第3章

函数与泛函分析

现有的人工智能方法通过模型优化从历史数据中发现观测对象与评定结果之间的对应关系，并将其用于评定新观测对象。也就是说，人工智能方法的本质就是尝试找到一个尽可能逼近真实对应关系 $\hat{f}$ 的对应关系 f，将每个观测对象 x，映射成一个评定结果 y，使其尽可能与真实评定 $\hat{y}$ 相似，即 $y \approx \hat{y}$。本章介绍人工智能模型设计、优化求解、评估验证过程中，涉及的函数与泛函相关知识。

3.1 集　　合

通常，数据驱动的人工智能方法从历史经验数据中发现规律，并将其用于对应场景中，实现智能分析。发现规律的过程统称为机器学习。当下，针对特定应用场景的数据收集更容易、规模更大。即便如此，为确保人工智能方法的性能，不能简单粗暴地直接将模型投放到实际应用中。一方面，这将给用户带来许多负面体验。另一方面，模型的调优变得更加烦琐，可操作性变差。不难理解，历史数据来源于实际应用，是应用场景中待解决问题对应数据的部分抽样。所以，只要数据量足够大，可假设收集到的数据的分布与该应用场景中待解决问题对应的数据分布完美契合。也就是说，抽样式的数据收集并未改变原始的数据分布特性，收集到的数据与待解决问题对应的数据分布相同。基于此，数据驱动的人工智能方法通常将已有历史数据划分为两部分，一部分用于模型训练、学习，另一部分用于模拟真实场景对模型的性能进行验证评估。需要说明的是，对某个特定问题来说，数据的划分方式可能不唯一。例如，交叉验证法对数据进行多次划分，并各自独立训练、验证，再由性能均值对设计的模型进行评估。实际上，以上描述中涉及的数量可数的历史数据构成一个集合。划分后的用于模型训练的数据构成该集合的子集，称作训练集；用于模型性能评估的另一部分数据也构成该历史数据集的子集，称作验证集。那么，从数学的角度来说，到底什么是集合？集合具备哪些特性？集合与集合之间具备什么样的关系呢？本节接下来将围绕以上问题介绍集合相关的数学知识及其在人工智能领域中的应用。

注

更多关于模型性能评估与验证的内容详见第 10 章。

3.1.1 定义与表示

人类可感知到的客观存在以及思维中包含的事物或抽象符号，都可称作对象。如上文所述，收集到的历史数据是可数的。那么，每一个数据可称作一个对象。需要指出的是，这里所说的对象其实是有应用场景的。对待解决的问题来说，收集到的每个数据可视作一个对象。但是，收集到的每个数据可能均由多个子项构成。那么，构成数据对象的各个子项，也可各自视作一个对象。其实，这些数据子项构成了一个整体。数学上，把能够确定的多个对象构成的整体，称作由这些对象构成的集合。通常地，集合采用大写拉丁字母表示，例如，所有 n 维实数向量构成的集合 R^n、训练数据集合 T、验证数据集合 V 等。与之对应地，集合中的每个对象，称作“元素”。元素通常用小写拉丁字母表示，如 n 维实数向量集元素 $\boldsymbol{x}$、训练集元素 t、验证集元素 v 等。元素与集合之间只有属于与不属于两种关系。若元素 a 属于集合 A，则记作 $a\in A$。否则，记作 $a\notin A$。需要指出的是，若不存在一个历史数据既属于训练集又属于验证集的情况，也就是说，训练集与验证集不存在数据重叠，那么训练集中的任意元素肯定不属于验证集。反之亦然。有必要指出的是，集合中元素的个数，称作集合的大小，是对集合的一种测度。给定集合 A，则其大小记作 $\mathrm{card}(A)$。不含任何元素的集合，称作“空集”。空集通常记作 Φ，即 $\mathrm{card}(A)=0$；至少含有一个元素的集合，称作“非空集”。含有有限可数个元素的集合，称作“有限集”；元素个数不可数的集合，则称作“无限集”。数据驱动的人工智能分析方法处理的数据对象一般构成有限集。

注

有必要说明的是，测度论是研究集合上的测度和积分的理论，不是本书的重点，感兴趣的读者可查阅相关资料。

不难发现，仅采用拉丁符号表示集合，不利于对集合构成特点的表达。构成集合的所有元素已知的情况下，可采用穷举所有元素，并将其用大括号括起来的方式表示集合。例如，验证集 V 由 a、b、c 三个数据对象构成，则集合 V 可表示为 $V=\{a,b,c\}$。需要说明的是，空集是一个特殊集合，其内部没有任何元素。采用元素穷举法时，空集可写成 $\Phi=\{\}$ 的形式。显然，若集合为无限集，或其元素个数较多，则不便用穷举法表示。此时，可采用元素公共属性描述法表示集合。也就是说，一个集合可写成如下形式：$A=\{$元素一般式 $x\,|$ 公共属性描述 $D\}$。例如，昨天收集的数据构成的集合 $A=\{x\,|\,x$ 是昨天采集的$\}$；正实数构成的集合 $B=\{x\,|\,x>0$ 并且 $x\in R\}$。

3.1.2 元素特性

由集合与元素的关系可知，在集合 A 已知的情况下，任给一个元素 a，该元素或者属

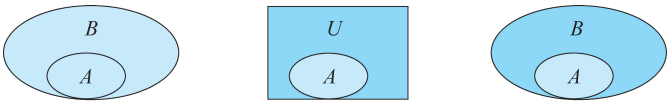
于 A 或者不属于 A，二者必取其一，不存在模棱两可的情况。这称作集合元素的确定性。也就是说，“皮肤白皙的美女”“个子很高的帅哥”均具有不确定性，不能构成集合。实际上，因为不确定性的信息很难用数据表达，所以收集到的历史数据通常是确定的。严格来说，构成集合的元素不仅具有确定性，相互之间还是不相同的。也就是说，集合中的元素具有互异性，它们在集合中最多只出现一次，是没有重复的。需要说明的是，由于历史数据是对真实待解决问题的抽样，一方面，取样过程中不可避免地引入离散化操作；另一方面，抽样结果的数据表达也不可避免地存在舍入误差。因此，历史经验数据构成的集合中可能存在不唯一的数据元素。集合元素的另外一个特性是，在集合内部元素地位等同，它们之间是没有先后顺序的。也就是说，集合$\{a,b,c\}$与集合$\{c,b,a\}$是同一个集合。

注

实际上，模糊集合论中集合元素的隶属关系不是绝对确定的，而是更加松弛。但是，作为一个全新的数学分支，模糊集合论不是本书重点关注内容。感兴趣的读者可自行查阅相关书籍或资料。

3.1.3　集合运算

给定两个集合 A 与 B，若集合 A 中的元素肯定存在集合 B 中，即对于 $\forall x\in A$，$x\in B$ 必然成立，则称 A 是 B 的子集，记作 $A\subseteq B$。若此时，集合 B 中存在集合 A 中没有的元素，即 $\exists x\in B$ 但 $x\notin A$，如图 3-1(a)所示，则称 A 是 B 的真子集，记作 $A\subset B$。例如，验证数据集与训练数据集均是历史数据集的真子集。需要指出的是，空集$\varnothing$是所有集合的真子集。若集合 A 与 B 互为子集，即 $A\subseteq B$ 与 $B\subseteq A$ 同时成立，则称 A 与 B 相等，记作 $A=B$。



(a) $A\subset B$　　(b) $\overline{A}$　　(c) $B-A$

图 3-1　子集与补集

一般地，收集到的历史数据构成的集合，在待解决问题求解过程中，称作全集。也就是说，全集是待研究对象的全体构成的集合。设全集为 U，集合 A 是 U 的一个子集，则 U 中所有不属于 A 的元素构成的集合，称作 U 中子集 A 的补集，或简称 A 的补集，记作 $\overline{A}$，如图 3-1(b)所示。显然，A 的补集的补集为集合 A 本身。上文提到的验证数据集与训练数据集互为补集。以上定义的补集有时又被称作绝对补集。这是因为，如图 3-1(c)所示，若 A 与 B 是两个集合，属于 B 但不属于 A 的元素构成的集合，称作 A 在 B 中的相对补集，记作 $B-A$。显然，若 $A=B$，则 $B-A=\varnothing$且 $A-B=\varnothing$。若记收集到的历史数据构成的集合为全集 U，则验证数据集 V 在 U 中的相对补集，其实就是它的绝对补集：训练数据集 T。

若集合 A 与 B 有共同元素，即 $\exists x \in A$ 且 $x \in B$，则其所有共同元素构成的集合，称作集合 A 与 B 的交集，记作 $A \cap B$，如图 3-2(a)所示。显然，上文提及的训练集与验证集的交集一般为空集$\varnothing$。与之对应地，将集合 A 与 B 中所有元素构成的集合，称作集合 A 与 B 的并集，记作 $A \cup B$，如图 3-2(b)所示。显然，上文提及的训练集与验证集的并集即是整个历史经验数据集。另外，若 $A \subseteq B$，则显然 $A \cap B = A$ 且 $A \cup B = B$。

(a) $A \cap B$　　(b) $A \cup B$

图 3-2　集合的交、并运算

有必要说明的是，集合运算是有规律可循的。具体地，集合的交、并运算满足交换律，也就是说，$A \cap B = B \cap A$ 且 $A \cup B = B \cup A$；集合的交、并复合运算满足结合律，也就是说，$(A \cap B) \cap C = A \cap (B \cap C)$且$(A \cup B) \cup C = A \cup (B \cup C)$；集合交、并混合运算满足分配律，也就是说，$(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$且$(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$；除此之外，集合交、并运算与求补集运算之间满足德摩根律，即$\overline{A \cup B} = \overline{A} \cap \overline{B}$ 且$\overline{A \cap B} = \overline{A} \cup \overline{B}$。

3.1.4　凸集分离定理

除上文介绍的集合运算定律之外，集合元素的分布特性对深入分析集合间的相互关系也有重要作用。给定一个由描述待观测对象特征的向量构成的集合 S，取 S 中任意两个元素 $\boldsymbol{x}$ 与 $\boldsymbol{y}$，将两元素在特征空间内连接在一起，若连线上的点对应的向量全部在集合 S 中，则称 S 是凸集。形式化地，设集合 $S \subset \mathbf{R}^n$，对于任意两个元素 $\boldsymbol{x} \in S$ 与 $\boldsymbol{y} \in S$，以及任意实数 λ，若与 n 维向量 $\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}$ 对应的向量元素 $\boldsymbol{z} \in S$ 恒成立，其中，$0 \leqslant \lambda \leqslant 1$，则称 S 为凸集。图 3-3 给出几个凸集与非凸集的示例。

