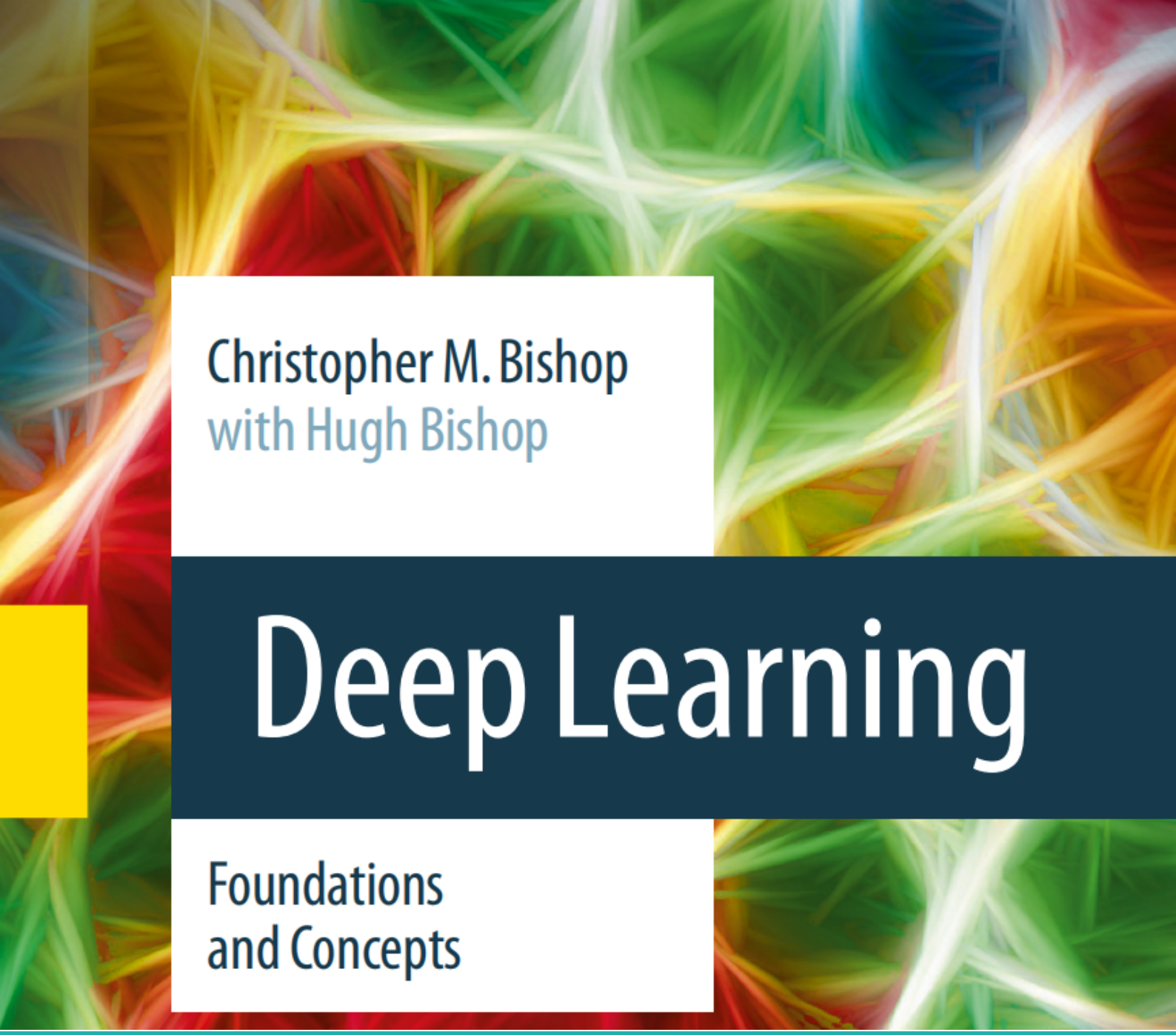


Deep Learning 学习笔记 Ch01&02

作者：东八区的鼠鼠
组织：交大門
时间：July 9, 2024
版本：0.1
用书：Deep Learning Foundations and Concepts (Christopher M. Bishop, Hugh Bishop)

目录

第一章 深度学习变革 **1**

1.1 深度学习带来的冲击 . . . 1

1.2 一个示例 . . . 1

1.2.1 合成数据 (Synthetic data) . . . 1

1.2.2 线性模型 . . . 1

1.2.3 误差函数 (Error function) . . . 1

1.2.4 模型复杂度 (Model complexity) . . . 1

1.2.5 正则化 (Regularization) . . . 3

1.2.6 模型选择 (Model selection) . . . 4

1.3 基于神经网络的机器学习方法发展简史 . . . 4

1.3.1 单层神经网络 (Single-layer networks) . . . 5

1.3.2 反向传播算法 (Backpropagation) . . . 6

1.3.3 深度网络 (Deep networks) . . . 6

第二章 概率论 (Probabilities) **8**

2.1 概率论基础 . . . 8

2.1.1 加法法则与乘法法则 . . . 8

2.1.2 贝叶斯定理 (Bayes' theorem) . . . 9

2.1.3 贝叶斯定理在医学筛查中的应用 . . . 9

2.1.4 先验 (Prior) 概率与后验 (Posterior) 概率 . . . 10

2.1.5 独立变量 . . . 10

2.2 概率密度 (Probability Densities) . . . 11

2.2.1 期望 (Expectation) 与协方差 (Covariance) . . . 11

2.3 高斯分布 (Gaussion Distribution) . . . 12

2.3.1 似然函数 (Likelihood function) . . . 12

2.3.2 最大似然估计的有偏性 . . . 13

2.3.3 线性回归 . . . 14

2.3.4 密度函数变换 . . . 14

2.4 信息论 (Information Theory) . . . 14

2.4.1 熵 (Entropy) . . . 14

2.4.2 物理学的角度 . . . 15

2.4.3 微分熵 (Differential entropy) . . . 15

2.4.4 最大熵 (Maximum entropy) . . . 15

2.4.5 相对熵 (Relative entropy) . . . 16

2.4.6 条件熵 (Conditional entropy) . . . 17

2.4.7 互信息 (Mutual information) . . . 17

2.5 贝叶斯概率 (Bayesian Probability) . . . 17

2.5.1 模型参数 . . . 18

2.5.2 正则化 . . . 18

2.5.3 贝叶斯机器学习 . . . 18

2.6 课后题选做 . . . 19

第一章 深度学习变革

1.1 深度学习带来的冲击

- 医学诊断：皮肤癌恶性瘤与良性瘤分类问题（监督学习 (supervised learning)+ 迁移学习 (transfer learning)）
- 蛋白质结构：计算蛋白质的三维结构（监督学习 + 预测模型）
- 图像合成：（无监督学习 (unsupervised learning)+ 生成模型 (generative model)）
- 大语言模型 (LLM)：自监督学习 (self-supervised learning)

