第3章

基础软硬件平台

3.1 智能芯片

经过长期的发展和探索，近几年人工智能不断取得突破性进展，几乎逐渐涵盖了人类生产、生活的方方面面。人工智能的背后需要强大的计算能力作为支撑，因此提供核心运算能力的智能芯片成为实现人工智能的关键。

3.1.1 智能芯片的定义

1. 芯片的概念

芯片是半导体元件产品的统称，又称微电路、微芯片、集成电路，是指内含集成电路的硅片，体积很小，常常是计算机或其他电子设备的一部分。半导体是一类材料的总称，集成电路是用半导体材料制成的电路的大型集合，芯片是由不同种类型的集成电路或者单一类型集成电路形成的产品。

2. 人工智能芯片的定义

从广义上讲，只要能够运行人工智能算法的芯片都可以称为人工智能芯片，但是，通常意义上的人工智能芯片指的是，针对人工智能算法做了特殊加速设计的芯片。因此，人工智能芯片也称为 AI 加速器（人工智能加速器）或计算卡，即专门用于处理人工智能应用中的大量计算任务的模块。

3. 人工智能芯片与传统芯片的区别

传统的CPU(中央处理器)芯片不适合人工智能算法的执行。传统芯片计算指令遵循串行执行的方式,背负着指令调度、指令寄存、指令翻译、编码、运算核心和缓存等与人工智能算法无关的任务,运算能力是受限的。

GPU(图形处理器)芯片在传统CPU芯片的基础上做了简化,处理的数据类型单一,加入了更多的浮点运算单元,更加适合大量算术计算而少逻辑运算的场合。在进行AI运算时,在性能、功耗等很多方面远远优于CPU,所以才经常被拿来"兼职"处理AI运算,但功耗较大,且成本昂贵。

人工智能芯片的设计思想是基于算法的角度精简GPU架构,加入更多的运算单元,在应用场景和算法相对确定的基础上,使得硬件设计上更加专门化。传统芯片和人工智能芯片最大的不同在于:前者是为通用功能设计的架构,后者是为专用功能优化的架构。这一区别决定了即便是最高效的GPU,与人工智能芯片相比,在时延、性能、功耗、能效比等方面也是有差距的,因而研发人工智能芯片是必然趋势,也将带来更高的性能、更低的功耗。

3.1.2 智能芯片的分类

人工智能芯片分类一般有按技术架构分类、按功能分类、按应用场景分类三种分类方式,如图3.1所示。

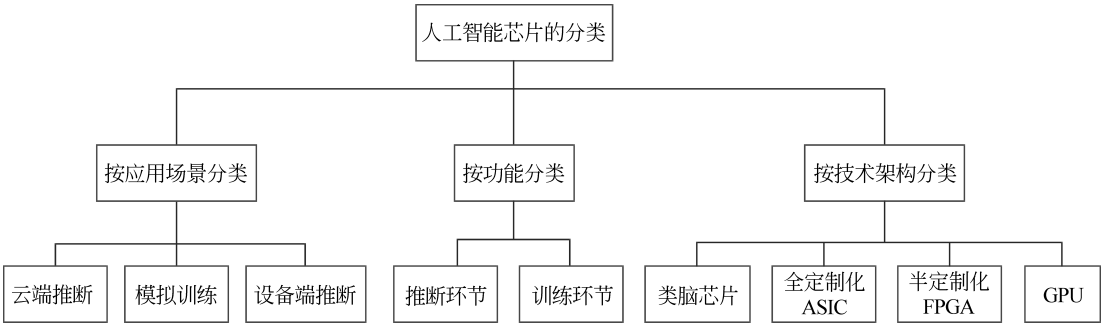

图3.1 人工智能芯片分类

1. 按技术架构分类

智能芯片按照技术架构,可以分为通用芯片、半定制化芯片、全定制化芯片和类脑芯片。

(1) 通用芯片(GPU):GPU是单指令、多数据处理,主要处理图像领域的运算加速。GPU不能单独使用,它只是处理大数据计算时的能手,必须由CPU进行调用,下达指令才能工作。而CPU可单独作用,处理复杂的逻辑运算和不同的数据类型,但当需要处理大数据计算时,则可调用GPU进行并行计算。

(2) 半定制化芯片(FPGA):FPGA适用于多指令、单数据流的分析,这与GPU相反。FPGA是用硬件实现软件算法的,因此在实现复杂算法方面有一定的难度,缺点是价格比较高。与GPU不同,FPGA同时拥有硬件流水线并行和数据并行处理能力,适用于以硬件流水线方式处理一条数据,且整数运算性能更高,因此常用于深度学习算法中的推断阶段。将

FPGA 和 CPU 对比可以发现两个特点，一是 FPGA 不需要频繁访问内存进行存储、读取等操作，效率更高；二是 FPGA 没有读取指令操作，所以功耗更低。劣势是价格比较高、编程复杂、整体运算能力不是很高。

