

第8章图像识别

图像的变换、增强、恢复等技术都是对输入图像的某种有效的改善，其输出仍然是一幅完整的图像。随着数字图像处理技术的发展和实际应用的需求，出现了另一类问题，就是不要求其结果输出是一幅完整图像的本身，而是将经过上述处理后的图像，再经过分割和描述提取有效的特征，进而加以判决分类。

8.1 图像识别概述

“识别”这两个字分开来解释有“认识”和“区别”的含义。说“识别某物件”包含有认识它而且能从一堆物件中把它与别的物件区别开来的意思。通常我们说：“你认识某事物”，这一定是你的经历中曾经见过它或接触过它，因而了解它的某些特性。一旦你认识了它，自然也能把它与其他事物区别开来。人就具有这样的本领，例如，桌上放了一堆文具，而你需要一支笔，你会很快地从它们中间抓起一支笔。人的识别能力在于，不管桌上放的是什么形式的第一铅笔、钢笔、圆珠笔或彩色笔——你总能将它从一堆文具中挑选出来。尽管这支笔的色彩和它的独特的造型你从前并未见过。人何以具有这样的本领呢？这是因为你从前见到过笔，使用过笔，对它能写字的功能有认识，而且对它便于手握的外形也了解。这些特征，经过你的实践构成一个笔的概念存储在你的大脑里。或者说通过你以前对它的接触，有一个“笔的模式”存储在你的大脑里，这个“模式”就是你用来刻画笔的一个、两个或多个特征。当下次遇到这类物件时，如一支具体的笔，尽管你以前可能并未见到过，但是你也能从它的特征中去鉴别。符合存储在你的大脑中的这个模式的，就能判定它就是笔；否则，就不是。故有人称这种识别为“模式识别”。

图像识别，可以认为就是图像的模式识别，它是模式识别技术在图像领域中的具体运用。模式识别的研究对象基本上可概括为两大类：一类是有直觉形象的如图像、相片、图案、文字等等；一类是没有直觉形象而只有数据或信息波形如语声、心电脉冲、地震波等等。但是，对模式识别来说，无论是数据、信号还是平面图形或立体景物，都是除掉他们的物理内容而找出它们的共性，把具有同一共性的归为一类，而具有另一种共性者归为另一类。模式识别研究的目的是研制能够自动处理某些信息的机器系统，以便代替人完成分类和辨识的任务。狭义地讲，图像识别所研究的模式就是图像。

一个图像识别系统可分为四个主要部分，其框图如图 8.1 所示：

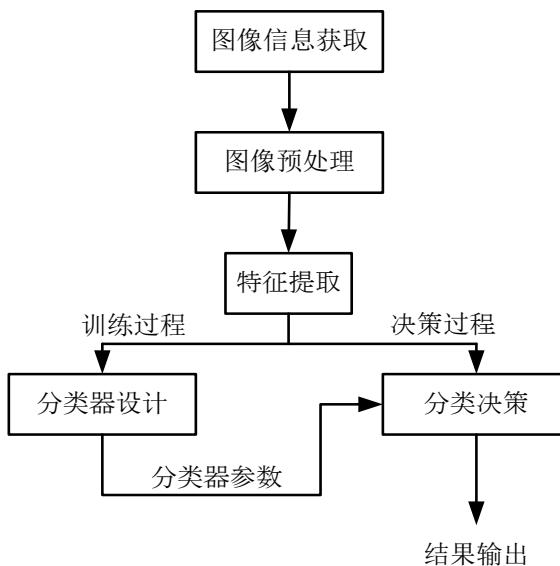


图 8.1 图像识别系统框图

第一部分是图像信息的获取。它相当于对被研究对象的调查和了解，从中得到数据和材料。对图像识别来说，就是把图片等信息经系统输入设备数字化后输入计算机以备后续处理。

第二部分是图像预处理。预处理的目的是去除干扰、噪声及差异，将原始图像变成适合于计算机进行特征提取的形式。它包括图像的变换、增强、恢复等，这些内容在本书的前面章节已作过介绍。第三部分是图像特征提取。它的作用在于把调查了解到的数据材料进行加工、整理、分析、归纳以去伪存真，去粗取精，抽出能反映事物本质的特征。当然，提取什么特征，保留多少特征与采用何种判决有很大关系。第四部分是分类判决。即根据提取的特征参数，采用某种分类判别函数和判别规则，对图像信息进行分类和辨识，得到识别的结果。这相当于人们从感性认识上升到理性认识而做出结论的过程。

8.2 判别函数和判别规则

模式识别系统的基本功能是能判别各模式所归属的类别，完成这一功能的重要方法之一是采用判别函数。判别函数有线性和非线性之分，本节将介绍几种典型的判别函数及相应的判别规则。

8.2.1 线性判别函数

线性判别函数是应用较广的一种判别函数。所谓线性判别函数，是指判别函数是图像所有 N 个特征量的线性组合。设它的组合系数为 ω_{i0} ，则对于 M 类问题，其任一类 i 的线性识别函数为：

$$D_i(X) = \sum_{k=1}^N \omega_{ik} X_k + \omega_{i0} \quad i = 1, 2, \dots, M \quad (8.1)$$

式中： $D_i(X)$ 代表第 i 个判别函数， ω_{ik} 是系数或权， ω_{i0} 为常数或称为阈值。在 ω_i 和 ω_j 两类之间的判决分界处有 $D_i(X) = D_j(X)$ ，所以边界方程为：

$$D_i(X) - D_j(X) = 0 \quad (8.2)$$

该方程在二维空间是直线，在三维空间是平面，在 N 维空间则是超平面。 $D_i(X) - D_j(X)$ 可以写成以下的形式：

$$D_i(X) - D_j(X) = \sum_{k=1}^N (\omega_{ik} - \omega_{jk}) X_k + (\omega_{i0} - \omega_{j0}) \quad (8.3)$$

相应的判别规则为：如果 $D_i(X) > D_j(X)$ 或 $D_i(X) - D_j(X) > 0$ ，则 $X \in \omega_i$ ；如果 $D_i(X) < D_j(X)$ 或 $D_i(X) - D_j(X) < 0$ ，则 $X \in \omega_j$ 。

用线性判别函数构造的分类器是线性分类器。任何 m 类问题都可以分解为 $(m-1)$ 个 2 类识别问题。方法是先把模式空间分为某一类和其他类的组合，即两两相比，如此进行下去即可实现最终分类。因此，两类线性分类器是最简单和最基本的。

分离两类的分界线由 $D_1(X) - D_2(X) = 0$ 表示。对于任何特定输入的特征向量 X ，都必须判定 $D_1(X)$ 大还是 $D_2(X)$ 大。若考虑某函数 $D(X) = D_1(X) - D_2(X)$ ，对于 1 类模式 $D(X)$ 为正，对于 2 类模式 $D(X)$ 为负。于是，只要处理与 $D(X)$ 相应的一组权值和特征向量的元素，并判断输出符号即可进行分类。执行这种运算的分类器原理框图如图 8.2 所示，其

中 Σ 是累加器。

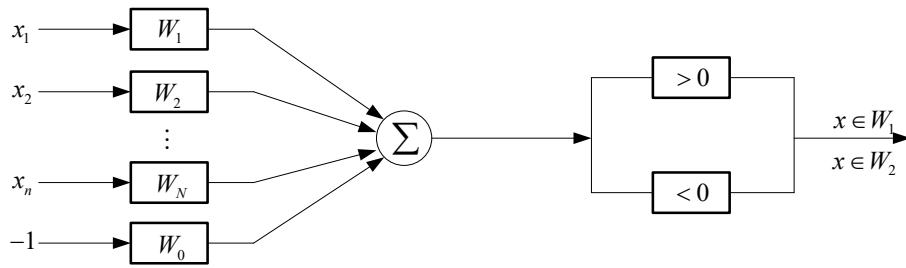


图 8.2 两类的线性分类器

在线性分类器中要找到合适的权值，才能使分类尽可能不出差错，有效的方法就是实验法。例如，先设所有的权值为 1，把已经分类的标准样本输入分类器进行判别，根据分类结果不断调整权值系数，直到实际分类结果和标准样本的分类结果基本吻合，这个过程称为线性分类器的训练或学习过程。

这里简要介绍一下调整权值系数的方法。为了便于表达，将上述线性识别函数用矩阵形式表示：将 N 个特征向量值 X 和 1 放在一起记为 Y，N+1 个权重系数记为 W，它们的矩阵形式表示为：

$$\mathbf{Y} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \\ 1 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_N \\ W_0 \end{bmatrix} \quad (8.4)$$

则线性函数可改写为：

$$D(X) = \sum_{k=1}^N W_k X_k + W_0 = [X_1, X_2, \dots, X_N, 1]^T \times [W_1, W_2, \dots, W_N, W_0] = \mathbf{Y}^T \mathbf{W} \quad (8.5)$$

式中： \mathbf{Y}^T 表示 Y 的转置矩阵。考虑有两类的图像($M=2$)，假设此时有两个训练集，即两组已经分类的标准样本集 T_1 和 T_2 ，两个训练集合是线性可分的，这意味着存在一个权值向量 W，使得：

$$\begin{cases} \mathbf{Y}^T \mathbf{W} > 0 & \text{当 } \mathbf{Y} \in T_1 \\ \mathbf{Y}^T \mathbf{W} < 0 & \text{当 } \mathbf{Y} \in T_2 \end{cases} \quad (8.6)$$

如果分类器的输出不能满足式(8.6)的条件，说明权值向量不符合分类要求，必须加以调整。

可以通过误差修正的方法对权值系数进行调整。例如，如果第一类模式 $\mathbf{Y}^T \mathbf{W}$ 不大于零，则说明系数不够大，可用加大系数 a 的方法进行误差修正。具体修正方法如下：

对于任一个 $\mathbf{Y} \in T_1$ ，若 $\mathbf{Y}^T \mathbf{W} \leq 0$ ，则使：

$$\mathbf{W}' = \mathbf{W} + \alpha \mathbf{Y} \quad (8.7)$$

对于任一个 $\mathbf{Y} \in T_2$ ，若 $\mathbf{Y}^T \mathbf{W} > 0$ ，则使：

$$\mathbf{W}' = \mathbf{W} - \alpha \mathbf{Y} \quad (8.8)$$

通常使用的误差修正方法有固定增量规则、绝对修正规则和部分修正规则：

(1) 固定增量规则: 选择 α 为一个固定的非负数。

(2) 绝对修正规则: 取 α 为一个最小整数, 它是使 $\mathbf{Y}^T \mathbf{W}$ 值刚大于零的一个阈值。公式为:

$$\alpha = \frac{|\mathbf{Y}^T \mathbf{W}|}{\mathbf{Y}^T \mathbf{W}} \quad (8.9)$$

