上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Cloud Computing 云计算

马汝辉 副教授

计算机科学与工程系

上海交通大学

课程基本情况

课程教师

- 马汝辉，计算机系，ruhuima@sjtu.edu.cn，电院3号楼229

课程助教

- 蔡子诺、施宏建
- 电院3号楼229

课程网站：

- https://aisigsjtu.github.io/ICE6405P-260/

课程评价

- 出勤+课堂互动：10%
- 两次综述报告：15%*2
- 两次实践作业：30%*2

群聊：云计算 2024

该二维码7天内(9月29日前)有效，重新进入将更新

课程大纲

- **11次授课**
  - 云计算应用：联邦学习
  - 云计算基石：虚拟化
  - 云计算前沿：无服务器计算、量子计算
- **5次实践**
  - 联邦学习课程项目
  - 无服务器计算课程项目

周次	课程内容	课程作业
2	云计算概论 & 机器学习概论	
3	联邦学习概论	
4	联邦学习研究	
5	联邦学习前沿	综述报告
6	联邦学习实践（1）	
7	联邦学习实践（2）	课程实践
8	云计算与虚拟化概述	
9	更多虚拟化：IO、网卡与异构硬件	
10	轻量级虚拟化 & 云计算实践（1）	
11	无服务器计算概论	
12	无服务器计算前沿	
13	System for AI: 以无服务器计算为例	综述报告
14	云计算实践（2）	
15	云计算实践（3）	课程实践
16	量子计算概论	
17	量子计算前沿	

11次授课
5次实践

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Lecture 1: 云计算概论 & 机器学习概论

马汝辉 副教授

计算机科学与工程系

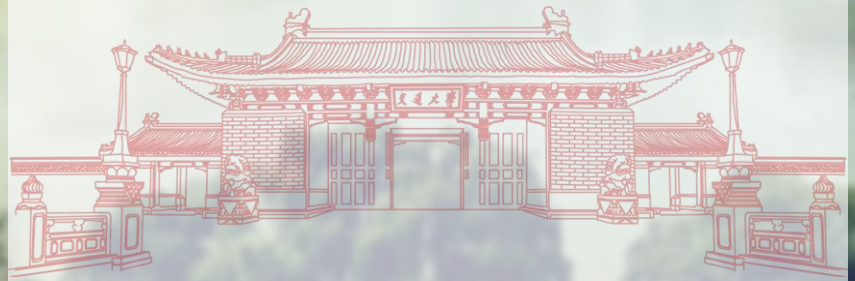
上海交通大学

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录

01

云计算背景



Gartner 技术成熟度曲线

- “Gartner分析师对全球技术前瞻趋势的分析以及关于行业用户的准确洞察力，能够高效助力于企业的战略布局、技术选型以及不同市场企业目标的制定及完成。”
- 每个技术成熟度曲线都将技术的生命周期划分为五个关键阶段。

Gartner®

技术萌芽期 → 期望膨胀期 → 泡沫破裂谷底期 → 稳步爬升复苏期 → 生产成熟期

- 分析技术成熟度曲线，可以：
  - 将宣传炒作与技术商业前景的真正驱动因素区分开
  - 降低技术投资决策的风险
  - 对技术业务价值的理解与经验丰富的 IT 分析师的客观评价进行对比

2022年新兴技术成熟度曲线

主题 1: 发展/扩展沉浸式体验

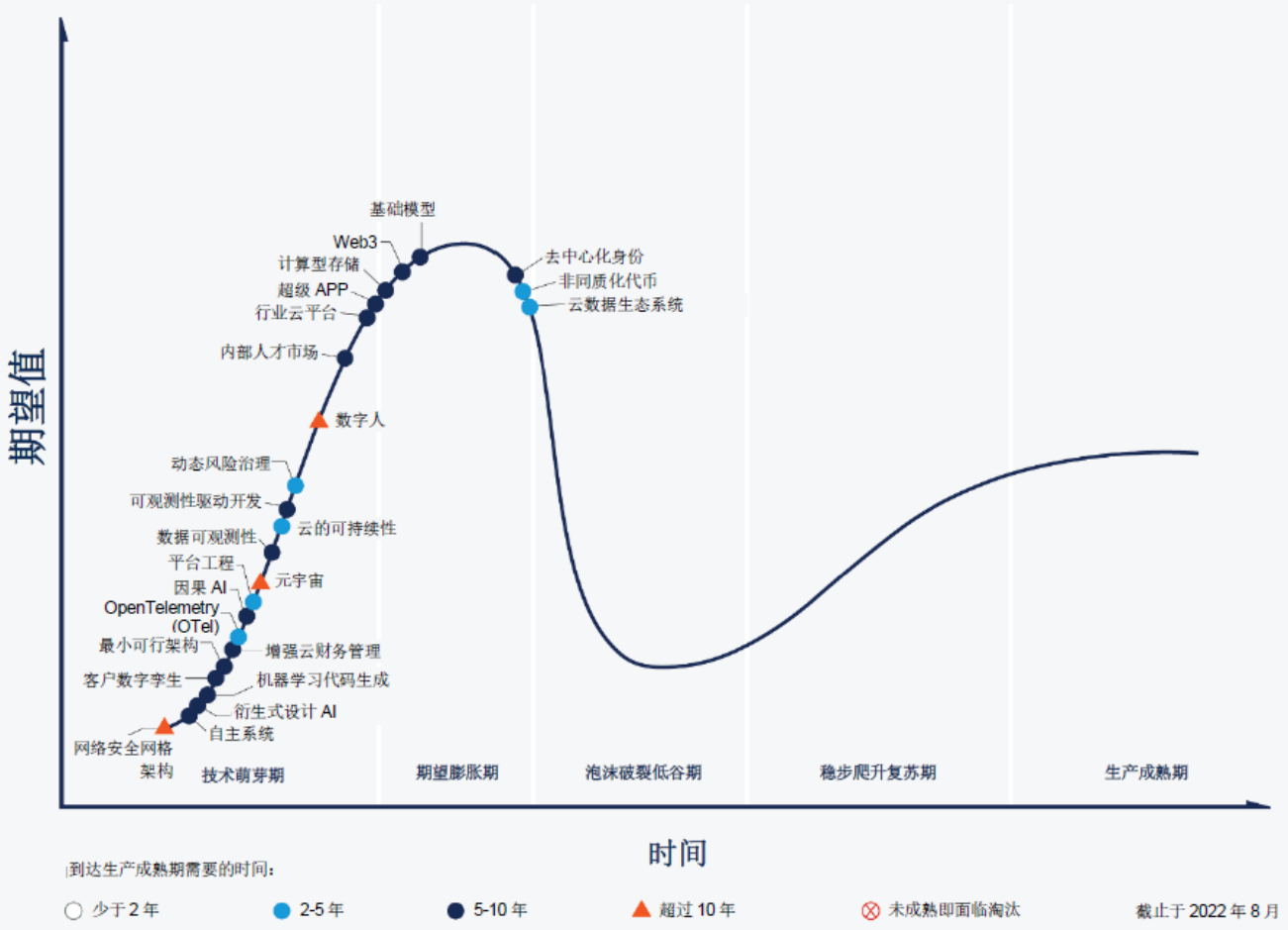
- 去中心化身份、数字人、内部人才市场、元宇宙、非同质化代币、超级APP、Web3

主题2：加速人工智能自动化

- 因果AI、基础模型、衍生式设计 AI、机器学习代码生成

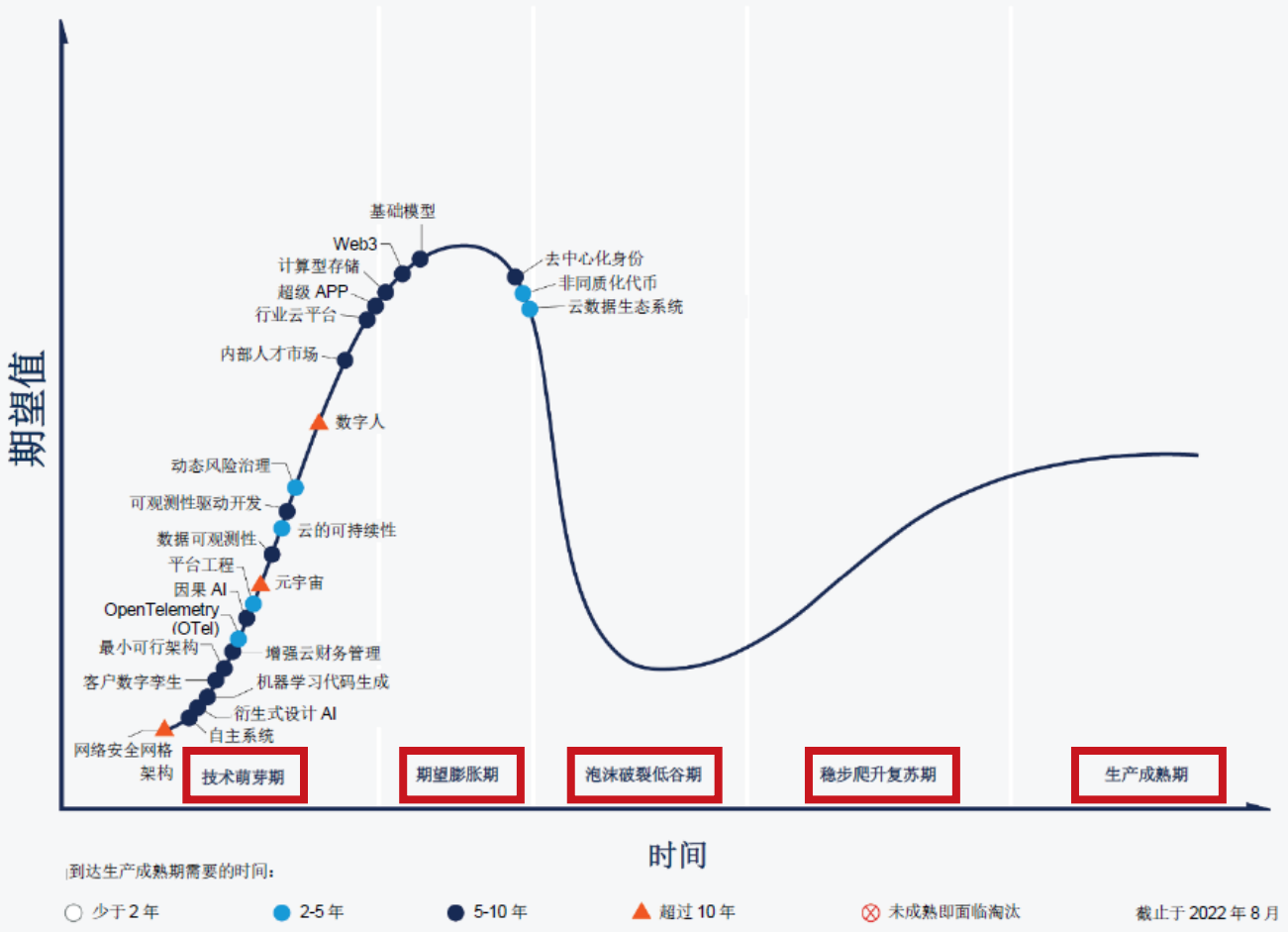
主题3：优化的技术人员交付

- 增强型云财务管理、云的可持续性、计算型存储、网络安全网格结构、数据可观察性、动态风险治理、行业云平台、最小可行架构、可观察性驱动开发、OpenTelemetry、平台工程

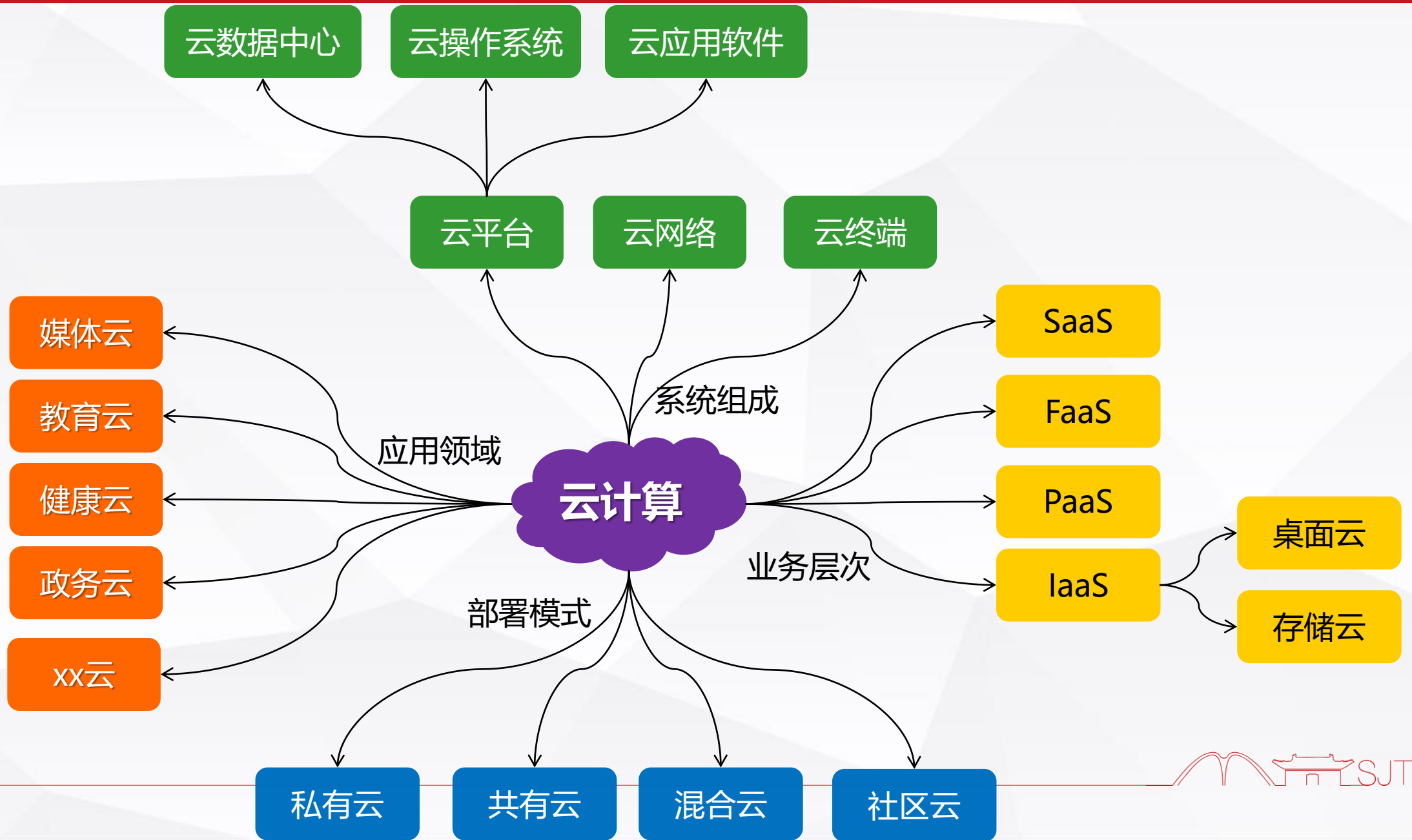


2022年新兴技术成熟度曲线

- 2022年Gartner新兴技术成熟度曲线（Hype Cycle™）列出了25项值得关注的技术创新，它们能够帮助企业建立差异化竞争优势。
- 这些技术中只有一小部分可能在两年内获得广泛采用，其中许多技术都将需要发展10年乃至更长时间。
- 由于它们处在雏形阶段，部署这些技术或为企业带来更多风险，但早期采用者也可能从这些技术中获得更大收益。