1.2 一个示例

以下是多项式拟合的示例（假设都是实值）

1.2.1 合成数据 (Synthetic data)

设 x = 输入 (input) 变量，t = 目标 (target) 变量，设两变量在实轴上取连续值，训练集中含有观测值 $x_1, x_2, \cdots, x_N$，其对应的 t 值为 $t_1, t_2, \cdots, t_N$，用一个数据集之外的新的 x 来预测 t，称为是泛化 (generalization)。

对 $[0, 1]$ 区间上均匀分布的 $x_1, \cdots, x_N$，先计算 $sin(2\pi x_i)$，再加上小水平的随机噪声得到对应的 t_i。（在现实中的数据总有各种随机噪声，主要矛盾是找出其潜在规律性(underlying regularity)）

1.2.2 线性模型

目标是利用训练集来预测 $\hat{x}$ 所对应的 $\hat{t}$，隐式地找出函数 $sin(2\pi x)$ 非常困难，先采用如下多项式拟合：

$$y(x, \mathbf{w}) = w_0 + w_1 x + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

M 称为多项式的阶 (order)，上式是关于 $\mathbf{w}$ 的线性函数。

1.2.3 误差函数 (Error function)

误差函数用于表征预测函数与真实数量关系不相合的程度，系数的值 $\mathbf{w}$ 可以通过极小化误差函数来得到：

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

其中系数取为 $\frac{1}{2}$ 是为了方便。上式非负，为零 $\Longleftrightarrow$ 曲线穿过每一个数据点。由于上式为一个关于 w 的二次函数，可以解得极小点 w^*，得到函数 $y(x, \mathbf{w}^*)$。

1.2.4 模型复杂度 (Model complexity)

采用阶 $M = 0, 1, 3, 9$ 拟合所得的函数图像如下，$M = 0, 1$ 较差，$M = 3$ 最好，$M = 9$ 过拟合 (over-fitting)。

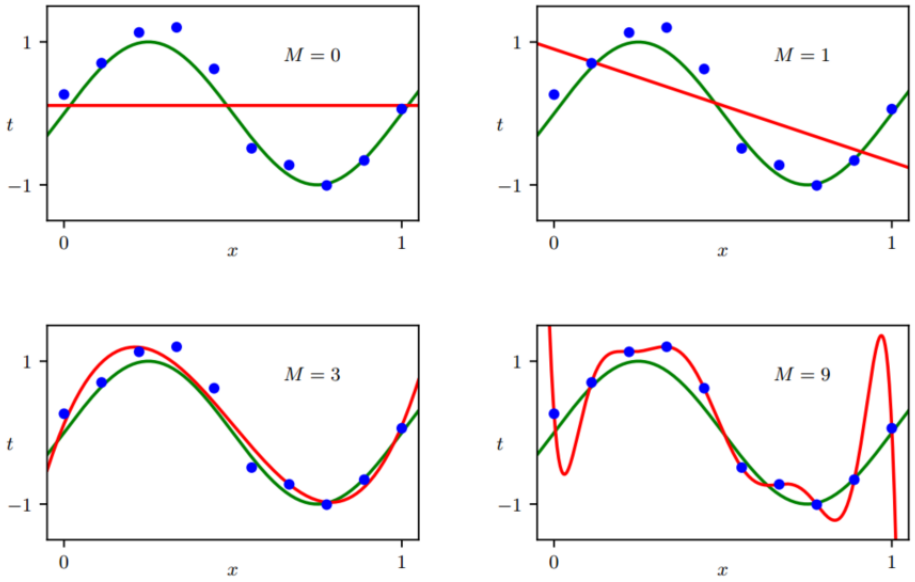
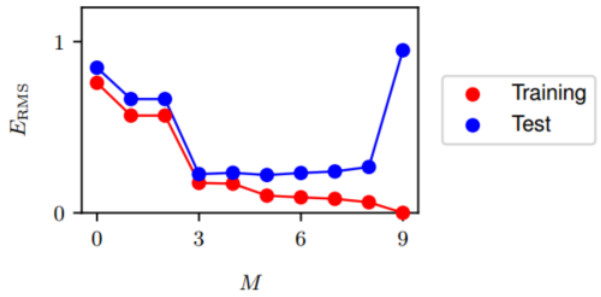


Figure 1.6 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.4 by minimizing the error function (1.2).

使用测试集 (test set) 来定量地描述泛化性对 M 的依赖程度，引入如下均方根 (RMS:root-mean-square) 误差函数：

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2}$$

Figure 1.7 Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set, and on an independent test set, for various values of M.



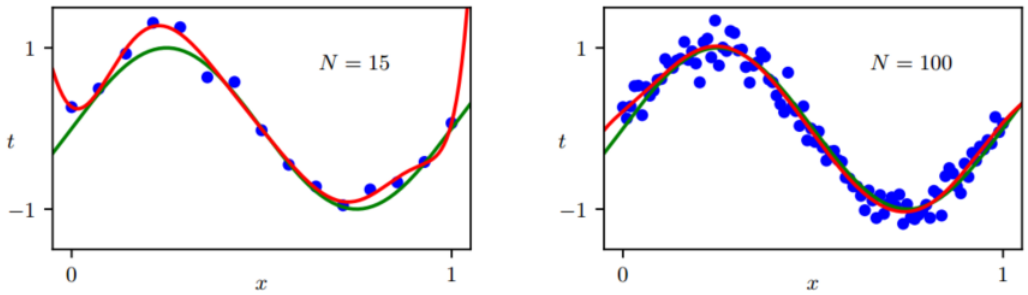


Figure 1.8 Plots of the solutions obtained by minimizing the sum-of-squares error function (1.2) using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

上图可以看出：数据点数量应不少于模型中可学习参数 (5-10 倍)，增大数据集可以有效降低过拟合问题。

1.2.5 正则化 (Regularization)

需要根据问题的复杂程度来选择模型的复杂程度，正则化技术可以有效防止过拟合现象。

正则化技术：在误差函数中添加惩罚项 (penalty term), 防止系数 (coefficients) 出现较大幅度变化。系数 w_0 通常从正则化器 (regularizer) 中省略（否则可能导致得到的模型被初值干扰）；也可以留下，但是自己单独使用一个正则化系数 (regularization coefficient)。