(3) 部分修正规则: 取 a 为下式:

$$\alpha = \lambda \frac{|\mathbf{Y}^T \mathbf{W}|}{\mathbf{Y}^T \mathbf{W}} \quad 0 < \lambda \leq 2 \quad (8.10)$$

8.2 最小距离判别函数

在图像识别中, 线性分类器一种很重要的方法就是将未知类别的图像和特征空间中作为模板的点 (标准样本的中心) 之间的距离作为分类的准则。对于 M 类模板, 未知类别图像与哪一类距离最近就属于那一类。

假定图像类别数为 M , 分别为 W_1, W_2, \dots, W_M 。每类有一个标准图像模板特征向量, 则共有 M 个模板特征向量, 表示为 Z_1, Z_2, \dots, Z_M 。则未知类别图像的特征向量 X 和 W_i 类的模板特征向量 Z_i 之间的欧几里德距离为:

$$D_i(\mathbf{X}) = d(\mathbf{X}, \mathbf{Z}_i) = \|\mathbf{X} - \mathbf{Z}_i\| = \sqrt{(\mathbf{X} - \mathbf{Z}_i)^T (\mathbf{X} - \mathbf{Z}_i)} \quad i = 1, 2, \dots, M \quad (8.11)$$

相应的判别规则为: 把未知图像的特征向量 X 和 M 类图像的模板特征向量分别求距离, 可得到一个距离集 D_1, D_2, \dots, D_M , 将 X 分到与它距离最近的类别。换句话说, 对所有的 $j \neq i$, 若 $D_i(X) < D_j(X)$, 则 X 就属于 W_i 类, 即 $X \in W_i$ 。基于最小距离判别函数的分类器称为最小距离分类器。

方程(8.11)可改写为:

$$\begin{aligned} D_i^2(\mathbf{X}) &= \|\mathbf{X} - \mathbf{Z}_i\|^2 = (\mathbf{X} - \mathbf{Z}_i)^T (\mathbf{X} - \mathbf{Z}_i) \\ &= \mathbf{X}^T \mathbf{X} - 2\mathbf{X}^T \mathbf{Z}_i + \mathbf{Z}_i^T + \mathbf{Z}_i \\ &= \mathbf{X}^T \mathbf{X} - 2(\mathbf{X}^T \mathbf{Z}_i - \frac{1}{2} \mathbf{Z}_i^T \mathbf{Z}_i) \quad i = 1, 2, \dots, M \end{aligned} \quad (8.12)$$

因为所有距离都是正的, 所以 D_i^2 最小, 也就是 D_i 最小。在方程(8.12)中, $\mathbf{X}^T \mathbf{X}$ 与 i 无关, 则最小化 D_i^2 等于最大化 $(\mathbf{X}^T \mathbf{Z}_i - \frac{1}{2} \mathbf{Z}_i^T \mathbf{Z}_i)$ 。因此可以定义判别函数:

$$D_i(X) = \mathbf{X}^T \mathbf{Z}_i - \frac{1}{2} \mathbf{Z}_i^T \mathbf{Z}_i \quad i = 1, 2, \dots, M \quad (8.13)$$

相应的判别规则为: 对所有的 $j \neq i$, 若 $D_i(X) > D_j(X)$, 则 X 属于 W_i 类, 即 $X \in W_i$ 。

由式(8.12)和(8.13)可见, $D_i(x)$ 是一个线性函数, 因此最小距离分类器也是一个线性分类器。在最小距离分类中, 在决策边界上的点与相邻两类都是等距离的, 这种方法就难于解决。此时, 必须寻找新的特征, 重新分类。

8.2.3 最近邻域判别函数

最近邻域判别函数是最小距离判别函数概念的延伸。上述最小距离判别函数是取一个

最标准的向量作为模板，而在最近邻域判别函数中，每类图像的模板不是取一个点为代表，而是取一组点代表一类。未知类别图像与模板的距离是一个点和一组点之间的距离。如有 M 类图像 W_1, W_2, \dots, W_M ，某一类 w_i 中含有 N_i 个标准模板，我们把这个标准模板集合用符号 Z_i 表示，则 Z_i 集合中的模板可表示为 $Z_i = \{Z_1^1, Z_1^2, \dots, Z_1^{N_i}\}$ ，也就是属于集合 Z_i 的一组空间点。输入图像与 w_i 类的距离可用下式表示：

$$d(x, Z_i) = \min(d(x, Z_i^k)) = \min \|x - Z_i^k\| \quad k = 1, 2, \dots, N_i \quad (8.14)$$

也就是说，输入图像 x 与模板集 Z_i 的距离就是 x 与 Z_i 中一组空间点的距离中最小的距离。空间点之间距离的求取方法与最小距离分类器中距离的求法相同。

在采用这种判别函数的图像识别中，决策边界将是分段线性的。例如，有一个两类问题如图 8.3 所示， $w1$ 类模板集由两个模板组成，即 $Z_1 = \{Z_1^1, Z_1^2\}, N_1 = 2$ ； $w2$ 类模板集由 3 个模板组成，即 $Z_2 = \{Z_2^1, Z_2^2, Z_2^3\}, N_2 = 3$ ，其分界线是由分段直线组成，用实线表示；虚线连接产生两类分界线的空间点。

识别系统判定对象类别时，首先要计算输入图像与每个空间点的距离，然后找出最小距离。这种方法概念简单，分段的线性边界可以分段拟合成很复杂的曲线，把本来是曲线的边界用分段直线来近似代替，降低了识别的复杂性。

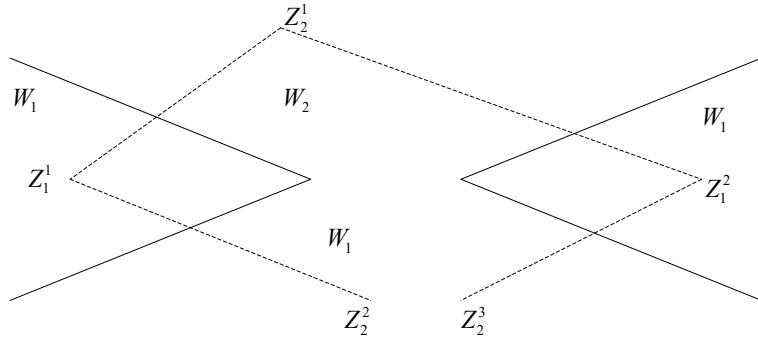


图 8.3 两类最近邻域分类示意图

8.2.4 非线性判别函数

线性判别函数很简单，但也有缺点，它对于较复杂的分类往往不能胜任。在较复杂的分类问题中，往往需要提高判别函数的次数，因此根据问题的复杂性，可将判别函数从线性推广到非线性。非线性判别函数可写成下式的形式：

$$\begin{aligned} D(x) &= \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_N x_N + \omega_{12} x_1 x_2 + \omega_{13} x_1 x_3 \\ &\quad + \dots + \omega_{1N} x_N + \omega_{11} x_1^2 + \omega_{22} x_2^2 + \dots + \omega_{NN} x_N^2 \\ &= \omega_0 + \sum_{k=1}^N \omega_{kk} x_k^3 + \sum_{k=1}^N \omega_k x_k + \sum_{k=1}^N \sum_{i=1}^N \omega_{ki} x_k x_i \end{aligned} \quad (8.15)$$

式(8.15)一个二次型判别函数，通常二次型判别函数的决策边界是一个超二次曲面。

8.3 特征的提取和选择

在图像识别中，对获得的图像直接进行分类是不现实的。首先，图像数据占用很大的存储空间，直接进行识别费时费力，其计算量无法接受；其次，图像中含有许多与识别无关的信息，如图像的背景等，因此必须进行特征的提取和选择，这样就能对被识别的图像数据进行大量压缩，有利于图像识别。提取特征和选择特征很关键，特征若提取得不恰当，分类就不能很精确，甚至无法分类。

良好的特征应具有四个特点：

(1) 可区别性。对于属于不同类别的图像，它们的特征值应具有明显的差异。

(2) 可靠性。对于同类的图像，它们的特征值应比较相近。

(3) 独立性。所使用的各特征之间应彼此不相关。

(4) 数量少。图像识别系统的复杂度随着特征的个数迅速增长，尤为重要的时用来训练分类器和测试结果的样本数量随特征的数量呈指数关系增长。

我们能够利用很多词语描述一个西瓜，例如色泽、根蒂、纹理、触感、敲声等，但是有经验的人往往只需要看根蒂、听听敲声就知道是否是好瓜，换而言之，对于一个任务来说，其中有些描述词语可能很关键、很有用，另外一些词语可能没什么用。我们将这些词语当作特征，对于当前任务有用的词语称为相关特征，没用的词语称为无关特征。特征选择的过程就是从给定的词语集合中选出相关特征的过程。

特征选取的方法是很多的。从一个模式中提取什么特征，将因不同的模式而异，并且与识别的目的、方法等有直接关系。有关图像特征的各种检测方法，都在前面章节中已经介绍过，这里不再赘述。需要说明的是，特征提取和选择并不是截然分开的，有时可以先将原始特征空间映射到维数较低的空间，在这个空间中再进行选择以进一步降低维数。也可以先经过选择去掉那些明显没有分类信息的特征，再进行映射以降低维数。

特征提取和选择的总原则是：尽可能减少整个识别系统的处理时间和错误识别概率。当两者无法兼得时，需要人们作出相应的平衡。或者缩小错误识别的概率，以提高识别精度，但会增加系统运行时间；或者提高整个系统速度以适应实时需要，但会增加错误识别的概率。

8. 4 统计模式识别方法

当图像特征分布的统计性质已知或能够推断时，可采用统计模式识别法。在对样本图像的实际分类时，测量到的数据总会不同程度的受到噪声的影响，这时测量到的特征有可能不代表图像。此时为了客观地描述图像，就要用统计的方法。统计方法最基本内容为贝叶斯分析理论。

8.4.1 基本概念

先介绍统计模式识别法中一些符号的含义：

$p(\omega_i)$ 为 ω_i 的先验概率。是检测对象属于 ω_i 类的概率的预先了解。

$p(x | \omega_i)$ 为 x 属于 ω_i 类的条件概率。此时把检测对象的特征 x 看作是一个其分布依赖于类别状态的随机变量，随着检测对象特征值 x 的不断变化，其落入 ω_i 类的可能性（概率）也不断变化。这种可能性的变化可以用函数来描述。