多种视角看云计算


云数据中心
云操作系统
云应用软件
云平台
云网络
云终端
媒体云
教育云
健康云
政务云
xx云
应用领域
系统组成
云计算
部署模式
业务层次
SaaS
FaaS
PaaS
IaaS
桌面云
存储云
私有云
共有云
混合云
社区云

IT发展面临三大问题

- **信息处理需求——巨大**
  - 百度索引：今年百亿 - 明年千
  - 腾讯带宽需求：07年200G - 2

大数据

- **企业IT系统——高昂**
  - 大企业IT成本不断增加，需要减负
    - 超过70%的大中型企业正在
  - 中小企业、创新企业IT外包需
    - 硬件成本、软件成本、运营成本、管理成本

高成本

- **资源利用率——过低**
  - 传统的按物理服务器集群分配 源的大量浪费
  - 我国互联网企业的服务器平均
  - 较低的资源利用效率 VS较高的设备增长率 =企业很大的成本压力

低效率

传统方式无法完全解决问题

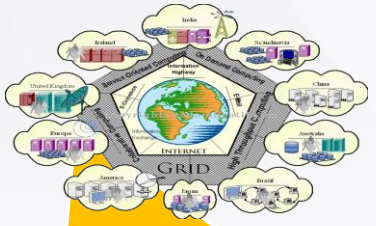
可以解决大处理量问题，但成本高

大型机

可以解决大处理量问题，但成本高，应用门槛高，应用领域较窄

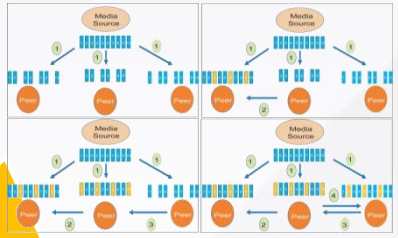
HPC

可以解决大处理量问题，但商业模式不清，无法形成商业服务



网格

仅能解决大数据量存储问题，且缺乏商业模式



P2P

需要能够进行海量数据处理，而且成本低、具有清晰商业模式的解决方案

云计算概念逐渐丰富

应对IT领域的诸多问题，云计算理念逐步走向成熟

2000

谷歌

MapReduce

GFS　BigTable

计算能力在“云”中

2001

Salesforce

CRM

ERP

企业信息化软件在“云”中

2007

AWS

EC2　SimpleDB

S3

IT软、硬件资源在“云”中

2009

Cisco/EMC

/VMWare

vBlock

企业信息化系统在“云”中

2010-2011

“云”手机

“云”存储

“云”桌面

“云”数据中心

……

“云”杀毒

依靠x86主机分布式计算技术解决低成本海量数据处理

通过网络化多租户软件系统提供低成本信息化系统软件

通过虚拟化技术提高系统利用率，并对外提供商业租用

改造传统IT系统，降低企业IT系统成本，提高安全性

云计算理念进一步扩展，涵盖从终端到应用的各个方面

云计算具有服务模式和技术实现的两层含义

云计算是一种通过网络实现对各种IT能力进行灵活调用的服务模式。 → 服务模式

云计算通过分布式计算、虚拟化等关键技术，构建用于资源和任务统一管理调度的资源控制层，将分散的ICT资源集中起来形成资源池，动态按需分配给应用使用。 → 技术实现

云计算是一种获得IT服务的模式，这种服务模式是随着IT产业的发展，信息技术逐步普及化，向社会基本需求转变的必然结果，也是IT产业由用户自给自足向社会化服务模式发展趋势的体现。

- 通过网络使用IT服务
- IT服务按量计费
- 服务规模可以按需变化
- 所有者使用者分离

用户角度：使用模式

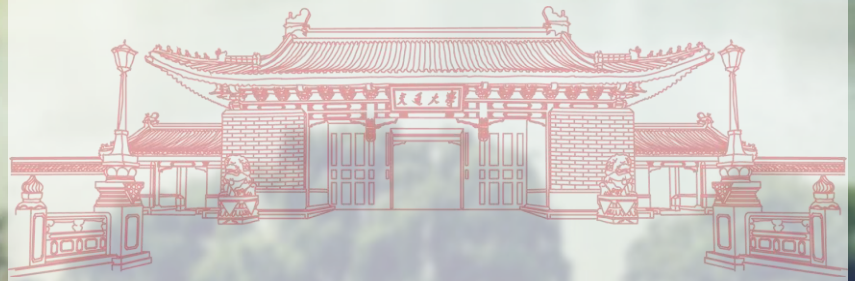
云计算是通过技术的发展和对各种已有IT技术的综合利用，实现产业核心从提供产品到提供服务的转变，并实现信息系统或运营的自动化和总体成本的有效降低。

- 多租户
- 系统规模平滑扩展
- IT资源池化，对用户实现统一的自动调度

业务提供者角度：技术实现与管理

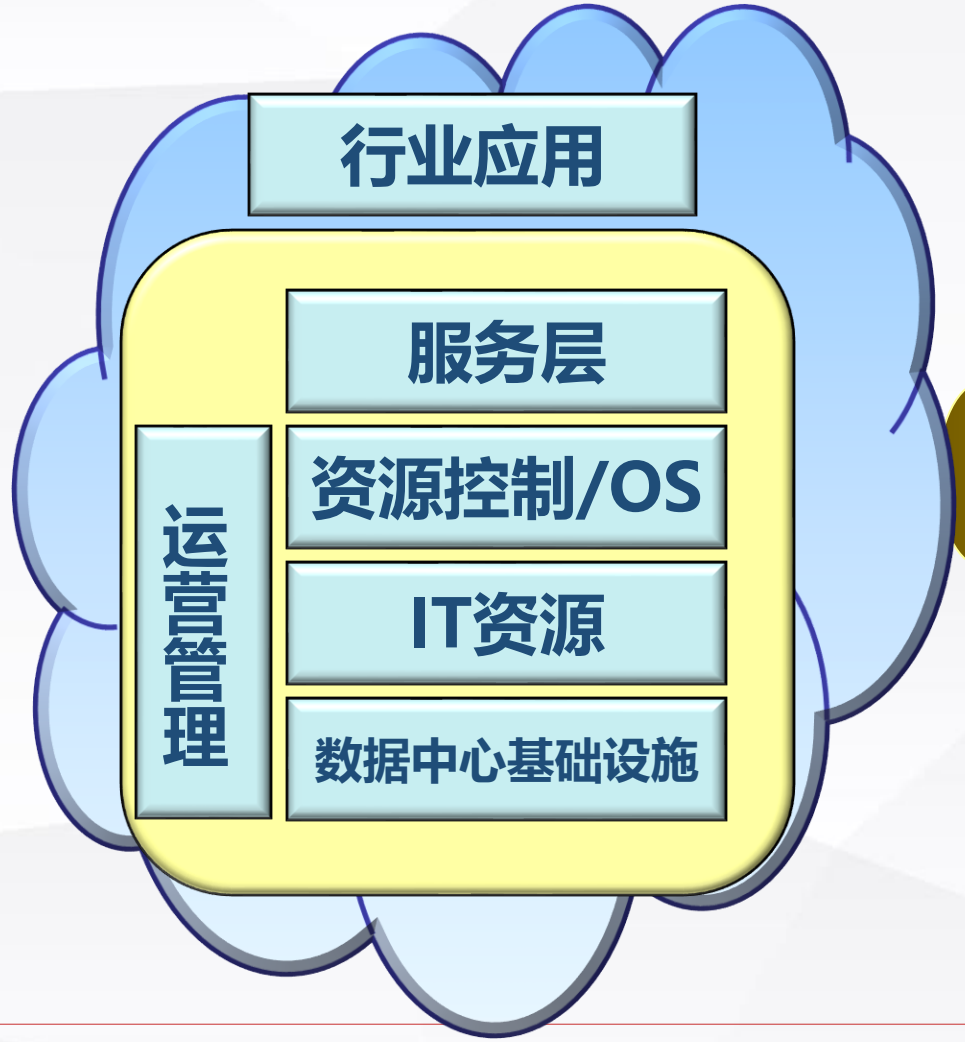
02

云计算技术



云计算在未来ICT领域技术中的位置

云 —— 管 —— 端



智能管道

云计算将对“管”和“端”技术产生重要影响

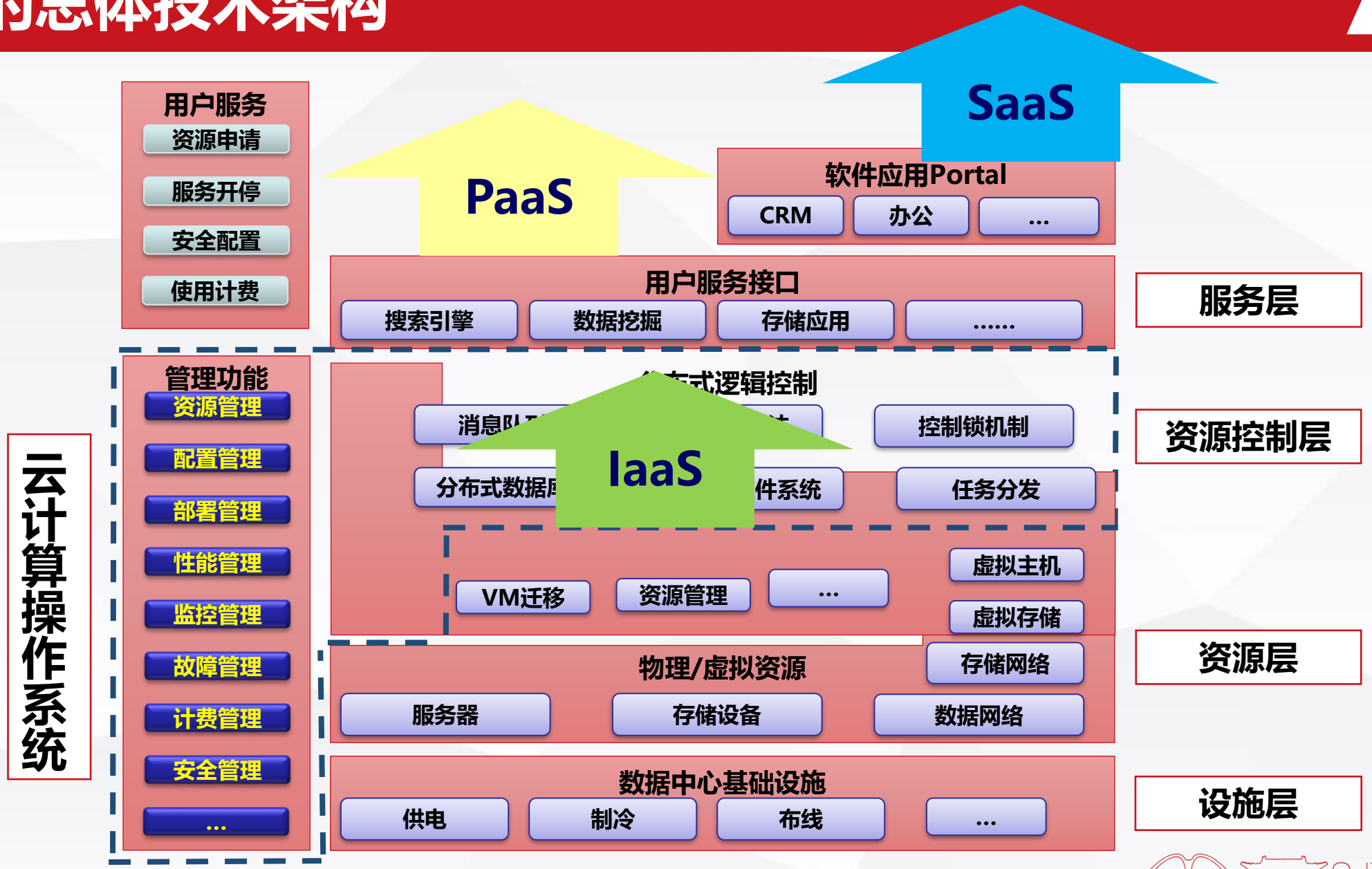
智能管道

云计算对广域网智能管道的需求：
——感知 Awareness
——按需部署 On demand provisioning
——流量优化 Optimization
——开放网络
云计算对数据中心网络技术的需求：
——数据中心内部网络
——数据中心互联 Dada center bridging