(3) 全定制化芯片(ASIC)：ASIC 是为实现特定场景应用要求而定制的专用 AI 芯片。除了不能扩展以外，它在功耗、可靠性、体积方面都有优势，尤其在高性能、低功耗的移动设备端。定制的特性有助于提高 ASIC 的性能功耗比，缺点是电路设计需要定制，开发周期相对长，功能难以扩展。但在功耗、可靠性、集成度等方面都有优势，尤其在要求高性能、低功耗的移动应用端体现明显。

(4) 类脑芯片：类脑芯片架构是一款模拟人脑的神经网络模型的新型芯片编程架构，这一系统可以模拟人脑功能进行感知方式、行为方式和思维方式。真正的人工智能芯片未来的发展方向是类脑芯片。

2. 按功能分类

根据机器学习算法步骤，可分为训练(Training)和推断(Inference)两个环节。

(1) 训练环节通常需要通过大量的数据输入，训练出一个复杂的深度神经网络模型。训练过程由于涉及海量的训练数据和复杂的深度神经网络结构，运算量巨大，需要庞大的计算规模，对于处理器的计算能力、精度、可扩展性等性能要求很高。

(2) 推断环节是指利用训练好的模型，使用新的数据去“推断”出各种结论。这个环节的计算量相对训练环节少很多，但仍然会涉及大量的矩阵运算。在推断环节中，除了使用 CPU 或 GPU 进行运算外，FPGA 以及 ASIC 均能发挥重大作用。

3. 按应用场景分类

智能芯片按照用途可以分为三类，分别是模拟训练、云端推断、设备端推断，如图 3.2 所示。

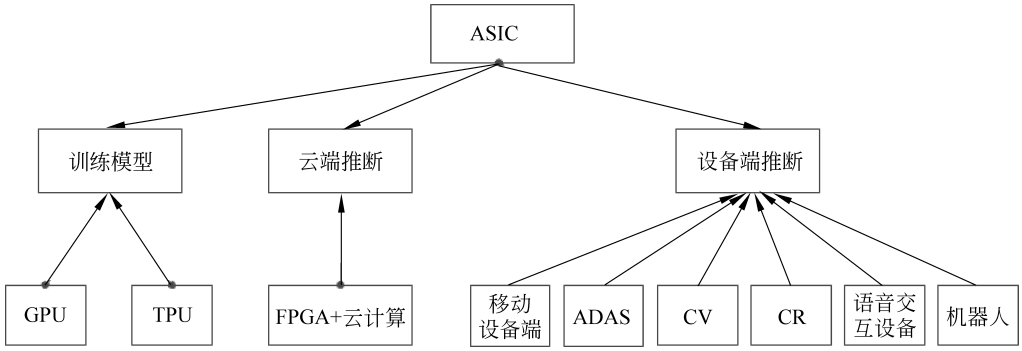

图 3.2　按应用场景分类

(1) 模拟训练环节的芯片：这个过程要处理海量的数据和复杂的深度神经网络。

(2) 云端推断的芯片：目前主流的 AI 应用需要通过云端提供服务，将采集到的数据传到云端服务器，在服务器的 CPU、GPU、TPU 处推断任务，然后再将处理结果返回终端。

(3) 设备端推断的芯片：也可称为嵌入式设备的芯片，比如智能手机、智能安防摄像头、机器人等设备就是采用这类芯片。

3.1.3 智能芯片的组成

在人工智能处理方面，虽然 GPU 的表现通常比 CPU 更好，但并不完美。因此，芯片设计人员现在正在努力创建针对执行人工智能算法而优化的处理单元。这些单元有很多名称，例如 NPU、TPU 等，我们可以采用人工智能处理单元（AI PU）这个笼统的术语进行概述。虽然 AI PU 构成了芯片上人工智能系统（SoC）的大脑，但它也只是组成芯片的一系列复杂组件的一部分。下面将介绍 AI SoC，包括 AI PU 和一些与之配对的组件，以及它们如何协同工作。

1. AI PU

如上所述，AI PU 是执行 AI SoC 核心操作的神经处理单元或矩阵乘法引擎，需要指出的是，对于人工智能芯片制造商来说，这也是任何 AI SoC 从所有其他 AI SoC 中脱颖而出的关键。

2. 控制器

控制器通常基于 RISC-V（由加州大学伯克利分校设计）、ARM（由 ARM 公司设计）或自定义逻辑指令集架构（ISA），用于控制所有其他模块和外部处理器并与之通信。

3. SRAM

SRAM 是指用于存储模型或输出的本地存储器，尽管存储空间很小，但可以快速方便地获取数据或将数据放回去。在某些用例中，尤其是与边缘人工智能有关的情况下，处理速度至关重要。例如，当行人突然出现在路上时，自动驾驶汽车必须及时刹车。芯片中包含多少 SRAM 取决于成本与性能。更大的 SRAM 存储池将会增加前期成本。