$p(\omega_i | x)$ 为特征 x 属于 ω_i 的后验概率。它表示把检测对象特征 x 的状态作观察后判断属于 ω_i 的可能性。此时 x 是特定的值，它是由未知类别的检测对象经特征提取产生的。

8. 4. 2 贝叶斯(Bayes)分类器

对于一个实际的图像识别问题来说，由于存在噪声的干扰，造成检测对象的几何分布常常不是线性可分的，甚至在同一区域内有可能出现不同类检测对象的情况，因而不可避免地会出现错分现象。贝叶斯准则就是基于错分概率或风险最小的准则，按 Bayes 准则建立起来的 Bayes 判别规则称为 Bayes 分类器。

假定有一模式 X ， X 属于 W_i 类的概率为 $P(W_i | X)$ 。如果模式 X 实际是属于 W_i 类的，

而分类器把它分到 W_j 类，于是就会产生损失，记作 L_{ij} 。由于模式 X 可能属于所研究的 M 类中的任何一类，于是分配 X 为 W_j 类所发生的期望损失为：

$$r_j(X) = \sum_{i=1}^M L_{ij} P(W_i / X) \quad (8.16)$$

在判别理论中，常把 $r_j(X)$ 称为条件平均风险。

对于每个给定模式，分类器有 M 种可能的分法。如果对每个模式 X 计算 $r_i(X)$ ，

$r_1(X), \dots, r_M(X)$ ，并且将它分到条件平均风险最小的那一类，这样的分类器，就称它为 Bayes 分类器。

利用 Bayes 公式：

$$P(W_i / X) = \frac{P(W_i)P(X / W_i)}{P(X)} \quad (8.17)$$

式中： $P(W_i)$ 为 W_i 类出现的先验概率； $P(X / W_i)$ 为在 W_i 类中出现 X 的条件概率；

$P(W_i / X)$ 为 X 属于 W_i 类的后验概率。

于是式(8.16)可以进一步表示为：

$$r_j(X) = \frac{1}{P(X)} \sum_{i=1}^M L_{ij} P(W_i) p(X / W_i) \quad (8.18)$$

由于在求 $j=1,2,\dots,M$ 的 $r_j(X)$ 值时， $1/P(X)$ 是一个公共因子，所以可将它从式中略去。于是平均风险的表达式可以简化为：

$$r_j(X) = \sum_{i=1}^M L_{ij} P(W_i) p(X / W_i) \quad (8.19)$$

对于两类问题， $M=2$ 。如果将一模式 X 分到类别 1，于是：

$$r_1(X) = L_{11} p(x / \omega_1) p(\omega_1) + L_{21} p(x / \omega_2) p(\omega_2) \quad (8.20)$$

如果分到类别 2：

$$r_2(X) = L_{12} p(x / \omega_1) p(\omega_1) + L_{22} p(x / \omega_2) p(\omega_2) \quad (8.21)$$

如上所述，Bayes 分类器是将一模式分配到 r 值最小的那一类。因此，当 $r_1(x) < r_2(x)$ ，也就是当

$$L_{11} p(x / \omega_1) p(\omega_1) + L_{21} p(x / \omega_2) p(\omega_2) < L_{12} p(x / \omega_1) p(\omega_1) + L_{22} p(x / \omega_2) p(\omega_2) \quad (8.22)$$

或写成当

$$(L_{21} - L_{12}) p(x / \omega_2) p(\omega_2) < (L_{12} - L_{11}) p(x / \omega_1) p(\omega_1) \quad (8.23)$$

时，将 x 判别为 ω_1 类；否则，将 x 判别为 ω_2 类。

对于多类问题，当 $j=1,2,\dots,M$, $j \neq i$, $r_i(x) < r_j(x)$ 时，模式 x 被分配到 ω_i 类，也就是说，当

$$\sum_{k=1}^M L_{ki} p(x/\omega_k) p(\omega_k) < \sum_{q=1}^M L_{qi} p(x/\omega_q) p(\omega_q) \quad j=1,2,\dots,M, j \neq i \quad (8.24)$$

时， x 属于 ω_i 类

在大多数模式识别问题中，对于正确的判别，分类损失 r_{ij} 为零；而对于所有错误的判别，则有相同的损失。因此，损失函数可以表示为：

$$L_{ij} = 1 - \delta_{ij} \quad (8.25)$$

其中：当 $i=j$ 时， $\delta_{ij}=1$ ；当 $i \neq j$ 时， $\delta_{ij}=0$ 。上式说明：当模式分类不正确时，具有值等于 1 的归一化损失；而在分类正确时，则无损失。

将式 (8.25) 代入 (8.19) 中，可得：

$$r_j(X) = \sum_{i=1}^M (1 - \delta_{ij}) P(W_i) p(X/W_i) = P(X) - P(X/W_j) P(W_j) \quad (8.26)$$

于是 Bayes 判决规则可以写成，当

$$P(X) - P(X/W_i) P(W_i) < P(X) - P(X/W_j) P(W_j) \quad j=1,2,\dots,M; j \neq i \quad (8.27)$$

或

$$P(X/W_i) P(W_i) > P(X/W_j) P(W_j) \quad j=1,2,\dots,M; j \neq i \quad (8.28)$$

时，模式 X 属于 W_i 类。

根据判决函数的意义可知，式(8.28)的 Bayes 判别规则实际上也是判决函数：

$$D_i(X) = P(X/W_i) P(W_i) \quad i=1,2,\dots,M \quad (8.29)$$

的执行过程。当某一模式 X 对所有 $j \neq i$ 时的 $D_i(X) > D_j(x)$ ，则该模式属于 W_i 类。

从上述论述可以看出，作为 Bayes 分类器的特殊情况，规定正确分类时的损失为零，不正确分类时的损失都相等，对这种特殊情况所作的最优判决使分类错误概率为最小。

如果假定概率密度函数 $P(X/W_i)$ 是多变量正态分布的，则其概率密度函数为：

$$P(X/W_i) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{X} - \mathbf{M}_i)^T (\Sigma_i)^{-1} (\mathbf{X} - \mathbf{M}_i) \right] \quad (8.30)$$

式中： $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$ 为模式的特征向量；

$\mathbf{M} = [m_1, m_2, \dots, m_n]^T$ 为数学期望向量，其中 $m_i = \frac{1}{L} \sum_k x_{ik}$ ， x_{ik} 表示第 i 类第 k 个像

素的灰度值, L 为第 i 类像素数;

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}$$

其中

$$\sigma_{ij} = \frac{1}{L} \sum_k (x_{ik} - m_i)(x_{ik} - m_j)$$

由于式(8.29)中 $P(X/W_i)$ 是正态密度函数, 它是指数形式的。为了便于计算, 可以用取对数的方式来处理, 因而将判决函数写成:

$$D_i(X) = \ln P(X/W_i)P(W_i) = \ln P(X/W_i) + \ln P(W_i) \quad i=1, 2, \dots, M \quad (8.31)$$

由于 \ln 是一个单调增加的函数, 因此从分类效果来看, 式(8.31)与(8.29)完全等效。将式(8.30)代入(8.31), 得到:

$$D_i(X) = \ln P(W_i) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} [(\mathbf{X} - \mathbf{M}_i)^T (\Sigma_i)^{-1} (\mathbf{X} - \mathbf{M}_i)] \quad i=1, 2, \dots, M$$

(8.32)

由于 $(n/2)\ln 2\pi$ 项与 i 无关, 可把它从表达式中略去, 于是 $D_i(X)$ 变成:

$$D_i(X) = \ln P(W_i) - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} [(\mathbf{X} - \mathbf{M}_i)^T (\Sigma_i)^{-1} (\mathbf{X} - \mathbf{M}_i)] \quad i=1, 2, \dots, M \quad (8.33)$$

式(8.33)即为 Bayes 分类判决函数。相应的 Bayes 分类判决规则为:

若对于所有可能的类 j ($j=1, 2, \dots, N$; $j \neq i$) 有 $d_i(X) > d_j(X)$, 则 X 属于 W_i 类。

8.5 深度神经网络图像识别

1、深度神经网络

2016 年, Google 旗下的 DeepMind 公司开发的围棋程序 AlphaGo 与世界围棋冠军李世石、柯洁相继进行人机大战, 最终分别以 4:1 和 3:0 的总比分赢得比赛。随着媒体的争相报道, 人工智能 (Artificial Intelligence, AI)、机器学习 (Machine Learning, ML)、深度学习 (Deep Learning, DL) 这 3 个概念得到了迅速普及。图 8.4 很好的概括了这三者间的关系和各自兴起年代。人工智能的概念从提出后就一直广受关注和期待, 机器学习是实现人工智能的一个分支, 也是人工智能领域发展最快的一个分支。而深度学习是一种机器学习方法, 和传统的机器学习方法一样, 都可以根据输入的数据进行分类或者回归。但是随着数据量的增加, 传统的机器学习方法表现的不尽如人意, 而此时深度学习表现出了优异的性能, 迅速受到学术界和工业界的重视, 现在甚至有人开始担心未来人工智能是否会危害人类。

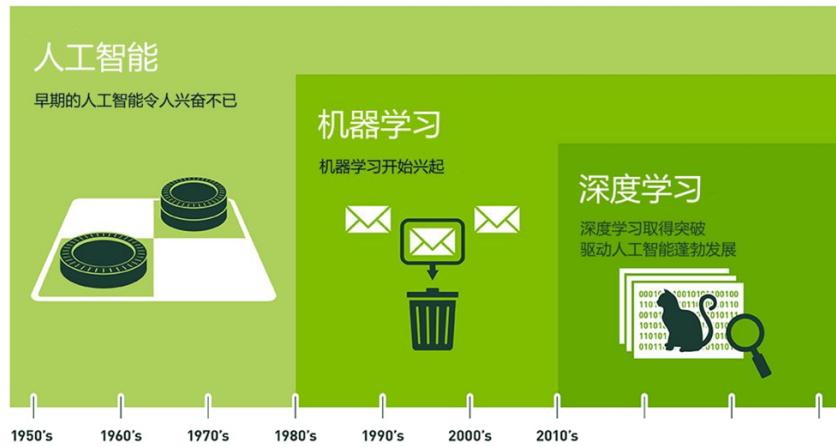


图 8.4 人工智能、机器学习、深度学习三者之间关系

神经网络是人工智能研究的重要对象，因为研究者相信人类的智能主要体现在大脑的神经网络之中。神经网络的本质是一个数学模型，用的是完成将输入数据映射成输出结果值的函数功能。与普通函数不同的是，神经网络函数中的函数结构和参数在学习过程中不断调整和优化的。深度学习是多层神经网络，利用简单的表示，逐层表达复杂现象。深度神经网络不是凭空出现的，是许多学者长期以来对神经网络坚持不懈的研究得来的，是目前神经网络的集大成者。神经网络的发展可以大致分为 3 个阶段：

