedding）**
- 此时，LSTM 实际上扮演了一个**编码器（Encoder）的角色：**
  它接收词序列作为输入，通过不断更新隐藏状态，最终生成一个固定维度的向量来表示整个句子的语义。



语言表示的高成本：监督学习的局限

极客时间 训练营

- **有监督任务中的 LSTM 表示学习机制**
- 在文本分类、情感分析等有监督任务中，LSTM 通常通过**提取末位隐状态（hidden state）**来获得整段文本的语义表示，并将其输入到分类头（如 MLP）以预测标签。
- 该机制依赖于大量**人工标注数据**，通过最小化分类损失，LSTM 被训练为**压缩时序信息并对齐类别标签的语义结构**。模型逐步学会将语义相似的文本映射到相同类别，从而构建出针对任务的判别性表示空间。
- 然而，这种训练方式**标签依赖强**：缺乏标签时难以学习有效表示；

- 因此，该方式虽有效，但**难以支撑通用语言理解任务中对广义语义表示的需求**，为后续的自监督预训练机制埋下伏笔。

语言表示的高成本：监督学习的局限

极客时间 训练营

- **一个自然的问题是：**
- 是否可以**不依赖标签**，仅通过语言自身的结构规律，训练 LSTM 自动学习通用的语言表示?

- 答案是肯定的。
  只需设定恰当的**自监督学习任务**，例如：
- **预测下一个词**；
- 重构被遮盖的词；
- 判断两个句子的顺序关系；
- 就能让模型**在没有标签的情况下，自发学习语言的结构与语义规律**。

从预测下一个词到语义表示

- 在众多自监督学习策略中，Causal Language Modeling（CLM，自回归语言建模）是一种最基础也最常用的语言建模范式。
- **任务定义：**
- 给定一段文本的前缀序列，预测下一个词出现的概率。原文：“**我今天心情很好**”
- **示例：**
- 输入序列为：
- “我今天心情很”
- 模型应预测的下一个词为：
- “好”
- 即使没有标签，模型也可以通过“预测未来”的方式从大规模文本中自动构造训练样本，这正是**自监督学习**的核心思想。

从预测下一个词到语义表示

- **方法优势：**
- 完全基于语言本身的结构，**无需人工标注**；
- 任务形式与人类语言生成方式一致，**符合自然语言的生成顺序**；

- 训练后可作为通用的语言表示器或生成器，适用于文本分类、生成、问答等多种下游任务。
- 因此，CLM 是构建预训练语言模型（如 GPT 系列）的基础任务之一，通过“预测下一个词”的训练目标，引导模型自动学习语言的**语法结构、语义关联与上下文依赖**。

从预测下一个词到语义表示

- **模型结构：使用 LSTM 构建自回归语言模型（CLM）**
- 在 Causal Language Modeling（因果语言建模）中，可以使用 LSTM 作为核心架构，构建一个端到端的语言建模系统。该系统的目标是在给定前缀的条件下，预测下一个 token，从而学习语言的生成分布。其基本结构包含以下三个关键模块：

- **1. 标记化层（Embedding Layer）**
- 输入：一个由离散 token 组成的序列，通常通过分词器（Tokenizer）将文本切分为词（word）、子词（subword）或字符级 token。每个 token 对应词典（Vocabulary）中的一个唯一索引。

```
Many words map to one token, but some don't: indivisib
le. Unicode characters like emojis may be split into m
any tokens containing the underlying bytes: 🤚
Sequences of characters commonly found next to each ot
her may be grouped together: 1234567890
```

从预测下一个词到语义表示

极客时间 训练营

- from torchtext.vocab import build_vocab_from_iterator
- from torchtext.data.utils import get_tokenizer

- tokenizer = get_tokenizer("basic_english")
- tokens = tokenizer("I feel very good today")

- # 映射为token id
- vocab = build_vocab_from_iterator([tokens])
- ids = vocab(tokens)

- # 输入Embedding
- embedding = nn.Embedding(num_embeddings=len(vocab), embedding_dim=64)
- embedded = embedding(torch.tensor(ids))

从预测下一个词到语义表示

极客时间 训练营

- **2. LSTM 网络主体**
- 顺序处理每个词向量，逐步更新隐藏状态 h_t；
- 隐藏状态携带到当前位置为止的全部上下文语义。

```
lstm = nn.LSTM(input_size=64, hidden_size=128, batch_first=True)
output, (hn, cn) = lstm(embedded)
```