(a) 凸集　　(b) 非凸集

图 3-3　集合凸性

有必要指出的是，n 维特征空间的任意超平面 H，将该特征空间划分为 H^+ 与 H^- 两部分。具体地，设超平面 H 的表达式为 $\boldsymbol{w}\boldsymbol{x}^{\mathrm{T}} = 0$，则半空间 H^+ 与 H^- 可分别由无限集 $H^+ = \{\boldsymbol{x} \mid \boldsymbol{w}\boldsymbol{x}^{\mathrm{T}} \geqslant 0\}$与 $H^- = \{\boldsymbol{x} \mid \boldsymbol{w}\boldsymbol{x}^{\mathrm{T}} \leqslant 0\}$表示。需要说明的是，超平面方程采用的是将截距项作为权重向量 $\boldsymbol{w}$ 的附加分量，将向量 $\boldsymbol{x}$ 延长一个常数分量维度的齐次表达形式。也就是说，$\boldsymbol{w}$ 与 $\boldsymbol{x}$ 均为 $n+1$ 维行向量。设 S_1 与 S_2 为 $\mathbf{R}^n$ 空间中两个不相交的非空凸集，则必然存在 $n+1$ 维向量 $\boldsymbol{w}$，使得 $S_1 \subseteq H^+$ 与 $S_2 \subseteq H^-$ 或者 $S_2 \subseteq H^+$ 与 $S_1 \subseteq H^-$ 同时成立，并且$[w_1, w_2, \cdots, w_n] \neq \boldsymbol{o}$。也就是说，对于 $\mathbf{R}^n$ 空间中两个不相交的非空凸集 S_1

与 S_2，存在一个超平面将它们分离开。换言之，$\mathbf{R}^n$ 空间中两个不相交的非空凸集 S_1 与 S_2 线性可分。

不难理解，若 $S_1 \subseteq H^+$ 与 $S_2 \subseteq H^-$，则

$$\begin{cases} \boldsymbol{w}\boldsymbol{s}_1^{\mathrm{T}} \geqslant 0, & \forall \boldsymbol{s}_1 \in S_1 \\ \boldsymbol{w}\boldsymbol{s}_2^{\mathrm{T}} \leqslant 0, & \forall \boldsymbol{s}_2 \in S_2 \end{cases} \tag{3-1}$$

设集合 S_1 与 S_2 之间的距离定义为

$$\operatorname{dist}(S_1, S_2) = \min_{\boldsymbol{s}_1 \in S_1, \boldsymbol{s}_2 \in S_2} \left\| \boldsymbol{s}_1 - \boldsymbol{s}_2 \right\|_2^2 \tag{3-2}$$

其中，$\left\| \boldsymbol{s}_1 - \boldsymbol{s}_2 \right\|_2^2$ 为 $\boldsymbol{s}_1$ 与 $\boldsymbol{s}_2$ 欧氏距离的平方。令

$$\boldsymbol{s}_1^*, \boldsymbol{s}_2^* = \arg\min_{\boldsymbol{s}_1 \in S_1, \boldsymbol{s}_2 \in S_2} \left\| \boldsymbol{s}_1 - \boldsymbol{s}_2 \right\|_2^2 \tag{3-3}$$

并记 $\boldsymbol{a} = \boldsymbol{s}_1^* - \boldsymbol{s}_2^*$、$b = -(\| \boldsymbol{s}_1^* \|_2^2 - \| \boldsymbol{s}_2^* \|_2^2)/2$，则对于 $\forall \boldsymbol{s}_1 \in S_1$，有

$$\boldsymbol{a}\boldsymbol{s}_1^{\mathrm{T}} + b \geqslant 0 \tag{3-4}$$

对于 $\forall \boldsymbol{s}_2 \in S_2$，有

$$\boldsymbol{a}\boldsymbol{s}_2^{\mathrm{T}} + b \leqslant 0 \tag{3-5}$$

由于非空凸集 S_1 与 S_2 不相交，即 $S_1 \cap S_2 = \varnothing$，所以，$\boldsymbol{s}_1^* \neq \boldsymbol{s}_2^*$，即 $\boldsymbol{a} \neq \boldsymbol{0}$。有必要指出的是，实际上 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}} + b = 0$ 是 $\boldsymbol{s}_1^*$ 与 $\boldsymbol{s}_2^*$ 连线的“中垂面”。这是因为，不难证明 $\boldsymbol{a} = \boldsymbol{s}_1^* - \boldsymbol{s}_2^*$ 是 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}} + b = 0$ 的法线。又 $(\boldsymbol{s}_1^* + \boldsymbol{s}_2^*)/2$ 是 $\boldsymbol{s}_1^*$ 与 $\boldsymbol{s}_2^*$ 连线的中点，令 $\boldsymbol{x} = (\boldsymbol{s}_1^* + \boldsymbol{s}_2^*)/2$ 代入 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}} + b$，得 $\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}} + b = 0$。得证。

为证式(3-4)，假设 $\exists \boldsymbol{s}_1 \in S_1$，使得 $\boldsymbol{a}\boldsymbol{s}_1^{\mathrm{T}} + b < 0$，即

$$\begin{aligned} \boldsymbol{a}\boldsymbol{s}_1^{\mathrm{T}} + b &= (\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)\boldsymbol{s}_1^{\mathrm{T}} - \frac{\left\| \boldsymbol{s}_1^* \right\|_2^2 - \left\| \boldsymbol{s}_2^* \right\|_2^2}{2} \\ &= (\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)\left(\boldsymbol{s}_1^{\mathrm{T}} - \frac{(\boldsymbol{s}_1^*)^{\mathrm{T}} + (\boldsymbol{s}_2^*)^{\mathrm{T}}}{2} \right) \\ &= (\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)\left((\boldsymbol{s}_1^{\mathrm{T}} - (\boldsymbol{s}_1^*)^{\mathrm{T}}) + \frac{(\boldsymbol{s}_1^*)^{\mathrm{T}} - (\boldsymbol{s}_2^*)^{\mathrm{T}}}{2} \right) \\ &= (\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)(\boldsymbol{s}_1^{\mathrm{T}} - (\boldsymbol{s}_1^*)^{\mathrm{T}}) + \frac{\left\| \boldsymbol{s}_1^* - \boldsymbol{s}_2^* \right\|_2^2}{2} \\ &< 0 \end{aligned} \tag{3-6}$$

由于 $(\| \boldsymbol{s}_1^* - \boldsymbol{s}_2^* \|_2^2)/2 \geqslant 0$，所以 $(\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)(\boldsymbol{s}_1 - \boldsymbol{s}_1^*)^{\mathrm{T}} < 0$。对于 $\boldsymbol{s}_1^*$ 与 $\boldsymbol{s}_1$ 连线上另外一点 $\boldsymbol{p}$，有 $\boldsymbol{p} = \lambda \boldsymbol{s}_1 + (1-\lambda)\boldsymbol{s}_1^*$。其中，$0 \leqslant \lambda \leqslant 1$。由于 S_1 是凸集，所以 $\boldsymbol{p} \in S_1$。此时，$\boldsymbol{p}$ 点与 $\boldsymbol{s}_2^*$ 的欧氏距离平方

$$\begin{aligned} \left\| \boldsymbol{p} - \boldsymbol{s}_2^* \right\|_2^2 &= \left\| \lambda \boldsymbol{s}_1 + (1-\lambda)\boldsymbol{s}_1^* - \boldsymbol{s}_2^* \right\|_2^2 \\ &= \left\| \boldsymbol{s}_1^* - \boldsymbol{s}_2^* + \lambda(\boldsymbol{s}_1 - \boldsymbol{s}_1^*) \right\|_2^2 \\ &= \left\| \boldsymbol{s}_1^* - \boldsymbol{s}_2^* \right\|_2^2 + \lambda\left(2(\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)(\boldsymbol{s}_1 - \boldsymbol{s}_1^*)^{\mathrm{T}} + \lambda \left\| (\boldsymbol{s}_1 - \boldsymbol{s}_1^*) \right\|_2^2 \right) \end{aligned} \tag{3-7}$$

显然，若 λ 取值为一个很小的正数，即

$$\lambda < -\frac{2(\boldsymbol{s}_1^* - \boldsymbol{s}_2^*)(\boldsymbol{s}_1 - \boldsymbol{s}_1^*)^{\mathrm{T}}}{\left\| (\boldsymbol{s}_1 - \boldsymbol{s}_1^*) \right\|_2^2} \tag{3-8}$$

时，一定有$\|\boldsymbol{p}-\boldsymbol{s}_2^*\|_2^2<\|\boldsymbol{s}_1^*-\boldsymbol{s}_2^*\|_2^2$。又因为，$\boldsymbol{p}\in S_1$，这与式(3-3)矛盾。也就是说，原假设$\exists \boldsymbol{s}_1\in S_1$，使得$\boldsymbol{a}\boldsymbol{s}_1^{\mathrm{T}}+b<0$不成立。式(3-4)得证。类似地，可证式(3-5)。

3.2 区　　间

对于许多数据驱动的人工智能方法来说，除收集描述历史对象特征的数据之外，通常还需要对收集的数据给予评定。例如，银行要求购房者向其提供收入证明g、征信报告h、银行流水r等数据信息，银行依据这些信息决定是否批复贷款。这里的“可批复贷款”或“不可批复贷款”即是对收集数据的评价。当然，此实例指的是贷款批复人工智能方法的应用场景。实际上，为了能够设计一个行之有效的人工智能批复贷款模型，需要对每个历史数据对象给出如上文所示的是否可批复贷款的评定，用于训练过程中校正模型学习误差。需要说明的是，以上实例中，对于贷款申请人偿还能力只有肯定与否定两种评定。或者认定申请人肯定能偿还，或者认定申请人不能偿还，存在一刀切问题。特别地，基于这种模式，银行并不能评估通过贷款申请给自己带来的金融风险，也不能评估若不通过贷款给自己带来的经济损失。相应地，若能对申请人的偿还能力进行等级评定，或者进一步细化，对其偿还能力进行0～100分值的打分，则一定程度上可缓解以上问题。为了使得银行对通过贷款带来的风险能量化评定，依据收集到的贷款申请人的收入证明g、征信报告h、银行流水r等数据信息，对其收入支持如期偿还贷款的可能性进行估计有重要意义。例如，若贷款申请人可如期偿还的可能性是100%，则银行批复贷款后无风险；若可能性是80%，则有一定风险；若可能性小于50%，则有较高风险。显然，此时数据对象评定取值范围为0%～100%内的任意实数。