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

其中 $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = w_0^2 + w_1^2 + \cdots + w_n^2$，$\lambda$ 决定了正则项与平方和误差项的相对重要性。

上式中的误差函数可以以封闭形式 (closed-form) 最小化，这些技术称为收缩方法 (shrinkage methods)，因为这些方法减少了多项式系数的值。在神经网络中，这种方法称为权重衰减(weight decay)

Table 1.1 Table of the coefficients $\mathbf{w}^\star$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.11	0.90	0.12	0.26
$w_1^\star$		−1.58	11.20	−66.13
$w_2^\star$			−33.67	1,665.69
$w_3^\star$			22.43	−15,566.61
$w_4^\star$				76,321.23
$w_5^\star$				−217,389.15
$w_6^\star$				370,626.48
$w_7^\star$				−372,051.47
$w_8^\star$				202,540.70
$w_9^\star$				−46,080.94

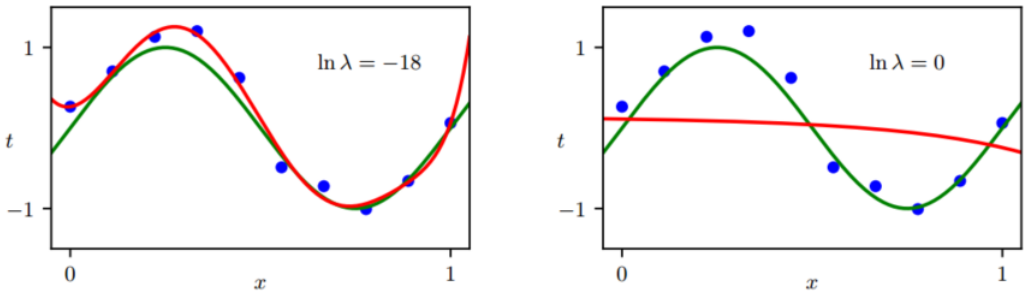
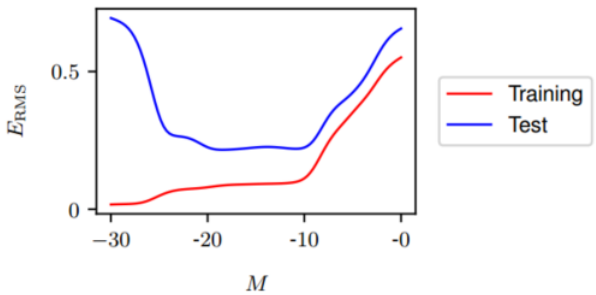


Figure 1.9 Plots of $M = 9$ polynomials fitted to the data set shown in Figure 1.4 using the regularized error function (1.4) for two values of the regularization parameter λ corresponding to $\ln\lambda = -18$ and $\ln\lambda = 0$. The case of no regularizer, i.e., $\lambda = 0$, corresponding to $\ln\lambda = -\infty$, is shown at the bottom right of Figure 1.6.

由下图，λ 控制着模型的有效复杂度 (effective complexity)，因此决定了过拟合的程度。

Figure 1.10 Graph of the root-mean-square error (1.3) versus $\ln\lambda$ for the $M = 9$ polynomial.



1.2.6 模型选择 (Model selection)

λ 是一个超参数 (hyperparameter)，在极小化误差函数以确定模型参数 w 的过程中，其值是固定的，所以不能简单地通过联合极小化误差函数来确定 λ 的值，因为这将得到一个 $\lambda \to 0$ 和训练误差很小或为 0 的过拟合模型。同样，多项式的阶 M 也是模型的一个超参数，简单地针对 M 来优化训练集误差将导致 M 过大以及过拟合问题。

因此，我们需要找到一种方法来确定超参数的合适值。上面的结果建议了一种实现这一目标的简单方法，即获取可用数据并将其划分为一个训练集 (用于确定系数 $\mathbf{w}$) 和一个单独的验证集 (validation set)（也称为留出集 (hold-out set 留出法得到的集合) 或开发集 (development set)）。

Table 1.2 Table of the coefficients $\mathbf{w}^\star$ for $M = 9$ polynomials with various values for the regularization parameter λ. Note that $\ln\lambda = -\infty$ corresponds to a model with no regularization, i.e., to the graph at the bottom right in Figure 1.6. We see that, as the value of λ increases, the magnitude of a typical coefficient gets smaller.

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.26	0.26	0.11
$w_1^\star$	−66.13	0.64	−0.07
$w_2^\star$	1,665.69	43.68	−0.09
$w_3^\star$	−15,566.61	−144.00	−0.07
$w_4^\star$	76,321.23	57.90	−0.05
$w_5^\star$	−217,389.15	117.36	−0.04
$w_6^\star$	370,626.48	9.87	−0.02
$w_7^\star$	−372,051.47	−90.02	−0.01
$w_8^\star$	202,540.70	−70.90	−0.01
$w_9^\star$	−46,080.94	75.26	0.00

然后我们选择在验证集上误差最小的模型（如果使用有限大小的数据集多次迭代模型设计，那么可能会出现一些对验证数据的过拟合，因此可能有必要保留第三个测试集，以便最终评估所选模型的性能）。

在一些场景，用于训练和测试的数据供应将是有限的。为了建立一个好的模型，我们应该使用尽可能多的可用数据进行训练。然而，如果验证集太小，它将给出一个相对嘈杂的预测性能估计。使用交叉验证(cross-validation)可以有效解决这个问题。

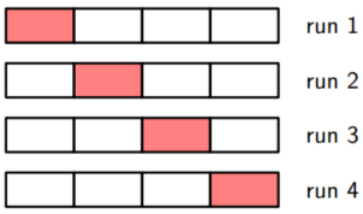


Figure 1.11 The technique of S-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into S groups of equal size. Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure is then repeated for all S possible choices for the held-out group, indicated here by the red blocks, and the performance scores from the S runs are then averaged.