终端

云计算对终端的影响：
——对终端功能的无限扩展
——对终端软件架构的影响：无缝调用网络资源
——对终端联网的影响：永远在线，宽带

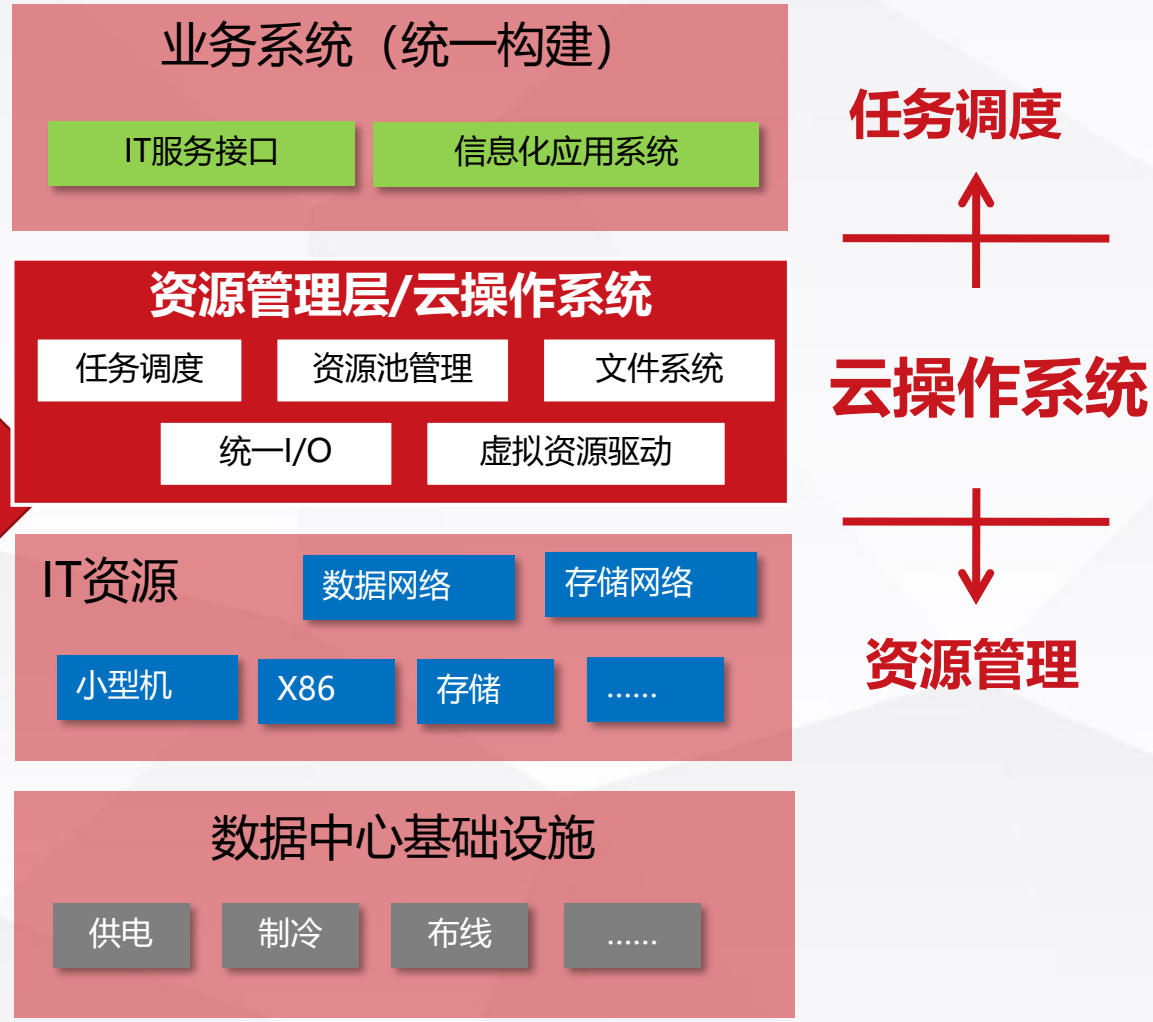
云的总体技术架构


SaaS
PaaS
IaaS
用户服务
资源申请
服务开停
安全配置
使用计费
软件应用Portal
CRM
办公
...
用户服务接口
搜索引擎
数据挖掘
存储应用
.....
服务层
管理功能
资源管理
配置管理
部署管理
性能管理
监控管理
故障管理
计费管理
安全管理
...
云计算操作系统
控制锁机制
任务分发
VM迁移
资源管理
...
虚拟主机
虚拟存储
资源控制层
物理/虚拟资源
存储网络
服务器
存储设备
数据网络
资源层
数据中心基础设施
供电
制冷
布线
...
设施层

资源控制（操作系统）技术是云计算技术的核心

传统信息化平台/系统

云计算平台/系统



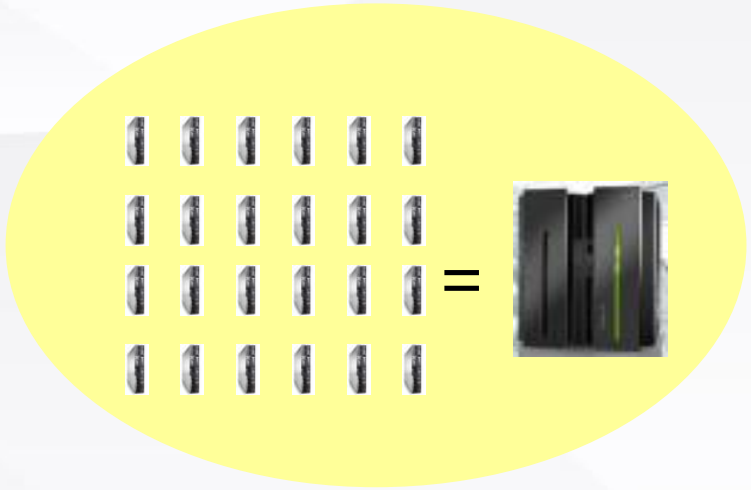
任务调度

云操作系统

资源管理

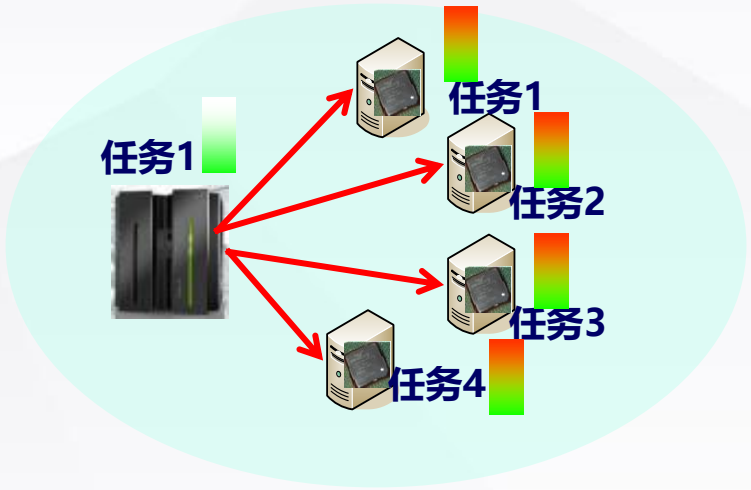
云操作系统有两种面向不同场景的实现模式

低性能资源“多合一”



主要目的：将较小的计算资源聚合，统一调度**完成**大规模计算任务，
代表：
✓**谷歌集群计算系统**
✓开源平台Apache Hadoop（Yahoo）
✓**阿里巴巴云平台**

高性能资源“一虚多”



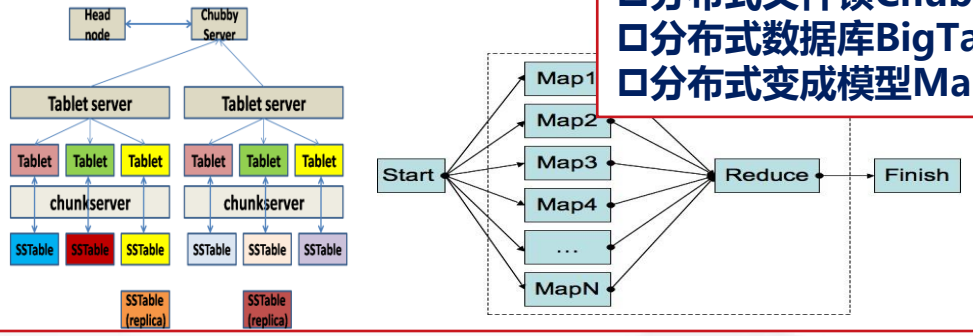
主要目的：将较强大的物理资源分割为虚拟资源，统一管理，提高资源利用效率，代表：
✓VMware vCloud
✓Amazon EC2
✓华为UVP

——分别面向大规模计算和资源精细管理两种不同的应用场景
——在实际系统中也可以结合使用，如Hadoop over EC2（弹性Hadoop）

云操作系统的模式#1：低性能资源“多合一”

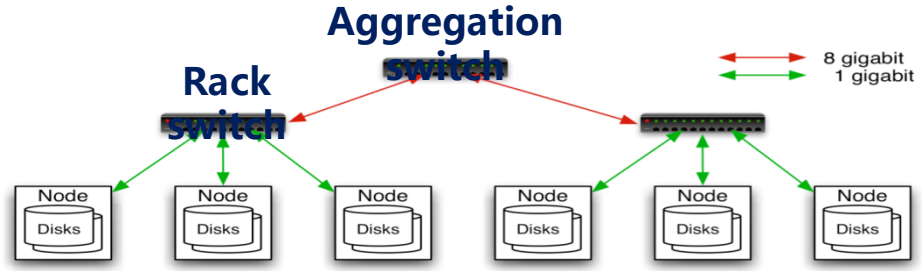
大规模计算任务
网页检索、数据挖掘、日志分析等

分布式文件系统及任务调度



谷歌系统的四大法宝：
- 分布式文件系统GFS（CFS）
- 分布式文件锁Chubby
- 分布式数据库BigTable
- 分布式变成模型MapReduce

计算机集群（Cluster）



廉价的分散资源（典型规模1000-5000台）

资源控制层

特点：
- 分布式架构
- 为特定任务设计
- 需要特定编程模型支持

资源层

特点：
- 多机集群
- 网状互联
- 节点定制化

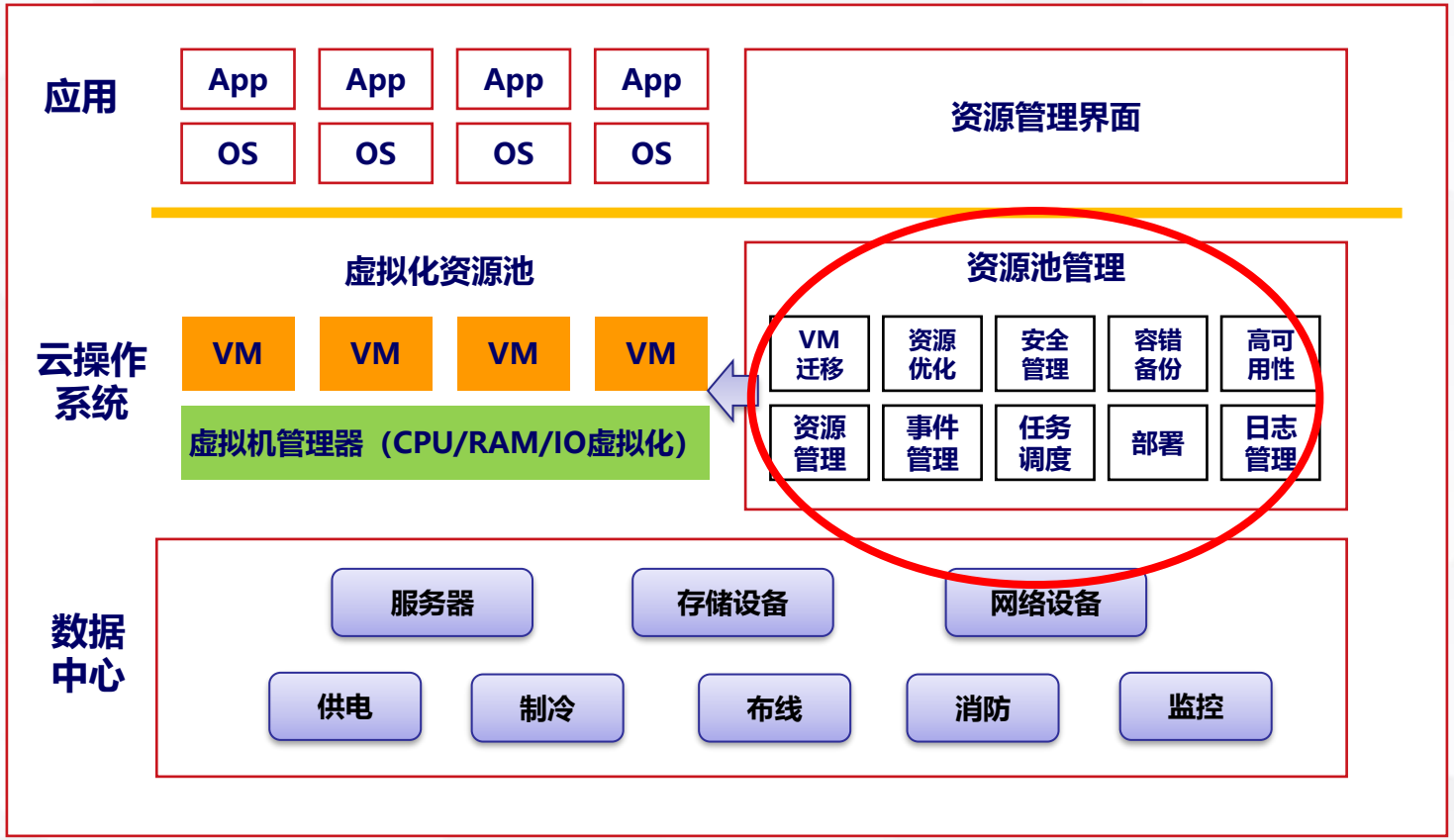
云操作系统的模式#2：高性能资源“一虚多”

多租户的各种一般IT应用

统一管理的资源池/云操作系统

性能较强的物理资源虚拟化

服务器虚拟化、存储虚拟化和网络虚拟化



以VMWare vSphere为例

技术上并无太大突破，是原有虚拟化系统的规模化、商业化升级，同时与原有IT系统结合形成新的“私有云”解决方案

我国在云计算操作系统关键技术方面的基础和差距

“多合一模式”

分布式计算

文件系统　数据库

集群消息　任务分发

□谷歌在分布式计算（集群计算）方面保持较大优势，集群规模可以达到5000台主机以上，多集群[illegible]-30000台主机

□国内阿里巴[illegible]布式计算领域基于开源平台（Hadoop等[illegible]有较好技术基础，形成自有技术体系，但规模上较小，单集群规模在2000台左右