1) 1943 年，McCulloch 和 Pitts 提出了人工神经网络的概念。人工神经网络(artificial neural network)的研究最初是受生物神经系统启发的，是以模拟人体神经系统的结构和功能为基础而建立的一种信息处理系统。由生物学知识可以知道，人体神经系统由神经细胞(即神经元)构成，每一个神经元包括细胞体、树突、轴突三个部分。神经元之间通过树突和轴突的相互连接(称为突触)形成神经网络。其中，细胞体相当于一个初等处理器，它对来自其他神经元的信号进行处理(例如进行求和运算)，然后产生一个输出信号；树突是神经元的输入部分，接收其他神经元的信号；轴突则是神经元的输出部分，产生输出信号并通过突触传送给其他神经元。1958 年前后，感知机和自适应线性单元等线性模型的提出，让人工神经网络走进现实，并获得人们的关注和期望。然而，线性模型有很多局限性，如它们无法学习异或逻辑函数。因此人工神经网络受到抵触而研究陷入停滞。

2) 从 20 世纪 70 年代开始人们对神经网络的研究热情不断下降，特别是 1995 年，最受欢迎的支持向量机算法被提出，并进一步的削弱了深度神经网络的影响力，但是深度学习算法的发展并没有停止。在 1989 年，Yann LeCun 等人将标准的反向传播算法应用于卷积神经网络中实现了自动提取图像特征，成功的完成了手写数字识别。1992 年前后，研究人员使用神经网络进行序列建模方面取得了重要进展。长短期记忆网络 (LSTM) 通过在神经单元中引入书入门、遗忘门、输出门等门的概念，来选择对长期信息和短期信息的提取，提升网络的记忆能力。但是受神经网络训练时间长，参数设置严重依赖经验等多种因素的影响，最终并没有引起人们足够的重视。然而，这期间的几个重要贡献如今仍然有深远影响，反向传播的思想仍是现在深度模型训练的主导方法，LSTM 在许多序列建模任务中广泛应用，特别是自然语言处理任务。

3) 2006 年，Hinton 提出了深度神经网络，使用一种称为“贪婪逐层预训练”的策略来有效的训练，通过非监督学习来学习出网络结构，再由后向传播算法学习网络内部参数。2009 年，基于长短期记忆的递归神经网络获得了 ICDAR 手写字体识别大赛的冠军，尤其是在 2012 年的计算机视觉领域的著名比赛—ImageNet ILSVRC 分类比赛中，基于深度学习的 AlexNet 以高于第二名 10 个百分点的战绩高调获得第一名。自此，迎来了深度学习的伟大复兴。在很短的时间内，各种基于深度学习的方法刷新了很多领域的最好成绩，并取得了巨

大的进步。截止目前为止，深度学习成为了人工智能的主要研究方向。同时伴随着各种硬件设备，如 GPU, TPU 等运算设备的大力发展，各种深度学习框架如 Caffe, Pytorch, Tensorflow, Keras Theano, MXNet, CNTK 的开源，现在深度学习的流行程度和易用性得到了极大的提升。

2、卷积神经网络

卷积神经网络是非常适合计算机视觉应用的模型，早在 20 世纪 90 年代，卷积神经网络就已经被广泛应用，深度学习的复兴也归因于卷积神经网络的成功应用。接下来我们将简单介绍下典型的构成卷积神经网络的层，包括卷基层，池化层，激活函数，全连接层等，以及经典的卷积神经网络模型。

1) 卷积层

通常在计算机处理数据时，时间是离散的，因此只考虑离散卷积，其数学定义如式 8.34 所示是对两个实变函数之间的内积运算。在卷积网络的术语中，我们称 x 为输入（Input）， ω 为核函数（Kernel Function），输出 s 为特征映射（Feature Map）。

$$s(t) = \sum_{a=-\infty}^{\infty} x(a)\omega(t-a) \quad (8.34)$$

在人工智能的应用中使用卷积运算有三个好处：稀疏交互（Sparse Interaction）、参数共享（Parameter Sharing）和等价表示（Equivariant Representation）。此外，卷积也提供了一种处理输入尺寸可变的方法。

稀疏交互也叫稀疏连接，是指通过尺寸远小于输入数据尺寸的卷积核来进行卷积操作，建立输入数据和输出特征映射之间的稀疏连接。这样能减少存储参数，减少模型的训练参数，从而加快模网络的运行速度。

参数共享是指模型中的多个函数可以共用同一组参数。在一个卷积层中，一个卷积核会作用于输入的所有位置，这样不需要针对每个位置学习一个单独的参数集合，如图 8.5 所示，针对一个二维图像使用 2×2 的卷积核，步长为 1 的运算过程。进一步的减少了存储需求和模型的参数量。为了提高对卷积层对输入的特征提取能力和表达能力，可以在一个卷积层中使用多个卷积核，分别获得不同的特征映射，如图 8.6 所示，利用不同的卷积核获得不同的特征映射。

卷积神经网络的参数共享使得卷积层具有输入发生变化输出也随之发生同样变化的变换等价性。

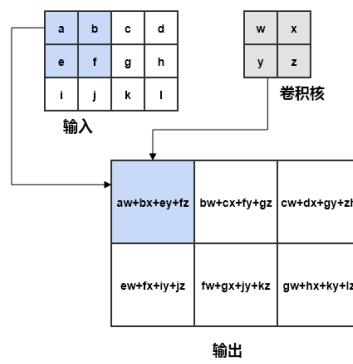


图 8.5 二维卷积运算过程示意图

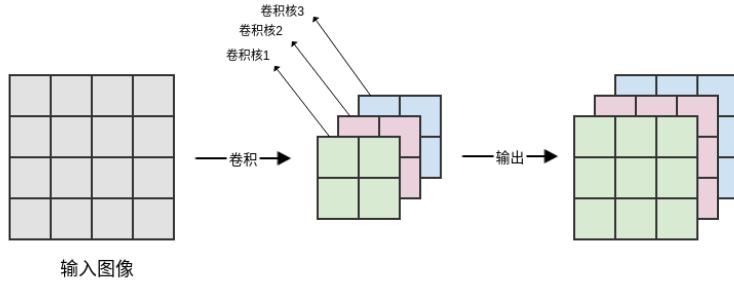


图 8.6 多核卷积示意图

2) 池化层

池化函数使用某一位置所在局部区域的总体统计特征来代替网络在该区域的输出。如果用改区域的平均值代替该区域的输出，则称为平均值池化，如果使用最大值则称为最大值池化。这两种池化是目前最常用的 2 中池化方式，如图 8.7 所示，对输入数据进行滑动窗口为 2×2 ，步长为 2 的最大值池化和平均值池化。

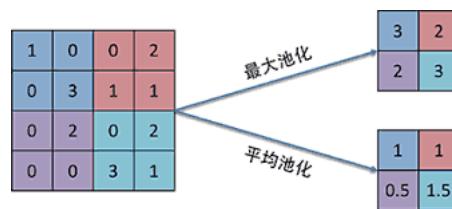


图 8.7 最大值池化和平均值池化

池化函数能对数据进行进一步的浓缩，缓解计算时内存压力，同时能帮助让网络具备局部平移不变性。

3) 激活函数

激活函数是一种模仿生物神经元的函数，是生物神经元的抽象和简化，旨在帮助网络学习数据中的复杂模式。与人类大脑中神经元模型类似，一个节点的激活函数定义了该节点在给定的输入或输入集合下的输出。

现实中的很多问题是线性不可分的，在神经网络中引入非线性激活函数，能让神经网络的表达能力更加强大了。常用的激活函数公式及图像如图 8.8 所示。

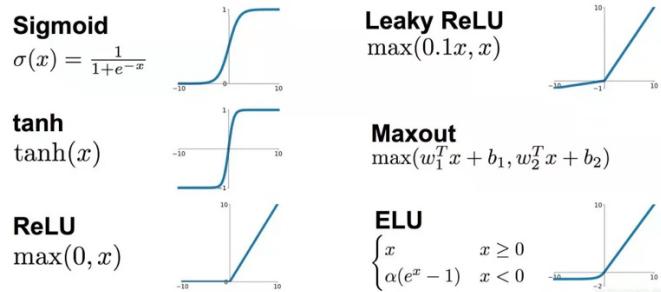


图 8.8 常用激活函数

4) 全连接层

全连接层是每一个结点都与上一层的所有结点相连，其所有的输出和该层的输入都有连接，即每个输入节点对所有的输出节点都有影响，如图 8.9 所示。由于其全相连的特性，一般全连接层的参数也是最多的。

卷积神经网络的末端，通常会由多个全连接层组成。全连接层在整个卷积神经网络中起到“分类器”的作用。如果说卷积层、池化层和激活函数等操作是将原始数据映射到隐层特征空间的话，全连接层则起到将学到的特征空间映射到样本标记空间的作用。

在实际使用中，全连接层可由卷积核为 1×1 的卷积卷积操作实现。

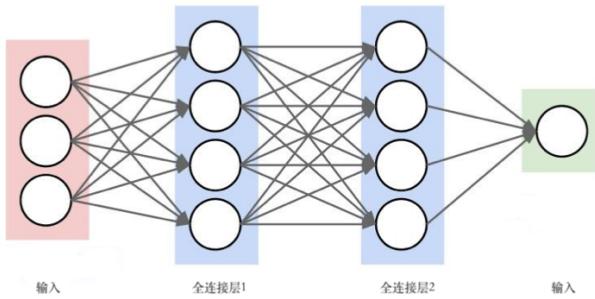


图 8.9 全连接层

3、经典的图像分类卷积神经网络模型

图像相比文字能够提供更加生动、容易理解及更具艺术感的信息，是人们传递与交换信息的重要来源。接下来我们将专注于图像领域的一个重要问题，即图像分类。

图像分类是根据图像的语义信息将不同类别图像区分开来，是计算机视觉中重要的基本问题，也是图像检测、图像分割、物体跟踪、行为分析等其他高层视觉任务的基础。传统的图像分类是通过提取图像的手工特征，再利用分类器判别物体类别。而基于深度学习的图像分类方法，可以通过有监督或无监督的方式学习层次化的特征描述，从而取代了手工设计或选择图像特征的工作。深度学习模型中的卷积神经网络(Convolution Neural Network, CNN)近年来在图像领域取得了惊人的成绩，CNN 直接利用图像像素信息作为输入，最大程度上保留了输入图像的所有信息，通过卷积操作进行特征的提取和高层抽象，模型输出直接是图像识别的结果。这种基于“输入-输出”直接端到端的学习方法取得了非常好的效果，得到了广泛的应用。