3.2.1 定义与表示

数学上，将具有特定属性的实数集合，称作区间。这里的特定属性指的是，给定实数集合S，若实数x与y均是集合S中的元素，即$x\in S$且$y\in S$，则x与y之间的任意实数z均属于集合S，即$x\leqslant\forall z\leqslant y, z\in S$恒成立。那么，实数集合$S$，称作区间。特别地，集合$S$中的最小实数，称作该区间的下确界；集合$S$中的最大实数，称作该区间的上确界。如图3-4(a)所示，若记S区间的下确界为i，上确界为s，则区间S称作闭区间，可用如下符号表示：$S=[i\rightarrow s]$。若采用集合记法，则$S=\{x\mid i\leqslant x\leqslant s\}$。不难发现，区间是元素为实数的无限集，并且实数元素取值是连续不间断的。需要指出的是，许多教材采用逗号分隔上下界的方式标记区间。这与本书二维向量的表示方法相同。为以示区别，采用右向箭头分隔上下确界。有必要说明的是，如图3-4(b)所示，与集合$\{x\mid i<x\leqslant s\}$对应的区间，称作左开右闭区间，记作$(i\rightarrow s]$。此时，区间没有下确界，i为它的下界。类似地，如图3-4(c)所示，与集合$\{x\mid i\leqslant x<s\}$对应的区间，称作左闭右开区间，记作$[i\rightarrow s)$。此时，区间没有上确界，s为它的上界。进一步地，如图3-4(d)所示，与集合$\{x\mid i<x<s\}$对应的区间，称作开区间，记作$(i\rightarrow s)$。此时，区间没有下确界与上确界，i与s分别是它的下界与上界。一般地，用无穷符号∞表示区间在某方向上是无界的。例如，$(-\infty\rightarrow a]$、

$[b\rightarrow+\infty)$。特别地，$(0\rightarrow+\infty)$表示正实数集，记作 R^+。显然，上文所述的贷款申请人收入情况支持如期偿还贷款的可能性构成闭区间$[0\rightarrow1]$。除上述集合表示法之外，如图 3-4 所示，区间还可用数轴法来表示。区间内元素在数轴上围成的区域，称作区间内部。数轴的其他区域，称作区间外部。

(a) 闭区间　(b) 左开右闭　(c) 左闭右开　(d) 开区间

图 3-4　区间的开闭性

3.2.2　元素特性

如上文所述，区间是一类特殊的集合。与集合类似，给定任意一个区间，即便其内部实数个数无限多，但是值是确定的。这称作区间元素的确定性。换言之，给定任意一个实数，它或者落入指定区间内部，或者落入该区间外部。实际上，给定实数对 i 与 s，由其作为端点的区间内的元素是确定的。由于不存在两个不同实数具有相等的数值，所以，区间元素是互不相同的。这称作区间元素的互异性。实际上，数值相等的两个元素在区间内被视作实数轴上的同一点。除了元素确定与互异之外，又因为可比较实数大小，所以区间内确定且互异的元素是有大小关系的。例如，0.2 就比 0.3 更靠近闭区间$[0\rightarrow1]$的左端。这称作区间元素的有序性。由区间的数学定义不难发现，区间内实数元素之间是连续不间断的。也就是说，区间元素具有连续性。

3.2.3　区间算术

如前文所述，区间是一段连续实数构成的集合。若变量 x 为区间内任意元素，则 x 的取值具有不确定性。例如，向银行提交贷款申请后，银行最终给出的风险评定值具有不确定性。一般地，我们之前接触的算术中参与运算的变量取值均是确定的。例如，$3+5$、4×2 等。那么，对于取值具有不确定性的区间来说，是否也可以进行加、减、乘、除运算呢？区间算术就是指以区间为操作数的四则运算。普通四则运算中参与运算的操作数是具有确定值的整数、实数、向量等，其运算结果也是对应类型的值。类似地，区间算术中参与运算的操作数是区间，运算结果仍为一个区间。一般地，区间由两个界值唯一确定。那么，区间算术运算结果区间的界与参与运算的区间操作数的界之间具有什么样的函数关系呢？就区间加法来说，设参与运算的两个区间分别为 $S_1=[i_1\rightarrow s_1]$与 $S_2=[i_2\rightarrow s_2]$，对于 $\forall x\in S_1$、$\forall y\in S_1$，定义变量 $z=x+y$ 的所有可能的取值构成的区间 S 为区间 S_1 与 S_2 相加的结果，即 $S=S_1+S_2$。不难证明，若 $S_1=[i_1\rightarrow s_1]$、$S_2=[i_2\rightarrow s_2]$，则 $S_1+S_2=[i_1+i_2\rightarrow s_1+s_2]$。类似地，对于 $\forall x\in S_1$、$\forall y\in S_1$，定义变量 $z=x-y$ 的所有可能的取值构成的区间 S 为区间 S_1 与 S_2 相减的结果，即 $S=S_1-S_2$。不难证明，$S_1-S_2=[i_1-s_2\rightarrow s_1-i_2]$。类似地，$S_1\times S_2=[i,s]$，其中，$i=\min(i_1i_2,i_1s_2,s_1i_2,s_1s_2)$、$s=\max(i_1i_2,i_1s_2,s_1i_2,s_1s_2)$；$S_1/S_2=[i,s]$，其中，$i=\min(i_1/i_2,i_1/s_2,s_1/i_2,s_1/s_2)$、$s=\max(i_1/i_2,i_1/s_2,s_1/i_2,s_1/s_2)$。显然，跨 0 区间不应该作为区间除法的除数。有必要指出的是，区

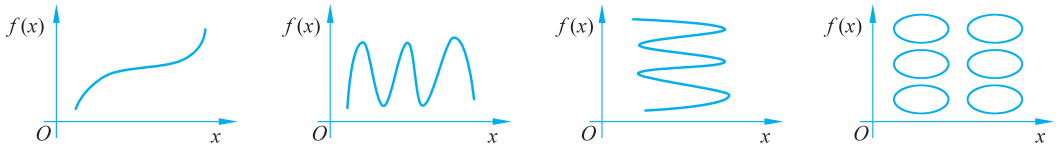
间加法和乘法符合交换律、结合律。

3.3 函数映射

收集数据的根本目的是从中发现规律，并将其用于评定新观测对象。显然，每个观测对象 x 与一个评定结果 y 相对应。这里的对应关系，记作 $\hat{f}$，即是人类智能对客观对象 x 的认识 $\hat{y}$。人类认知能力的来源、本质以及遵循的法则仍是个谜，人类更多地将其称为本能。与之对应地，人工智能就是要使得计算机具备与人类智能一样或类似的认知能力。也就是说，人工智能方法就是要找到一个尽可能逼近 $\hat{f}$ 的对应关系 f，将每个观测对象 x，映射成一个评定结果 y，使得 $y \approx \hat{y}$。不难理解，观测对象与评定结果分别构成两个集合，记作 A 与 B，其中，$x \in A$、$y \in B$。数学上，将以上对应关系 $\hat{f}$ 或 f 称作映射，用于指示观测对象集合 A 中任意元素与评定集合 B 中元素的对应关系，记作 $f: A \to B$。数学上，将这种映射关系称作函数，记作 $y = f(x)$ 或 $f(A) = \{y \mid y = f(x), x \in A\} = B$。

根据对应关系中观测对象集合元素个数与评定集合元素个数的不同，如图 3-5 所示，函数映射又分为一对一、多对一、一对多、多对多四种模式。顾名思义，一对一映射指的是，对于集合 A 中的任意元素 x，在集合 B 中都有一个唯一的 y 与之对应，反之亦然。对于人工智能方法待解决的实际应用问题来说，一对一映射并不常见。这是因为，若不同观测对象的评定结果不同，不同评定值对应的观测对象也不相同，则历史经验数据是随机分布的、无规律可循的。这与数据驱动的人工智能方法尝试从中发现规律，并用于解决实际问题相悖。但是在人工智能方法处理数据过程中，一对一映射很常见。例如，卷积神经网络中输入图像与输出图像分辨率相同时，前者像素与后者像素存在一一对应关系。与一对一映射不同，多对一映射在数据驱动的人工智能领域很常见。例如，对于对象类别识别任务来说，多个相似对象被归为同一类别，每个类别相当于对待识别对象的评定结果。也就是说，观测对象集合中有多个元素与评定集合中的一个元素相对应。为实现精准识别，人工智能方法通常从不同角度提取待观测对象的多个特征值，再对特征值进行智能分析。不难发现，观测对象与其描述特征之间的对应关系是典型的一对多映射。有必要说明的是，在基于卷积神经网络的图像识别任务中，多个卷积层的组合应用，本质是用于提取图像不同层次的特征。第一个卷积层的输入通常为一幅图像，其输出一般为多个特征映射。与之不同的是，经典神经元网络中隐层神经元间的对应关系是典型的多对多映射。另外，用于图像识别任务的卷积神经网络中间卷积层的输入与输出之间的对应关系也是多对多映射。

有必要指出的是，数学上，一般将一对一映射、多对一映射称作函数。给定任意函数 $y = f(x)$，即确定了一个以观测对象集合 A 为定义域、以评定集合 B 为值域的元素对应关系。反过来说，给定评定集合 B 中任意元素，也总能在观测对象集合 A 中找到与之对应的元素，这构成了另一个函数映射。数学上，若原函数 $y = f(x)$ 为一对一映射，则其反向映射构成一个新函数，称作 $y = f(x)$ 的反函数，记作 $y = f^{-1}(x)$。