4. I/O

I/O 模块用于将 SoC 连接到 SoC 之外的组件，例如外部处理器。这些接口对于 AI SoC 最大化其潜在性能和应用程序至关重要，否则会造成瓶颈。

5. 互连结构

互连结构是处理器（AI PU、控制器）和 SoC 上所有其他模块之间的连接。与 I/O 一样，互连结构对于提取 AI SoC 的所有性能至关重要。无论处理器有多快，都只在互连结构能够保持正常运行且不会造成阻碍整体性能延迟的情况下起作用。比如，如果城市道路上没有足够的车道，就会在上下班时段造成交通拥堵。

所有这些组件都是人工智能芯片的关键部分。虽然不同的芯片可能有额外的组件，或者对这些组件的投资有不同的优先级，但这些基本组件以共生的方式协同工作，以确保人工智能芯片能够快速有效地处理人工智能模型。不过，与 CPU 和 GPU 不同，AI SoC 的设计还远未成熟。这一部分的产业正在持续快速发展，人们将会看到 AI SoC 设计方面的进步。

3.1.4 智能芯片的特征

1. 新型计算范式

AI 计算既不脱离传统计算，也具有新的计算特质，包括处理的内容往往是非结构化数

据，例如视频、图像及语音等。这类数据很难通过预编程的方法得到满意的结果。因此，需要通过新的计算方式来处理相应数据。

2. 大数据处理能力

人工智能的发展高度依赖海量的数据。满足高效能机器学习的数据处理要求，是 AI 芯片需要考虑的最重要因素。一个无法回避的现实是，运算单元与内存之间的性能差距越来越大，内存子系统成为芯片整体处理能力提高的瓶颈。人工智能工作负载大多是数据密集型，需要大量的存储和各层次存储器间的数据搬移，导致上述问题更加突出。为了弥补计算单元和存储器之间的差距，学术界和工业界正在两个方向上进行探索：一是富内存的处理单元，另一个方向是研究具备计算能力的新型存储器。

3. 数据精度

低精度设计是 AI 芯片的一个趋势，在针对推断的芯片中更加明显。对一些应用来说，降低精度的设计不仅加速了机器学习算法的训练和推断，甚至可能更符合神经形态计算的特征。已经证明，对于学习算法和神经网络的某些部分，使用尽可能低的精度就足以达到预期效果，同时可以节省大量内存，降低能量消耗。通过对数据上下文数据精度的分析和对精度的舍入误差敏感性，来动态地进行精度的设置和调整，将是 AI 芯片设计优化的必要策略。

4. 可重构能力

人工智能各领域的算法和应用还处在高速发展和快速迭代的阶段，考虑到芯片的研发成本和周期，针对特定应用、算法或场景的定制化设计很难适应变化。针对特定领域而不针对特定应用的设计，将是 AI 芯片设计的一个指导原则，具有可重构能力的 AI 芯片可以在更多应用中大显身手，并且可以通过重新配置，适应新的 AI 算法、架构和任务。

5. 软件工具

就像传统的 CPU 需要编译工具的支持，AI 芯片也需要软件工具链的支持，才能将不同的机器学习任务和神经网络转换为可以在 AI 芯片上高效执行的指令代码。基本处理、内存访问以及任务的正确分配和调度，将是工具链中需要重点考虑的因素。对 AI 芯片来说，构建一个集成化的流程，将 AI 模型的开发和训练、硬件无关和硬件相关的代码优化、自动化指令翻译等功能无缝地结合在一起，将是关键要求。

3.1.5　智能芯片的应用模式

1. 常规功能

(1) 训练与推理：人工智能本质上是使用人工神经网络对人脑进行的模拟，人工神经网络旨在替代人们大脑中的生物神经网络。神经网络由大量节点组成，可以调用这些节点以执行模型。这就是人工智能芯片发挥作用的地方。人工智能芯片尤其擅长处理这些人工神经网络，并且被设计为对它们做两件事：训练和推理。原始的神经网络最初是通过输入大量数据来开发和训练的。训练非常耗费计算资源，因此需要专注于训练的人工智能芯片，这些芯片旨在能够快速有效地处理这些数据。芯片功能越强大，网络学习的速度就越快。

一旦网络经过训练，它就需要为推理而设计的芯片，以便在现实世界中使用数据。可以将训练视为构建一个字典，而推理类似于查找单词并理解如何使用它们。这两者都是必要且共生的。值得注意的是，为训练而设计的芯片也可以进行推理，但是推理芯片无法进行训练。

(2) 云计算与边缘计算：人们需要知道的人工智能芯片的另一方面是，它是为云计算用例还是边缘计算用例设计的？而对于这些用例，人们是需要采用推理芯片还是训练芯片？云计算的可访问性非常有用，因为它的功能可以完全在场外使用，不需要采用设备上的芯片来处理这些用例中的推理，从而可以节省功耗和成本。但是，在隐私和安全性方面存在弊端，因为数据存储在可能被黑客入侵或处理不当的云计算服务器上。对于推理用例，它的效率也可能较低，因为它不如边缘计算芯片那么专业。

