# 第 2 单元　机器学习基础知识

## 支撑的课程目标

1. 能够基于智能信息处理的基本理论和技术，识别和理解数据处理与分析等问题的相关特性.
2. 能够运用智能信息处理的相关原理和专业知识，设计实验方案，为解决数据处理与分析等问题提供支持.

## 基本要求

1. 能够详述监督学习与非监督学习的区别，理解两种模型的特点，根据处理数据的特性选择合适的机器学习方法.
2. 能够应用评估参数评价模型优劣，运用交叉验证和损失函数正则化来确定模型参数.

## 教学重点与难点

重点：　机器学习的关键组件；监督学习、无监督学习.

难点：　机器学习的关键组件.

## 教学过程设计

新课导入、知识讲授、教学目标达成考核、总结.

## 教学过程设计

本单元教学通过“互动、开放”的课堂形式，采用探究式学习、问题导入的教学方法，激发学生的学习兴趣，促成课程目标的达成.

## 教学学时

4 学时.

一、导入新课（10 分钟）

* 能否编写一个应用程序，接受地理信息、卫星图像和一些历史天气信息，并预测明天的天气；

* 能否编写一个应用程序，接受自然文本表示的问题，并正确回答该问题；

* 能否编写一个应用程序，接受一张图像，识别出该图像所包含的人；

* 能否编写一个应用程序，向用户推荐他们可能喜欢，但在自然浏览过程中不太可能遇到的产品。

在这些情况下，即使是顶级程序员也无法提出完美的解决方案。原因可能各不相同。有时候任务可能遵循一种随着时间推移而变化的模式，我们需要程序来自动调整。有时任务内的关系可能太复杂（比如像素和抽象类别之间的关系），需要数千或数百万次的计算。即使人类的眼睛能毫不费力地完成这些难以提出完美解决方案的任务，这其中的计算也超出了人类意识理解范畴。

机器学习（machine learning，ML）是一类强大的可以从经验中学习的技术。通常采用观测数据或与环境交互的形式，机器学习算法会积累更多的经验，其性能也会逐步提高。

相反，对于电子商务平台，如果它一直执行相同的业务逻辑，无论积累多少经验，都不会自动提高，除非开发人员认识到问题并更新软件。本单元将带大家开启机器学习之旅，并特别关注深度学习（deep learning，DL）的基础知识。深度学习是一套强大的技术，它可以推动计算机视觉、自然语言处理、医疗保健和基因组学等不同领域的创新。

## 二、新课讲授 (160 分钟)

本单元要点

* 日常生活中的机器学习

* 机器学习的关键组件

  – 数据

  – 模型

  – 目标函数

  - 优化算法

* 各种机器学习问题

  - 监督学习
  - 非监督学习

* 机器学习与深度学习

### 1. 日常生活中的机器学习

机器学习应用在日常生活中的方方面面。现在，假设有同学想给朋友张三打电话。这位同学拿起一部手机，对它说道："给张三打电话!"，手机的语音识别系统就自动触发语音转文字功能，并在通讯录中搜索张三的电话号码，启动拨号功能，拨通张三的电话号码.....

由此可见，机器学习渗透在生活中的方方面面，在短短几秒钟的时间里，人们与智能手机的日常互动就可以涉及几种机器学习模型。

现在，假如需要我们编写程序来响应一个"唤醒词"（比如"小爱同学"和"小度小度"），如图 1。我们试着用一台计算机和一个代码编辑器编写代码。问题看似很难解决：麦克风每秒钟将收集大约 44000 个样本，每个样本都是声波振幅的测量值。而该测量值与唤醒词难以直接关联。那又该如何编写程序，令其输入为麦克风采集到的原始音频片段，输出为 {是, 否}（表示该片段是否包含唤醒词），如何做到可靠预测呢？我们对编写这个程序毫无头绪，这就是需要机器学习的原因。