(a) 一对一　(b) 多对一　(c) 一对多　(d) 多对多

图 3-5　函数映射

3.3.1　自变量与因变量

不难理解，客观事物是导致人类认知对外界刺激做出本能反应的根本。同类事物之间既有相似性，又有区别。人类智能就是时时刻刻在不知不觉中发现客观事物的区别与联系。类似地，观测对象的不同是导致人工智能方法给出不同评定结论或做出不同决策的根本原因。也就是说，观测对象 x 与其评定结果 y 的对应关系中，一般前者是客观存在，观测对象 x 的改变导致评定结果 y 的不同。因此，数学上将观测对象 x 称作映射关系 f 的自变量。对应地，观测对象集合 A 中的元素是自变量 x 在映射关系 f 中所有可能的取值。因此，将集合 A 称作映射 f 的定义域。有必要指出的是，函数映射的定义并未限定自变量的数据类型。也就是说，自变量可以是标量、向量，甚至矩阵。

如上文所述，观测对象 x 的改变导致评定结果 y 的不同。也就是说，评定结果 y 的取值因自变量 x 的取值不同而不同。因此，数学上将评定结果 y 称作映射关系 f 的因变量。因变量的每个取值，称作对应函数的函数值。对应地，将评定结果 y 所有可能的取值构成的集合 B 称作映射 f 的值域。有必要指出的是，函数映射的定义并未限定其因变量的数据类型，理论上来说，因变量可以是标量、向量，甚至矩阵。但是，常见的因变量评定结果一般是标量。

3.3.2　多元函数

数据驱动的人工智能方法很少直接操作收集到的待观测对象数据，而是从不同角度提取描述待观测对象特征的多个值，将特征值组成特征向量，再基于特征向量对待观测对象做出评定。不难发现，在这种情况下，人工智能方法就是要找到一个尽可能逼近真实情况的函数 f，用于将每个观测对象的特征向量 $\boldsymbol{x}=[x_1,x_2,\cdots,x_n]$ 与某个评定结果 y 之间建立映射关系，即使得 $y=f(\boldsymbol{x})$ 恒成立。不难理解，此时映射函数 f 的自变量不再是一个标量值，而是由多个值 $x_1,x_2,\cdots,x_n$ 组成。具有这种特性的函数，称作多元函数。为了体现其多元属性，有时也将 $y=f(\boldsymbol{x})$ 写作 $y=f(x_1,x_2,\cdots,x_n)$ 的形式。也有书籍将多元函数称作多自变量函数，其中，$x_1,x_2,\cdots,x_n$ 均是函数 $y=f(x_1,x_2,\cdots,x_n)$ 的自变量。需要指出的是，在卷积神经网络中，输出图像像素值通常由输入图像对应位置周围几个像素的权重均值决定。若将输入图像对应位置及其周围像素视作多个自变量，将作为输出图像像素值的权重均值视作因变量，则图像像素值与卷积结果之间构成一个多元函数映射。有必要说明的是，当卷积核尺寸为 1×1 时，卷积核仅由一个标量值构成，若考虑权重归一化问题，则此标量值为 1。此时输入图像与输出图像像素值之间的对应关系退

化为单元函数映射。有必要说明的是，多元函数并不要求函数的自变量均为同一数据类型的变量。也就是说，一个三元函数的自变量可以是标量、向量、矩阵的任意组合。例如，对于线性分类问题来说，人工智能方法通常是基于已知经验数据集优化模型函数 $\sum_{x,\hat{y}}(\hat{y}-\boldsymbol{w}\boldsymbol{x}^{\mathrm{T}}-b)^2$。不难理解，在观测对象集合中所有元素 $\boldsymbol{x}$ 及人类对其智能认知评定 $\hat{y}$ 已知的情况下，权重行向量 $\boldsymbol{w}$ 与截距标量 b 均是该线性分类优化函数的自变量。也就是说，该线性分类优化模型可记作 $f(w,b)=\sum_{x,\hat{y}}(\hat{y}-\boldsymbol{w}\boldsymbol{x}^{\mathrm{T}}-b)^2$。

注

关于卷积的更多内容详见第 6 章，关于优化的更多内容详见第 8 章。

3.3.3 复合函数

不难发现，上文涉及的函数表达式均比较简单，且其因变量即是最终评定结果。换言之，函数的因变量直接作为对待观测对象的评定，而不再作为其他函数的输入自变量。数学上，将这类函数称作简单函数。在人工智能领域，许多处理方法涉及更复杂的函数形式。它们通常由某些简单函数的输出作为其他简单函数的输入进而复合而成，称作复合函数。给定任意函数 $u=g(x)$ 与 $y=f(u)$，若 $u=g(x)$ 的值域与 $y=f(u)$ 的定义域相等，则称 $y=f(g(x))$ 为函数 $u=g(x)$ 与 $y=f(u)$ 的复合函数，简记作 $y=f(g(x))$。

注

①有必要提出的是，复合函数并不严格要求 $u=g(x)$ 的值域与 $y=f(u)$ 的定义域相等。实际上，只要后者与前者交集为非空集合，则可定义复合函数 $y=f(g(x))$。此时，复合函数的定义域为 $u=g(x)$ 的值域与 $y=f(u)$ 的定义域的交集。②有书籍将函数 $u=g(x)$ 与函数 $y=f(u)$ 的复合函数记作 $y=(f\circ g)(x)$。本书不采用这种记法的原因是第 1 章中已将$\circ$定义为分量乘法运算符。

具体地，复合函数一个直接的例子是多层全连接神经网络中多结点输入与多结点输出的对应关系。设输入层结点个数为 n_0，第 i 个结点的输入值为 x_i，记 $x=[x_1,x_2,\cdots,x_{n_0}]$。第一隐层结点个数为 n_1，记第一隐层中第 j 个结点的输入为 y_j，其与输入层第 i 个结点的连接权重记作 $w_{i,j}^1$，其中，$j=1,2,\cdots,n_1$。显然，$y_j=\boldsymbol{w}_{:,j}^1\boldsymbol{x}^{\mathrm{T}}-b_j^1=\sum_i^{n_0}w_{i,j}^1x_i-b_j^1$，其中，$b_j^1$ 为第一隐层第 j 个结点对应线性模型的截距。记 $\boldsymbol{y}=[y_1,y_2,\cdots,y_{n_1}]$、$\boldsymbol{W}^1=[(\boldsymbol{w}_{:,1}^1)^{\mathrm{T}},(\boldsymbol{w}_{:,2}^1)^{\mathrm{T}},\cdots,(\boldsymbol{w}_{:,n_1}^1)^{\mathrm{T}}]$、$\boldsymbol{b}^1=[b_1^1,b_2^1,\cdots,b_{n_1}^1]$，则输入层结点与第一隐层结点输入值之间构成函数映射关系，记作 $\boldsymbol{y}=f_1(x,\boldsymbol{W}^1,\boldsymbol{b}^1)$。若令第二隐层结点个数为 n_2，记第二隐层中第 k 个结点的输入为 z_k，其与第一隐层第 j 个结点的连接权重记作 $w_{j,k}^2$，其中，$k=1,2,\cdots,n_2$。不难证明，$z_k=\boldsymbol{w}_{:,k}^2\boldsymbol{y}^{\mathrm{T}}-b_k^2=\sum_j^{n_1}w_{j,k}^2y_j-b_k^2$，其中，$b_k^2$ 为第二隐层第 k 个结点对应线性模型的截距。记 $\boldsymbol{z}=[z_1,z_2,\cdots,z_{n_2}]$、$\boldsymbol{W}^2=[(\boldsymbol{w}_{:,1}^2)^{\mathrm{T}},(\boldsymbol{w}_{:,2}^2)^{\mathrm{T}},\cdots,$

$(\boldsymbol{w}_{:,n_2}^2)^{\mathrm{T}}]$、$\boldsymbol{b}^2=[b_1^2,b_2^2,\cdots,b_{n_2}^2]$，则第一隐层输出与第二隐层结点输入值之间构成函数映射关系，记作 $\boldsymbol{z}=f_2(\boldsymbol{y},\boldsymbol{W}^2,\boldsymbol{b}^2)$。考虑第一隐层中第 j 个结点的输入是由输入层所有结点的加权均值 $\sum_i^{n_0} w_{i,j}^1 x_i$ 平移 b_j^1 得到，第二隐层中的第 k 个结点的输入 z_k 可表示为输入层结点的函数，即 $z_k=\sum_j^{n_1} w_{j,k}^2\left(\sum_i^{n_0} w_{i,j}^1 x_i-b_j^1\right)-b_k^2$。实际上，由输入层与第一隐层之间、第一隐层与第二隐层之间的两个函数映射，可直接得到输入层与第二隐层之间的函数映射关系，即 $\boldsymbol{z}=f_2(f_1(\boldsymbol{x},\boldsymbol{W}^1,\boldsymbol{b}^1),\boldsymbol{W}^2,\boldsymbol{b}^2)$。随着神经网络深度的增加，最终输出与输入层之间的函数映射关系复合程度越高，表达式越复杂，表达能力越强。

注

需要说明的是，若给定函数的自变量与因变量都可表示成另外一组变量的函数，则给定函数是该组变量的复合函数。分别给出自变量与因变量相对于该组变量的函数表达式，将其称作给定函数的参数方程。