从 2012 年 AlexNet 大幅度的刷新 ImageNet ILSVRC 分类结果获得比赛冠军开始，卷积神经网络获得了大量的关注和研究，后续不少优秀的研究者不断刷新新纪录，提出了很多优秀的模型：VGGNet，GoogleNet，ResNet 等。这些经典的卷积神经网络模型形状、深度各异，但他们还是有很多共同点，因此先从最早的 CNN 模型 LeNet-5 开始介绍。

1) LeNet-5

LeNet-5 是 1998 年由 LeCun 提出用于手写字符识别的。LeNet-5 的网络结构如图 8.1 所示。LeNet-5 网络一共包含 7 层，输入图像为 32x32 的灰度图像，经过卷积层 C1 后输出 6 个 28x28 的特征映射，再经过一个下采样层 S2 后输出 6 个 14x14 的特征映射，通过卷积层 C3 和下采样层 S4 后输出 16 个 5x5 的特征映射，经过 C5 和 F6 两个全连接层后，特征映射维度变为 84 维，最后一层为包含 10 个神经元的输出层，对手写数字进行分类，即 0 到 9 一共 10 类。

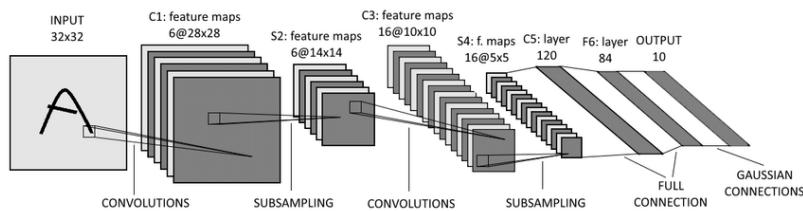


图 8.10 LeNet-5 网络结构图

2) AlexNet

2012 年，Alex Krizhevsky 提出的 AlexNet 开启了神经网络的新篇章，奠定了深度学习在计算机视觉领域的霸主地位，在同年的 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 图像分类竞赛（1000 个类别，128 万张图片的分类任务）中一举刷新了记

录，精度远远超过当年的第二名 ISI。AlexNet 的网络结构如图 8.11 所示。AlexNet 是一个包含 8 层的卷积神经网络，5 个卷积层和 3 个全连接层。每个卷基层中包含一个 ReLU 激活函数以及局部响应归一化 (LRN)，卷积计算后通过最大值池化层达到降维效果。最后一个全连接层为包含 1000 个神经元的输出层，输出 1000 类图像类别。

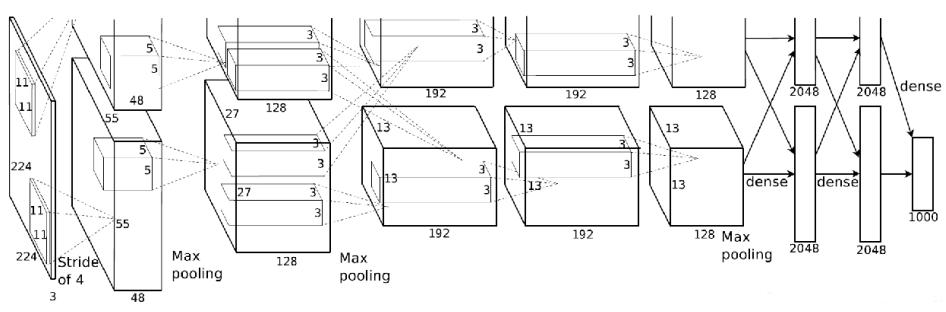


图 8.11 AlexNet 网络结构

由于当时 GPU 的计算性能有限，AlexNet 最初的设计是分为两路，由 2 块 GPU 并行训练，一块 GPU 负责顶部模型部分，一块 GPU 负责底部模型部分，两块 GPU 在某些层间互相通信。但是随着计算性能的提升，现在完全可以单卡训练。

AlexNet 网络的输入是 3 通道 RGB 彩色图像，图像大小为 224x224。经过多次卷积和池化后输出 256 个 13x13 的特征映射，展平 (flatten) 后经过多个全连接层获得分类结果，最后通过 softmax 函数将分类结果转换为概率。

3) VGGNet

VGGNet 是由牛津大学 (Oxford) 计算机视觉组 (Visual Geometry Group) 于 2014 年提出，并取得了当年 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 竞赛的第二名。VGGNet 一共有四种不同深度层次的卷积神经网络，分别是 11, 13, 16, 19 层，网络结构图如图 8.12 所示，一簇卷积层之后是三个全连接层，前两个包含 4096 个神经单元，第三个对应 ImageNet ILSVRC 竞赛的 1000 个分类采用 1000 个神经单元。VGGNet 继承了 AlexNet 的很多结构。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

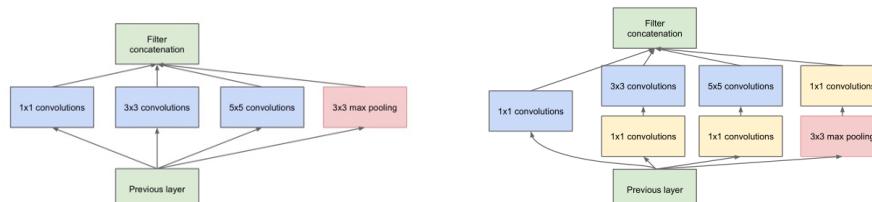
图 8.12 VGG 系列网络结构

在 VGGNet 的多种网络结构中，最常用的就是 VGG-16（图中网络结构 D）和 VGG-19（图中网络结构 E）。从图 8.12 可以看出，这两种网络的最大区别就是网络深度的不同。网络输入是一个固定尺寸为 224x224 的 RGB 图像，通过减去均值除以方差等预处理后输入神经网络的卷积层，VGGNet 采用多个感受野较小的 3x3 的卷积核，而没有像 AlexNet 一样采用感受野较大的卷积核。VGG 的作者认为 2 个 3x3 的卷积核堆叠获得的感受野大小相当于一个 5x5 的卷积核，而 3 个 3x3 的卷积核堆叠获得的感受野大小相当于一个 7x7 的卷积核。使用小的卷积核一方面可以减小参数，另一方面增加了更多的非线性映射，可以进一步的增加网络的拟合能力。

4) GoogleNet

GoogleNet，又称为 Inception，是由 Google DeepMind 公司于 2014 年提出，在同年 ImageNet ILSVRC 竞赛中获得冠军。GoogleNet 通过精巧的网络结构设计，在保持一定计算开销的前提下增加了网络深度和宽度，一共 22 层，且没有全连接层，与 AlexNet 相比，GoogleNet 在精度上获得显著提升，同时参数减少将近 12 倍。

GoogleNet 的核心是 Inception 模块，分别为简单的 Inception 模块和维度减小的 Inception 模块，如图 8.13 所示。和简单的 Inception 模块相比，维度减小的 Inception 模块在 3x3，5x5 的卷积前面和 3x3 的池化层后面分别添加了 1x1 的卷积进行降维，从而减少计算量和修正线性特性。



(a)简单的 inception 模块

(b) 维度减小的 inception 模块

图 8.13 inception 模块

GoogleNet 中还添加了 2 个辅助分类器，增加低层网络的分类能力，同时能防止梯度消失。GoogleNet 的网络设计充分考虑了计算效率和实用性，因此可以在更多设备上使用，其详细网络结构如图 8.14 所示，其中 type 表示每一层的类型；patch size 表示 kernel 大小；stride 表示步长；#nxn 表示对应 nxn 卷积核数量；#nxn reduce 表示用于 nxn 卷积层前 1x1 卷积核数量；pool proj 表示维度减少的 inception 模块中池化层后面 1x1 卷积核数量；params 表示参数量，ops 表示操作量。随着相关技术和理论的发展，GoogleNet 后续产生了几个改进版本，如 2015 年提出的 Inception-v2，2016 年提出的 Inception-v3 和 2017 年提出的 Inception-v4。

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

图 8.14 GoogleNet 详细网络结构

5) Resnet

Resnet 是 2015 年由微软亚洲研究院所提出的深度残差网络，在 2015 年的 ImageNet ILSVRC 和 COCO 的 5 个领域都获得了冠军。

深度退化 在经验上，网络的深度是影响模型的性能重要因素，当网络层数增加时，网络可以进行更加复杂的特征模式的提取，所以理论上更深的模型具有更好的表现。那么更深的网络是否更好？回答这个问题的一大障碍是梯度爆炸/消失。当更深的网络开始收敛时，“退化”（Degradation）现象便暴露出来了：网络深度增加时，网络准确度出现饱和，甚至出现下降。如图 8.15 所示：56 层网络的训练误差和测试误差都比 20 层网络要高，即增加网络深度后，模型的效果反而变差了。这并非过拟合的问题，因从训练误差上可以看出 56 层网络并没有达到过拟合的程度，而是因为随着网络深度增加出现了退化现象。

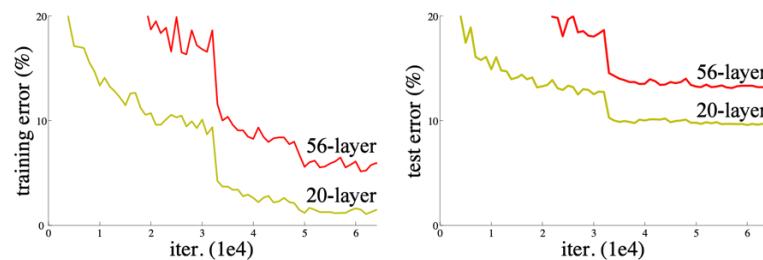


图 8.15 更深的模型同时具有更高的训练误差和测试误差

Resnet 一个最重要的贡献是解决了超深层 CNN 网络的训练问题。之前的 GoogleNet 只有 22 层，VGGNet 多个版本中最多是 19 层，而 Resnet 能够多达 152 层，甚至尝试了 1000 层。ResNet 通过引入“深度残差学习”的框架来解决退化问题。接下来我们详细介绍下“深度残差学习”。