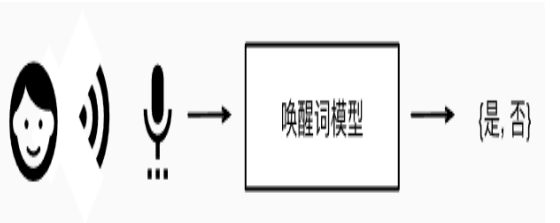


图 1: 识别唤醒词

通常，即使我们不知道怎样明确地告诉计算机如何从输入映射到输出，大脑仍然能够自己执行认知功能。换句话说，即使我们不知道如何编写计算机程序来识别“小度”这个词，大脑自己也能够识别它。有了这一能力，我们就可以收集一个包含大量音频样本的数据集（dataset），并对包含和不包含唤醒词的样本进行标记。利用机器学习算法，我们只需要定义一个灵活的程序算法，其输出由许多参数（parameter）决定，然后使用数据集来确定当下的“最佳参数集”，这些参数通过某种性能度量方式来达到完成任务的最佳性能。

那么到底什么是参数呢？参数可以被看作旋钮，旋钮的转动可以调整程序的行为。任一调整参数后的程序被称为模型（model）。通过操作参数而生成的所有不同程序（输入-输出映射）的集合称为“模型族”。使用数据集来选择参数的元程序被称为学习算法（learning algorithm）。

在开始用机器学习算法解决问题之前，我们必须精确地定义问题，确定输入（input）和输出（output）的性质，并选择合适的模型族。在本例中，模型接收一段音频作为输入，然后在是或否中选择一个作为输出。如果一切顺利，经过一番训练，模型对于“片段是否包含唤醒词”的预测通常是正确的。

但如果模型所有的按钮（模型参数）都被随机设置，就不太可能识别出“小度”或任何其他单词。在机器学习中，学习（learning）是一个训练模型的过程。通过这个过程，我们可以发现正确的参数集，从而使模型强制执行所需的行为。换句话说，我们用数据训练（train）模型。训练过程通常包含如下步骤：

(1) 从一个随机初始化参数的模型开始，这个模型基本没有“智能”；

(2) 获取一些数据样本（例如，音频片段以及对应的是或否标签）；

(3) 调整参数，使模型在这些样本中表现得更好；

(4) 重复第（2）步和第（3）步，直到模型在任务中的表现令人满意。

总而言之，我们没有编写唤醒词识别器，而是编写了一个“学习”程序。如果我们用一个巨大的带标签的数据集，它很可能可以“学习”识别唤醒词。这种“通过用数据集来确定程序行为”的方法可以被看作“用数据编程”（programming with data）。比如，我们可以通过向机器学习系统，提供许多猫和狗的图片来设计一个“猫图检测器”。检测器最终可以学会：如果输入是猫的图片就输出一个非常大的正数，如果输入是狗的图片就会输出一个非常小的负数。如果检测器

不确定输入的图片中是猫还是狗，它会输出接近于零的数..... 这个例子仅仅是机器学习常见应用的冰山一角，而深度学习是机器学习的一个主要分支，机器学习的基础知识也适应于深度学习。

### 2. 机器学习的关键组件

首先介绍一些核心组件。无论什么类型的机器学习问题，都会遇到这些组件：

（1）可以用来学习的数据（data）；

（2）如何转换数据的模型（model）；

（3）一个目标函数（objective function），用来量化模型的有效性；

（4）调整模型参数以优化目标函数的算法（algorithm）。

#### 2.1 数据

毋庸置疑，如果没有数据，那么数据科学毫无用武之地。每个数据集由一个个“样本”组成，大多时候，它们遵循独立同分布 (independently and identically distributed, i.i.d.)。样本有时也叫做数据点（data point）或者数据实例（data instance），通常每个样本由一组称为特征（features）的属性组成。机器学习模型会根据这些属性进行预测。在监督学习问题中，要预测的是一个特殊的属性，它被称为标签（label）。

当处理图像数据时，每一张单独的照片即为一个样本，它的特征由每个像素数值的有序列表来表示。比如，$200 \times 200$ 彩色照片由 $200 \times 200 \times 3 = 120000$ 个数值组成，其中的“3”对应于每个空间位置的红、绿、蓝通道的强度。再比如，对于一组医疗数据，给定一组标准的特征（如年龄、生命体征和诊断），此数据可以用来尝试预测患者是否会患某种疾病。