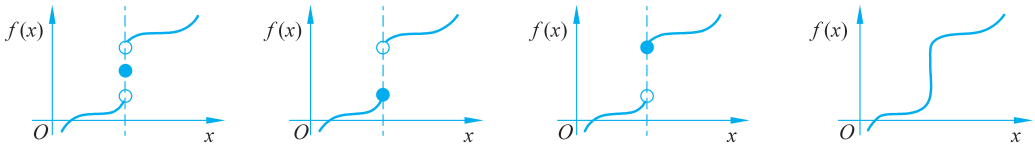
3.3.4　连续性、单调性、奇偶性

对于人工智能应用来说，以观测对象的特征向量为自变量，以对观测对象的评定作为因变量构造的函数映射通常具有很好的性质。n 维特征空间内与给定点 P 的欧氏距离不大于 δ 的所有点构成的集合，称作 P 点的 δ 邻域。独立元素点 P 构成的集合在 P 点的 δ 邻域中的相对补集，称作 P 点的 δ 去心邻域。给定定义在 n 维特征子空间 A 上的任意函数 f，对于定义域 A 内任意元素 $\boldsymbol{x}_0$，若 f 在 $\boldsymbol{x}_0$ 去心邻域内均有定义，且存在一个与 f 函数值数据类型相同的值 $\boldsymbol{c}$，对于任意给定的不论多小的正实数 ε，总存在正实数 δ，使得当 $\boldsymbol{x}$ 满足不等式 $\|\boldsymbol{x}-\boldsymbol{x}_0\|<\delta$ 时，对应的函数值 $f(\boldsymbol{x})$ 都满足不等式 $\|f(\boldsymbol{x})-\boldsymbol{c}\|<\varepsilon$，那么值 $\boldsymbol{c}$ 称作当 $\boldsymbol{x}\to\boldsymbol{x}_0$ 时函数 f 的极限，记作 $\lim\limits_{x\to x_0} f(\boldsymbol{x})=\boldsymbol{c}$。进一步地，若函数 f 在 $\boldsymbol{x}_0$ 处的函数值 $f(\boldsymbol{x}_0)=\boldsymbol{c}$，则称函数 f 在 $\boldsymbol{x}_0$ 处连续，将 $\boldsymbol{x}_0$ 称为函数 f 的连续点，否则称 $\boldsymbol{x}_0$ 为函数 f 的间断点。

与之对应地，给定函数 f 定义域内任意一点 $\boldsymbol{x}_0=[x_{0,1},x_{0,2},\cdots,x_{0,j},\cdots,x_{0,n}]$，对于指定维度分量 x_i 以及任意正实数 δ，若集合 $S=\{\boldsymbol{x}=[x_1,x_2,\cdots,x_i,\cdots,x_n]\mid\|\boldsymbol{x}-\boldsymbol{x}_0\|<\delta\}$ 是函数 f 定义域的子集，其中对于任意的 $j\neq i$ 有 $x_j=x_{0,j}$，则称集合 S 为 $\boldsymbol{x}_0$ 点沿维度 i 方向的 δ 邻域。集合 $\{\boldsymbol{x}_0\}$ 在 $\boldsymbol{x}_0$ 点沿维度 i 方向的 δ 邻域中的相对补集，称作 $\boldsymbol{x}_0$ 点沿维度 i 方向的 δ 去心邻域。对于定义域内任意元素 $\boldsymbol{x}_0$，若 f 在 $\boldsymbol{x}_0$ 点沿维度 i 方向的去心邻域内均有定义，且存在一个与 f 函数值数据类型相同的常数 $\boldsymbol{c}$，对于任意给定的不论多小的正实数 ε，总存在正实数 δ，使得当与元素 $\boldsymbol{x}_0$ 的 i 维分量不同其他分量相同的 $\boldsymbol{x}$ 满足不等式 $\|\boldsymbol{x}-\boldsymbol{x}_0\|<\delta$ 时，对应的函数值 $f(x)$ 均满足不等式 $\|f(x)-\boldsymbol{c}\|<\varepsilon$，那么常数值 $\boldsymbol{c}$ 称作当 $\boldsymbol{x}$ 沿维度 $\boldsymbol{i}$ 方向靠近 $\boldsymbol{x}_0$ 时函数 f 的极限，记作 $\lim\limits_{x\to x_{0,i}} f(\boldsymbol{x})=\boldsymbol{c}$。进一步地，若函数 f 在 $\boldsymbol{x}_0$ 处的函数值 $f(\boldsymbol{x}_0)=\boldsymbol{c}$，则称函数 f 在 $\boldsymbol{x}_0$ 处沿维度 i 方向连续，将 x_0 称为函数 f 的沿维度 i 方向的连续点。不难理解，当函数 f 在 $\boldsymbol{x}_0$ 处沿各维度方向的极限值均存

在，且与其在该处的函数值均相等时，称函数 f 在 $\boldsymbol{x}_0$ 处连续，将 $\boldsymbol{x}_0$ 称为函数 f 的连续点，否则称 $\boldsymbol{x}_0$ 为函数 f 的间断点。

若将上述自变量不等式描述中的欧氏距离替换为自变量任意元素 $\boldsymbol{x}$ 与 $\boldsymbol{x}_0$ 的差，考虑差值的符号，将不等式划分为左右两部分，分别对应 $\boldsymbol{x}_0$ 点的左邻域与右邻域，则函数 f 在 $\boldsymbol{x}_0$ 处沿维度 i 方向连续又可分为左连续与右连续，对应的函数极限，称作函数 f 沿维度 i 方向的左极限与右极限。不难理解，左右极限相等是函数 f 沿维度 i 方向连续的充要条件。为便于理解，图 3-6 给出一个自变量与因变量均为一维标量的函数连续性示例。



(a) 间断点　(b) 左连续　(c) 右连续　(d) 双侧连续

图 3-6　函数连续性

给定任意函数 $\boldsymbol{y}=f(\boldsymbol{x})$，若其自变量 $\boldsymbol{x}$ 与因变量 $\boldsymbol{y}$ 各自对应的数据之间均可比较大小，且因变量 $\boldsymbol{y}$ 随自变量 $\boldsymbol{x}$ 增大而增大，则称函数 f 为单调增函数；反之，若因变量 $\boldsymbol{y}$ 随自变量 $\boldsymbol{x}$ 增大而减小，则称函数 f 为单调减函数。单调增函数和单调减函数，统称为单调函数。以上描述可形式化为：设函数 $\boldsymbol{y}=f(\boldsymbol{x})$ 的定义域为 A，对任意元素 $\boldsymbol{x}_1\in A$ 与 $\boldsymbol{x}_2\in A$，满足不等式 $\boldsymbol{x}_1>\boldsymbol{x}_2$ 时，$f(\boldsymbol{x}_1)>f(\boldsymbol{x}_2)$ 恒成立，则称函数 f 为单调增函数；若任意元素 $\boldsymbol{x}_1\in A$ 与 $\boldsymbol{x}_2\in A$，满足不等式 $\boldsymbol{x}_1>\boldsymbol{x}_2$ 时，$f(\boldsymbol{x}_1)<f(\boldsymbol{x}_2)$ 恒成立，则称函数 f 为单调减函数。

注

若将上述定义中的不等式符号对应修改为大于或等于或者小于或等于符号，则分别称函数为增函数或者减函数。

若取反操作对其自变量 $\boldsymbol{x}$ 与因变量 $\boldsymbol{y}$ 各自对应的数据有效，则称自变量 $\boldsymbol{x}$ 取反时因变量 $\boldsymbol{y}$ 也取反的函数 f 为奇函数；反之，若自变量 $\boldsymbol{x}$ 取反时，因变量 $\boldsymbol{y}$ 取值不变，则称函数 $\boldsymbol{y}=f(\boldsymbol{x})$ 为偶函数。以上描述可形式化为：设函数 $\boldsymbol{y}=f(\boldsymbol{x})$ 的定义域为 A，对任意元素 $\boldsymbol{x}\in A$，若 $-\boldsymbol{x}\in A$ 且 $f(-\boldsymbol{x})=-f(\boldsymbol{x})$ 恒成立，则称函数 f 为奇函数；对任意元素 $\boldsymbol{x}\in A$，若 $-\boldsymbol{x}\in A$ 且 $f(-\boldsymbol{x})=f(\boldsymbol{x})$ 恒成立，则称函数 f 为偶函数。不难理解，奇函数值域空间关于原点对称，偶函数值域空间关于 $x=0$ 超平面对称。

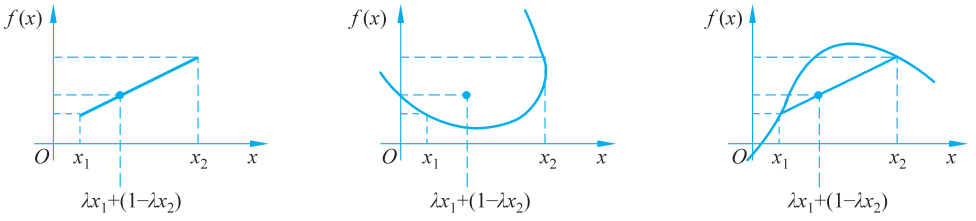
3.3.5　函数凸性与极值

目前，人工智能方法求解逼近真实映射函数的过程，通常由最优化一个目标函数来实现。更多内容详见第 8 章。有必要指出的是，不是所有的函数都适合用作目标函数，设计中需考虑目标函数的凸性与极值。给定定义在 n 维特征子空间 A 上的任意函数 f，对于定义域 A 内任意元素 $\boldsymbol{x}_0$，若存在正实数 δ 使得 f 在 $\boldsymbol{x}_0$ 的 δ 邻域内均有定义，且当 $\boldsymbol{x}$ 满足

不等式 $\|\boldsymbol{x}-\boldsymbol{x}_0\|<\delta$ 时，对应的函数值 $f(\boldsymbol{x})$ 均满足不等式 $f(\boldsymbol{x})<f(\boldsymbol{x}_0)$，则称 $\boldsymbol{x}_0$ 为函数 $f(\boldsymbol{x})$ 的局部极大值点，$f(\boldsymbol{x}_0)$ 为函数 $f(\boldsymbol{x})$ 的局部极大值。对应地，若 $\boldsymbol{x}$ 满足不等式 $\|\boldsymbol{x}-\boldsymbol{x}_0\|<\delta$ 时，对应的函数值 $f(\boldsymbol{x})$ 均满足不等式 $f(\boldsymbol{x})>f(\boldsymbol{x}_0)$，则称 $\boldsymbol{x}_0$ 为函数 $f(\boldsymbol{x})$ 的局部极小值点，$f(\boldsymbol{x}_0)$ 为函数 $f(\boldsymbol{x})$ 的局部极小值。需要指出的是，函数的局部极大值与局部极小值，均称作函数的局部极值；对应的极大值点与极小值点，统称为局部极值点。有必要说明的是，若函数 $f(\boldsymbol{x})$ 的值域是有界的，则其极大值与所有局部极大值中的最大值相等，其极小值与所有局部极小值中的最小值相等。