残差学习 若将输入设为 x ，网络层设为 H ，那么 $H(x)$ 为输入通过此层的输出。传统的思路是拟合 $y=H(x)$ ，即通过训练学习出函数 H 的表达。而残差学习是拟合残差 $F(x)=H(x)-x$ ，即致力于学习输入和输出之间的残差，原始映射就变成 $H(x)=F(x)+x$ 。残差学习的模块如图 8.16 所示，残差学习模块由 $F(x)$ 和 x 两部分组成，图中的类似于电路“短路”的连接称为短路连接（shortcut connection），这是一种恒等映射（identity mapping），即 x 的部分；除去短路连接后剩余的部分称为残差映射（residual mapping），即 $F(x)$ 部分。

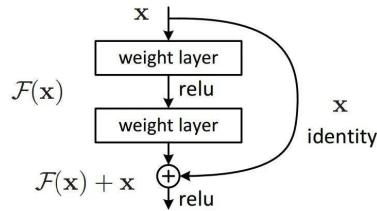


图 8.16 残差学习模块

Resnet 和其他网络结构在 ImageNet ILSVRC 测试集上测试错误率如图 8.17 所示，随着网络深度的增加，Resnet 网络取得了更加好的效果，152 层时取得最优结果。Resnet 的残差学习是如何解决退化问题？Resnet 的作者认为残差映射比原始映射更容易优化。当网络达到最优时，残差映射会趋向于 0，此时残差学习模块仅仅做了恒等映射，网络性能不会下降，然而实际上残差不会为 0，这会使得残差学习模块能学习到新的特征，从而拥有更好的性能，即网络的性能随着深度增加而增加。

method	top-1 err.	top-5 err.
VGG (ILSVRC'14)	-	8.43 [†]
GoogLeNet (ILSVRC'14)	-	7.89
VGG (v5)	24.4	7.1
PReLU-net	21.59	5.71
BN-inception	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	19.38	4.49

图 8.17 Resnet 和其他网络结构在 ImageNet ILSVRC 测试集上错误率（结果越小越好）。

受 VGGNet 启发，Resnet 中卷基层主要采用 3×3 的卷积核，并遵循如下两个设计原则：(1) 相同尺寸的特征图输出层之间具有相同的卷积核数量；(2) 每一层具有相同的时间复杂度，即特征图尺寸减半，卷积核数量翻倍。相比于 VGGNet，Resnet 的卷积核更少，复杂度更低，34 层的 Resnet 网络包含 360 万次乘加操作，仅是 VGG-19 的操作量的 18%。VGG-19 与 34 层普通网络及 Resnet-34 的网络结构图如 8.18 所示，Resnet 中存在 2 种短路连接，即图中的实线曲线和虚线曲线。实线的短路连接的部分的输入和输出具有相同的尺寸和通道数，因此恒等映射和残差映射采用直接相加的计算方式： $y=F(x)+x$ 。虚线

的短路连接的部分的输入和输出具有不同的尺寸和通道数，即恒等映射和残差映射具有不同的尺寸和通道数，无法直接相加，因此通过步长为 2 的 1x1 卷积核改变恒等映射的尺寸和通道数，最终的计算方式为： $y=F(x)+Wx$ ，其中 W 是卷积操作，用来调整 x 的尺寸和通道数。

后续许多优秀的研究人员不断提出新的卷积网络结构，如 ResneXt、DenseNet、DPN、EfficientNet 等，这些网络结构都具有优秀的表达能力，不断刷新各种竞赛结果。从 AlexNet 到 Resnet，再到 DPN 深度学习模型的性能在不断提升，各种卷积层，归一化层，优化方式等也在不断的丰富，在这里就不再一一介绍。接下来将介绍一些简单实用的深度学习训练技巧。

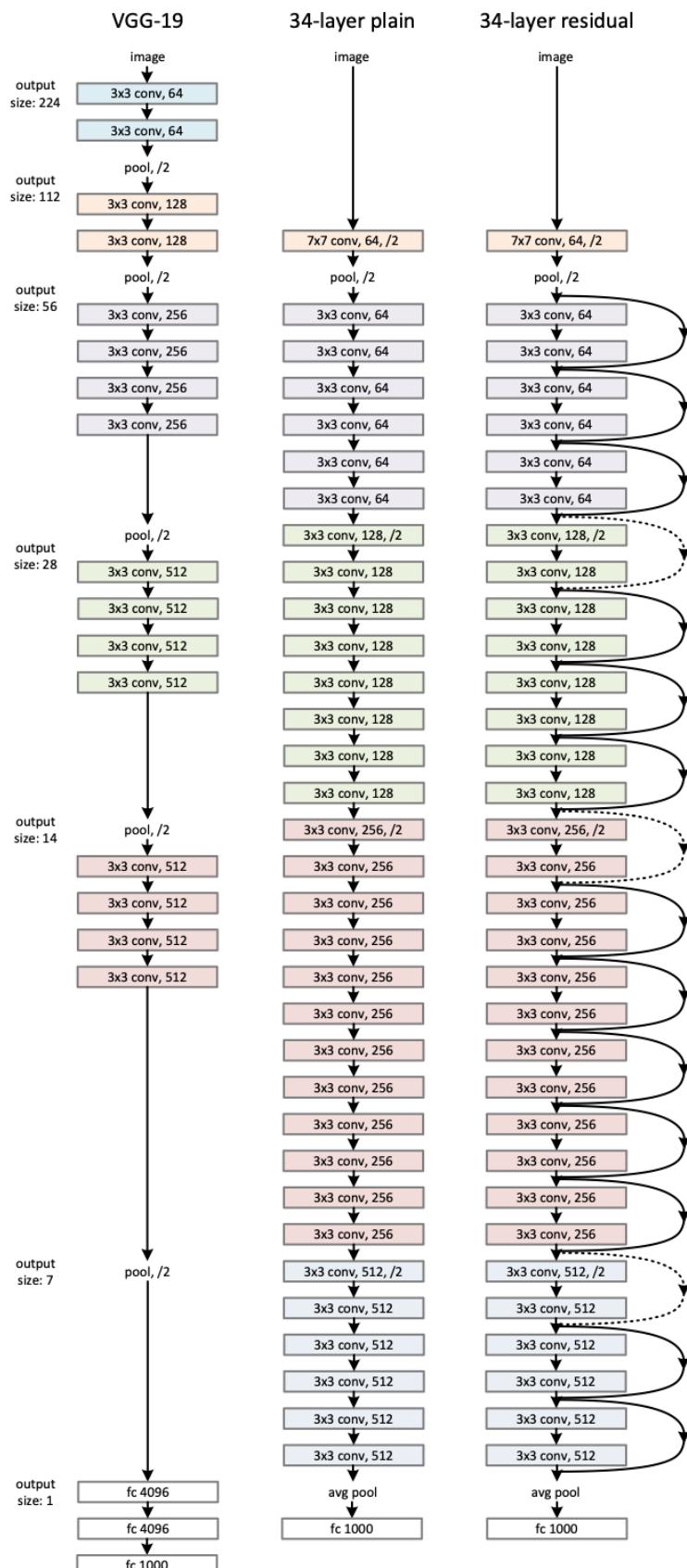


图 8.18 VGG-19 与 34 层普通网络及 Resnet-34 的对比

4、过拟合

深度学习模型的过拟合是指模型在训练数据和测试数据上的表现差距很大，即模型在训练集上表现很好，但在测试集上却表现很差。深度学习网络通过大量的可训练参数是网络具有很强的表示能力，在很多领域都取得了突破性的进展，但是众多的参数也带来了深度学习网络更容易过拟合，导致模型的泛化能力很差。

深度学习模型是否过拟合会很大程度上影响模型性能的好坏，然而准确有效的检测模型是否过拟合是比较困难的，并且深度学习模型通常需要很长的训练时间，这让试错的成本很大，因此众多研究者总结和发现了一些防止过拟合的方法。接下来我们将简单介绍下这些策略。

数据增强 很多时候，训练数据的规模和丰富程度决定了模型性能的好坏，因为更深、更宽的神经网络，需要更多的训练参赛，需要更多的训练数据才能获得稳定的训练结果。因此，克服过拟合最简单、直接的方法就是增加训练数据。对于训练数据有限的情况，可以通过数据增强技术对现有的数据集进行扩充。通用的数据增强技术包括裁剪、平移、尺度变化、水平翻转、随机光照、色彩变换。

常用的数据增强方式是图像裁剪和水平翻转的组合。例如，在 256x256 的图像随机裁剪尺寸为 224x224 的图像块，再随机对这些图像块进行水平翻转，利用这些图像块训练网络。图像裁剪和水平翻转的组合使得训练数据扩大了 2048 倍。在测试阶段，在输入图片的四角和中心共裁减出 5 个 224x224 的图像块及它们的水平翻转都送入网络进行预测，最后对 10 个输出结果进行平均作为最终的预测结果。

ReLU 激活函数 在上面介绍激活函数时，我们已经提及了 ReLU 激活函数，事实上这是 ReLU 激活函数的首次应用。在此之前，常用的激活函数是 $\tanh(x)$ 或者 $\text{sigmoid}(x)$ ，但是无论是 \tanh 还是 sigmoid 函数，都会在 x 取值很大或者很小时进入饱和区，此时神经元梯度会接近 0，容易造成梯度消失，而非饱和的 ReLU 激活函数能够在一定程度上克服这个问题，如图 8.8 所示。ReLU 及其变种激活函数在实际应用中被广泛采用，因为其既具有非线性的特点，使得信息整合能力大大增强；在一定范围内又具有线性的特点，使其训练简单、快速。

Dropout 层是深度学习网络中最常用的正则化层，即以一定的概率将一个神经元的输出设置为 0，通过忽略一定百分比的神经元数量（通常是一半的神经元），减轻网络的过拟合现象。以这种方式被关闭的神经元在前向、后向传播过程中都不起作用，这样训练得到的网络的鲁棒性会更强，模型的泛化性更强，不会过度依赖某些局部特征。Dropout 的工作示意图如图 8.19 所示，通过 Dropout 操作，会随机减少网络中神经元和连接权重的数量。