2. 应用程序和芯片配对应用

(1) 云计算＋训练：这种配对的目的是开发用于推理的人工智能模型。这些模型最终被细化为特定用例的人工智能应用程序。这些芯片功能强大，运行成本高，其设计的目的是尽可能快地进行训练，例如，英特尔 Habana 的 Gaudi 芯片。人们每天接触到的需要大量训练的应用程序示例包括 Facebook 照片或 Google 翻译。

(2) 云计算＋推理：这种配对的目的是推理需要大量处理能力，以至无法在设备上进行推理。这是因为应用程序使用更大的模型并处理大量数据。例如，高通公司的 Cloud AI 100，这是用于大型云平台的人工智能芯片。

(3) 边缘计算＋推理：使用边缘计算设备的芯片进行推理可以消除任何与网络不稳定或延迟有关的问题，并且可以更好地保护所使用数据的私密性和安全性。使用上传大量数据(尤其是像图像或视频之类的视觉数据)所需的带宽并没有相关的成本，因此，只要平衡成本和能效，它就可以比云计算＋推理更便宜、更高效，例如，英特尔 Movidius 和 Google 的 Coral TPU。其使用案例包括面部识别监控摄像头、用于行人和危险检测的车辆摄像头，以及语音助理的自然语言处理。

这些不同类型的芯片及其不同的实现、模型和用例，对于未来人工智能的发展至关重要。

3.1.6 智能芯片的发展现状

1. 国内现状

当前，我国人工智能芯片行业正处于起步阶段，主要原因是国内人工智能芯片行业起步较晚，整体销售市场正处于快速增长阶段前夕，传统芯片的应用场景逐渐被人工智能专用芯片所取代，市场对于人工智能芯片的需求将随着云、边缘计算、智慧型手机和物联网产品一同增长，并且在这期间，国内的许多企业纷纷发布了自己的专用 AI 芯片。由于我国特殊的环境和市场，国内 AI 芯片的发展同时也呈现出百花齐放、百家争鸣的态势，AI 芯片的应用领域也遍布股票交易、金融、商品推荐、安防、早教机器人以及无人驾驶等众多领域，催生了大量的人工智能芯片创业公司。

2. 国际现状

尽管国内 AI 芯片发展快速，但根据芯片技术结构分类来看，各种类的人工智能芯片领

域几乎都能看到国外半导体巨头的影子。反观国内的人工智能芯片企业，由于它们大部分是新创公司，所以在人工智能芯片领域的渗透率较低。从应用领域分类来看，NVIDA 在全球云端训练芯片市场一家独大，目前 NVIDA 的 GPU＋CUDA 计算平台是最成熟的 AI 训练方案。此外还有第三方异构计算平台 OpenCL＋AMD GPU 以及云计算服务商自主研发加速芯片这两种方案。全球各芯片厂商基于不同方案，都推出了针对云端训练的人工智能芯片。全球云端推断芯片竞争格局方面，云端推断芯片百家争鸣，各有千秋。相比训练芯片，推断芯片考虑的因素更加综合，包括单位功耗算力、时延、成本等。

3.1.7　智能芯片的未来发展趋势

目前主流 AI 芯片的核心主要是利用 MAC（Multiplier and Accumulation，乘加计算）加速阵列来实现对 CNN（卷积神经网络）中最主要的卷积运算的加速。这一代 AI 芯片主要有如下问题。

（1）深度学习计算所需数据量巨大，造成内存带宽成为整个系统的瓶颈，即所谓的 Memory Wall 问题。

（2）与第一个问题相关，内存大量访问和 MAC 阵列的大量运算，造成 AI 芯片整体功耗的增加。

（3）深度学习对算力要求很高，要提升算力，最好的方法是做硬件加速，但是同时深度学习算法的发展也是日新月异，新的算法可能在已经固化的硬件加速器上无法得到很好的支持，即性能和灵活度之间的平衡问题。

因此，我们可以预见，下一代 AI 芯片将有如下的几个发展趋势。

1. 更高效的大卷积解构/复用

在标准 SIMD 的基础上，CNN 由于其特殊的复用机制，可以进一步减少总线上的数据通信。而复用这一概念，在超大型神经网络中就显得格外重要。如何合理地分解、映射这些超大卷积到有效的硬件上，成了一个值得研究的方向。

2. 更低的推理计算/存储位宽

AI 芯片最大的演进方向之一可能就是神经网络参数/计算位宽的迅速减少——从 32 位浮点到 16 位定点、8 位定点，甚至是 4 位定点。在理论计算领域，2 位甚至 1 位参数位宽都已经逐渐进入实践领域。

3. 更多样的存储器定制设计

当计算部件不再成为神经网络加速器的设计瓶颈时，如何减少存储器的访问延时将会成为下一个研究方向。通常，离计算越近的存储器速度越快，每字节的成本也越高，同时容量也越受限，因此新型的存储结构也将应运而生。

4. 更稀疏的大规模向量实现

神经网络虽然大，但是，实际上有很多以零为输入的情况，此时稀疏计算可以高效地减少无用能效。来自哈佛大学的团队就该问题提出了优化的五级流水线结构，以达到减少无用功耗的目的。