系统规模小

“一虚多”模式

资源池管理

VM迁移　系统资源监控

故障容灾　生命周期管理

□VMware、微软、IBM、Amazon等公[illegible]资源池管理技术方面拥有丰富经验，且与商用[illegible]

□国内企业如[illegible]通过自主研发基本掌握资源池管理技术，但缺乏[illegible]商用系统的结合。

商用经验少

IT设备虚拟化

CPU虚拟化　内存虚拟化

I/O虚拟化　网络虚拟化

□在主机虚拟化（CPU虚拟化、内存虚拟化[illegible]O虚拟化）方面VMware、Citrix、微软等[illegible]决方案，国内华为、中兴、天云等公[illegible]nux KVM）基础上开发虚拟化系统

□网络虚拟化思科公司以FEX等技术继续保持技术优势，华为、中兴等公司也有类似技术

技术有差距

云操作系统领域开源渐成趋势，但需深入研究

Hadoop、Eucalyptus、OpenNebula、OpenStack、OpenQRM、XenServer、CloudStack、ConVirt

谷歌的分布式计算思想

Yahoo!

hortonworks

核心推动者

hadoop

Apache Hadoop社区

二次开发者

Cloudera/MapR……兼容Hadoop API的基础上进行局部改良或代码重构

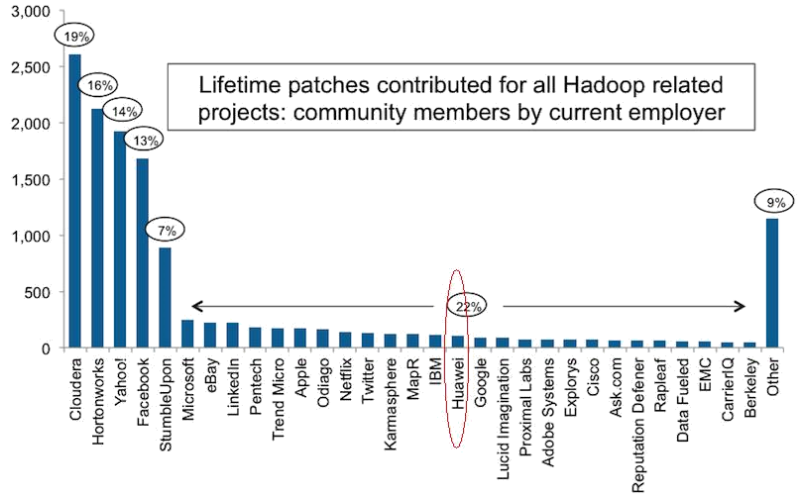
Big Data用户

使用者

Facebook/Linkedin/Amazon/EMC/eBay/Tweeter/IBM/Microsoft/Apple/HP/淘宝/腾讯/百度……

Hadoop生态系统初步形成

国内企业积极参与开源社区



开源不等于免费使用，同样面临很多问题

技术

Hadoop：

- 可扩展性和可靠性不足（NameNode单点瓶颈），单点故障高、集群规模上不去
- 通用大数据平台，对大规模迭代和循环等操作不优化
- 编程复杂（MapReduce编程模型复杂）

知识产权

开源社区的许可证制约着商业化应用：

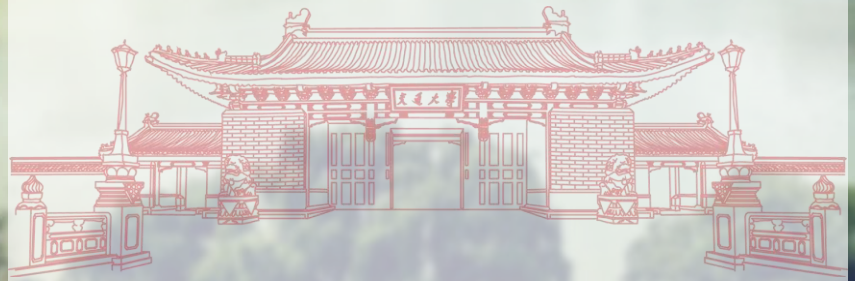
- 严格开源的许可证（如GPL许可证，Eucalyptus、OpenQRM等使用）面临技术流失的风险
- 弱开源的许可证（如Apache许可证，Hadoop、OpenStack等使用）面临专利、著作权侵犯风险

小结

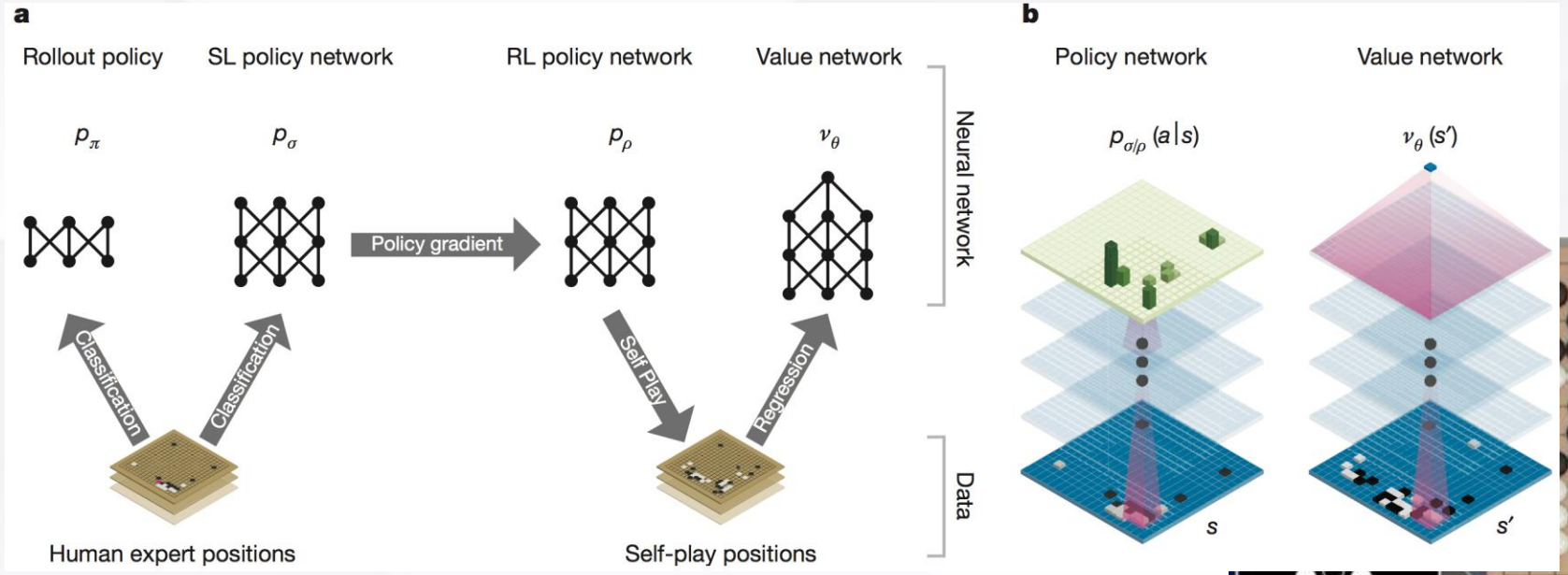
- 云计算正在引发数据中心、服务器、应用软件、操作系统等技术的重大变革，也将带动网络和终端的技术创新；
- 云计算操作系统是云技术的核心，进行大数据处理的“多合一”平台是技术创新的焦点，大公司技术封闭，开源技术成为热点；
- 国内云操作系统以互联网企业为核心，形成了一定的技术能力，未来存在技术突破的机会，但需注意开源系统的风险；
- 云计算带来数据中心内部及外部网络的技术变革，Cisco等国外厂商技术储备雄厚，我国企业处于跟随态势；
- 数据中心是云计算的载体，数据中心技术正在向高密度、绿色化、模块化方向发展，我国在新一代数据中心方面技术集成和应用水平较低。