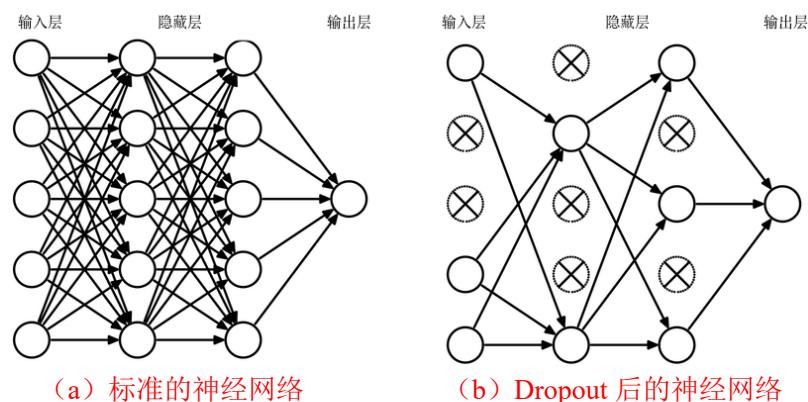


图 8.19 Dropout 示意图

8.6 图像分类常用评价指标

科学的评价指标是衡量网络性能的重要依据，也是对模型进行调整的重要依据。常用的图像分类评价指标如下：准确率、错误率、精确率和召回率、F-score、PR 曲线、ROC 曲线、AUC 等。接下来将一一介绍这些评价指标。

在介绍这些评价指标前，首先介绍几个基本概念：TP(True Positive)、FP(False Positive)、FN(False Negative)、TN(True Negative)。

考虑一个二分类问题，将样本分为真（positive），假（negative）两类，那么 TP、FP、TN、FN 的概念如下：

TN: True Negative, 被预测为负样本，事实上也是负样本，属于正确预测。

TP: True Positive, 被预测为正样本，事实上也是正样本，属于正确预测。

FN: False Negative, 被预测为负样本，但事实上是正样本，属于错误预测。

FP: False Positive, 被预测为正样本，但事实上是负样本，属于错误预测。

从上面概念可以看出 TN+FN 为预测负样本总数，TP+FP 为预测正样本总数，TP+FN 为实际正样本总数，TN+FP 为实际负样本总数，TN+TP 为正确预测的样本总数，FN+FP 为错误预测的总数，如图 8.20 二分类混淆矩阵所示。

		预测类别	
		真	假
真实类别	真	True Positive	False Negative
	假	False Positive	True Negative

图 8.20 二分类混淆矩阵

结合这些基本概念，我们来介绍下这些评价指标。

准确率（Accuracy）：

准确率是指总样本中正确预测样本的比例，计算方式如式 8.35 所示，分子是正确预测的样本数之和，分母是总样本数。准确率一般用来评估模型的全局准确程度，不能包含太多信息，无法全面评价一个模型性能。

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8.35)$$

误分类率：

误分类率是指总样本中错误预测样本的比例，计算方式如式 8.36 所示。误分类率和准确率之和为 1，分别衡量的是错误预测和正确预测的比例。

$$\text{误分类率} = \frac{FP+FN}{TP+TN+FP+FN} = 1 - \text{Accuracy} \quad (8.36)$$

精确率（Precision）：

精确率是指预测为正的样本中真正正样本的比例，计算方式如式 8.37 所示，分子为正确预测的正样本数量，分母为预测正样本总数。精确率衡量的是预测结果，可以反映一个类别的预测正确率。

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8.37)$$

召回率（Recall）：

召回率是指真正的正样本中被正确预测的比例，计算方式如式 8.38 所示，分子为正确预测的正样本数量，分母为实际正样本总数。召回率衡量的是样本中的正例有多少被预测正确。

$$Recall = \frac{TP}{TP+FN} \quad (8.38)$$

精确率和召回率这两个评价指标很容易混淆，它们的唯一区别是计算方式中的分母，一个是预测正样本的总数，一个是实际正样本的总数，这也决定了这两个评价指标关注的是不同的信息：精确率关注的是预测为正样本中实际为真的比例，如果预测的正样本全部为真的正样本，那么精确率的值为 1。召回率关注的是真正的正样本中我们错过了多少，如果所有的正样本都预测出来了，召回率值就为 1。

从计算方式上可以看出，精确率和召回率其中任何一个单独都不足以成为好的指标，用最笨的分类器就可以使上面的单个指标达到 100% 或者 0%。比如说将所有样本都预测为真的系统可以得到 100% 的召回率，此时精确率等于样本中正样本的比例。那么如何利用精确率和召回率去组合出一个合理的评价指标呢？

准确率和召回率是互相影响的，因为如果想要提高准确率就会把预测的置信率阈值调高，此时召回率会降低。一般情况下准确率高、召回率就低，召回率低、准确率就高，如果两者都低，就是网络模型出问题了。

PR 曲线 (Precision x Recall Curve):

PR 曲线即准确率 (Precision) 与召回率 (Recall)，以召回率为坐标 x 轴，准确率为坐标 y 轴，从而画出了一条曲线，如图 8.21 所示，曲线与 $y=x$ 的交点即为以 0.5 作为正负样本概率划分时的召回率和准确率。P-R 图直观地显示出分类系统在样本总体上的召回率和准确率。在进行比较时，若一个分类系统的 P-R 曲线完全被另一个分类系统的曲线完全“包住”，则我们就可以断言后者的性能优于前者。如果一个分类系统的性能比较好，那么它应该有如下的表现：在 Recall 值增长的同时，Precision 的值保持在一个很高的水平。而性能比较差的分类器可能会损失很多 Precision 值才能换来 Recall 值的提高。

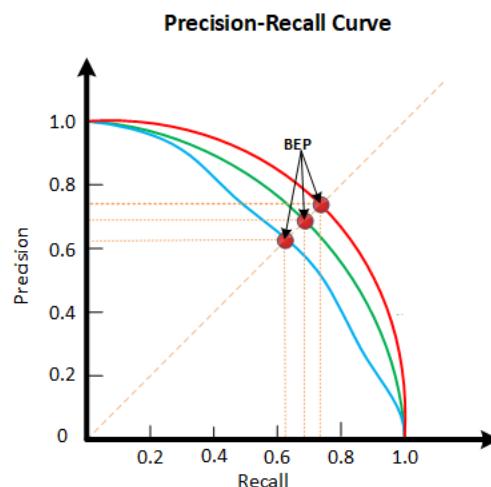


图 8.21 PR 曲线

F-Score:

F-Score 是精确率和召回率加权调和平均，是信息检索领域常用的一个评价标准，常用于评价分类模型的好坏。计算方式如式 8.39 所示，其中 P 表示精确率 (Precision), R 表示召回率 (Recall), β 表示平衡因子，用于平衡精确率和召回率。当 β 为 1 时，精确率和召回率具有相同权重，此时 F-Score 特殊化为 F1-Score，如式 8.40 所示，可以看到，当

Recall 或者 Precision 的值很小的时候, F1 值也将很小, 且不管其中一个值多高, 只要另一个值很小, F1 就会很小, 只有当 Recall 和 Precision 都取得一个高的值时, F1 才会比较高, 因此能对精确率和召回率进行很好的平衡。F 的计算中没有使用到 TN, 而在不均衡样本预测中, TN 很可能就是不被关注的信息, 这也是准确率不可靠的原因。

$$F_{\beta} = \frac{(\beta^2+1)*P*R}{\beta^2*P+R} \quad (8.39)$$

$$F_1 = \frac{2*P*R}{P+R} \quad (8.40)$$

ROC (Receiver Operating Characteristic) 曲线:

对于样本数据, 分类系统会给出每个样本为真的概率, 设定一个置信度阈值, 当某个样本被判断为真的概率大于这个阈值时, 认为该样本为正样本, 小于则为负样本, 然后通过计算我们就可以得到一个(TPR, FPR)对, 即图像上的一个点, 我们通过不断调整这个阈值, 就得到若干个点, 从而画出一条曲线。ROC 曲线的横坐标为假正样本率 (False Positive Rate, FPR), 纵坐标为真正样本率 (True Positive Rate, TPR), 如图 8.22 所示。TPR 和 FPR 的计算方式如式 8.41 所示。

$$FPR = \frac{FP}{FP+TN}, \quad TPR = \frac{TP}{TP+FN} \quad (8.41)$$

AUC (Area Under Curve):

AUC 是指 ROC 曲线下的面积, 显然这个面积小于 1, 又因为 ROC 曲线一般都处于 $y=x$ 这条直线的上方, 所以 AUC 一般在 0.5 到 1 之间。使用 AUC 值作为评价标准是因为很多时候 ROC 曲线并不能清晰的说明哪个分类器的效果更好, 而作为一个数值, 对应 AUC 更大的分类器效果更好。

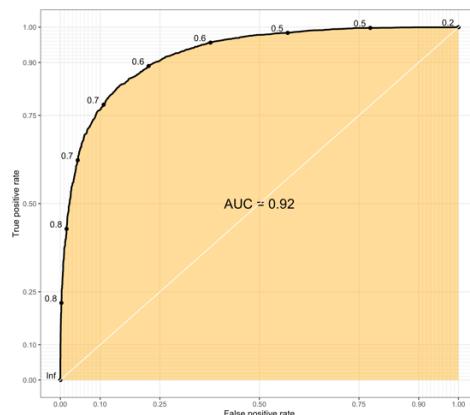


图 8.22 ROC 曲线和 AUC 示例图

对于图 8.22 中的 ROC 曲线有 4 个关键点:

- 1) 点 $(0, 1)$, 即即 $FPR=0, TPR=1$, 这意味着 $FN=0$, 并且 $FP=0$ 。这是完美的分类系统, 它将所有的样本都正确分类。
- 2) 点 $(1, 0)$, 即 $FPR=1, TPR=0$, 类似地分析可以发现这是一个最糟糕的分类系统, 因为它成功避开了所有的正确答案。
- 3) 点 $(0, 0)$, 即 $FPR=TPR=0$, 即 $FP=TP=0$, 可以发现该分类系统预测所有的样本都为负样本 (negative)。
- 4) 点 $(1, 1)$, 即 $FPR=TPR=1$, 即 $FN=TN=0$, 分类系统实际上预测所有的样本都为正样本。

经过以上的分析, ROC 曲线越接近左上角, 该分类系统的性能越好。

从 AUC 判断分类系统（预测模型）优劣的标准：

AUC = 1，是完美分类系统，采用这个预测模型时，存在至少一个阈值能得出完美预测。绝大多数预测的场合，不存在完美分类器。

$0.5 < \text{AUC} < 1$ ，优于随机猜测。这个分类系统（模型）妥善设定阈值的话，能有预测价值。

AUC = 0.5，跟随机猜测一样（例：丢铜板），模型没有预测价值。

AUC < 0.5，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。

在这里以二分类为例介绍了常用的一些分类评价指标，这些评价指标都可以推广到多分类任务。在面对不同的场景和目的，可以选择合适的评价指标。

8.7 常用图像分类数据库

MNIST 手写数字识别数据集

MNIST 数据集(Mixed National Institute of Standards and Technology database)是美国国家标准与技术研究院收集整理的大型手写数字数据库。MNIST 数据集是由 0-9 手写数字图片和数字标签所组成的，由 60000 个训练样本和 10000 个测试样本组成，每个样本都是一张 $28 * 28$ 像素的灰度手写数字图片。如下图 8.23 所示。