如上图，采用K 折交叉验证(k-fold cross validation)，将原数据集均分为 K 组，使用其中 $K-1$ 组用于训练模型，用剩下的一组作为测试集评估模型，将此过程进行 K 次（K 组轮流作为测试集），将 K 次运行所得结果取平均作为最终结果。特别地，当数据集过小时，可以采用留一法(leave-one out technique)，即留出一个数据点作为测试集。

交叉验证的缺点：计算成本过大，在较差情形下会关于超参数数量呈指数级增长；只能在较小范围内探索超参数取值，严重依赖通过较小模型以及启发式 (heuristics) 算法获得的经验。

1.3 基于神经网络的机器学习方法发展简史

通过模拟神经元的特性，心理学家 W.S.McCulloch 和数理逻辑学家 W.Pitts 建立了人工神经网络 (artificial neural networks, ANNs)：使用其他神经元输出的线性组合来描述单个神经元特性，再使用非线性函数进行转化。

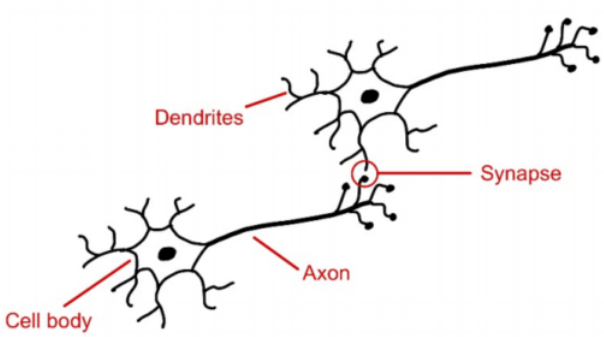


Figure 1.12 Schematic illustration showing two neurons from the human brain. These electrically active cells communicate through junctions called synapses whose strengths change as the network learns.

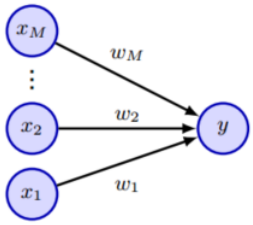


Figure 1.13 A simple neural network diagram representing the transformations (1.5) and (1.6) describing a single neuron. The polynomial function (1.1) can be seen as a special case of this model.

写成数学形式即为：

$$a = \sum_{i=1}^{M} w_i x_i$$

$$y = f(a)$$

其中 $x_1, x_2, \cdots, x_M$ 表示 M 个输入，$w_1, w_2, \cdots, w_M$ 表示 M 个权重（用于模拟相关突触信号强度）；a 称为预激活(pre-activation)，非线性函数 $f(\cdot)$ 称为激活函数(activation function)，输出 y 称为激活函数值 (activation)，上图即是其图表形式。

1.3.1 单层神经网络 (Single-layer networks)

人工神经网络发展按照网络的复杂程度 (sophistication) 以及处理层 (layers) 的数量大致分为三个阶段。

Figure 1.14 Illustration of the Mark 1 perceptron hardware. The photograph on the left shows how the inputs were obtained using a simple camera system in which an input scene, in this case a printed character, was illuminated by powerful lights, and an image focused onto a 20×20 array of cadmium sulphide photocells, giving a primitive 400-pixel image. The perceptron also had a patch board, shown in the middle photograph, which allowed different configurations of input features to be tried. Often these were wired up at random to demonstrate the ability of the perceptron to learn without the need for precise wiring, in contrast to a modern digital computer. The photograph on the right shows one of the racks of learnable weights. Each weight was implemented using a rotary variable resistor, also called a potentiometer, driven by an electric motor thereby allowing the value of the weight to be adjusted automatically by the learning algorithm.

图 1.1: 上图即是 Mark 1 感知机硬件

单层神经网络的代表模型是感知机 (perceptron)，其采用的激活函数为：

$$f(a) = \begin{cases} 0, & \text{if a} \leqslant 0 \\ 1, & \text{if a} > 0 \end{cases}$$

虽然典型的感知机具有多层处理数据的结构，但是只有一层能够从数据中学习，因此被认为是“单层”的。现代神经网络有时也被称为多层感知机(multilayer perceptron, MLP)。

1.3.2 反向传播算法 (Backpropagation)

具有一层以上可学习参数的神经网络归功于微分 (differential calculus) 以及基于梯度的优化算法的引入，其显著改进有两处：其一，是将激活函数由阶跃函数改进为具有非零梯度的连续可微函数 (continuous differentiable)；其二，是引入了可微的误差函数，用于表征给定参数值在训练集预测目标变量的好坏程度。

Figure 1.15 A neural network having two layers of parameters in which arrows denote the direction of information flow through the network. Each of the hidden units and each of the output units computes a function of the form given by (1.5) and (1.6) in which the activation function $f(\cdot)$ is differentiable.

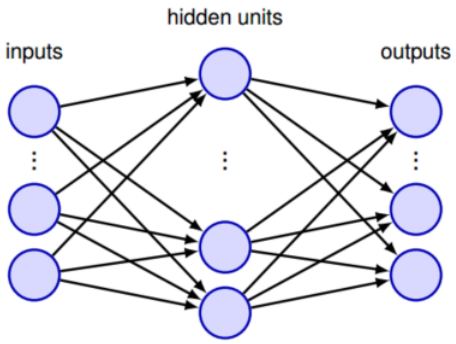


图 1.2: 具有两层参数的神经网络结构：输入层–> 隐藏层–> 输出层

上图结构包含了两个处理层，中间层的节点称为隐藏单元 (hidden units)，因为它们的值不会出现在训练集中，而只是提供输入输出值。上图结构信息是沿箭头再网络中向前流动，于是称为是前馈神经网络(feed-forward neural networks, FNNs)。

为了训练这样的网络，首先要随机数生成器初始化参数，再使用基于梯度的优化算法对参数进行迭代更新。在上述过程中，需要计算误差函数的导数，我们可以采用误差反向传播算法(error backpropagation) 来有效完成此问题。对于基于梯度的优化算法，目前最流行的方法就是随机梯度下降法(stochastic gradient descent, SGD)

按照上述过程可以发现，在多层网络中，只有最后两层的权重能够学习到有用的值。除卷积神经网络 (convolutional neural networks, CNNs) 外，很少有超过两层的应用。为了实现较好的性能，会进行手动设计的预处理 (pre-processing)，有时称作是特征提取(feature extraction)。

1.3.3 深度网络 (Deep networks)

第三阶段，也就是当前的阶段，具有多层权重的神经网络称为深度神经网络 (Deep neural networks, DNNs)，专注于这种网络的机器学习称为深度学习 (Deep learning)。