5. 计算和存储一体化

计算和存储一体化(Process-in-Memory)技术,其要点是通过使用新型非易失性存储(如 ReRAM)器件,在存储阵列里面加上神经网络计算功能,从而省去数据搬移操作,即实现了计算存储一体化的神经网络处理,在功耗性能方面可以获得显著提升。现在的计算机,采用的都是冯·诺依曼架构。它的核心架构就是处理器和存储器是分开布局的,所以 CPU(中央处理器)和内存条没有集成在一起,只是在 CPU 中设置了容量极小的高速缓存。而类人脑架构是模仿人脑神经系统模型的结构,人脑中的神经元既是控制系统,同时又是存储系统。因此 CPU、内存条、总线、南北桥等,最终都必将集成在一起,形成类人脑的巨大芯片组。

3.2 人工智能系统软件

人工智能系统软件是管理智能硬件与软件资源的计算机程序,在作用上对应于人工智能操作系统(Artificial Intelligence Operating System,AIOS),上可支配应用,下可控制硬件,它控制着智能系统中硬件和应用软件之间的联系,也控制着智能设备的整个生态。人工智能操作系统的理论前身为 20 世纪 60 年代末由斯坦福大学提出的机器人操作系统。人工智能操作系统应具有通用计算机操作系统所具备的所有功能,并且包括语音识别、机器视觉、执行器系统和认知行为系统。

3.2.1 人工智能系统软件的特征

1. 并行计算

计算机系统处理指令集合的方式可以分为串行计算和并行计算。串行计算是指系统同一时刻只处理一个指令的算法,如图 3.3 所示。并行性是指两个或多个事件在同一时刻发生。所以并行计算相对于串行计算而言,是一种同一时刻可以执行多个指令的算法,如图 3.4 所示。

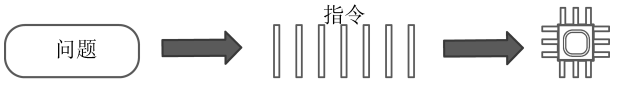

图 3.3 串行计算

并行计算是同时使用多种系统资源解决计算问题,提高计算机系统计算速度和处理能力的有效方法。其基本思想是,将求解问题分解为若干个部分,将系统一块较大的处理资源分为若干较小部分处理资源,使用每个小的系统计算处理单元解决每一小部分的问题,进而协同求解一个较大的问题。

并行计算可以分为时间上的并行和空间上的并行。时间并行是指流水线技术,例如食品工厂生产步骤可以分为清洗、加工、分割和包装。如果不采用流水线工作,一个食品只能等上一个加工完成之后才能开始制作,而采用时间并行方式可以在同一时间启动两个以上的操作,从而提高计算效率。空间并行是指多个计算单元能够协同处理同一个任务,即在一

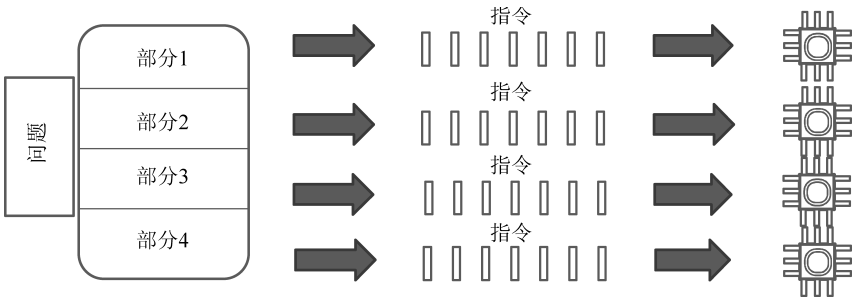

图 3.4　并行计算

段时间内，任务的不同部分在系统的不同计算单元同时运行。例如，小明需要耗费 3 小时打扫三间教室，而他求助了两个朋友，三个人一起打扫总共耗费 1 小时就完成打扫任务。

2. 学习、感知、认知能力

现在的人工智能系统软件处于感知智能到认知智能的发展阶段，智能系统软件具有较为成熟的感知能力和部分认知能力。

（1）学习：不完全依赖代码编程的学习。人工智能系统软件学习的首要特征就是随时间学习的能力，并且不需要明确编程。与人类一样，人工智能学习算法通过探索与实践学习，而不是遵循一步步的指令。在后面的章节中将会具体了解到，人工智能学习方式可以分为监督学习、无监督学习以及强化学习，人工智能系统软件应该具备这三种学习方式，才能在使用的过程中适应各种不同类型的信息输入。

（2）感知：解释周边世界。感知是指能够将接收到的自然界信息转换成自身能够处理的信息，并对转化后的信息产生响应，感知智能即人工智能系统通过“视觉”“听觉”“触觉”等感知能力与自然界进行交互。