从预测下一个词到语义表示

极客时间 训练营

- **3. 输出层（Softmax 分类头）**
- 将当前隐藏状态 h_t 映射为一个与词表等长的 logits 向量；
- 通过 Softmax 计算每个词作为下一个词的概率。

```
linear = nn.Linear(128, len(vocab))  # 输出维度等于词表大小
logits = linear(output)
pred_prob = torch.softmax(logits, dim=-1)

loss_fn = nn.CrossEntropyLoss()
loss = loss_fn(logits.view(-1, vocab_size), targets.view(-1))
```

获取句子 Embedding 的方法

- **模型应用：使用训练好的 LSTM 提取句子表示**
- 在 CLM 训练完成后，我们可以**移除预测用的 Softmax 层**，仅保留词嵌入层与 LSTM 主体，将其作为一个语言编码器（Encoder）使用。

- **输入示例：**

input = ["我", "今天", "心情", "很", "好"]

embedding = LSTM(input)[–1] # 提取最后一个隐藏状态 h_T

- 这里的 h_T 表示模型在完整处理完输入序列后形成的**最终隐藏状态**，它压缩了整个句子的语义信息，可作为该句子的语义表示（Sentence Embedding）。

获取句子 Embedding 的方法

- **用途广泛的通用表示向量：**
- 通过这种方式获得的句子向量，可以广泛用于下游任务：
- **文本分类**（如新闻类别识别）
- **情感分析**（如“积极 / 消极”判断）
- **语义检索**（如相似句匹配、问答候选排序）
- **迁移初始化**（作为其他任务模型的预训练特征）
- 不依赖标签，仅基于语言结构训练，即可获得具有迁移能力的通用语言表示。
- 这标志着 LSTM 已从“序列预测器”转变为一个**结构压缩的语言理解模块**，具备了表示学习与迁移学习的基本能力。

表示学习的标准动作

- **标准动作一：在大规模数据上预训练通用编码器**
- 在表征学习中，第一步往往是构建一个**通用编码器（Universal Encoder）**，使其能够从原始输入（如图像、文本）中自动学习**结构化、可迁移的中间表示**。
- **核心目标：**
- 利用一个预定义的任务（如分类或预测），倒逼模型自主学习底层结构与语义模式，即使任务本身不是最终用途，它也能驱动表示学习的形成。

表示学习的标准动作

极客时间 训练营

- **视觉领域的典型做法：**
- 在 **ImageNet** 上训练 **ResNet**；
- 使用 **图像分类** 作为训练任务；
- 最终获得的是一个能够提取**多层次视觉特征**的卷积编码器。

- **语言领域的对应策略：**
- 在海量语料上训练 **LSTM** 或 **Transformer**；
- 采用 **CLM（自回归）或 MLM（遮蔽预测）** 作为训练目标；
- 模型被迫学习词汇、语法、语义的统计规律，进而构建出**可泛化的语言表示空间**。

表示学习的标准动作

- **这一步就是所谓的“预训练阶段”**，其核心解决两个关键问题：
- **表示压缩**：如何将原始数据（如像素或词序列）编码为高密度、信息丰富的表示；
- **无监督驱动**：如何在缺乏人工标签的条件下，利用**任务结构**引导模型学习有用的中间特征。
- 无需手工特征、无需大量标签，预训练让模型“自学成才”，形成具备泛化能力的结构表达。

表示学习的标准动作

- **标准动作二：提取模型中间层作为高维语义表示**
- 完成预训练后，下一步就是**显性化模型的内部表征**，将其转化为可用于下游任务的通用特征向量。
- **核心目标：**
- 从模型中提取“隐层输出”，作为具有语义压缩能力的高维表示（Embedding）。
- 这些表示不仅包含原始输入的信息，还蕴含了模型在预训练任务中学到的结构与语义规律。

表示学习的标准动作

- **视觉任务中的典型做法：**
- 从 CNN 的**卷积特征图**或**倒数第二层的全连接输出**中提取图像表示；
- 例如：ResNet-50 的 **隐层输出** 常被用作图像的通用特征向量。

- **语言任务中的对应做法：**
- 从 LSTM 的**最终隐藏状态 h_T** 提取整句语义；
- 得到的向量可用于分类、聚类、匹配等任务。

表示学习的标准动作

极客时间 训练营

- **这些中间表示具备三个关键属性：**
- **压缩性**：将高冗余输入转化为固定维度、信息密集的表示；
- **结构性**：编码了层级、语义、上下文等深层语言结构；
- **迁移性**：可跨任务、跨领域使用，无需重新训练整个模型。