深度学习诞生的一个重要主题就是（以参数数量来衡量的）神经网络规模的显著增加。

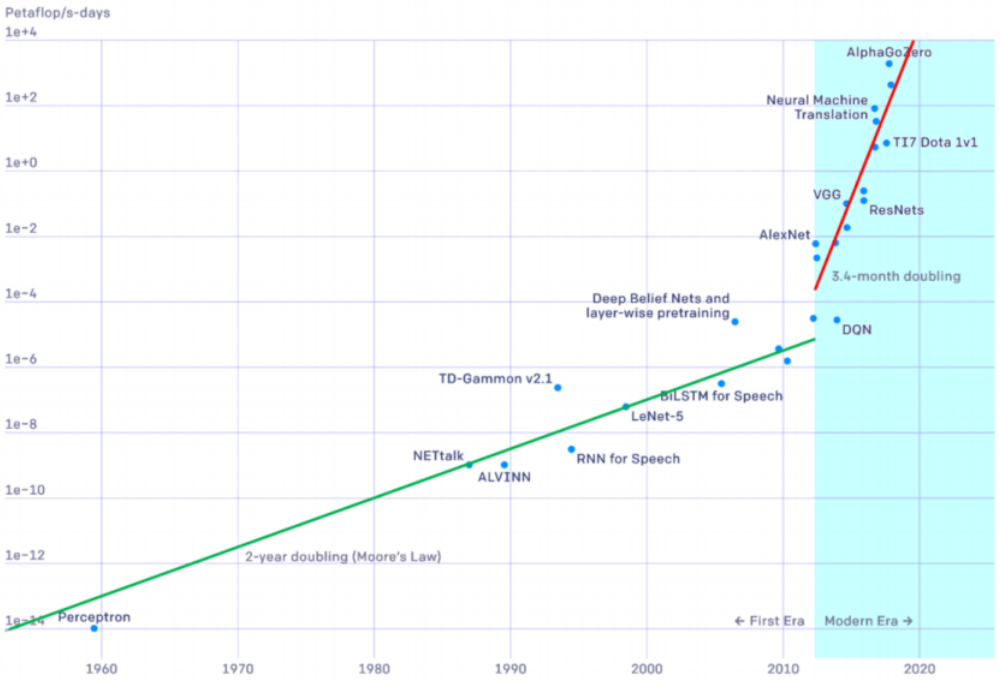


Figure 1.16 Plot of the number of compute cycles, measured in petaflop/s-days, needed to train a state-of-the-art neural network as a function of date, showing two distinct phases of exponential growth. [From OpenAI with permission.]

图 1.3: 神经网络的计算能力随时间增长图，明显分为两个不同的指数增长阶段

深度 (depth) 在神经网络中发挥着重要作用。实际上隐藏层也发挥着重要作用，表征学习 (特征学习, 表示学习, Representation Learning) 就是其作用的体现。它通过将输入数据转换为新的表征，从而为最后一层或几层创建一个更容易解决的问题，这种内部表征可以被重新利用，用于迁移学习。

除了增大数据规模，还有其他的方式来提高深度学习的有效性。例如，在简单神经网络中，训练信号在反向传播时，可能会变得很弱。解决此问题有以下两种方法：其一，引入残差连接(residual connection)，这有助于训练数百层的网络；其二，使用自动微分法(automatic differentiation)，这使得研究人员可以快速对不同结构的神经网络进行试验，并组合不同的结构元素（因为只需要写出相对简单的向前传播函数代码即可）。

第二章 概率论 (Probabilities)

在几乎所有的机器学习应用中，我们都要处理不确定性问题。不确定性分为两类。

第一种是认知不确定性(epistemic uncertainty)，源于希腊语 episteme，意思是知识，用于衡量我们对于正确模型参数的无知程度，有时也被称为系统不确定性(systematic uncertainty)。

第二种是偶然不确定性(aleatoric uncertainty)，这样的误差并不是我们所选用模型带来的，而是因为我们只能观测到世界的部分信息，导致数据本身就存在误差————数据集里的偏差 (bias) 越大，我们的偶然不确定性就越大，因此也称为 intrinsic or stochastic uncertainty，或是称为噪声(noise)。

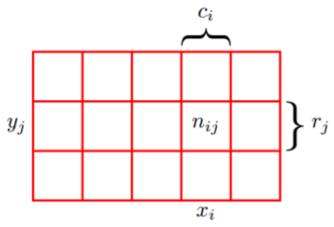
而这两种不确定性都可以通过使用加法规则 (sum rule) 和乘法规则 (product rule) 在概率论的框架下进行处理。

2.1 概率论基础

2.1.1 加法法则与乘法法则

设存在随机变量 X、Y(random variables or stochastic variables)，分别有取值 $x_1, x_2, \cdots, x_L$ 与 $y_1, y_2, \cdots, y_M$，设在 N 次试验中出现 $X = x_i$ 且 $Y = y_j$ 的次数为 n_{ij}，用 c_i 来表示满足 $X = x_i$ 的试验次数，用 r_j 来表示满足 $Y = y_j$ 的试验次数。

2.4 We can derive the sum and product rules of probability by considering a random variable X, which takes the values $\{x_i\}$ where $i = 1, \ldots, L$, and a second random variable Y, which takes the values $\{y_j\}$ where $j = 1, \ldots, M$. In this illustration, we have $L = 5$ and $M = 3$. If we consider the total number N of instances of these variables, then we denote the number of instances where $X = x_i$ and $Y = y_j$ by n_{ij}, which is the number of instances in the corresponding cell of the array. The number of instances in column i, corresponding to $X = x_i$, is denoted by c_i, and the number of instances in row j, corresponding to $Y = y_j$, is denoted by r_j.



则有：

$$p(X = x_i) = \frac{c_i}{N}$$

$$\sum_{i=1}^{L} p(X = x_i) = 1$$

$$\sum_{i=1}^{M} p(Y = y_i) = 1$$

定义联合概率(joint probability) 为：

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

定义边际概率(marginal probability) 为：

$$p(X = x_i) = \sum_{j=1}^{M} p(X = x_i, Y = y_j)$$

在发生 $X = x_i$ 的条件下，发生 $Y = y_j$ 的概率，称为条件概率(conditional probability)：

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

由此得到

定理 2.1 (加法法则与乘法法则)

$$\text{sum rule:} \quad p(X) = \sum_{Y} p(X,Y)$$

$$\text{product rule:} \quad p(X,Y) = p(Y|X)p(X)$$

♡

2.1.2 贝叶斯定理 (Bayes' theorem)

由 $p(X,Y) = p(Y,X)$ 可以得到如下条件概率之间的关系：

定理 2.2 (贝叶斯定理)

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

♡

2.1.3 贝叶斯定理在医学筛查中的应用

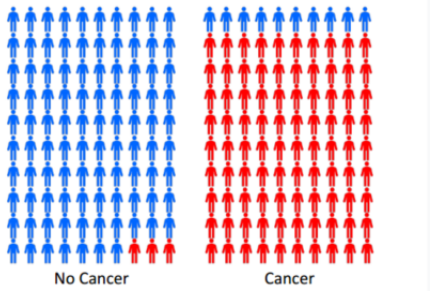


Figure 2.3 Illustration of the accuracy of a cancer test. Out of every hundred people taking the test who do not have cancer, shown on the left, on average three will test positive. For those who have cancer, shown on the right, out of every hundred people taking the test, on average 90 will test positive.