图 8.23 MNIST 数据示例

CIFAR10/CIFAR100

CIFAR-10 和 CIFAR-100 是 8000 万个微小图像数据集的带标签的子集。他们由 Alex Krizhevsky, Vinod Nair 和 Geoffrey Hinton 收集。

CIFAR-10 数据集由 10 个类的 60000 个 32×32 彩色图像组成，每个类有 6000 个图像。有 50000 个训练图像和 10000 个测试图像。如图 8.24 所示，CIFAR-10 中的 10 个类别以及来自每个类的 10 个随机图像。

CIFAR-100 数据集有 100 个类，每个类包含 600 个图像。每类各有 500 个训练图像和 100 个测试图像。CIFAR-100 中的 100 个类被分成 20 个超类。每个图像都带有一个“精细”标签（它所属的类）和一个“粗糙”标签（它所属的超类）。CIFAR-100 中的类别列表，如图 8.25 所示。

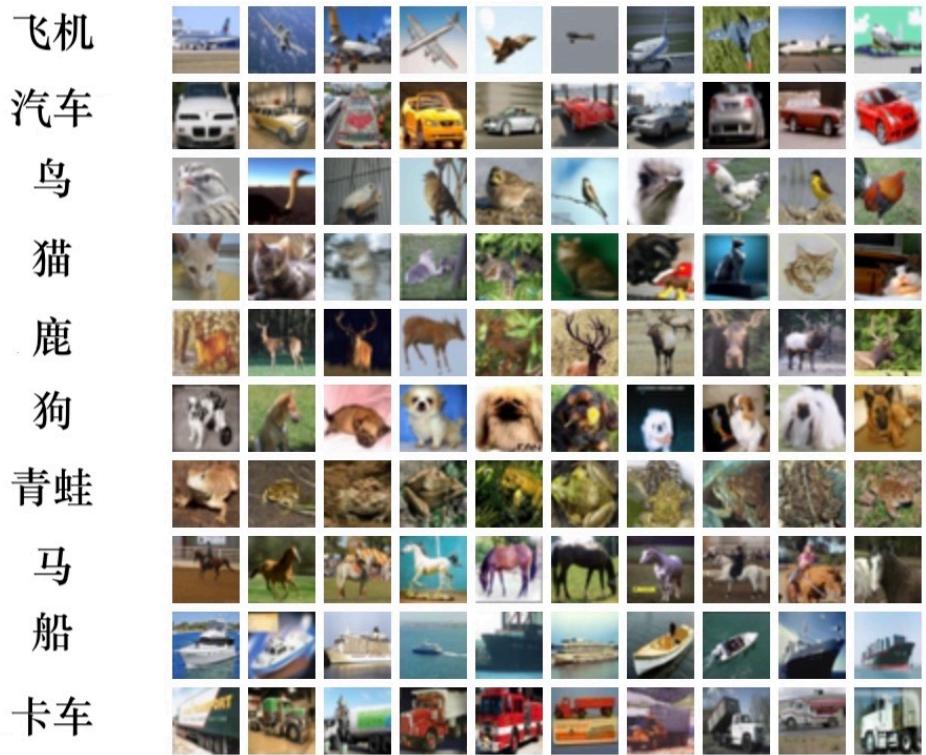


图 8.24 CIFAR-10 数据库示例

超类	类别
水生哺乳动物	海狸, 海豚, 水獭, 海豹, 鲸鱼
鱼	水族馆的鱼, 比目鱼, 射线, 鲨鱼, 鳕鱼
花卉	兰花, 罂粟花, 玫瑰, 向日葵, 郁金香
食品容器	瓶子, 碗, 罐子, 杯子, 盘子
水果和蔬菜	苹果, 蘑菇, 橘子, 梨, 甜椒
家用电器	时钟, 电脑键盘, 台灯, 电话机, 电视机
家用家具	床, 椅子, 沙发, 桌子, 衣柜
昆虫	蜜蜂, 甲虫, 蝴蝶, 毛虫, 蟑螂
大型食肉动物	熊, 豹, 狮子, 老虎, 狼
大型人造户外用品	桥, 城堡, 房子, 路, 摩天大楼
大自然的户外场景	云, 森林, 山, 平原, 海
大杂食动物和食草动物	骆驼, 牛, 黑猩猩, 大象, 袋鼠
中型哺乳动物	狐狸, 豪猪, 负鼠, 洗熊, 臭鼬
非昆虫无脊椎动物	螃蟹, 龙虾, 蜗牛, 蜘蛛, 蠕虫
人	宝贝, 男孩, 女孩, 男人, 女人
爬行动物	鳄鱼, 恐龙, 蜥蜴, 蛇, 乌龟
小型哺乳动物	仓鼠, 老鼠, 兔子, 母老虎, 松鼠
树木	枫树, 橡树, 棕榈, 松树, 柳树
车辆1	自行车, 公共汽车, 摩托车, 皮卡车, 火车
车辆2	割草机, 火箭, 有轨电车, 坦克, 拖拉机

图 8.25 CIFAR-100 类别示意图

Caltech101/Caltech256

Caltech101/Caltech256 数据集是加利福尼亚理工学院收集整理的数据集，该数据集选自 Google Image 数据集，并手工删除了不符合其类别的图片。

Caltech101 数据集包含了 101 类的图像，每类大约有 40~800 张图像，大部分是 50 张/类，每张图像的大小大约是 300x200，一共含有 9145 张图片。在 2003 年 9 月由 Fei-Fei Li, Marco Andreetto 和 Marc 'Aurelio Ranzato 收集。

Caltech 256 数据集是 Caltech101 数据集的改进版图片数据集，该数据集收集了 256 个类的 30607 张图片。有几处改进：

- 1、类别数量增加一倍以上；
- 2、类别中图像的最小数量从 31 增加到 80；
- 3、避免因图像旋转造成的伪影；



a) Caltech 101



b) Caltech 256

图 8.26 Caltech 101 和 Caltech 256 数据库示例

ImageNet

ImageNet 数据集是一个计算机视觉数据集，是由斯坦福大学的李飞飞教授带领创建。该数据集包含 1400 多万幅图片，涵盖 2 万多个类别。ImageNet 数据集一直是评估图像分类算法性能的基准。

ImageNet 数据集是为了促进计算机图像识别技术的发展而设立的一个大型图像数据集。ImageNet 数据集中已经超过千万张图片，每一张图片都被手工标定好类别。ImageNet 数据集中的图片涵盖了大部分生活中会看到的图片类别。如图图 8.27 所示，它包含了各种各样的图像，并且每张图像都被关联了标签（类别名）。

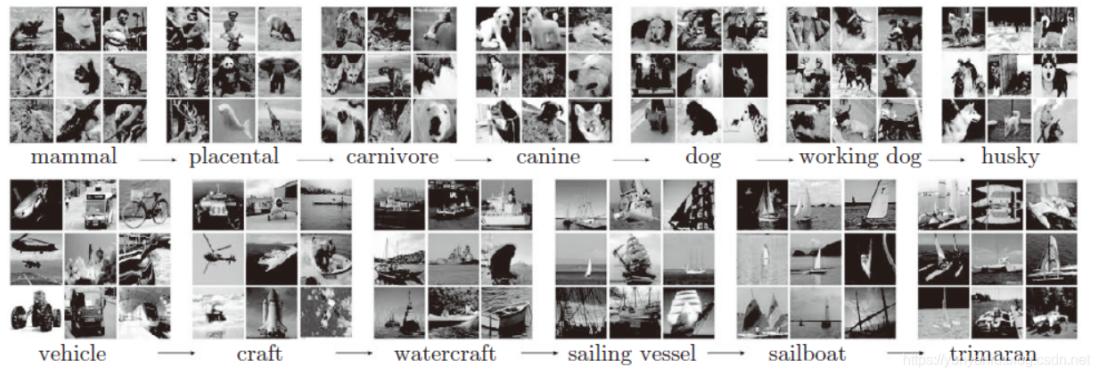


图 8.27 ImageNet 数据集示例

从 2010 年起，每年 ImageNet 的项目组织都会举办一场基于 ImageNet 的大规模视觉识别竞赛（ImageNet Large Scale Visual Recognition Challenge , ILSVRC）。ILSVRC 使用 ImageNet 的一个子集，含有 1000 个类别，总共有大约 120 万个训练图像，50,000 个验证图像，以及 150,000 个测试图像。在 ILSVRC 竞赛中诞生了许多成功的图像识别方法，其中很多是深度学习方法，比如前面介绍的 AlexNet, VGGNet, GoogleNet, ResNet 等，它们在赛后又会得到进一步发展与应用。可以说，ImageNet 数据集和 ILSVRC 竞赛大大促进了计算机视觉技术，乃至深度学习的发展，在深度学习的浪潮中占有举足轻重的地位。

Quick Draw

Quick Draw 数据集是由 Google 的 Magenta 团队发布的一个包含 5000 万张人工手绘图的数据集，包含 345 个类别，这些绘图画都来自于 Quick, Draw! 游戏的玩家。这些手绘画都是加了时间戳的矢量图，并带有一些元数据标注，包括玩家被要求绘画的内容和玩家所在的国家。如图 8.28 所示。手绘图和图像有很多异同点，比如手绘图是人工手绘，由白色背景和黑色线条组成，图像是由彩色像素点构成；手绘图会因人而异，不同的人具有不同的认知和绘画习惯，最后绘画呈现效果区别很大，这也是手绘图需要记录笔画顺序等时间戳信息和绘画者地区的原因，因为手绘图也可以用来分析人的行为和心理；手绘图具有一定的形变和抽象性。对于手绘图识别即草图分类任务，一种做法是将手绘图作为图像，将矢量图转为图片，利用卷积神经网络对草图进行分类。



图 8.28 Quick Draw 数据库示例

习题:

1. 图像识别方法有哪些, 请各举一例。
2. 统计图像识别的步骤有哪些?
3. 试比较线性分类器和贝叶斯分类器。
4. 什么线性分类器的学习?
5. 试述统计图像识别与结构图像识别的区别与联系。
6. 卷积神经网络的好处有哪些?
7. 残差学习模块由哪两部分组成?
8. 应该用哪个评价指标实现“宁缺毋滥”的目的? 哪个评价指标适合“一个都不能少”的目的? 为什么?