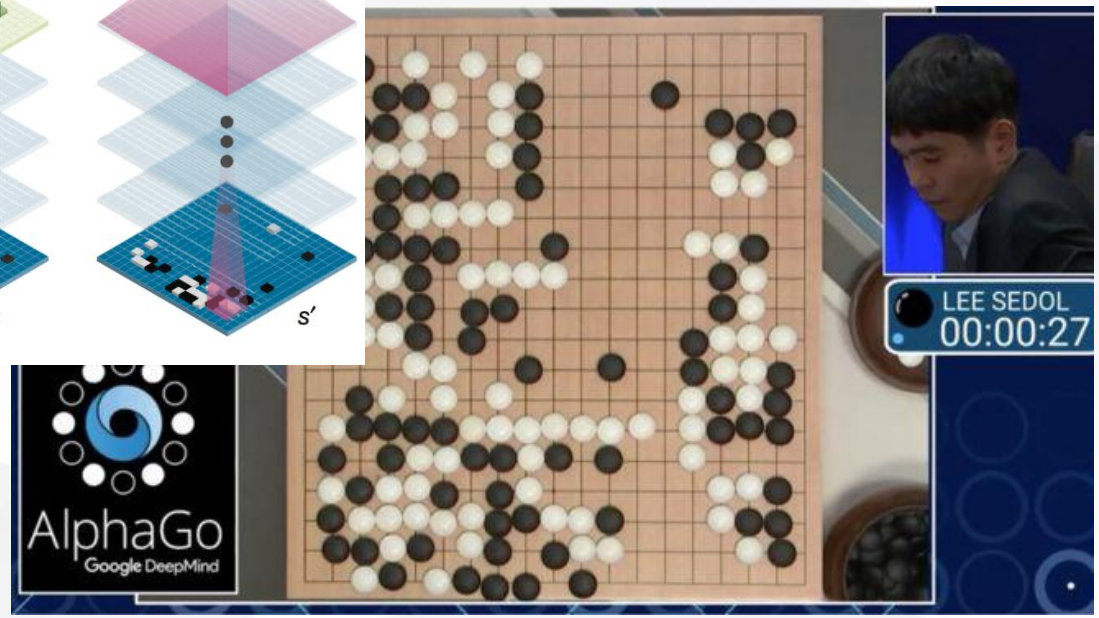
03

机器学习概论

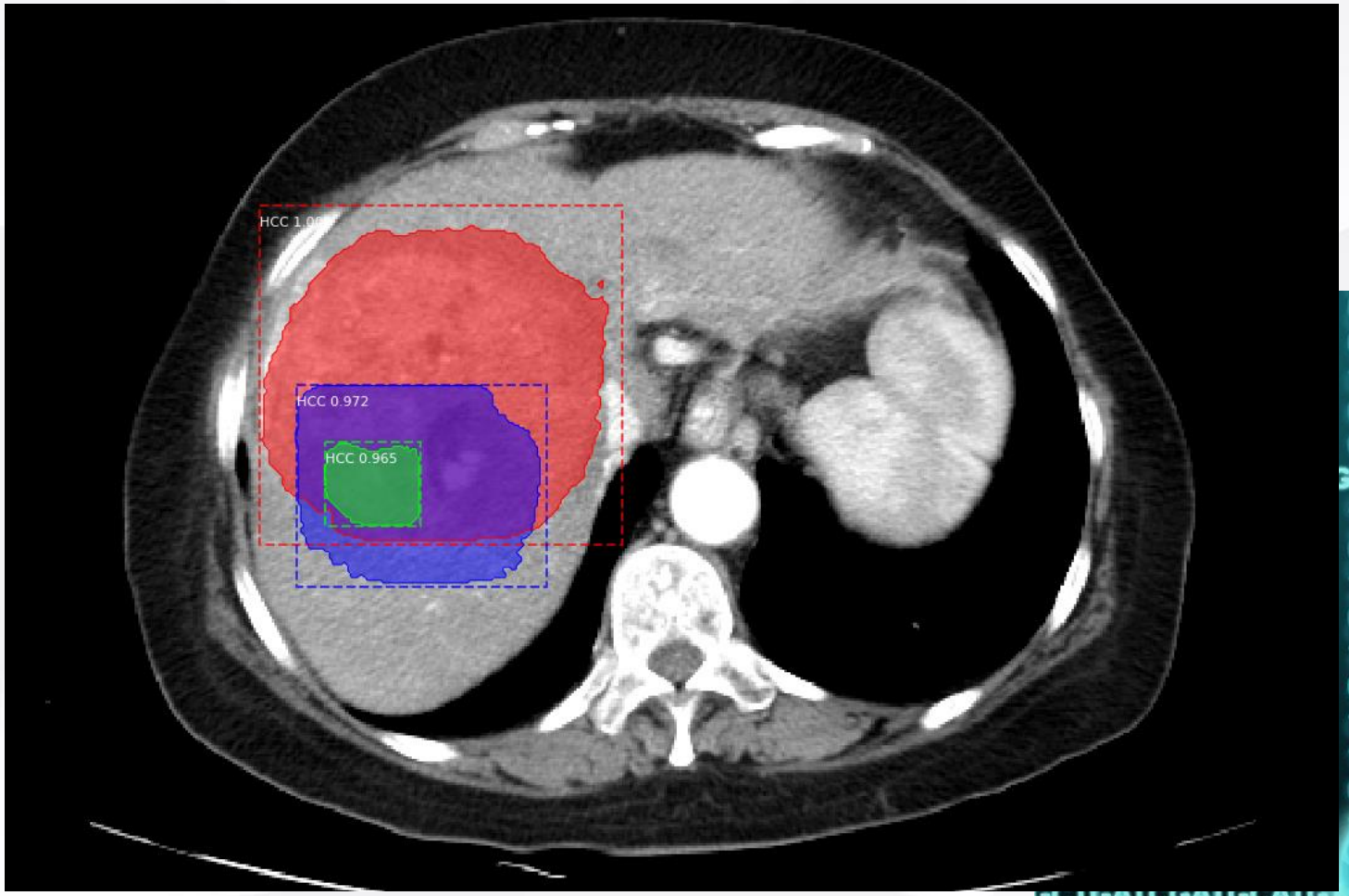


3.1 Application： AlphaGo

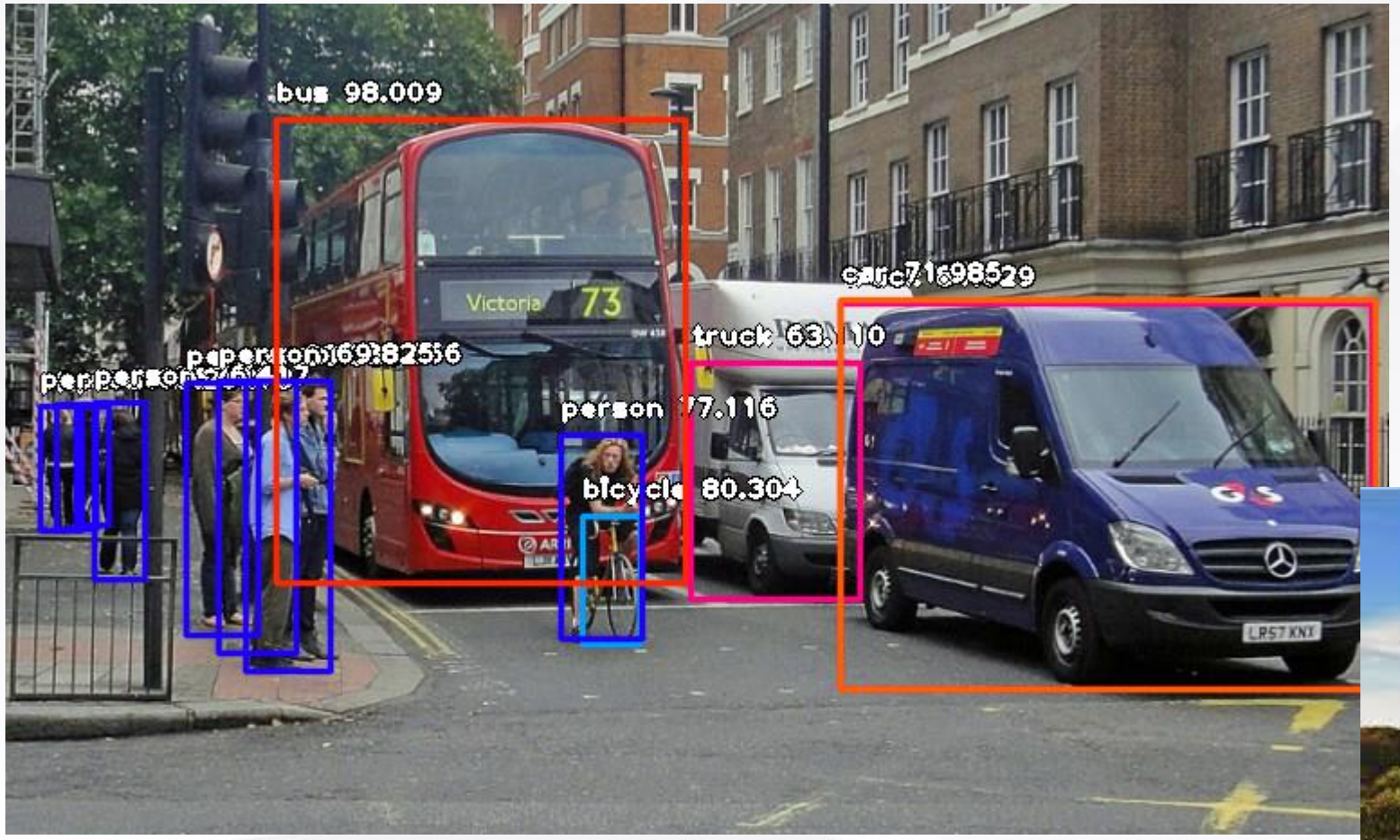




3.1 Application： Medical Assist Diagnosis

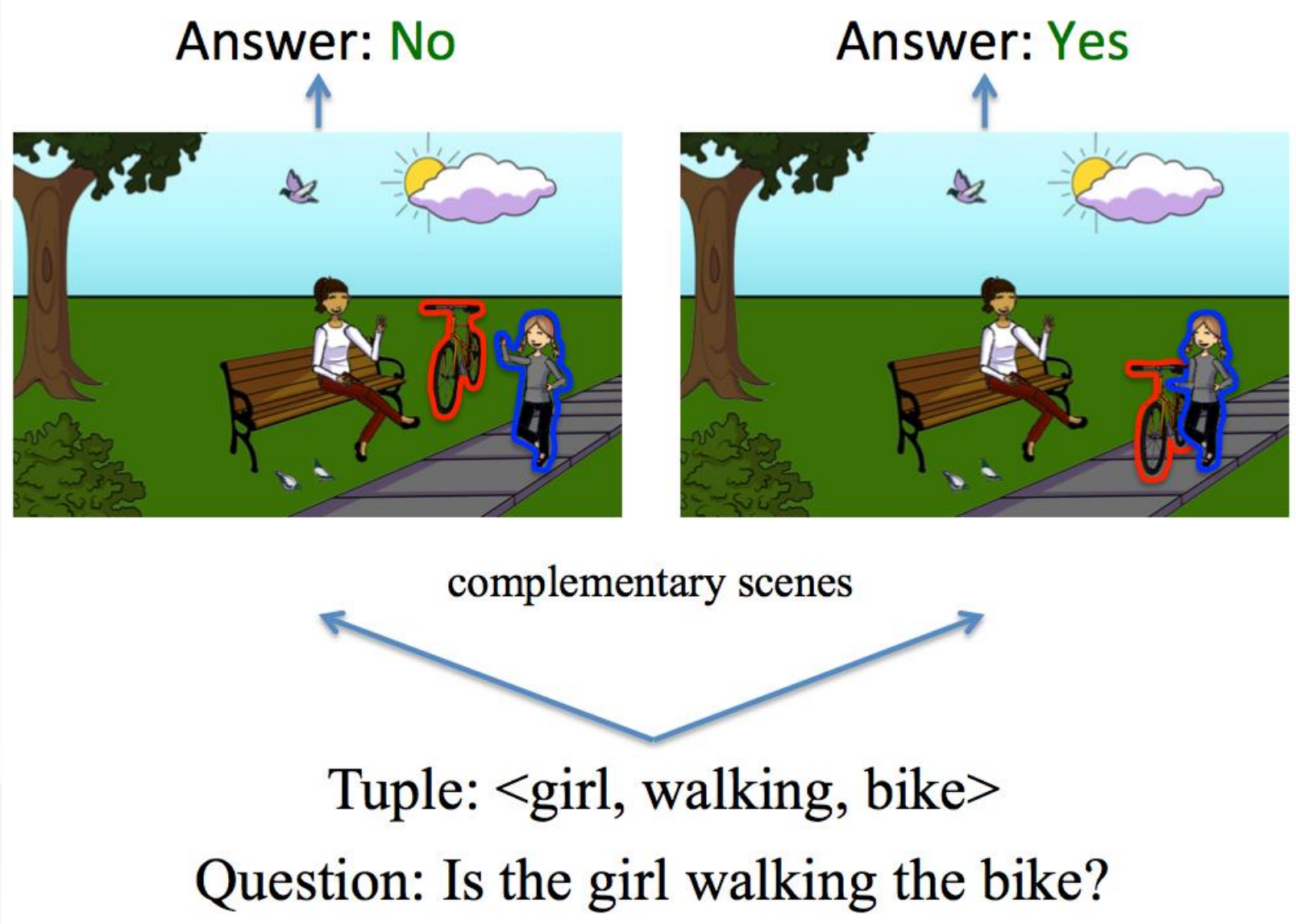


3.1 Application： Object Detection



3.1 Application： Visual Segmentation

3.1 Application： Q&A System



3.1 Application： Voice Assistant

3.1 日常生活中的机器学习

Hey Siri

图1.1.1 识别唤醒词

训练过程通常包含如下步骤

- 从一个随机初始化参数的模型开始，这个模型基本毫不“智能”。
- 获取一些数据样本（例如，音频片段以及对应的是否标签）。
- 调整参数，使模型在这些样本中表现得更好。
- 重复第2步和第3步，直到模型在任务中的表现令你满意。

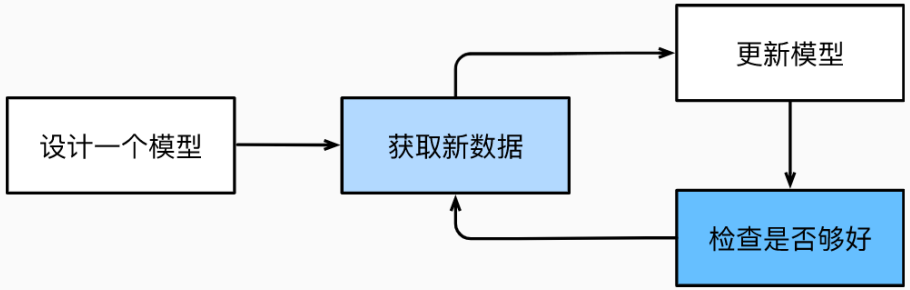


图1.1.2 一个典型的训练过程

3.1 What is Machine Learning?

- Machine learning is a subfield of computer science that originated in the study of artificial intelligence
- A field that combines computation and statistics and is closely related to information theory, signal processing, algorithms, control theory and optimization theory ——Michel Jordan
- Machine Learning = Matrices + Optimization + Algorithms + Statistics …
- Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn' —— Wikipedia

3.2 关键组件

数据

- 每个数据集由一个个**样本**（example, sample）组成
- 每个样本由一组称为**特征**（features，或协变量（covariates））的属性组成
- 监督学习问题中，要预测的是一个特殊的属性，它被称为**标签**（label，或目标（target））
- 每个样本的特征类别数量都是相同的时候，其特征向量是固定长度的，这个长度被称为数据的**维数**（dimensionality）

模型

- 简单模型：LR，SVM，Decision Tree，Random Forest，XGBoost, etc.
- 深度学习：由**神经网络**错综复杂的交织在一起，包含层层数据转换，因此被称为深度学习（deep learning）。
  - CNN
  - RNN
  - Transformer