（3）认知：基于数据进行推理。认知是感知的内在，是对内容更深层次的理解，它们不专注于感知周围的世界，而是对世界进行抽象并基于抽象进行推理。

当前，人工智能系统软件的认知特征在严格意义上属于弱认知，以往的数据仅仅是系统学习训练的前验知识，想要人工智能系统具备强认知能力，需要智能系统软件有完备的知识定义和存储形式，才能让系统和人类一样有自己的思维语言知识结构，对世界有足够的认识，而且需要解决两个或多个智能主体之间的通信问题。认知智能强依赖于外源知识，这些外源知识是通信过程的“背景知识”，无论是词的解释、常识还是领域知识等，都是智能系统主体之间共享的知识，不能从字面上解释。让知识在智能主题间流动起来才能创造价值，能够解决认知智能的问题。

3. 具有支持微型 MCU 和众多传感器的特性

微控制单元（Microcontroller Unit，MCU）也称为单片机，相当于芯片级的计算机。传感器和微控单元作为物联网的基础，与人工智能系统、机器人之间是相互影响促进的关系。传感器会产生大量的数据，庞大的数据量使人工智能系统的学习结果更加准确、更智能。智能系统生成的学习结果指导机器人更精确地执行任务，机器人行动的过程中又会触发传感器形成一个完整的良性循环。

人工智能系统软件与传感器的紧密结合能够使我们在未来的智能应用中自主做出以信

息为主导的决策。

4. 实时性

人工智能系统软件的实时性表现在通过系统本身强大的计算能力，将大量高维度的数据用最短时间来处理实现。人工智能系统软件未来必将会应用到越来越多的实际领域当中，如果没有以其超强算力为基础的实时计算能力，在面对日后愈加复杂的应用场景时将会很难落地实施。

3.2.2 人工智能系统软件的功能构成

人工智能系统软件应该具备传统计算机操作系统的典型功能，同时还应具备对智能功能的支撑。

1. 处理机管理功能

在传统的多道程序系统中，处理机的分配和运行都是以进程为基本单位，因而对处理机的管理可归结为对进程的管理；在引入了线程的 OS 中，也包含对线程的管理。处理机管理的主要功能是创建和撤销进程(线程)，对各进程(线程)的运行进行协调，实现进程(线程)之间的信息交换，以及按照一定的算法把处理机分配给进程(线程)。

2. 存储器管理功能

存储器管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，从而提高存储器的利用率以及能从逻辑上扩充内存。为此，存储器管理应具有内存分配、内存保护、地址映射和内存扩充等功能。

3. 设备管理功能

设备管理用于管理计算机系统中所有的外围设备，而设备管理的主要任务是：完成用户进程提出的输入/输出(Input/Output，I/O)请求，为用户进程分配其所需的 I/O 设备；提高 CPU 和 I/O 设备的利用率；提高 I/O 速度，方便用户使用 I/O 设备。为实现上述任务，设备管理应具有缓冲管理、设备分配和设备处理以及虚拟设备等功能。

4. 文件管理功能

在现代计算机管理中，总是把程序和数据以文件的形式存储在磁盘和磁带上，供所有的或指定的用户使用。为此，在操作系统中必须配置文件管理机制。文件管理的主要任务是对用户文件和系统文件进行管理，以方便用户使用，并保证文件的安全性。为此，文件管理应具有对文件存储空间的管理、目录管理、文件的读/写管理，以及文件的共享与保护等功能。

5. 语音识别功能

语音识别功能即与机器进行语音交流的能力。让机器明白你说什么，这是人们长期以来梦寐以求的事情。人工智能系统软件所需要的一般是大词汇量的、连续性的语音识别系统。语音识别系统应具备以下功能：

(1) 将连续的讲话分解为词、音素等单位来理解语义。

(2) 存储语音信息量大。语音模式不仅对不同的说话者不同，对同一说话者也是不同

的。例如，一个说话者在随意说话和认真说话时的语音信息是不同的，一个人的说话方式随着时间变化也会变化。

(3) 具备正确的识别率。说话者在讲话时不同的词听起来可能是相似的。这在英语和汉语中最常见。

(4) 具备良好的抗噪能力。环境噪声和干扰对语音识别有严重影响，致使识别率低。

目前，国外的应用一直以苹果的 Siri 为龙头，而国内方面，科大讯飞、云知声、盛大、捷通华声、搜狗语音助手、紫冬口译、百度语音等系统都采用了最新的语音识别系统，市面上其他相关的产品也直接或间接嵌入了类似的技术。

6. 机器视觉功能

机器视觉功能就是利用机器代替人眼来做各种测量和外部场景的精准判断。机器视觉的优点包括以下几点：

(1) 非接触测量，从而提高系统的可靠性和智能性。

(2) 具有较宽的光谱响应范围。例如，对人眼看不见的红外光的测量，扩展了人眼的视觉范围。

(3) 长时间稳定工作。

7. 认知功能

认知功能是指人工智能系统软件应具备提供机器的推理、认知的特性。认知是获取和处理知识的能力，它包含人脑用于推理、理解、解决问题、计划和决策的高层次概念。认知是对内容更深层次的理解，它不是专注于感知周围的世界，而是对世界进行抽象并基于抽象进行推理。