假设人群中 1% 会患有某种癌症。在理想情况下，我们对癌症的检测将对任何患有癌症的人给出阳性结果，而对没有癌症的人给出阴性结果。然而，测试并不是完美的，所以我们假设当测试给没有癌症的人时，3% 的人会检测出阳性。这些被称为假阳性。同样，当对患有癌症的人进行测试时，10% 的人会呈阴性。这些被称为假阴性。有了这些信息，我们可以使用贝叶斯定理来解决以下问题：

(1) 如果我们对人群进行筛查，某人检测呈阳性的概率是多少?

(2) 如果一个人的检测结果是阳性的，那么他确实得了癌症的概率是多少?

我们用变量 C 来表示癌症的存在或不存在:$C = 0$ 表示“无”，$C = 1$ 表示“有”；引入第二个随机变量 T 来表示筛选测试的结果，其中 $T = 1$ 表示测试阳性，$T = 0$ 表示测试阴性，则有：

$$p(C = 1) = \frac{1}{100}$$

$$p(C = 0) = \frac{99}{100}$$

$$p(C = 0) + p(C = 1) = 1$$

我们知道对于癌症患者，检测结果为阳性的概率为 90%，而对于未患癌症的人，检测结果为阳性的概率为 3%。因此，我们可以写出所有条件概率:

$$p(T=1|C=1)=\frac{90}{100}$$
$$p(T=0|C=1)=\frac{10}{100}$$
$$p(T=1|C=0)=\frac{3}{100}$$
$$p(T=0|C=0)=\frac{97}{100}$$

我们现在可以解决第一个问题，评估随机接受测试的人得到阳性测试结果的总体概率：

$$\begin{aligned}p(T=1)&=p(T=1|C=0)p(C=0)+p(T=1|C=1)p(C=1)\\&=\frac{3}{100}\times\frac{99}{100}+\frac{90}{100}\times\frac{1}{100}=\frac{387}{10000}=0.0387\\p(T=0)&=1\frac{387}{10000}=0.9613\end{aligned}$$

因此，如果一个人随机接受测试，那么大约有 4% 的几率测试结果是阳性的，尽管他们实际上有 1% 的几率患有癌症；大约有 96% 的几率其测试是阴性。

现在考虑第二个问题，这是一个条件概率：

$$\begin{aligned}p(C=1|T=1)&=\frac{p(T=1|C=1)p(C=1)}{p(T=1)}\\&=\frac{99}{100}\times\frac{1}{100}\times\frac{10000}{387}=\frac{90}{387}\approx 0.23\end{aligned}$$

因此他有 23% 的概率确实得了癌症。

2.1.4 先验 (Prior) 概率与后验 (Posterior) 概率

如果在接受检查之前，有人问我们某人是否有可能患癌症，那么我们所能得到的最完整的信息就是概率 $p(C)$，我们称其为先验概率(prior probability)，因为它是在我们观察到测试结果之前可用的概率。一旦我们被告知这个人接受了阳性测试的结果，我们就可以用贝叶斯定理来计算概率 $p(C|T)$，我们称之为后验概率(posterior probability)，因为它是我们观察到测试结果 T 后得到的概率。

在上述例子中，患癌症的先验概率是 1%。然而，一旦我们观察到测试结果是阳性的，我们发现癌症的后验概率是 23%。这是癌症的概率大大提高，正如我们直观地预期的那样。然而，我们注意到，一个测试呈阳性的人实际上患有癌症的可能性仍然只有 23%。虽然测试提供了癌症的有力证据，但这必须与使用贝叶斯定理的先验概率相结合，才能得到正确的后验概率。

2.1.5 独立变量

定义 2.1 (Independent variables)

对于随机变量 X,Y，若有 $p(X,Y)=p(X)p(Y)$，则称 X,Y 独立。 ♣

2.2 概率密度 (Probability Densities)

对于连续型随机变量 X，将其概率密度函数 (probability density function) 记为 $p(x)$，将其累积分布函数 (cumulative distribution function) 记为 $P(z)$，于是有：

$$p\left(x \in (a,b)\right) = \int_a^b p(x)dx$$

$$p(x) \geqslant 0$$

$$\int_{-\infty}^{\infty} p(x)dx = 1$$

$$P(z) = \int_{-\infty}^{z} p(x)dx$$

对于连续型变量 X,Y，加法、乘法法则与贝叶斯定理仍然成立：

$$p(x) = \int p(x,y)dy$$

$$p(x,y) = p(y|x)p(x)$$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$$p(x) = \int p(x|y)p(y)dy$$

2.2.1 期望 (Expectation) 与协方差 (Covariance)

定义 2.2 (期望)

某函数 $f(x)$ 在概率分布 $p(x)$ 下的加权平均值称为 $f(x)$ 的期望，记作 $\mathbb{E}[f]$，有：

$$\text{离散型：}\mathbb{E}[f] = \sum_x p(x)f(x) \qquad \text{连续型：}\mathbb{E}[f] = \int p(x)f(x)dx$$

♣

类似可定义条件期望 (conditional expectation)：

定义 2.3 (条件期望)

某函数 $f(x)$ 在概率分布 $p(x)$ 下的对于某一 y 关于 x 的加权平均值称为 $f(x)$ 关于 x 的条件期望，记作 $\mathbb{E}_x[f(x,y)]$，这是一个关于 y 的函数，有：

$$\text{离散型：}\mathbb{E}_x[f|y] = \sum_x p(x|y)f(x) \qquad \text{连续型：}\mathbb{E}_x[f|y] = \int p(x|y)f(x)dx$$

♣

还可定义方差 (variance)。

定义 2.4 (方差)

$$var[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

♣

定义协方差 (covariance) 用于衡量二者相关程度。

定义 2.5 (协方差)

$$cov[x,y] = \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] = \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]$$

若 $cov[x,y] = 0$，则称变量 x,y 独立。

♣

2.3 高斯分布 (Gaussion Distribution)