当每个样本的特征维数都是相同的时候，其特征向量是固定长度的，这个长度被称为数据的维数（dimensionality）。固定长度的特征向量是一个方便的属性，它可以用来量化学习大量样本。

然而，并不是所有的数据都可以用“固定长度”的向量表示。以图像数据为例，如果它们全部来自标准显微镜设备，那么“固定长度”是可取的；但是如果图像数据来自互联网，它们很难具有相同的分辨率或形状。这时，将图像裁剪成标准尺寸是一种方法，但这种办法很局限，有丢失信息的风险。此外，文本

数据更不符合“固定长度”的要求。比如，对于亚马逊等电子商务网站上的客户评论，有些文本数据很简短（比如“好极了”），有些则长篇大论。与传统机器学习方法相比，深度学习的一个主要优势是可以处理不同长度的数据。

一般来说，拥有越多数据的时候，工作就越容易。更多的数据可以被用来训练出更强大的模型，从而减少对预先设想假设的依赖。数据集的由小变大为现代深度学习的成功奠定基础。在没有大数据集的情况下，许多令人兴奋的深度学习模型黯然失色。就算一些深度学习模型在小数据集上能够工作，但其效能并不比传统方法高。

请注意，仅仅拥有海量的数据是不够的，我们还需要正确的数据。如果数据中充满了错误,或者如果数据的特征不能预测任务目标,那么模型很可能无效。有一句古语很好地反映了这个现象:“输入的是垃圾,输出的也是垃圾。”(“Garbage in, garbage out.”）此外，糟糕的预测性能甚至会加倍放大事态的严重性。在一些敏感应用中，如简历筛选和用于贷款的风险模型，我们必须特别警惕垃圾数据带来的后果。一种常见的问题来自不均衡的数据集，比如在一个有关医疗的训练数据集中，某些人群没有样本表示。想象一下，假设我们想要训练一个皮肤癌识别模型，但它（在训练数据集中）从未“见过”黑色皮肤的人群，这个模型就会顿时束手无策。

## 2.2 模型

大多数机器学习会涉及到数据的转换。比如一个“摄取照片并预测笑脸”的系统。再比如通过一组传感器读数预测读数的正常与异常程度。虽然简单的模型能够解决如上简单的问题，但本课程关注的问题超出了经典方法的极限。深度学习与经典方法的区别主要在于:前者关注的功能强大的模型,这些模型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)。在讨论深度模型的过程中，本课程中也将提及一些传统方法。

## 2.3 目标函数

前面的内容将机器学习解释为“从经验中学习”。这里所说的“学习”，是指自主提高模型完成某些任务的效能。但是，什么才算真正的提高呢？在机器学习中，我们需要定义模型的优劣程度的度量标准，这个度量在大多数情况是“可优化”的，这被称之为目标函数（objective function)。我们通常定义一个目

标函数，并希望优化它到最低值。因为越低越好，所以这些函数有时被称为损失函数（loss function，或 cost function）。但这只是一个惯例，我们也可以取一个新的函数，优化到它的最高点。这两个函数本质上是相同的，只是翻转一下符号。

在预测数值的任务中，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。有些目标函数（如平方误差）很容易被优化，有些目标（如错误率）由于不可微性或其他复杂性难以直接优化。在这些情况下，通常会优化替代目标。

通常，损失函数是根据模型参数定义的，并取决于数据集。在一个数据集上，我们可以通过最小化总损失来学习模型参数的最佳值。该数据集由一些为训练而收集的样本组成，称为训练数据集（training dataset，或称为训练集（training set））。然而，在训练数据上表现良好的模型，并不一定在“新数据集”上有同样的性能，这里的“新数据集”通常称为测试数据集（test dataset，或称为测试集（test set））。

综上所述，可用数据集通常可以分成两部分：训练数据集用于拟合模型参数，测试数据集用于评估拟合的模型。然后我们观察模型在这两部分数据集的性能。“一个模型在训练数据集上的性能”可以被想象成“一个学生在模拟考试中的分数”。这个分数用来为一些真正的期末考试做参考，即使成绩令人鼓舞，也不能保证期末考试成功。换言之，测试性能可能会显著偏离训练性能。当一个模型在训练集上表现良好，但不能推广到测试集时，这个模型被称为过拟合（overfitting）的。就像在现实生活中，尽管模拟考试考得很好，真正的考试不一定百发百中。