3.3　人工智能开发框架

随着计算机技术的发展和应用范围的不断延伸，作为计算机灵魂的软件系统，其规模也在不断扩大，结构越来越复杂，代码越来越冗长，从过去的几百行代码，到几万、几十万甚至几百万行代码的软件系统比比皆是。为了解决这些问题，在软件开发中，将基础的代码进行封装，以形成模块化的代码，并提供相应的应用程序编程接口(Application Programming Interface，API)，开发者在软件开发过程中直接调用 API，不必再考虑太多的底层功能的操作，在此基础上进行后续的软件开发设计。这种在软件开发中对通用功能进行封装并且可重用的设计就是开发框架。

3.3.1　开发框架的作用

开发框架在软件开发的过程中起着不可或缺的作用，开发框架能够屏蔽掉底层烦琐的开发细节，给开发者提供简单的开发接口，在软件开发时只需调用框架就可以实现一定的功能。由于框架具有可复用特点，利用框架来实现软件开发的时候，不仅写起来容易，而且可读性很好，极大地降低了软件开发时的复杂度，提高了效率与质量。

人工智能软件比传统的计算机软件更加复杂，其复杂之处主要体现在比传统的计算机

软件更具智能性，人工智能软件开发任务也变得更加困难。人工智能的智能化主要依靠算法来实现。由于人工智能算法的复杂性，在没有框架以前，人工智能软件的开发只针对非常专业的人，一般的软件开发人员要进行人工智能软件开发是一件望尘莫及的事，能够进行人工智能软件开发的人也是凤毛麟角。而框架的出现，为人工智能开发提供了智能单元，实现了对人工智能算法的封装、数据的调用以及计算资源的调度使用，提升了效率，极大地降低了人工智能系统开发的复杂性。

3.3.2 人工智能开发框架的核心特征

利用恰当的框架来快速构建模型，而无需编写数百行代码。一个良好的人工智能开发框架应该具有良好的性能并且易于开发人员理解和使用，以减少计算，提高人工智能软件开发效率。人工智能开发框架的核心特征有如下几点。

(1) 规范化：一个良好的开发框架应严格执行代码开发规范要求，便于使用者理解与掌握。

(2) 代码模块化：开发框架一般都有统一的代码风格，同一分层的不同类代码，具有相似的模板化结构，方便使用模板工具统一生成，减少大量重复代码的编写。

(3) 可重用性：开发人员可以不做修改或稍加改动，就可以在不同环境下重复使用功能模块。

(4) 封装性(高内聚)：开发人员将各种需要的功能代码进行集成，调用时不需要考虑功能的实现细节，只需要关注功能的实现结果。

(5) 可维护性：成熟的开发框架对于二次开发或现有功能的维护来说，进行添加、修改或删除某一个功能时不会对整体框架产生不利影响。

3.3.3 典型的人工智能开发框架

框架的使用极大降低了人工智能系统开发的复杂性。业界主流的开发框架主要有以下几种。

1. TensorFlow

TensorFlow 由谷歌人工智能团队谷歌大脑(Google Brain)开发和维护，是谷歌针对第一代分布式机器学习的学习框架 DistBelief 总结经验教训而形成的，自 2015 年 11 月 9 日起，开放源代码。它还拥有强大、活跃的社区支持，使得质量得到了快速提升。TensorFlow 支持多种编程语言，包括 Python、C、C++ 、Go、R、Java 等。

TensorFlow 采用计算图实现算法编程和用于数值计算的框架。计算图用节点(node)和线(edge)的有向图来描述数学计算，其中节点在图中表示数学操作，也可表示数据的输入(input)和输出(output)，线在图中表示节点间相互联系的多维数据数组，即张量(tensor)。TensorFlow 程序一般分为两个阶段，构建计算图阶段和执行计算图阶段。

TensorFlow 灵活的架构以及支持多种编程语言，可以在众多平台上进行部署，简化了真实场景应用学习难度。同时，其创建的大规模的人工智能应用场景也十分广泛，包括自然语言处理、图像识别、语音识别、智能机器人、知识图谱等场景，并取得了优异的成果。由于

Tensoflow 的每个计算流必须构建成图，没有符号循环，因此使一些计算变得困难。

2. Keras

Keras 由谷歌人工智能研究员 Francois Chollet 开发，并于 2015 年 3 月开源。Keras 是由 Python 编写而成的高级神经网络应用程序编程接口（API），以 TensorFlow、Theano 或 CNTK 作为后端运行，于 2017 年成为 TensorFlow 的高级别框架。Keras 支持多种编程语言，例如 Python、R 语言等，以及适用各个平台，例如 UNIX、Windows、Mac OS X 等。

Keras 的开发目的是支持快速实验，使用户在最短的时间内将想法变成实验结果。模块化的操作，使各个高级功能模块尽可能少地组装到一起，在用户需要时简单拼接就形成了一个新的模型，减少了用户阅读代码的时间，从而更加专注于实验，同时 Keras 又具备优秀的可扩展性，能够更加轻松地创建新的先进模块。