高斯分布也称为正态分布 (normal distribution)。设正态分布 $\mathcal{N}(\mu, \sigma^2)$ 的密度函数为 $p(x)$，则：

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$p(x|\mu, \sigma^2) > 0$$

$$\int_{-\infty}^{\infty} p(x|\mu, \sigma^2) dx = 1$$

可以计算出其期望与方差：

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} p(x|\mu, \sigma^2) x dx = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} p(x|\mu, \sigma^2) x^2 dx = \mu^2 + \sigma^2$$

$$var[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

2.3.1 似然函数 (Likelihood function)

假设有一个观测数据集表示为行向量 $\mathbf{x} = (x_1, x_2, \cdots, x_N)$，表示变量 x 的 N 个观测值，构造一个 D 维列向量 $\mathbf{y}_k = (y_{1k}, y_{2k}, \cdots, y_{Dk})^T$，表示在某一观测值 x_k 下的 D 个分量的值。由于只有有限数据集，无法确定其概率分布，下面将其限制为高斯分布进行讨论。

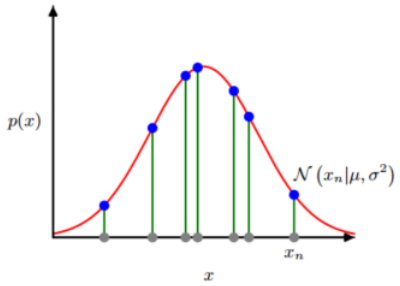


Figure 2.9 Illustration of the likelihood function for the Gaussian distribution shown by the red curve. Here the grey points denote a data set of values $\{x_n\}$, and the likelihood function (2.55) is given by the product of the corresponding values of $p(x)$ denoted by the blue points. Maximizing the likelihood involves adjusting the mean and variance of the Gaussian so as to maximize this product.

独立同分布 (independent and identically distributed) 通常缩写为 *i.i.d.* 或 *IID*，则有：

$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} p(x_n|\mu, \sigma^2)$$

将上式中的 μ、σ^2 当作主元，称为是高斯分布的似然函数(likelihood function) 取对数有：

$$\ln p(\mathbf{x}|\mu, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi)$$

对 μ 进行最大化，得到最大似然解 (maximum likelihood solution)：

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

此即样本均值 (sample mean)。

类似地，对 σ^2 进行最大化：

$$\sigma^2_{ML} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{ML})^2$$

此即样本方差 (sample variance)。

2.3.2 最大似然估计的有偏性

最大似然估计(MLE: Maximum Likelihood Estimation) 具有一些局限性，可以用单变量高斯分布来说明。

设有对应于数据集 $x_1, x_2, \cdots, x_N$ 的最大似然解 μ_{ML}、σ^2_{ML}，对于每个值 x_k 都是独立于实际参数 μ、σ^2 的高斯分布生成的。考虑 μ_{ML} 和 σ^2_{ML} 的期望（先证明一道课后题）：

✍ **练习 2.1 课后题 2.16** 若 x_n、x_m 独立同分布，服从 $\mathcal{N}(\mu, \sigma^2)$，求证：$\mathbb{E}[x_n x_m] = \mu^2 + \delta_{mn}\sigma^2$（$\delta$ 是 Kronecker 符号）

证明

当 m = n 时，$\mathbb{E}[x_n x_n] = \mathrm{var}(x_n) + \mathbb{E}[x_n]^2 = \sigma^2 + \mu^2$

当 m ≠ n 时，$\mathbb{E}[x_m x_n] = \frac{1}{2}\mathbb{E}\left[(x_n + x_m)^2 - x_n^2 - x_m^2\right] = \frac{1}{2}\left[\mathbb{E}[(x_n + x_m)^2] - 2\mu^2 - 2\sigma^2\right]$

$$= \frac{1}{2}\left[\mathrm{var}(x_n + x_m) + [\mathbb{E}(x_m + x_n)]^2 - 2\mu^2 - 2\sigma^2\right] = \frac{1}{2}\left[4\mu^2 - 2\mu^2\right] = \mu^2$$

于是课后习题证毕。利用上题结论：

$$\mathbb{E}[\mu_{ML}] = \mathbb{E}[\frac{1}{N}\sum_{n=1}^{N} x_n] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[x_n] = \mu$$

$$\mathbb{E}[\sigma^2_{ML}] = \mathbb{E}[\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[x_n^2 - 2x_n\mu_{ML} + \mu^2_{ML}]$$

对于求和符号后的那一项：

$$\begin{aligned}
\mathbb{E}[x_n^2 - 2x_n\mu_{ML} + \mu^2_{ML}] &= \mathbb{E}[x_n^2 - 2x_n \cdot \frac{1}{N}\sum_{k=1}^{N} x_k + \frac{1}{N^2}\left(\sum_{n=1}^{N} x_n\right)^2] \\
&= \mathbb{E}[x_n^2] - \frac{2}{N}\mathbb{E}\left[x_n\sum_{k=1}^{N} x_k\right] + \frac{1}{N^2}\mathbb{E}\left[\sum_{k=1}^{N} x_k^2 + 2\sum_{i<j} x_i x_j\right] \\
&= (\sigma^2 + \mu^2) - \frac{2}{N}\left(N\mu^2 + \sigma^2\right) + \frac{1}{N^2}\left[N(\sigma^2 + \mu^2) + 2C_N^2\mu^2\right] \\
&= \left(\frac{N-1}{N}\right)\sigma^2
\end{aligned}$$

因此，

$$\begin{aligned}
\mathbb{E}[\sigma^2_{ML}] &= \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[x_n^2 - 2x_n\mu_{ML} + \mu^2_{ML}] = N \cdot \frac{1}{N}\left(\frac{N-1}{N}\right)\sigma^2 \\
&= \left(\frac{N-1}{N}\right)\sigma^2
\end{aligned}$$

可以看出，均值的最大似然解 = 实际均值，但是方差的最大似然估计 (maximum likelihood estimate) 却与真实值差了一个因子 $\frac{N-1}{N}$，这种随机变量估计值与真实值出现系统性的 (systematically) 不同的现象称为偏差(bias)。上面的例子即证明了最大似然估计的有偏性。

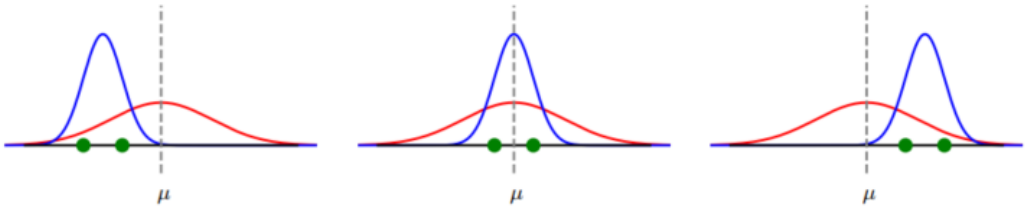


Figure 2.10 Illustration of how bias arises when using maximum likelihood to determine the mean and variance of a Gaussian. The red curves show the true Gaussian distribution from which data is generated, and the three blue curves show the Gaussian distributions obtained by fitting to three data sets, each consisting of two data points shown in green, using the maximum likelihood results (2.57) and (2.58). Averaged across the three data sets, the mean is correct, but the variance is systematically underestimated because it is measured relative to the sample mean and not relative to the true mean.