## 2.4 优化算法

当我们获得了一些数据源及其表示、一个模型和一个合适的损失函数，接下来就需要一种算法，它能够搜索出最佳参数，以最小化损失函数。深度学习中，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）。简而言之，在每个步骤中，梯度下降法都会检查每个参数，看看如果仅对该参数进行少量变动，训练集损失会朝哪个方向移动。然后，它在可以减少损失的方向上优化参数。

## 3. 各种机器学习问题

在机器学习的广泛应用中，问题回答、人脸识别、天气预报等例子，只是机器学习可以解决的众多问题中的一些例子。下面将列出一些常见的机器学习问题和应用，为之后学习做铺垫。接下来会经常引用前面提到的概念，如数据、模型和优化算法。

### 3.1 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）。我们的目标是生成一个模型，能够将任何输入特征映射到标签（即预测）。

举一个具体的例子：假设我们需要预测患者的心脏病是否会发作，那么观察结果“心脏病发作”或“心脏病没有发作”将是样本的标签。输入特征可能是生命体征，如心率、舒张压和收缩压等。

监督学习之所以能发挥作用，是因为在训练参数时，我们为模型提供了一个数据集，其中每个样本都有真实的标签。虽然监督学习只是几大类机器学习问题之一，但是在工业中，大部分机器学习的成功应用都使用了监督学习。这是因为在一定程度上，许多重要的任务可以清晰地描述为，在给定一组特定的可用数据的情况下，估计未知事物的概率。比如：

* 根据计算机断层扫描（Computed Tomography，CT）肿瘤图像，预测是否为癌症；

* 给出一个英语句子，预测正确的法语翻译；

* 根据本月的财务报告数据，预测下个月股票的价格；

监督学习的学习过程一般可以分为三大步骤：

(1) 从已知大量数据样本中随机选取一个子集，为每个样本获取真实标签。有时，这些样本已有标签（例如，患者是否在下一年内康复?）；有时，这些样本可能需要被人工标记（例如，图像分类）。这些输入和相应的标签一起构成了训练数据集；

(2) 选择有监督的学习算法，它将训练数据集作为输入，并输出一个“已完成学习的模型”；

(3) 将之前没有见过的样本特征放到这个“已完成学习的模型”中，使用模

型的输出作为相应标签的预测。

整个监督学习过程如图 2 所示。

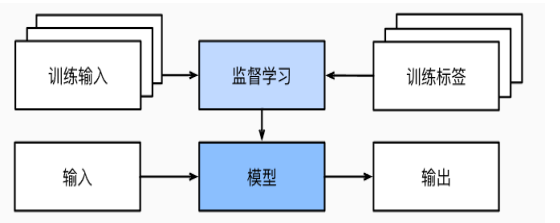


图 2: 监督学习

综上所述，即使使用简单的描述给定输入特征的预测标签，监督学习也可以采取多种形式的模型，并且需要大量不同的建模决策，这取决于输入和输出的类型、大小和数量。例如，我们使用不同的模型来处理“任意长度的序列”或“固定长度的序列”。

3.1.1 回归

回归（regression）是最简单的监督学习任务之一。假设有一组房屋销售数据表格，其中每行对应一个房子，每列对应一个相关的属性，例如房屋的面积、卧室的数量以及到大超市的步行距离，等等。每一行的属性构成了一个房子样本的特征向量。如果一个人住在森林半岛，那么他家的特征向量（房屋面积，卧室数量，步行距离）可能类似于：$[100, 3, 3]$。如果一个人住在蓝湾新城，这个特征向量可能更接近 $[120, 3, 2]$.....。当人们在市场上寻找新房子时，可能需要估计一栋房子的公平市场价值。为什么这个任务可以归类为回归问题呢？本质上是输出决定的。销售价格（即标签）是一个数值。当标签取任意数值时，我们称之为回归问题，此时的目标是生成一个模型，使它的预测非常接近实际标签值。