Keras 框架与 TensorFlow 一样使用数据流图进行数值计算，而且作为一个高级 API，为屏蔽作为后端的差异性而进行了层层封装，导致程序运行过于缓慢，而且用户很难学习到深度学习的内容。同时 Keras 运行会占用较大的 GPU 内存，容易导致 GPU 内存溢出（指应用系统中存在无法回收的内存或使用的内存过多，最终使得程序运行要用到的内存大于能提供的最大内存）。

3. Caffe

Caffe（Convolutional Architecture for Fast Feature Embedding）是一个清晰、高效的深度学习框架，由伯克利人工智能研究小组与伯克利视觉和学习中心开发。Caffe 最开始设计时的目标只针对图像，没有考虑文本、语音或时间序列的数据。

Caffe 一共有三个重要模块，分别为 Blob、Layer 和 Net。Blob 是 Caffe 中的数据操作基本单位，用来进行数据存储、数据交互和处理。Layer 是神经网络的核心，定义了许多层级结构，它将 Blob 视为输入输出。Net 是一系列 Layer 的集合，一个 Net 由多个 Layer 组合而成。

Caffe 的一大优势就是拥有大量训练好的经典模型，能够训练 state-of-the-art 的模型与大规模数据；其次 Caffe 底层是基于 C++ 的，可以在各种硬件环境编译并且具有良好的移植性。但是，与其他更新的深度学习框架相比，Caffe 确实不够灵活并且不适合非图像任务。

4. Torch

Torch 是一个高效的科学计算库，专注在 GPU 上计算的机器学习算法，2002 年发布了初版，在图像和视频领域应用广泛，由于 Facebook 开源了 Torch 的深度学习模块才被熟知。Torch 支持多种操作系统，例如 Windows、Linux 等，其目标是在保证使用方式简单的基础上，最大化算法的灵活性和速度，包含大量机器学习、深度学习、图像处理、语言处理等方面的库。Torch 是由 Lua 语言编写完成的。Lua 语言类似于 C 语言，支持类和面向对象，并且具有较高的运算效率，要使用 Torch 框架，必须先掌握 Lua 语言，这就限制了 Torch 的进一步发展。

5. PyTorch

PyTorch 广泛应用在图像处理领域，由 Facebook 人工智能研究院（FAIR）开发与维护，是 Facebook 在 Torch 基础上用 Python 重新开发的，于 2017 年 1 月发布。虽然其社区不如

TensorFlow 那么庞大，但依然确保了 PyTorch 的持续开发和更新，并且近几年开始变得非常流行。PyTorch 主要支持 Python、Java、C++ 等编程语言。

PyTorch 专注于快速原型设计和研究的灵活性，相比 Torch 框架，PyTorch 不仅仅是提供了一个新的 Python 接口，而是对张量上的全部模块进行重构，新增自动求导功能，同时又继承了 Torch 框架灵活、动态的编程环境和友好的开发界面。相比于 TensorFlow，PyTorch 采用动态图机制，具有更快的计算速度，并且简洁直观，底层代码也更容易看懂。

6. Theano

Theano 诞生于 2008 年，由蒙特利尔大学 Lisa Lab 团队开发并维护，是一个高性能的符号计算及深度学习库。Theano 核心是一个数学表达式的编译器，专门为处理大规模神经网络训练的计算而设计。Theano 是第一个有较大影响力的 Python 深度学习框架，并且是一个完全基于 Python 的符号计算库。Theano 派生出了很多基于它的深度学习库，包括一系列上层封装，例如 Keras、Lasagne 等。Theano 计算稳定性很好，可以精准地计算输出值很小的函数，但是在调试时输出的错误信息难以看懂，在 CPU 上的执行性能比较差。

7. CNTK

CNTK(Computational Network Toolkit)是微软研究院开源的深度学习框架，目前发展成一个通用的、跨平台的深度学习系统，在语音识别领域的使用尤其广泛。它把神经网络描述成一个有向图的结构来进行运算操作，叶子节点代表输入或者网络参数，其他节点代表各种矩阵运算。

CNTK 非常灵活，并且允许分布式训练。它与 Caffe 一样，也是基于 C++ 并且是跨平台的，大部分情况下，部署非常简单，支持 Linux、Mac OS X 和 Windows 系统，但目前不支持 ARM 架构，限制了在移动设备上的部署。

习题 3

一、简答题

1. 简述传统芯片与人工智能芯片的区别。
2. 智能芯片按技术架构和功能如何分类？
3. 简述智能芯片与应用程序的配对方式。
4. 简述智能芯片的特征。
5. 简述串行计算与并行计算的概念以及区别。

二、思考题

类脑芯片作为人工智能芯片未来发展的重要研究方向之一，你觉得在其今后的发展道路上会遇到哪些困难？研发人员能否研制出突破冯·诺依曼架构瓶颈的类脑芯片？