给定任意函数 f，若其定义域 A 是凸集，也就是说，对于定义域内任意元素 $\boldsymbol{x}_1\in A$ 与 $\boldsymbol{x}_2\in A$，以及任意实数 $0\leqslant\lambda\leqslant1$，$\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2\in A$ 恒成立。如图 3-7(b)所示，定义域 A 内元素 $\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2$ 的函数值与元素 $\boldsymbol{x}_1$ 与 $\boldsymbol{x}_2$ 的函数值之间若满足不等式 $f(\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2)\leqslant\lambda f(\boldsymbol{x}_1)+(1-\lambda)f(\boldsymbol{x}_2)$，则称函数 f 为凸集定义域 A 上的凸函数。若对于定义域内任意元素 $\boldsymbol{x}_1\in A$ 与 $\boldsymbol{x}_2\in A$，以及任意实数 $0\leqslant\lambda\leqslant1$，不等式 $f(\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2)<\lambda f(\boldsymbol{x}_1)+(1-\lambda)f(\boldsymbol{x}_2)$ 恒成立，则称 f 为凸集定义域 A 上的严格凸函数。与之对应地，如图 3-7(c)所示，若对于定义域内任意元素 $\boldsymbol{x}_1\in A$ 与 $\boldsymbol{x}_2\in A$，以及任意实数 $0\leqslant\lambda\leqslant1$，不等式 $f(\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2)\geqslant\lambda f(\boldsymbol{x}_1)+(1-\lambda)f(\boldsymbol{x}_2)$ 恒成立，则称 f 为凸集定义域 A 上的凹函数；若不等式 $f(\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2)>\lambda f(\boldsymbol{x}_1)+(1-\lambda)f(\boldsymbol{x}_2)$ 恒成立，则称 f 为凸集定义域 A 上的严格凹函数。不难发现，线性函数既是凸函数又是凹函数。这是因为，如图 3-7(a)所示，对于定义域内任意元素 $\boldsymbol{x}_1\in A$ 与 $\boldsymbol{x}_2\in A$，以及任意实数 $0\leqslant\lambda\leqslant1$，线性函数函数值之间满足等式 $f(\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2)=\lambda f(\boldsymbol{x}_1)+(1-\lambda)f(\boldsymbol{x}_2)$。

不难证明，若 $f(\boldsymbol{x})$ 为凸集定义域 A 上的凸/凹函数，则对于任意正实数 $\beta>0$，函数 $\beta f(\boldsymbol{x})$ 也是凸集定义域 A 上的凸/凹函数；若 $f_1(\boldsymbol{x})$ 与 $f_2(\boldsymbol{x})$ 均为凸集定义域 A 上的凸/凹函数，则函数 $f_1(\boldsymbol{x})+f_2(\boldsymbol{x})$ 也是凸集定义域 A 上的凸/凹函数；若 $f(\boldsymbol{x})$ 为凸集定义域 A 上的凸函数，则对于任意实数 β，集合 $A_\beta=\{\boldsymbol{x}\,|\,\boldsymbol{x}\in A\&f(\boldsymbol{x})\leqslant\beta\}$ 是凸集；与之对应地，若 $f(\boldsymbol{x})$ 为凹函数，则集合 $A_\beta=\{\boldsymbol{x}\,|\,\boldsymbol{x}\in A\&f(\boldsymbol{x})\geqslant\beta\}$ 是凸集；若 $f(\boldsymbol{x})$ 为凸集定义域 A 上的凸/凹函数，则 $f(\boldsymbol{x})$ 的任意一个局部极小值点/极大值点就是 $f(\boldsymbol{x})$ 的极小值点/极大值点，并且所有极值点构成的集合是凸集。



(a) 线性函数　　(b) 凸函数　　(c) 凹函数

图 3-7　函数凹凸性

注

有必要指出的是，可以证明，凸函数一定是连续函数。

3.3.6 激活函数

如前文所述，由输入层与第一隐层、第一隐层与第二隐层之间的函数映射，可得输入层与第二隐层之间的函数映射 $\boldsymbol{z}=f_2(f_1(\boldsymbol{x},\boldsymbol{W}^1,\boldsymbol{b}^1),\boldsymbol{W}^2,\boldsymbol{b}^2)$，其中，$z_k=\sum_j^{n_1} w_{j,k}^2\left(\sum_i^{n_0} w_{i,j}^1 x_i-b_j^1\right)-b_k^2$。不难发现，虽然随着神经网络层数增加，最终输出与初始输入变量 $\boldsymbol{x}$ 之间的函数映射关系复合程度变得更高，表达式更加复杂，但是其函数值仍然是输入自变量 $\boldsymbol{x}$ 的线性组合。也就是说，无论神经网络层数有多少，最终输出都是输入的线性组合，只是权重的表达更加复杂而已，这是最原始的感知机模型。由于权重值的个数与自变量 $\boldsymbol{x}$ 的取值一一对应，所以这样的神经网络相当于对线性模型的权重进行了细化。而其逼近真实函数的能力与没有隐藏层的神经网络本质上没有区别。有必要引入非线性函数对输出结果进行评定，从而增强多层神经网络的表达能力，使其不再只是输入变量的线性组合，几乎可以逼近任意函数。引入的对隐层输出结果进行非线性评定的函数，称作激活函数。这是因为，该类函数通常对大于一定阈值的隐层输出结果给出增强响应，而对于小于该阈值的输出结果进行抑制。

注

基于阈值实现函数输出影响值的控制是以人类智能为基础的。例如，人类视觉只能识别一定能量范围内的光，人类听觉只能辨别一定分贝的声音。

Sigmoid 函数是人工智能领域最早用于对隐层输出结果进行非线性评定的函数，也是最常用的非线性激活函数之一。对于任意的实数标量 x，Sigmoid 函数定义为

$$f_{\text{sig}}(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{3-9}$$

如图 3-8(a)所示，Sigmoid 函数实际为一个阶跃函数的平滑近似，将连续实型输入变换为 0 和 1 之间的实数输出。显然，其定义域为$(-\infty\rightarrow+\infty)$，值域为$(0\rightarrow+1)$。不难发现，当 x 远大于 0，即 $x>>0$ 时，$f_{\text{sig}}(x)\rightarrow 1$；当 x 远小于 0，即 $x<<0$ 时，$f_{\text{sig}}(x)\rightarrow 0$。另外，Sigmoid 函数的因变量均值为 0.5，不是 0 中心化的。以上两个特点给 Sigmoid 激活函数在应用中解决实际问题时带来不少负面影响。

注

关于 Sigmoid 激活函数以上两个特点给数据驱动的人工智能方法求解带来的负面影响，3.4.2 节将给出详细解释。另外，有必要说明的是，领域内 Sigmoid 函数又被称作 Logistic 函数，是线性回归模型的基础。有必要强调的是，表达变量取值的属性时，本书采用→表示逼近；表达区间属性时，本书采用→实现区间与二维向量的区分。

与之对应地，tanh 激活函数是一个函数值 0 中心化的函数，解决了非 0 中心化激活函数给数据驱动的人工智能方法求解带来的负面影响。其表达式为

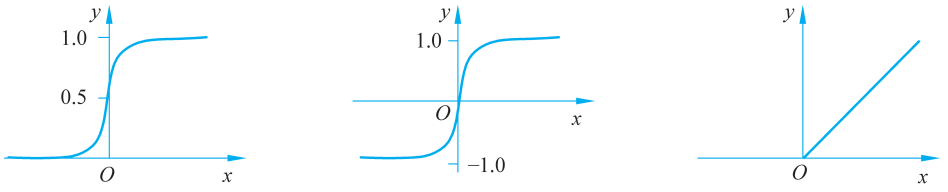
$$\tanh(x)=\frac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}} \tag{3-10}$$

如图 3-8(b)所示，tanh 函数也是一个阶跃函数的平滑近似。它将连续实型输入变换为−1～1 的实数输出。显然，其定义域为(−∞→+∞)，值域为(−1→+1)。不难发现，当 x 远大于 0，即 $x>>0$ 时，$\tanh(x)\to 1$；当 x 远小于 0，即 $x<<0$ 时，$\tanh(x)\to -1$。显然，tanh 函数为奇函数，其函数均值为 0。

Relu 激活函数解决了自变量绝对值大到一定程度时，因变量取值将基本保持不变的现象给数据驱动的人工智能方法求解带来的负面影响。其表达式为

$$\mathrm{Relu}(x)=\max(0,x) \tag{3-11}$$

如图 3-8(c)所示，Relu 函数将连续实型输入变换为 0～+∞的实数输出。显然，其定义域为(−∞→+∞)，值域为[0→+∞)。不难发现，当 $x>0$ 时，Relu 函数是一条与 x 轴夹角为 45°的直线；当 $x<0$ 时，Relu 函数是一条水平线，其函数值恒等于 0。$x=0$ 是 Relu 函数的连续点，且 Relu(0)=0。



(a) Sigmoid　(b) tanh　(c) Relu

图 3-8　激活函数

注

需要说明的是，以上只是激活函数家族的典型代表。读者需要了解，在人工智能领域还有许多其他类型的激活函数。感兴趣的读者可查阅相关资料。

3.4　导　　数

多数数据驱动的人工智能方法将待解决问题转换为目标函数的最优化问题。由前文函数的定义可知，任意函数的函数值均随自变量取值的变化而变化。而优化求解过程中往往需要考虑因变量随自变量的变化而变化的快慢程度。

3.4.1　函数可导与泰勒展开

数学上，上文提到的因变量随自变量的变化而变化的快慢程度，称作导数。形式化地，仅考虑自变量与因变量均为标量实数值的情况，给定任意函数 $y=f(x)$，设其在定义域内 x_0 点的某 $\delta>0$ 邻域内有定义，即当自变量 x 在 x_0 处有增量 $|\Delta x|<\delta$ 时，$x_0+\Delta x$ 也在定义域内，则函数 f 的因变量取得增量 $\Delta y=f(x_0+\Delta x)-f(x_0)$。若存在常数 c，对于任意给定的不论多小的正实数 ε，总存在正实数 δ，使得当 $|\Delta x|<\delta$ 成立时，不等式 $|\Delta y/\Delta x-c|<\varepsilon$ 恒成立。也就是说，Δx 趋向 0 时，因变量增量 Δy 与自变量增量 Δx 的比值极限存在，则称函数 f 在 x_0 处可导，并称此时的极限值 c 为函数 f 在 x_0 处的导数，