3.2 关键组件

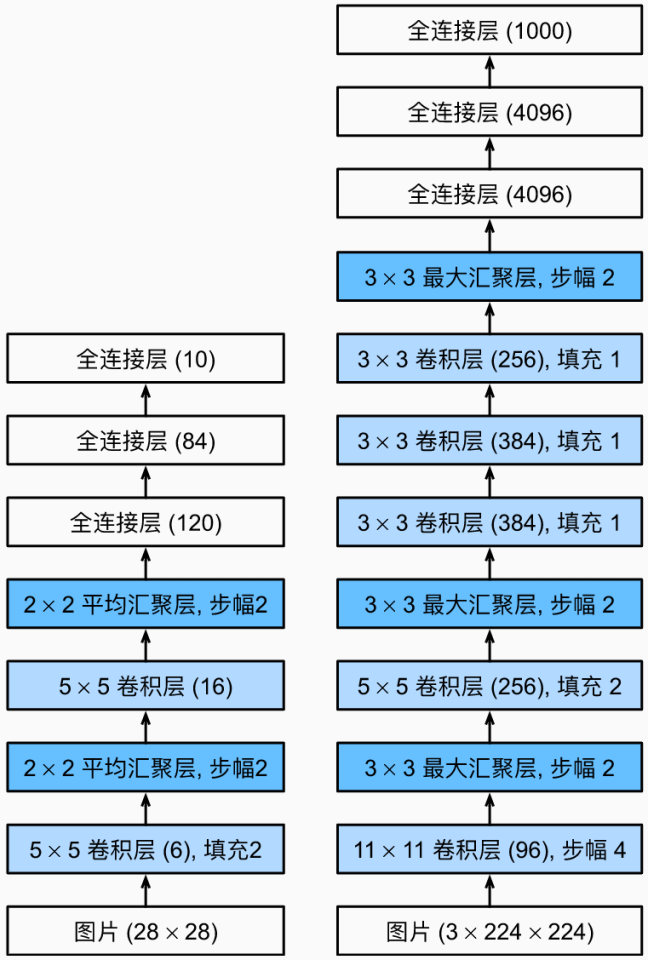


图7.1.2 从LeNet（左）到AlexNet（右）

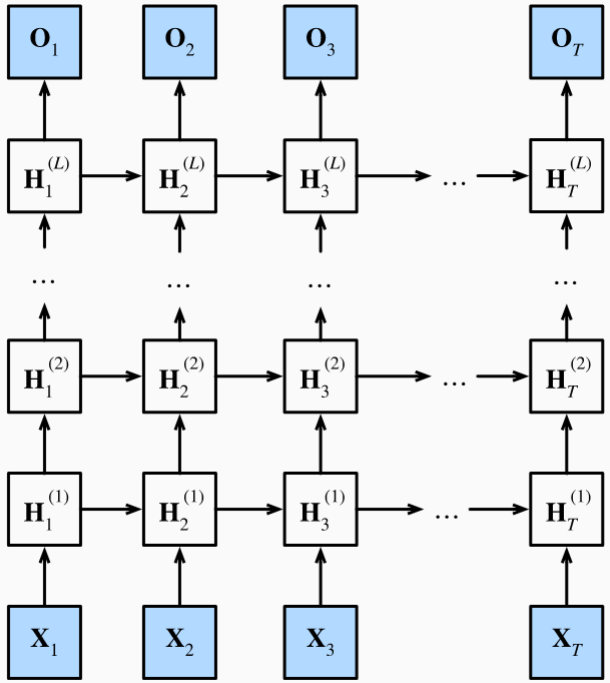


图9.3.1 深度循环神经网络结构

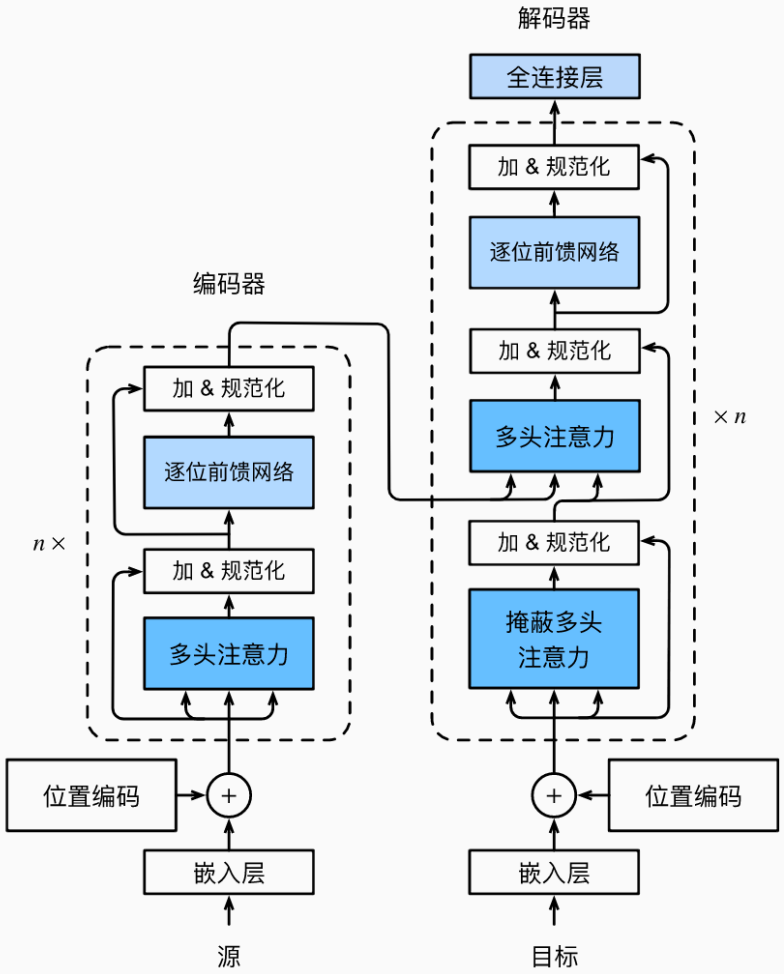


图10.7.1 transformer架构

3.2 关键组件

目标函数

- 我们需要定义模型的优劣程度的度量，这个度量在大多数情况是“可优化”的，我们称之为**目标函数（objective function）**
- 当任务在试图预测数值时，最常见的损失函数是**平方误差（squared error）**，即预测值与实际值之差的平方。
- 当试图解决分类问题时，最常见的目标函数是最小化**错误率**，即预测与实际情况不符的样本比例。

优化算法

- 优化算法能够搜索出最佳参数，以最小化损失函数。
- 大多流行的优化算法通常基于一种基本方法–**梯度下降**（gradient descent）。

监督学习

- 监督学习（supervised learning）擅长在“给定输入特征”的情况下预测标签。 每个“特征-标签”对都称为一个样本（example）。

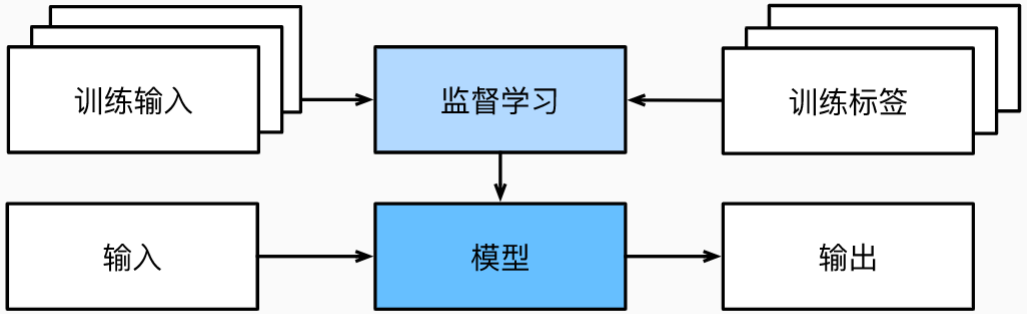


图1.3.1 监督学习

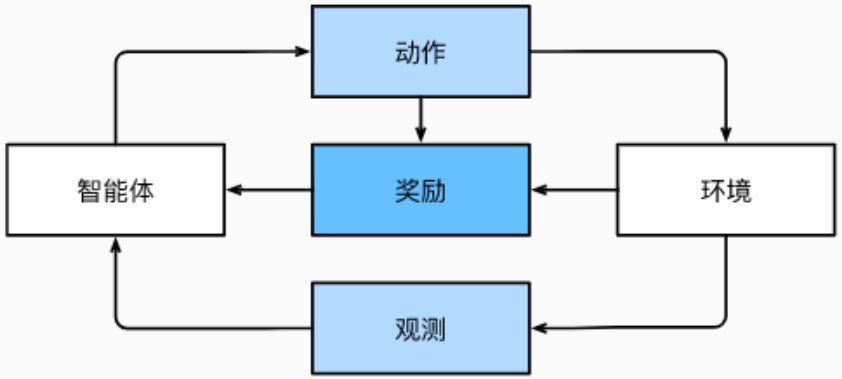
- 回归
- 分类
- 标记
- 搜索
- 推荐系统
- 序列学习

3.3 各种机器学习问题

无监督学习

- 聚类（clustering）问题：没有标签的情况下，我们是否能给数据分类呢?
- 主成分分析（principal component analysis）问题：我们能否找到少量的参数来准确地捕捉数据的线性相关属性?
- 因果关系（causality）和概率图模型（probabilistic graphical models）问题：我们能否描述观察到的许多数据的根本原因?
- 生成对抗性网络（generative adversarial networks）：为我们提供一种合成数据的方法，甚至像图像和音频这样复杂的非结构化数据。潜在的统计机制是检查真实和虚假数据是否相同的测试，它是无监督学习的另一个重要而令人兴奋的领域。

强化学习



3.4 小结

- **机器学习研究计算机系统如何利用经验（通常是数据）来提高特定任务的性能。**它结合了统计学、数据挖掘和优化的思想。通常，它是被用作实现人工智能解决方案的一种手段。
- **表示学习作为机器学习的一类，其研究的重点是如何自动找到合适的数据表示方式。**深度学习是通过学习多层次的转换来进行的多层次的表示学习。
- **深度学习不仅取代了传统机器学习的浅层模型，而且取代了劳动密集型的特征工程。**
- 最近在深度学习方面取得的许多进展，大都是由廉价传感器和互联网规模应用所产生的**大量数据，以及（通过GPU）算力的突破来触发的。**
- **整个系统优化是获得高性能的关键环节。**有效的深度学习框架的开源使得这一点的设计和实现变得非常容易。

04

从线性回归到深度学习

- 线性回归
- Softmax 回归
- 多层感知器
- 卷积神经网络

SJTU

4.1 线性回归

- **定义： 线性回归基于几个简单的假设： 首先，假设自变量x和因变量y之间的关系是线性的， 即可y以表示为中x元素的加权和，这里通常允许包含观测值的一些噪声； 其次，我们假设任何噪声都比较正常，如噪声遵循正态分布。**
- **举例：我们希望根据房屋的面积（平方英尺）和房龄（年）来估算房屋价格（美元）。**
  - 为了开发一个能预测房价的模型，我们需要收集一个真实的数据集。
    - 这个数据集包括了房屋的销售价格、面积和房龄。 在机器学习的术语中，该数据集称为训练数据集（training data set） 或训练集（training set）。
  - 每行数据（比如一次房屋交易相对应的数据）称为样本（sample）， 也可以称为数据点（data point）或数据样本（data instance）。
  - 我们把试图预测的目标（比如预测房屋价格）称为标签（label）或目标（target）。 预测所依据的自变量（面积和房龄）称为特征（feature）或协变量（covariate）。

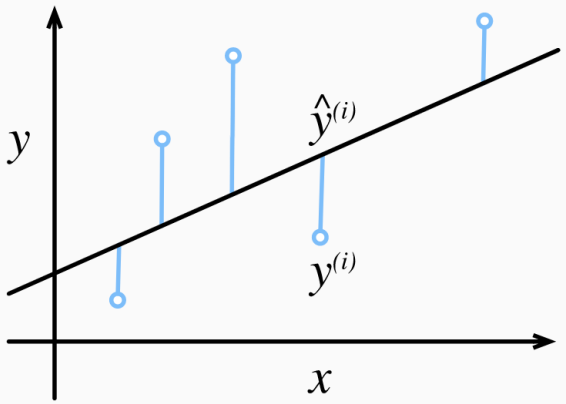
4.1 线性回归

基本元素

1. 线性模型

$$\text{price} = w_{\text{area}} \cdot \text{area} + w_{\text{age}} \cdot \text{age} + b.$$

2. 损失函数：损失函数（loss function）能够量化目标的实际值与预测值之间的差距。 通常我们会选择非负数作为损失，且数值越小表示损失越小，完美预测时的损失为0。 回归问题中最常用的损失函数是平方误差函数。$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2.$



基本元素

2. 损失函数

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2.$$

在训练模型时，我们希望寻找一组参数（w*, b*）， 这组参数能最小化在所有训练样本上的总损失。

$$\mathbf{w}^*, b^* = \operatorname*{argmin}_{\mathbf{w}, b}\ L(\mathbf{w}, b).$$

基本元素

3. 随机梯度下降

- 梯度下降最简单的用法是计算损失函数（数据集中所有样本的损失均值）关于模型参数的导数（在这里也可以称为梯度）。但实际中的执行可能会非常慢：因为在每一次更新参数之前，我们必须遍历整个数据集。因此，**我们通常会在每次需要计算更新的时候随机抽取一小批样本，这种变体叫做小批量随机梯度下降（minibatch stochastic gradient descent）。**
- 在每次迭代中，我们首先随机抽样一个小批量B，它是由固定数量的训练样本组成的。然后，我们计算小批量的平均损失关于模型参数的导数（也可以称为梯度）。最后，我们将梯度乘以一个预先确定的正数η，并从当前参数的值中减掉。

$$(\mathbf{w}, b) \leftarrow (\mathbf{w}, b) - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{(\mathbf{w},b)} l^{(i)}(\mathbf{w}, b).$$

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{\mathbf{w}} l^{(i)}(\mathbf{w}, b) = \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{x}^{(i)} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right),$$

$$b \leftarrow b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_b l^{(i)}(\mathbf{w}, b) = b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right).$$

4.2 softmax 回归

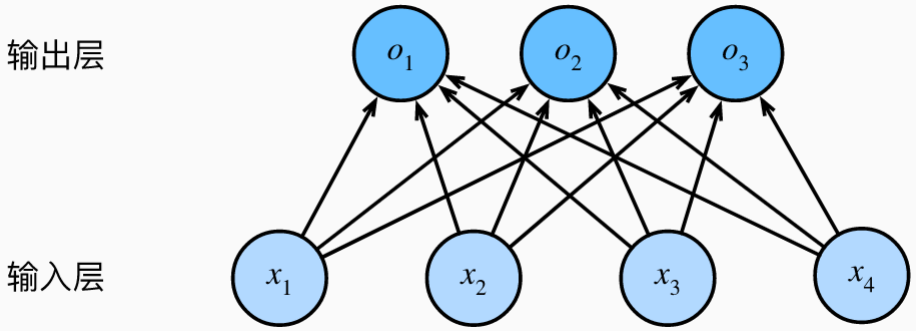
- **回归可以用于预测多少的问题。**
- **我们也对分类问题感兴趣：不是问“多少”，而是问“哪一个”：**
  - 某个电子邮件是否属于垃圾邮件文件夹?
  - 某个用户可能注册或不注册订阅服务?
  - 某个图像描绘的是驴、狗、猫、还是鸡?
  - 某人接下来最有可能看哪部电影?
- **问题示例：**我们从一个图像分类问题开始。 假设每次输入是一个2x2的灰度图像。 我们可以用一个标量表示每个像素值，每个图像对应四个特征。 此外，假设每个图像属于类别“猫”，“鸡”和“狗”中的一个。
- **关键组件**
  - 网络架构
  - 损失函数
  - 优化算法

4.2 softmax 回归

网络架构

- 为了估计所有可能类别的条件概率，我们需要一个有多个输出的模型，每个类别对应一个输出。

$$
\begin{aligned}
o_1 &= x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + b_1, \\
o_2 &= x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + b_2, \\
o_3 &= x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + b_3.
\end{aligned}
$$



4.2 softmax 回归

网络架构：softmax 运算

- 我们希望模型的输出y^j可以视为属于类j的概率，然后选择具有最大输出值的类别作为我们的预测。例如，如果y^1、y^2和y^3分别为0.1、0.8和0.1，那么我们预测的类别是2，在我们的例子中代表“鸡”。
- 因为将线性层的输出直接视为概率时存在一些问题：一方面，我们没有限制这些输出数字的总和为1。另一方面，根据输入的不同，它们可以为负值。
- 要将输出视为概率，我们必须保证在任何数据上的输出都是非负的且总和为1。

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}) \quad \text{其中} \quad \hat{y}_j = \frac{\exp(o_j)}{\sum_k \exp(o_k)}$$

4.2 softmax 回归

损失函数：交叉熵损失

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{j=1}^{q} y_j \log \hat{y}_j.$$

优化算法

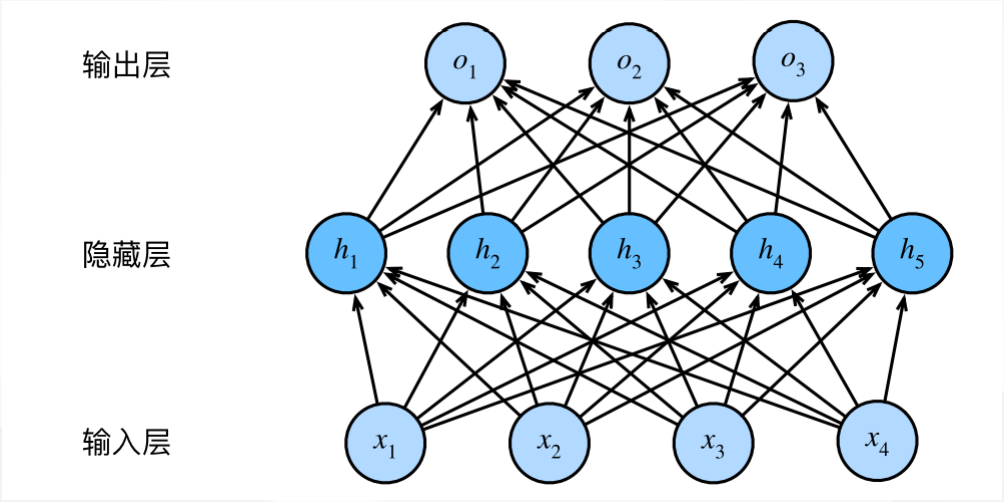
$$\begin{aligned} l(\mathbf{y}, \hat{\mathbf{y}}) &= -\sum_{j=1}^{q} y_j \log \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} \\ &= \sum_{j=1}^{q} y_j \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j \\ &= \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j. \end{aligned}$$

$$\partial_{o_j} l(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} - y_j = \mathrm{softmax}(\mathbf{o})_j - y_j.$$