- 这一阶段被称为“表示空间的显性化”：
- 模型原本“藏在网络内部”的知识，被导出为**可重用、可操作的向量表示**。
- 这使得预训练模型真正具备了**作为通用特征提取器**的能力，为后续微调与检索任务奠定基础。

表示学习的标准动作

极客时间 训练营

- **标准动作三：在新任务中“嫁接”使用预训练表示**
- 完成表示学习与显性化之后，最后一步是**将预训练得到的表示迁移到具体任务中使用**，
- 实现从“通用知识”向“任务能力”的适配与转化。
- **核心目标：**
- 利用已有语义表示，在目标任务中快速适配，减少标注依赖，提高训练效率。

表示学习的标准动作

- **迁移策略一：冻结编码器，仅训练任务头（Task Head）**
- 将预训练模型作为固定的特征提取器，冻结其全部参数；
- 在其输出之上叠加一个轻量的任务特定模块（如 MLP 或匹配网络）；
- 优势在于**计算开销小、训练样本需求低、适配速度快**。
- **典型应用：**
- 图像分类：使用 ResNet 表征 + 分类头；
- 文本情感分析：LSTM 表征 + 二分类头；
- 语义检索 / 推荐系统：双塔结构中固定编码器，仅训练相似度计算模块。

表示学习的标准动作

- **迁移策略二：微调整个模型（或部分层）**
- 解冻部分或全部预训练参数，在目标任务上进行端到端训练；
- 适用于**目标数据量较大或任务语域差异显著**的场景；
- 可显著提升模型在专业任务中的表现。

- **典型应用：**
- 医疗影像分类：对 ResNet 的高层进行微调以适配病理特征；
- 法律文书生成：对 Transformer 编码器进行微调以适应专业语言风格；
- 多语种翻译任务：在通用模型上局部调整以适配特定语言对。

表示学习的标准动作

极客时间 训练营

- 这一步是知识迁移的“**适配阶段**”，标志着模型从“表示学习器”转化为“任务专家”：
- 通用语义表示 → 任务优化模块
  抽象结构压缩 → 可执行行为输出

- 完成三步标准流程后，一个模型即具备了从海量无监督语料中学习、在具体任务中适配的完整能力路径。

表示学习的标准动作

- **迁移学习的本质**：
- 通过压缩数据结构 → 显性化语义表示 → 在新任务中重用
  构建一个**模块化、低资源、高效泛化**的学习体系。

- **连接全章主线：**
  LSTM/CLM 不是终点，而是语言建模从“压缩 → 泛化 → 适配”路径中的起点。

- 下节课我们将进入更强结构建模机制 ——
- **Transformer 与多头注意力：构建更灵活的上下文对齐结构**

实践练习：构建小型图像分类与检索系统

极客时间 训练营

- **练习 1：使用 ResNet 构建二分类器，判断图像是否属于指定类别**
- **任务说明：**
- 从互联网中收集一组图像（推荐 10~30 张），确保包含两类：
  - **正类**：目标类别（如“奥特曼”或某种动物）
  - **负类**：非目标类别的图像，作为对比样本
- 使用预训练的 ResNet 模型（如 ResNet-50）作为特征提取器：
  - 保留 ResNet 主干网络
  - 替换最后一层（分类头）为一个新的**二分类全连接层**
- 构建训练集并进行微调（fine-tuning）或仅训练新分类头（freeze主干网络）：
  - 输入：图像张量
  - 输出：是否属于目标类别（0/1）

实践练习：构建小型图像分类与检索系统

极客时间训练营

- **练习 2：使用 CLIP 构建图像检索系统（以图搜图）**
- **任务说明：**
- 利用 OpenAI 发布的 CLIP 模型 https://github.com/openai/CLIP 或其开源复现版本（如 open-clip）：
  - 提取每张图片的视觉特征向量（image embedding）
- 构建一个“图片索引库”：
  - 将上述图像的特征保存向量
- 给定一张查询图片：
  - 使用 CLIP 提取其图像向量
  - 计算与索引库中所有图片的余弦相似度
  - 返回最相似的 Top-K 图像作为检索结果
- 可视化检索效果（如：展示查询图 + Top-K 结果）

THANKS