记作 $f'(x_0)$、$y'|x=x_0$ 或 $\mathrm{d}f(x)/\mathrm{d}x|x=x_0$、$\mathrm{d}y/\mathrm{d}x|x=x_0$。以上表述可形式化为

$$f'(x_0)=\lim_{\Delta x\to 0}\frac{\Delta y}{\Delta x}=\lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} \tag{3-12}$$

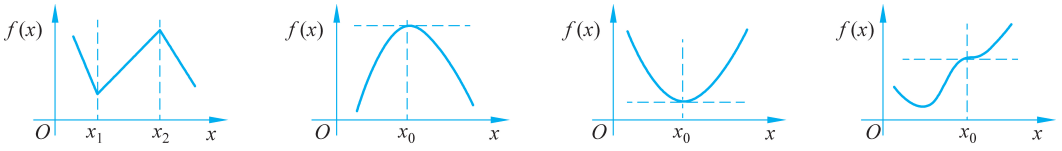
不难证明，f 在 x_0 处的导数为函数曲线在 x_0 处的切线的斜率。不难理解，若将自变量增量趋向 0 的方式分为由 0 的左侧趋向 0 与由 0 的右侧趋向 0，对应的增量比值极限仍然存在，则对应的极限称作函数 f 在 x_0 处的左导数与右导数，分别记作 $f'(x_0-0)$、$f'(x_0+0)$。有必要指出的是，函数 f 在 x_0 处可导的充要条件是其左右导数都存在且相等。若函数 $y=f(x)$在开区间 S 内每一点都可导，则称函数 $y=f(x)$在区间 S 上可导。不难理解，此时对于任意元素 $x\in S$，都存在唯一导数值 $f'(x)$与之对应。显然，区间内实数元素 x 与导数值 $f'(x)$之间构成一个新的函数映射，称之为函数 $y=f(x)$的导函数，记作 $f'(x)$、y'或 $\mathrm{d}f(x)/\mathrm{d}x$、$\mathrm{d}y/\mathrm{d}x$。对应地，函数 $y=f(x)$在 x_0 处的导数，记作 $f'(x_0)$、$y'|x_0$ 或 $\mathrm{d}f(x_0)/\mathrm{d}x$、$\mathrm{d}y/\mathrm{d}x|x_0$。不难发现，函数导数的定义与函数连续性的定义有许多相似之处。实际上，函数连续是其可导的必要非充分条件。例如，Relu 函数在 $x\neq 0$ 处既连续又可导，但在 $x=0$ 处是连续不可导的。这是因为，如图 3-11(c)所示，Relu 函数在 $x>0$ 区间内导数为 1，在 $x<0$ 区间内导数为 0，而在 $x=0$ 处 Relu 函数的左导数与右导数不相等，分别为 0 与 1。

注

有必要指出的是，若函数 $y=f(x)$在定义域内 x_0 处，左右导数 $f'(x_0-0)$与 $f'(x_0+0)$均存在，定义 $a=\min(f'(x_0-0),f'(x_0+0))$、$b=\max(f'(x_0-0),f'(x_0+0))$，则集合$[a\to b]$内任意元素定义为函数 $y=f(x)$在 x_0 处的次导数。显然，若函数$y=f(x)$在 x_0 处左右导数相等，则集合$[a\to b]$内只有一个元素。所以，函数 $y=f(x)$在 x_0 处可导是其在该处存在次导数的充分非必要条件。次导数的关键作用在于，可对不十分光滑的函数给出取极值条件。

由函数单调性及其导函数的定义不难发现，函数的单调性与导数的符号强相关。具体地，若函数 $y=f(x)$为单调增函数，则 $f'(x)>0$；若函数 $y=f(x)$为单调减函数，则 $f'(x)<0$；反过来，若 $f'(x)>0$，则存在 x 的邻域，函数 $y=f(x)$在此邻域内单调增；若 $f'(x)<0$，则存在 x 的邻域，函数 $y=f(x)$在此邻域内单调减。进一步地，若函数由增变减，则其导数必由正变负。考虑导函数的连续性，在单调增区间与单调减区间的邻接处必然存在一点 x_0，使得 $f'(x_0)=0$，则此点称作函数 $f(x)$的驻点。也就是说，若函数在局部极大值处可导，如图 3-9(b)所示，则其导数为 0；若函数由减变增，情况类似：函数在局部极小值处可导，如图 3-9(c)所示，则其导数为 0。有必要说明的是，如图 3-9(a)所示，函数极值点不是函数在此处导数为 0 的充分条件。另一方面，由定义域内 x_0 处导数为 0，不能得出 x_0 处为极值点的结论。也就是说，函数极值点不是函数在此处导数为 0 的必要条件。图 3-9(d)给出一个函数驻点非极值点的例子。但是，若定义域内 x_0 处导数为 0，且在 x_0 的左邻域与右邻域内导数变号，则 x_0 是极值点。

以上定义的函数 f 的导函数，称作 f 的一阶导函数。若函数 f 的导函数在同一区间上仍然可导，则该区间函数 f 的导函数值与导函数的导数值之间也构成新的函数映射。

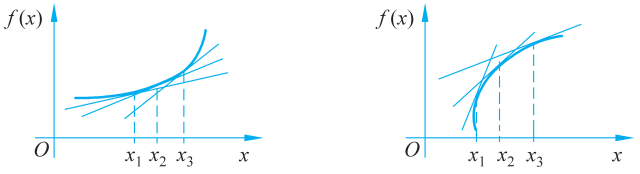


(a) 极值处导数非0　(b) 极大值处导数为0　(c) 极小值处导数为0　(d) 非极值导数为0

图 3-9　导数与极值的关系

此时，称导函数的导函数为原函数 f 的二阶导函数。类似地，可定义函数 f 的任意高阶导函数。数学上，将函数 f 的 n 阶导函数记作 $f^{(n)}$。为了一致性，有时也将原函数 f 写成 $f^{(0)}$。将前文定义的一阶导函数 f' 写成 $f^{(1)}$。

有必要说明的是，除上文提到的导数在目标函数最优化中的作用之外，其还可用于评定函数的凹凸性。如图 3-10 所示，凸函数因变量变化率随自变量增大而增大，凹函数因变量变化率随自变量增大而减小。因此，设函数 $y=f(x)$ 在 x_0 处存在二阶导数，若 f 为凸函数，则其二阶导函数 $f^{(2)}$ 在 x_0 处的取值大于 0；若 f 为凹函数，则其二阶导函数 $f^{(2)}$ 在 x_0 处的取值小于 0。



(a) 凸函数变化率递增　(b) 凹函数变化率递减

图 3-10　函数导数与凹凸性关系

注

有必要指出的是，关于优化问题以及导数在优化中的作用详见第 8 章。

如上文所述，函数的一阶导数可用于最优化问题中求目标函数的极值，二阶导数可用于判别函数的凹凸性。除此之外，其实函数的所有阶导数对于函数值的估计都具有指导性作用。具体地，若函数 $f(x)$ 在包含 x_0 的某闭区间 $[a\to b]$ 上具有任意阶导数，且在开区间 $(a\to b)$ 上具有高一阶导数，则对闭区间 $[a\to b]$ 内任意一点 x 有

$$f(x)=\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+\cdots \tag{3-13}$$

需要指出的是，式(3-13)称作函数 $f(x)$ 在 x_0 处的泰勒展开式。数学上，将若干项的和称作级数。显然，式(3-13)等号右端是一个无穷项级数。若函数 $f(x)$ 的高阶导数求解困难，甚至根本不存在，抑或是对函数值的评估不要求过高精度，则式(3-13)可改写为：

$$f(x)=\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n(x) \tag{3-14}$$

其中，$R_n(x)$称作泰勒余项，是$(x-x_0)^n$的高阶无穷小，即$\lim\limits_{x-x_0\to 0}(R_n(x)/(x-x_0)^n)=0$记作$o(x-x_0)^n$。不难发现，若定义函数$f(x)$为上式等号右侧前$n$项的和，则泰勒余项$R_n(x)$即为估计误差。可以证明，此误差与函数$f(x)$的$n+1$阶导数有关，即

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1} \tag{3-15}$$

其中，ξ介于x_0与x之间。

注

与一元函数类似，多元函数也有类似展开式。但形式更为复杂，感兴趣的读者可在3.4.3～3.4.5节找到一些蛛丝马迹，更多内容可查阅相关资料。另外，有必要说明的是，式(3-15)只是众多余项表达式中的一种。不同余项虽然表达式不同，但其相互间存在联系。感兴趣的读者可查阅相关资料。

3.4.2 求导法则

根据函数导数的定义不难证明，若函数$u(x)$与$v(x)$在x点处均可导，则它们的和、差、积、商在此处也可导，并且

$$\begin{aligned}(u(x)\pm v(x))'&=u'(x)\pm v'(x)\\(u(x)v(x))'&=u'(x)v(x)+u(x)v'(x)\\\left(\frac{u(x)}{v(x)}\right)'&=\frac{u'(x)v(x)-u(x)v'(x)}{v^2(x)}\end{aligned} \tag{3-16}$$