4.3 多层感知机

隐藏层

- 我们可以通过在网络中加入一个或多个隐藏层来克服线性模型的限制，使其能处理更普遍的函数关系类型。
- 最简单的方法是将许多全连接层堆叠在一起。每一层都输出到上面的层，直到生成最后的输出。我们可以把前L−1层看作表示，把最后一层看作线性预测器。这种架构通常称为多层感知机。



- 这个多层感知机有4个输入，3个输出，其隐藏层包含5个隐藏单元。

4.3 多层感知机

- 我们通过矩阵X ∈ R^(n*d)来表示n个样本的小批量，其中每个样本具有d个输入特征。
- 对于具有h个隐藏单元的单隐藏层多层感知机，用H ∈ R^(n*h) 表示隐藏层的输出。

$$\mathbf{H} = \mathbf{X}\mathbf{W}^{(1)} + \mathbf{b}^{(1)},$$
$$\mathbf{O} = \mathbf{H}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}.$$

$$\mathbf{O} = (\mathbf{X}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + \mathbf{b}^{(2)} = \mathbf{X}\mathbf{W}^{(1)}\mathbf{W}^{(2)} + \mathbf{b}^{(1)}\mathbf{W}^{(2)} + \mathbf{b}^{(2)} = \mathbf{X}\mathbf{W} + \mathbf{b}.$$

- 在仿射变换之后对每个隐藏单元应用非线性的激活函数（activation function）σ。

激活函数

- ReLU

$$\mathrm{ReLU}(x) = \max(x, 0).$$

- Sigmoid 函数

$$\mathrm{sigmoid}(x) = \frac{1}{1 + \exp(-x)}.$$

- Tanh 函数

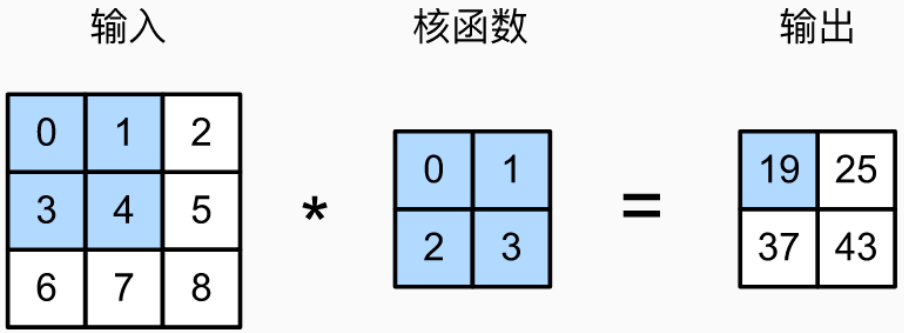
$$\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}.$$

4.4 从全连接层到卷积

- 我们之前讨论的多层感知机十分适合处理表格数据，其中行对应样本，列对应特征。对于表格数据，我们寻找的模式可能涉及特征之间的交互，但是我们不能预先假设任何与特征交互相关的先验结构。此时，多层感知机可能是最好的选择，然而对于高维感知数据，这种缺少结构的网络可能会变得不实用。
- 假设我们有一个足够充分的照片数据集，数据集中是拥有标注的照片，每张照片具有百万级像素，这意味着网络的每次输入都有一百万个维度，隐藏层维度降低到1000。

4.4 图像卷积

- 卷积神经网络（convolutional neural networks，CNN）是机器学习利用自然图像中一些已知结构的创造性方法。
- 互相关计算



- 有时，在应用了连续的卷积之后，我们最终得到的输出远小于输入大小。这是由于卷积核的宽度和高度通常大于1所导致的。比如，一个240*240像素的图像，经过10层的5*5卷积后，将减少到像素200*200。如此一来，原始图像的边界丢失了许多有用信息。而填充是解决此问题最有效的方法。
- 有时，我们可能希望大幅降低图像的宽度和高度。例如，如果我们发现原始的输入分辨率十分冗余。步幅则可以在这类情况下提供帮助。

填充

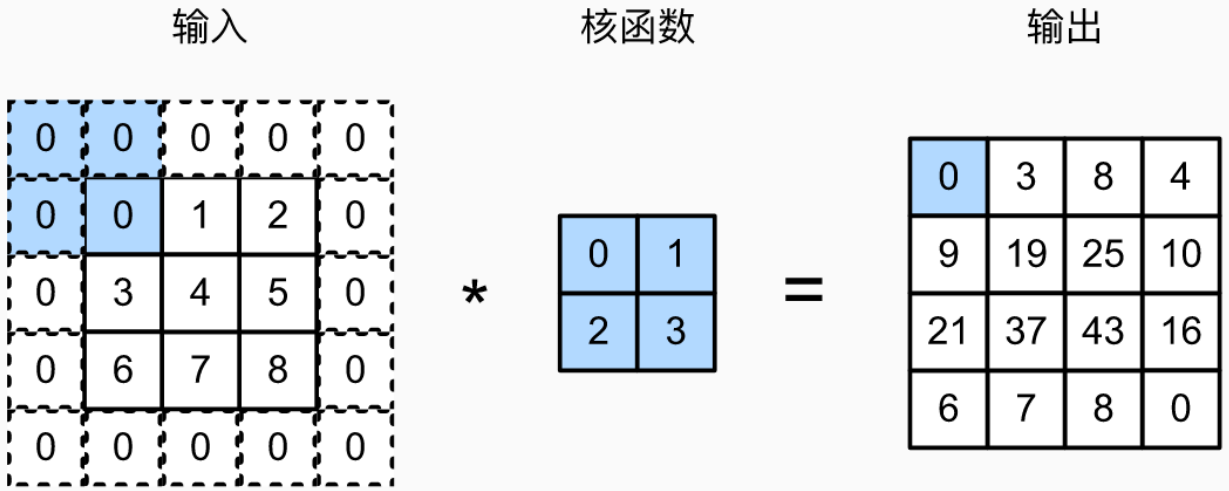


图6.3.1 带填充的二维互相关。

步幅

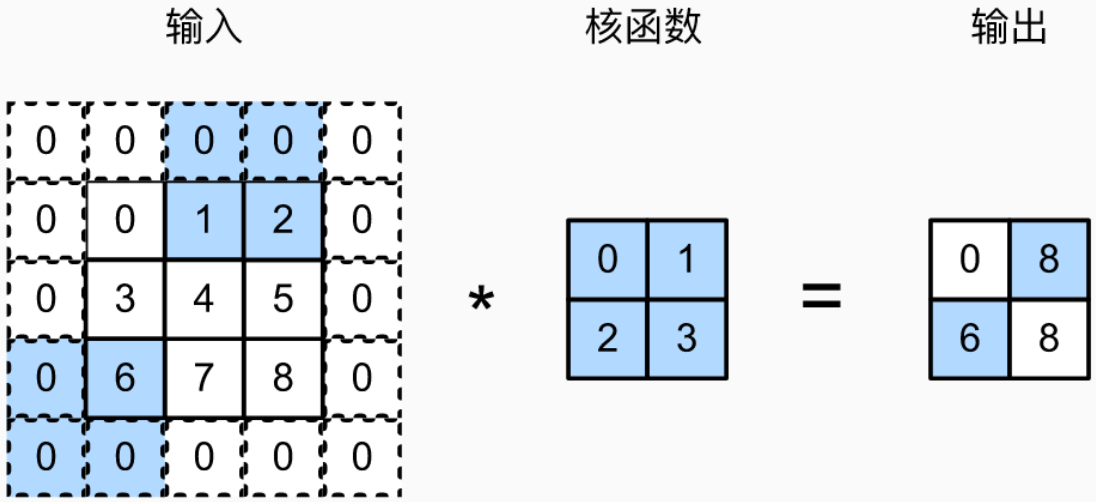


图6.3.2 垂直步幅为 3，水平步幅为 2 的二维互相关运算。

4.4 图像卷积

- 填充可以增加输出的高度和宽度。这常用来使输出与输入具有相同的高和宽。
- 步幅可以减小输出的高和宽，例如输出的高和宽仅为输入的高和宽的（是一个大于的整数）。
- 填充和步幅可用于有效地调整数据的维度。
- 图像的维度为 n，填充为 p，步幅为 s，卷积核的维度为 k，输出结果的维度为?

$$(N - k + 2 * p) / s + 1$$

4.4 图像卷积

多输入通道

- 当输入包含多个通道时，需要构造一个与输入数据具有相同输入通道数的卷积核，以便与输入数据进行互相关运算。

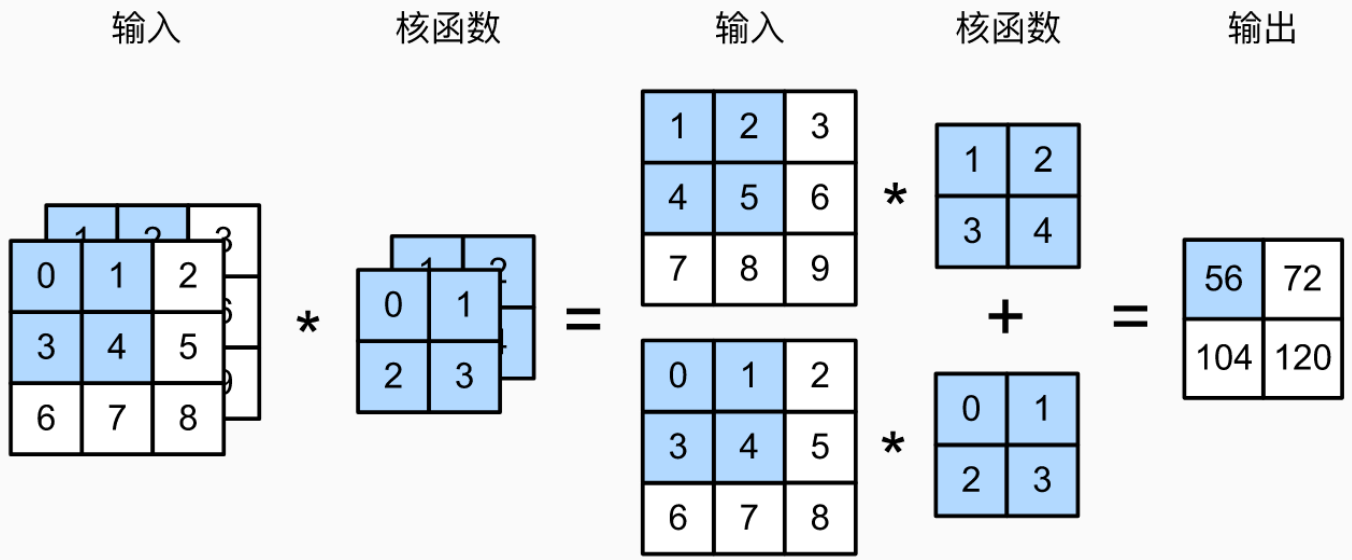


图6.4.1 两个输入通道的互相关计算。

4.4 图像卷积

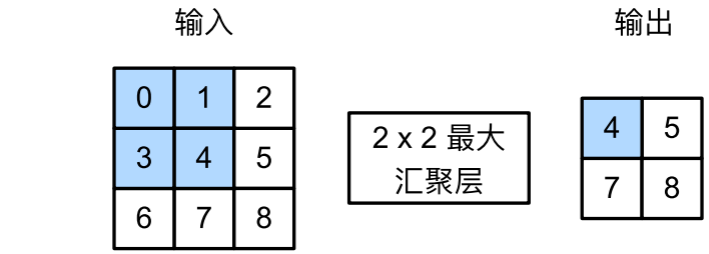
多输出通道

- 在最流行的神经网络架构中，随着神经网络层数的加深，我们常会增加输出通道的维数，通过减少空间分辨率以获得更大的通道深度。
- 直观地说，我们可以将每个通道看作是对不同特征的响应。而现实可能更为复杂一些，因为每个通道不是独立学习的，而是为了共同使用而优化的。
- 因此，多输出通道并不仅是学习多个单通道的检测器。
- 用ci和co分别表示输入和输出通道的数目，并让kh和kw为卷积核的高度和宽度。为了获得多个通道的输出，我们可以为每个输出通道创建一个形状为ci × kh × kw的卷积核张量，这样卷积核的形状是co × ci × kh × kw。在互相关运算中，每个输出通道先获取所有输入通道，再以对应该输出通道的卷积核计算出结果。

4.4 图像卷积

汇聚层

- 通常当我们处理图像时，我们希望逐渐降低隐藏表示的空间分辨率、聚集信息，这样随着我们在神经网络中层叠的上升，每个神经元对其敏感的感受野（输入）就越大。
- 汇聚（pooling）层，它具有双重目的：降低卷积层对位置的敏感性，同时降低对空间降采样表示的敏感性。
- 最大汇聚层和平均汇聚层



- 上图平均汇聚层的输出结果为多少?