生活中的许多问题都可归类为回归问题。比如，预测用户对一部电影的评分可以被归类为一个回归问题。这里有一个小插曲：在 2009 年，如果有人设计了一个很棒的算法来预测电影评分，那可能会赢得 [100 万美元的奈飞奖]。

总而言之，判断回归问题的一个很好的经验法则是，任何有关“有多少”的问题很可能就是回归问题。比如：

* 这个手术需要多少小时；

* 在未来 6 小时，这个镇会有多少降雨量。

在这些情况下，我们将尝试学习最小化“预测值和实际标签值的差异”的模型，即平方误差损失函数的最小化。

3.1.2 分类

虽然回归模型可以很好地解决“有多少”的问题，但是很多问题并非如此。例如，一家银行希望在其移动应用程序中添加支票扫描功能。具体地说，这款应用程序能够自动理解从图像中看到的文本，并将手写字符映射到对应的已知字符之上。这种“哪一个”的问题叫做分类（classification）问题。分类问题希望模型能够预测样本属于哪个类别（category，正式称为类（class））。例如，手写数字可能有 10 类，标签被设置为数字 0～9。最简单的分类问题是只有两类，这被称之为二项分类（binomial classification）。例如，数据集可能由动物图像组成，标签可能是 { 猫, 狗 } 两类。回归是训练一个回归函数来输出一个数值；分类是训练一个分类器来输出预测的类别。

然而模型怎么判断得出这种“是”或“不是”的硬分类预测呢？我们可以试着用概率语言来理解模型。给定一个样本特征，模型为每个可能的类分配一个概率。比如，之前的猫狗分类例子中，分类器可能会输出图像是猫的概率为 0.9。0.9 这个数字表达什么意思呢？可以这样理解：分类器 90 预测类别的概率的大小传达了一种模型的不确定性，后续单元将讨论其他运用不确定性概念的算法。

当有两个以上的类别时，我们把这个问题称为多项分类（multiclass classification）问题。常见的例子包括手写字符识别 $\{0, 1, 2, ...9, a, b, c, ...\}$。与解决回归问题不同，分类问题的常见损失函数被称为交叉熵（cross-entropy）。

3.2 无监督学习

到目前为止，所有的例子都与监督学习有关，即需要向模型提供巨大数据集：每个样本包含特征和相应标签值。打趣一下，“监督学习”模型像一个打工仔，有一份极其专业的工作和一位极其平庸的老板。老板站在身后，准确地告诉模型在每种情况下应该做什么，直到模型学会从情况到行动的映射。取悦这位老板很容易，只需尽快识别出模式并模仿他们的行为即可。

相反，如果工作没有十分具体的目标，就需要“自发”地去学习了。比如，

老板可能会给我们一大堆数据，然后要求用它做一些数据科学研究，却没有对结果有要求。这类数据中不含有“目标”的机器学习问题通常被为无监督学习（unsupervised learning），那么无监督学习可以回答什么样的问题呢？来看看下面的例子。

* 聚类（clustering）问题：没有标签的情况下，我们是否能给数据分类呢？比如，给定一组照片，我们能把它们分成风景照片、狗、婴儿、猫和山峰的照片吗？同样，给定一组用户的网页浏览记录，我们能否将具有相似行为的用户聚类呢？

* 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？比如，一个球的运动轨迹可以用球的速度、直径和质量来描述。再比如，裁缝们已经开发出了一小部分参数，这些参数相当准确地描述了人体的形状，以适应衣服的需要。

* 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？例如，如果我们有关于房价、污染、犯罪、地理位置、教育和工资的人口统计数据，我们能否简单地根据经验数据发现它们之间的关系？

* 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。

### 4. 机器学习与深度学习

为了解决各种各样的机器学习问题，深度学习提供了强大的工具。虽然许多深度学习方法都是最近才有重大突破，但使用数据和神经网络编程的核心思想已经研究了几个世纪。

神经网络（neural networks）的得名源于生物灵感。一个多世纪以来（追溯到 1873 年亚历山大·贝恩和 1890 年詹姆斯·谢林顿的模型），研究人员一直试图组装类似于相互作用的神经元网络的计算电路。随着时间的推移，对生物学的解释变得不再肤浅，但这个名字仍然存在。其核心是当今大多数网络中都可以找到的几个关键原则：