有必要说明的是，商的求导法需要确保分母不为0。

不难证明，给定任意函数$x=f(y)$，若其在定义域子集区间I_y上可导，导数$f'(y)$均不为0，且存在以其值域区间I_x为定义域的反函数$y=f^{-1}(x)$，则$y=f(x)$在区间I_x上也可导，并且

$$(f^{-1}(x))'=\frac{1}{f'(y)} \tag{3-17}$$

以上结论，称作反函数求导法则。

可以证明，若函数$u=\varphi(x)$在x点处可导，而函数$y=f(u)$在$u=\varphi(x)$处可导，则复合函数$y=f(\varphi(x))$在x点可导，并且

$$(f(\varphi))'(x)=f'(u)\varphi'(x) \tag{3-18}$$

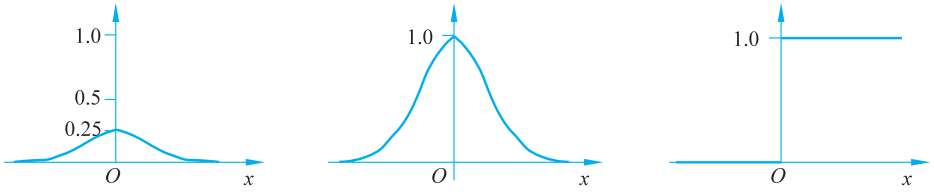
以上结论可推广到任意有限个函数复合的情形。此时，复合函数的导数等于有限个函数在对应点相对各自自变量导数的乘积。不难发现，复合函数的求导就像锁链一样一环套一环，故称作函数求导的链式法则。由前文关于Sigmoid激活函数的定义不难发现，Sigmoid函数可视作以下函数的复合：$r=f_1(x)=-x$，$u=f_2(r)=\mathrm{e}^r$，$v=f_3(u)=1+u$，$y=f_4(v)=1/v$。由链式法则可得，$f_{\mathrm{sig}}(x)=f_1(f_2(f_3(f_4(x))))$且

$$\begin{aligned}f'_{\mathrm{sig}}(x)&=f'_4(v)f'_3(u)f'_2(r)f'_1(x)\\&=\frac{-1}{v^2}\cdot 1\cdot \mathrm{e}^r\cdot(-1)\end{aligned}$$

$$=\frac{e^{-x}}{(1+e^{-x})^2}$$

$$=\frac{1}{1+e^{-x}}\left(1-\frac{1}{1+e^{-x}}\right) \tag{3-19}$$

不难发现，$f'_{sig}(x)=f_{sig}(x)(1-f_{sig}(x))$，其形状曲线如图 3-11(a)所示。类似地，可得 $\tanh(x)$ 函数的导函数等于 $1-\tanh^2(x)$，其形状曲线如图 3-11(b)所示。显然，激活函数 Sigmoid 与 tanh 的导函数均是其原函数的复合函数。这一性质也是 Sigmoid 与 tanh 函数常用作非线性激活函数的重要原因之一。但是，由 $f'_{sig}(x)$ 导函数曲线不难发现，其函数值均小于 1，最大值为 0.25。为提升神经网络逼近任意真实函数的能力，通常多层网络的每层输出均由激活函数进行复合。由复合函数求导法则不难发现，作为复合函数组成部分的激活函数，在求导过程中与一个乘数因子对应，小于 1 的导数值，使得复合函数的导数变小。复合层数越多，层数变小越明显。复合层数达到一定程度时，导数甚至接近消失。另一方面，考虑单实变量 x 的线性函数 $y=wx+b$。由函数求导法则，可得 $dy/dw=x$。由图 3-8(a)可知，Sigmoid 的函数值全大于 0，是非 0 中心化的。也就是说，若实变量 x 为上层 Sigmoid 激活函数的输出，则当前层反向求导时 $dy/dw>0$ 恒成立。显然，若模型训练过程中，采用 $\eta(dy/dw)$ 的步长更新权重因子 w 的值，则权重值 w 一直增长，其中，正实数 η 为学习率。以上两点使得 Sigmoid 激活函数给数据驱动的人工智能模型的优化求解带来困难。由图 3-8(b)可得，tanh 的输出是 0 中心化的。也就是说，若实变量 x 为激活函数 tanh 的输出，则 dy/dw 有正有负，更有利于权重值 w 的更新。但是由于 $0<\tanh'(x)\leqslant 1$，所以复合函数导数变小，甚至接近消失的问题仍未得到解决。不难理解，由于 Relu 函数的取值范围为 $[0\rightarrow+\infty)$，所以复合函数导数变小，甚至接近消失的问题得到很好的解决。但是，除因其原函数非 0 中心化的特点使得目标函数收敛速度变慢，甚至不能收敛之外，如图 3-11(c)所示，当 $x<0$ 时，Relu 函数的导数为 0，对应神经元将无法被激活。



(a) $f'_{sig}(x)$　(b) $\tanh'(x)$　(c) $\mathrm{Relu}'(x)$

图 3-11　激活函数的导函数

3.4.3　偏导数与雅可比矩阵

上文定义函数的导数时，仅考虑自变量与因变量均为标量实数值的情况。也就是说，可导函数是单变量函数。若一个函数是多元函数，例如，其自变量取值为描述待观测对象特征的向量，其因变量随各个自变量分量的变化而变化的快慢程度，在数学上称作偏导数。形式化地，给定任意多元函数 $y=f(\boldsymbol{x})$，其中，$\boldsymbol{x}=[x_1,x_2,\cdots,x_n]$，设其在 n 维定义

域空间内 $\boldsymbol{x}_0=[x_{0,1},x_{0,2},\cdots,x_{0,n}]$处沿 i 维方向的某 δ 邻域内有定义，则当自变量的第 i 维分量 $x_{0,i}$有增量 Δx_i 时，$[x_{0,1},x_{0,2},\cdots,x_{0,i}+\Delta x_i,\cdots,x_{0,n}]$也在定义域内，其中，$|\Delta x_i|<\delta$。此时，函数 $y=f(\boldsymbol{x})$的因变量取得增量为 $\Delta y=f([x_{0,1},x_{0,2},\cdots,x_{0,i}+\Delta x_i,\cdots,x_{0,n}])-f(\boldsymbol{x}_0)$。若存在常数 c，对于任意给定的不论多小的正实数 ε，总存在正实数 δ，使得当 $|\Delta x_i|<\delta$ 成立时，不等式 $|\Delta y/\Delta x_i-c|<\varepsilon$ 恒成立。也就是说，Δx_i 趋向 0 时，因变量增量 Δy 与自变量增量比值 Δx_i 的极限存在，则称函数 f 在 $x_0=[x_{0,1},x_{0,2},\cdots,x_{0,n}]$处对分量 x_i 可导，并称此时的极限值 c 为函数 f 在 $x_0=[x_{0,1},x_{0,2},\cdots,x_{0,n}]$处对 x_i 的偏导数，记作 $f'_{x_i}(x_0)$、$y'_{x_i}|x=x_0$ 或$\partial f(x_0)/\partial x_i$、$\partial y/\partial x_i|x=x_0$。以上表述可形式化为

$$f'_{x_i}(\boldsymbol{x}_0)=\lim_{\Delta x_i\to 0}\frac{\Delta y}{\Delta x_i}=\lim_{\Delta x_i\to 0}\frac{f([x_{0,1},x_{0,2},\cdots,x_{0,i}+\Delta x_i,\cdots,x_{0,n}])-f(\boldsymbol{x}_0)}{\Delta x_i} \tag{3-20}$$

不难证明，f 在 x_0 处对 x_i 的偏导数为函数曲线在 x_0 处的沿 x_i 方向切线的斜率。不难理解，若将自变量增量趋向 0 的方式分为由 0 的左侧趋向 0 与由 0 的右侧趋向 0，对应的增量比值极限仍然存在，则对应极限分别称作函数 f 在 x_0 处对 x_i 的左偏导数与右偏导数分别记作 $f'_{x_i}(\boldsymbol{x}_0-0)$、$f'_{x_i}(\boldsymbol{x}_0+0)$。有必要指出的是，函数 f 在 $\boldsymbol{x}_0$ 处对 x_i 可导的充分条件是其左右导数都存在且相等。若函数 $y=f(\boldsymbol{x})$在定义域内任意点处对 x_i 都可导，则称函数 $y=f(\boldsymbol{x})$在区间上对 x_i 可导。不难理解，此时对于任意元素 $\boldsymbol{x}\in S$，都存在唯一的偏导数值 $f'_{x_i}(\boldsymbol{x})$与之对应。显然，定义域内任意元素 $\boldsymbol{x}$ 与偏导数值 $f'_{x_i}(\boldsymbol{x})$之间构成一个新的函数映射，称为函数 $y=f(\boldsymbol{x})$对 x_i 的偏导函数，记作 $f'_{x_i}(\boldsymbol{x})$、y'_{x_i}或 $\partial f(\boldsymbol{x})/\partial x_i$、$\partial y/\partial x_i$。

注

有必要指出的是，若函数 $y=f(x)$在定义域内 x_0 处，对 x_i 的左右导数 $f'_{x_i}(\boldsymbol{x}_0-0)$与 $f'_{x_i}(\boldsymbol{x}_0+0)$均存在，定义 $a=\min(f'_{x_i}(\boldsymbol{x}_0-0),f'_{x_i}(\boldsymbol{x}_0+0))$、$b=\max(f'_{x_i}(\boldsymbol{x}_0-0),f'_{x_i}(\boldsymbol{x}_0+0))$，则集合$[a\to b]$内任意元素定义为函数 $y=f(x)$在 x_0 处对 x_i 的次导数。多元函数所有次导数构成的向量，称作次梯度。更多关于梯度的内容详见 3.4.4 节，更多关于次梯度的内容详见 8.8 节。

以上定义的函数 f 的偏导函数，称作 f 的一阶偏导函数。若函数 f 的偏导函数在定义域内对各自变量仍可导，则函数 f 的偏导函数值与偏导函数的偏导数值之间也构成新的函数映射。此时，称偏导函数的偏导函数为原函数 f 的二阶偏导函数。函数 f 的 x_i 偏导函数的 x_i 偏导函数，记作$\partial(\partial f(\boldsymbol{x})/\partial x_i)/\partial x_i$，可简记作$\partial^2 f(\boldsymbol{x})/\partial x_i^2$ 或 $f''_{x_i}(\boldsymbol{x})$。类似地，可定义函数 f 的任意高阶偏导函数。需要说明的是，函数的高阶偏导函数不要求求导变量为同一自变量。如函数 f 的 x_i 偏导函数的 x_j 偏导函数，记作$\partial(\partial f(\boldsymbol{x})/\partial x_i)/\partial x_j$，可简记作$\partial^2 f(\boldsymbol{x})/\partial x_i\partial x_j$ 或 $f''_{x_ix_j}(\boldsymbol{x})$。有必要指出的是，不难证明 $f''_{x_ix_j}(\boldsymbol{x})=f''_{x_jx_i}(\boldsymbol{x})$。也就是说，同阶偏导函数与求导次序无关。

给定任意多元函数 $y=f(x_1,x_2,\cdots,x_n)$，设其各自变量 x_i 均是另一变量 x 的函数，其中，$i\in\{1,2,\cdots,n\}$，则多元函数 $y=f(x_1,x_2,\cdots,x_n)$是自变量 x 的复合函数。由导数