4.4 卷积神经网络（LeNet）

- 它是最早发布的卷积神经网络之一，因其在计算机视觉任务中的高效性能而受到广泛关注。
- 这个模型是由AT&T贝尔实验室的研究员Yann LeCun在1989年提出的（并以其命名），目的是识别图像中的手写数字。
- 当时，Yann LeCun发表了第一篇通过反向传播成功训练卷积神经网络的研究，这项工作代表了十多年来神经网络研究开发的成果。

4.4 卷积神经网络（LeNet）

- 每个卷积块中的基本单元是一个卷积层、一个sigmoid激活函数和平均汇聚层。请注意，虽然ReLU和最大汇聚层更有效，但它们在20世纪90年代还没有出现。每个卷积层使用5*5卷积核和一个sigmoid激活函数。这些层将输入映射到多个二维特征输出，通常同时增加通道的数量。第一卷积层有6个输出通道，而第二个卷积层有16个输出通道。每个2*2池操作（步骤2）通过空间下采样将维数减少4倍。卷积的输出形状由批量大小、通道数、高度、宽度决定。

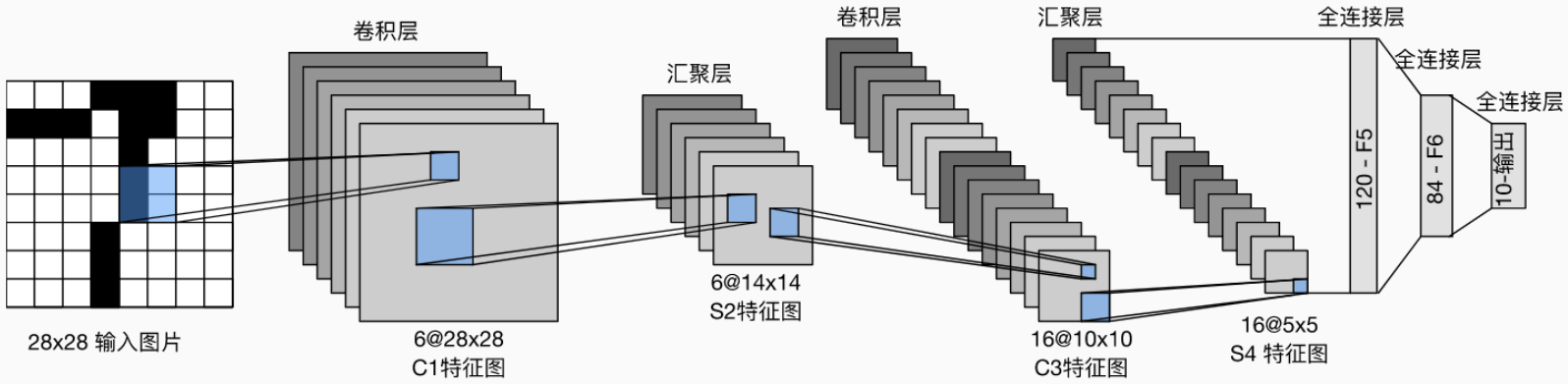


图6.6.1 LeNet中的数据流。输入是手写数字，输出为10种可能结果的概率。

4.4 AlexNet

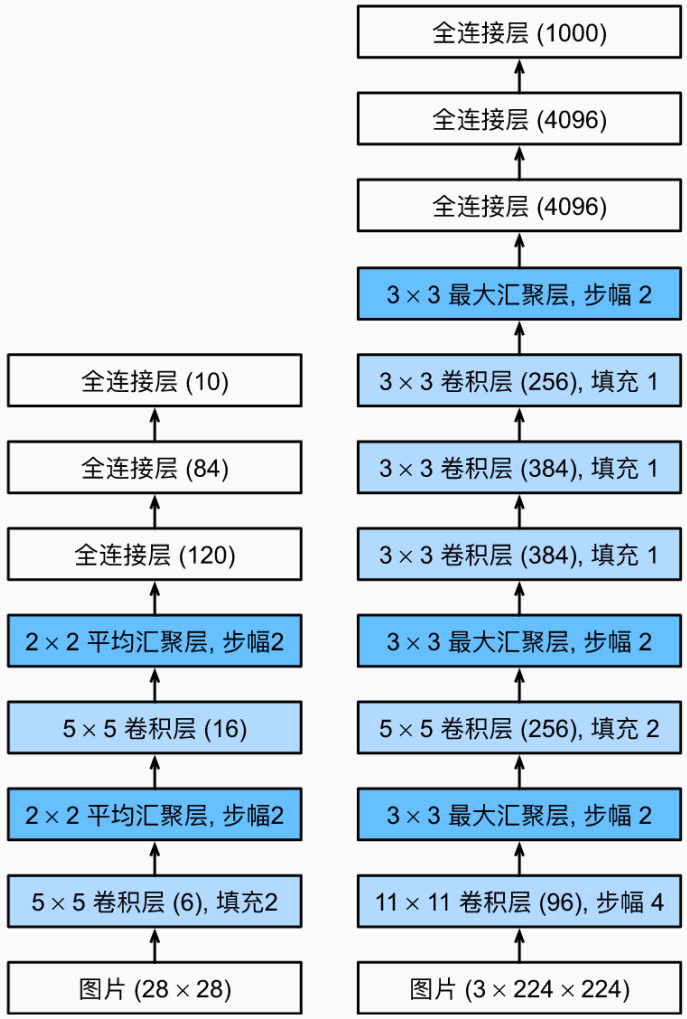


图7.1.2 从LeNet（左）到AlexNet（右）

4.4 VGG

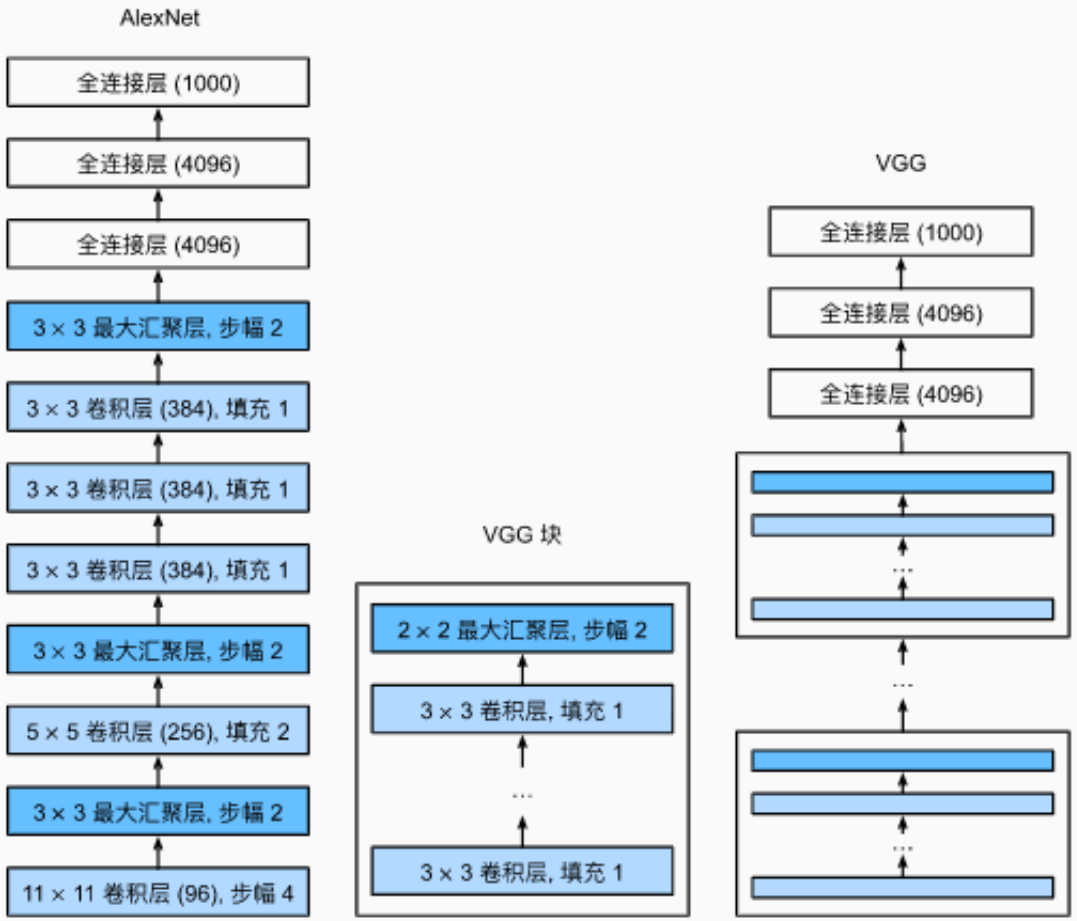


图7.2.1 从AlexNet到VGG，它们本质上都是块设计。

4.4 GoogleNet

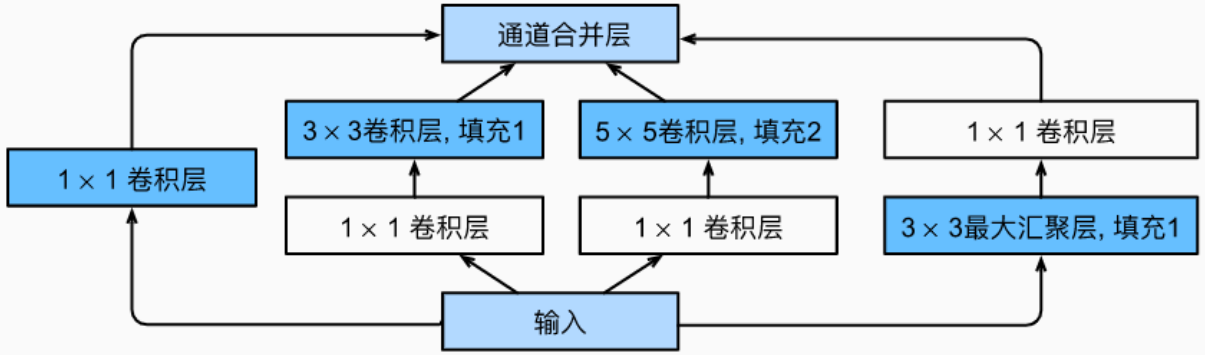


图7.4.1 Inception块的架构。

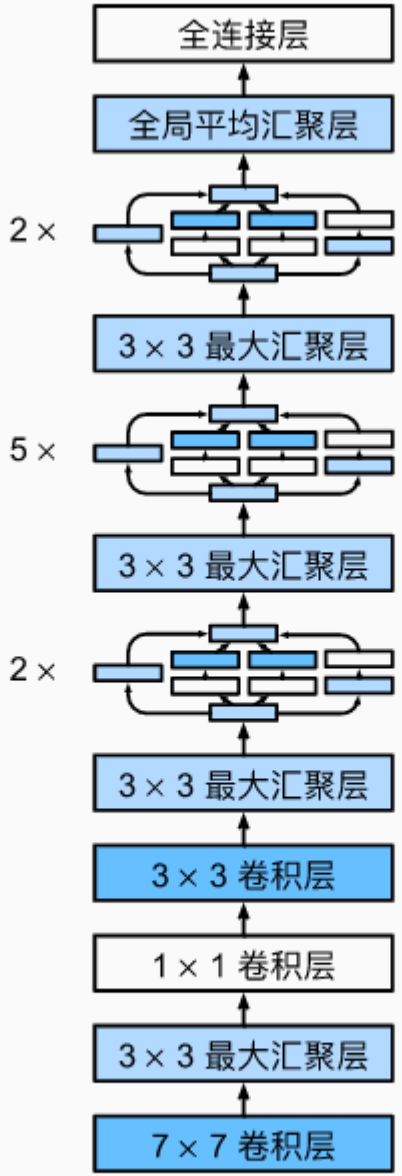
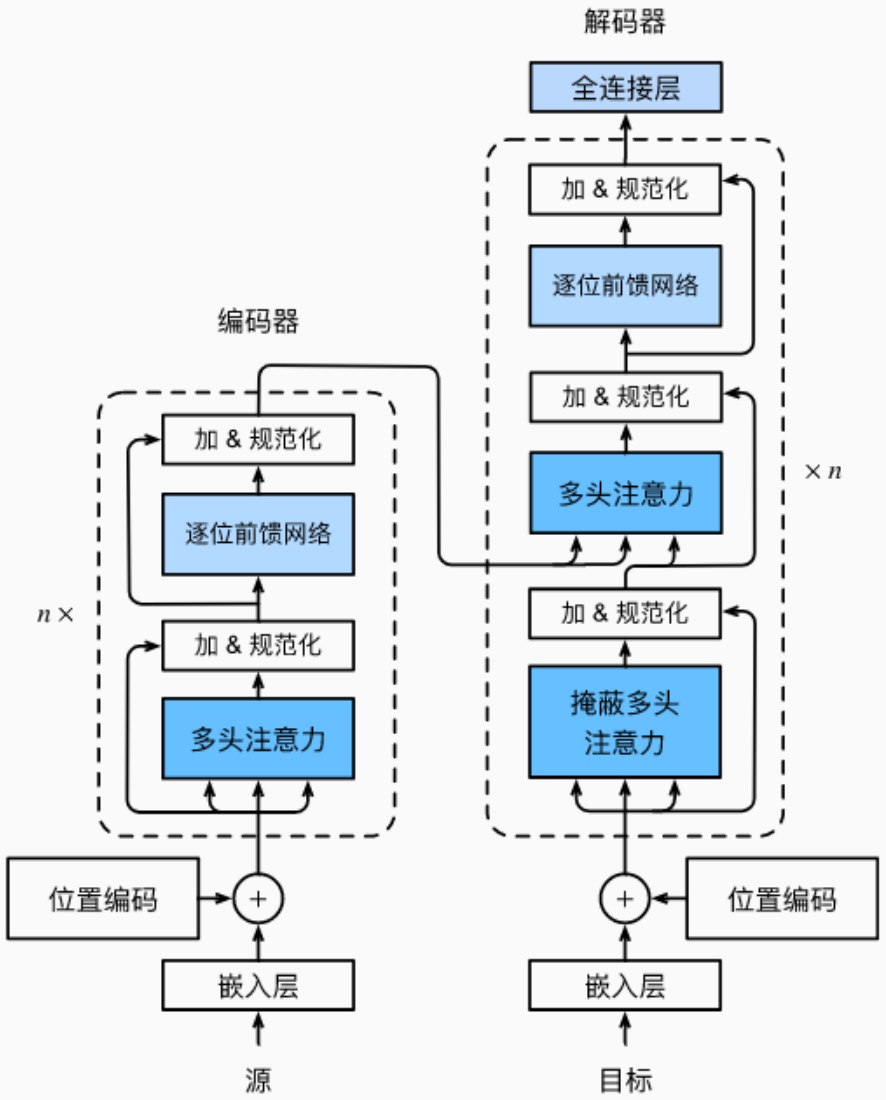


图7.4.2 GoogLeNet架构。

4.4 More Models

- ResNet
- MobileNet
- Transformer



谢谢！

饮 水 思 源　爱 国 荣 校