（1）线性和非线性处理单元的交替，通常称为层（layers）；

（2）使用链式规则（也称为反向传播（backpropagation））一次性调整网络

中的全部参数。

经过最初的快速发展，神经网络的研究从 1995 年左右开始停滞不前，直到 2005 年才稍有起色。这主要是因为两个原因。首先，训练网络（在计算上）非常昂贵。在上个世纪末，随机存取存储器（RAM）非常强大，而计算能力却很弱。其次，数据集相对较小。事实上，费舍尔 1932 年的鸢尾花卉数据集是测试算法有效性的流行工具，而 MNIST 数据集的 60000 个手写数字的数据集被认为是巨大的。考虑到数据和计算的稀缺性，核方法（kernel method）、决策树（decision tree）和图模型（graph models）等强大的统计工具（在经验上）证明是更为优越的。与神经网络不同的是，这些算法不需要数周的训练，而且有很强的理论依据，可以提供可预测的结果。

大约 2010 年开始，那些在计算上看起来不可行的神经网络算法变得热门起来，实际上是以下两点导致的：其一，随着互联网的公司的出现，为数亿在线用户提供服务，大规模数据集变得触手可及；另外，廉价又高质量的传感器、廉价的数据存储（克莱德定律）以及廉价计算（摩尔定律）的普及，特别是 GPU 的普及，使大规模算力唾手可得。

数据集 vs 计算机内存和计算能力，如图 3 所示。

| 年代 | 数据规模 | 内存 | 每秒浮点运算 |
|---|---|---|---|
| 1970 | 100 （鸢尾花卉） | 1 KB | 100 KF (Intel 8080) |
| 1980 | 1 K （波士顿房价） | 100 KB | 1 MF (Intel 80186) |
| 1990 | 10 K （光学字符识别） | 10 MB | 10 MF (Intel 80486) |
| 2000 | 10 M （网页） | 100 MB | 1 GF (Intel Core) |
| 2010 | 10 G （广告） | 1 GB | 1 TF (Nvidia C2050) |
| 2020 | 1 T （社交网络） | 100 GB | 1 PF (Nvidia DGX-2) |

图 3: 数据集 vs 计算机内存和计算能力

很明显，随机存取存储器没有跟上数据增长的步伐。与此同时，算力的增长速度已经超过了现有数据的增长速度。这意味着统计模型需要提高内存效率（这通常是通过添加非线性来实现的），同时由于计算预算的增加，能够花费更多时间来优化这些参数。因此，机器学习和统计的关注点从（广义的）线性模型和核方法转移到了深度神经网络。这也造就了许多深度学习的中流砥柱，如多层感知机（McCulloch.Pitts.1943）、卷积神经网络（LeCun.Bottou.Bengio.ea.1998）、长短期记忆网络（Graves.Schmidhuber.2005），在相对休眠了相当长一段时间之后，在过去十年中被“重新发现”。

深度学习的一个关键优势是它不仅取代了传统学习管道末端的浅层模型，而且还取代了劳动密集型的特征工程过程。此外，通过取代大部分特定领域的预处理，深度学习消除了以前分隔计算机视觉、语音识别、自然语言处理、医学信息学和其他应用领域的许多界限，为解决各种问题提供了一套统一的工具。

## 三、教学目标考核（20 分钟）

讨论：

1. 机器学习的关键组件有哪些？机器是如何进行自我学习的？

2. 你当前正在编写的代码的哪些部分可以“学习”，即通过学习和自动确定代码中所做的设计选择来改进？你的代码是否包含启发式设计选择？

3. 你遇到的哪些问题有许多解决它们的样本，但没有具体的自动化方法？这些可能是使用深度学习的主要候选者。

4. 如果把人工智能的发展看作一场新的工业革命，那么算法和数据之间的关系是什么？它类似于蒸汽机和煤吗？根本区别是什么？

## 四、总结（10 分钟）

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。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。