如果有

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

则：

$$\mathbb{E}[\hat{\sigma}^2] = \sigma^2$$

此时的估计就是无偏的(unbiased)。但是我们只有观测值，无法用 μ 的真实值进行带入计算，但经过变换后，对于高斯分布，下面的估计是无偏的。

$$\tilde{\sigma}^2 = \frac{N}{N-1}\sigma_{ML}^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{ML})^2$$

但是在神经网络等复杂模型中，对最大似然偏差的修正就不是这么容易。当 $N \to \infty$ 时，方差的最大似然解 = 实际方差。最大似然偏差问题与过拟合有密切关系。

2.3.3 线性回归

用误差极小化的思路解线性回归问题。输入变量为 x，目标变量为 t，输入值 $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ 以及其对应目标值 $\mathbf{t} = (t_1, t_2, \cdots, t_N)$

2.3.4 密度函数变换

概率密度的变换对于标准化流(normalizing flows) 至关重要。设有一维随机变量 x，其密度函数为 $p_x(x)$，且有 $x = g(y)$：

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right| = p_x(g(y))\left|\frac{dg}{dy}\right|$$

对于 D 维随机变量，$\mathbf{x} = (x_1, x_2, \cdots, x_D)^T$，$\mathbf{y} = (y_1, y_2, \cdots, y_D)^T$，且有多元函数 $\mathbf{g}$ 有 $\mathbf{x} = \mathbf{g}(\mathbf{y})$，设 $p_{\mathbf{x}}(\mathbf{x})$ 是 $\mathbf{x}$ 的密度函数，则：

$$p_{\mathbf{y}}(\mathbf{y}) = p_{\mathbf{x}}(\mathbf{x})\,|\det \mathbf{J}|$$

其中 $\mathbf{J}$ 是函数 $\mathbf{g}(\cdot)$ 的 Jacobi 矩阵。

2.4 信息论 (Information Theory)

2.4.1 熵 (Entropy)

信息熵是一个用于度量信息量 (amount of information) 的概念，其大小取决于事件发生的概率。（被告知某小概率事件会发生，这会使人更惊讶，于是这条信息具有更多的信息量）

设 $p(x)$ 是概率密度，定义函数 $h(\cdot)$ 作为衡量信息量的函数，有：

$$h(x) = -\log_2 p(x)$$

假设信息发送方想要将一个随机变量 x 的值传给接收方，设 $p(x)$ 是其概率密度，则在此过程中的平均信息量就是信息熵：

$$\mathrm{H}[x] = \mathbb{E}[-\log_2 p(x)] - \sum_x p(x)\log_2 p(x)$$

其中，$H[x]$ 称为随机变量 x 的熵(entropy)。由 $\lim_{\epsilon\to 0}\epsilon\ln\epsilon = 0$，规定：当 $p(x) = 0$ 时，有 $p(x)\ln p(x) = 0$。香农提出的无噪声编码原理 (Shannon 编码定理,1948) 指出，熵是传递随机变量状态所需的 bit 数的下界。在后文中使用 ln 来定义熵，其对应的单位为纳特 (nat)。（来自于 natural logarithm）

2.4.2 物理学的角度

热力学中使用熵表示体系的混乱程度。

假设有 N 个相同小球，分开装入箱子中，满足第 i 个箱子装着 n_i 个小球，则在这种情形下的方法数为：

$$W = \frac{N!}{\prod_i n_i!}$$

则定义熵：

$$\mathrm{H} = \ln W = \frac{1}{N}\ln N! - \frac{1}{N}\sum_i \ln n_i!$$

令 $N\to\infty$，由斯特林公式：

$$\ln N! \simeq N\ln N - N$$

于是：

$$\mathrm{H} = -\lim_{N\to\infty}\sum_i\left(\frac{n_i}{N}\right)\ln\left(\frac{n_i}{N}\right) = -\sum_i p_i\ln p_i$$

2.4.3 微分熵 (Differential entropy)

将熵的定义扩展到连续变量 x 上，设其密度函数 $p(x)$ 是连续的，则有：

$$\mathrm{H}[\mathbf{x}] = -\int p(\mathbf{x})\ln p(\mathbf{x})\mathrm{d}\mathbf{x}$$

这称为是 $\mathbf{x}$ 的微分熵。

2.4.4 最大熵 (Maximum entropy)

对于离散分布，最大熵的值需要约束一阶矩、二阶中心矩以及归一性条件：

$$\int_{-\infty}^{\infty} p(x)\mathrm{d}x = 1$$

$$\int_{-\infty}^{\infty} xp(x)\mathrm{d}x = \mu$$

$$\int_{-\infty}^{\infty} (x-\mu)^2 p(x)\mathrm{d}x = \sigma^2$$

使用拉格朗日乘子将约束最大化转化为无约束最大化：

$$\mathrm{L}(p,\lambda_1,\lambda_2,\lambda_3) = -\int_{-\infty}^{\infty} p(x)\ln p(x)\mathrm{d}x + \lambda_1\left(\int_{-\infty}^{\infty} p(x)\mathrm{d}x - 1\right) + \lambda_2\left(\int_{-\infty}^{\infty} xp(x)\mathrm{d}x - \mu\right) + \lambda_3\left(\int_{-\infty}^{\infty}(x-\mu)^2 p(x)\mathrm{d}x - \sigma^2\right)$$

于是可得：

$$\frac{\partial L}{\partial p} = -\frac{\partial}{\partial p}\left(\int_{-\infty}^{\infty} p(x)\ln p(x) - \lambda_1 p(x) - \lambda_2 xp(x) - \lambda_3(x-\mu)^2 p(x)\mathrm{d}x\right) \overset{\text{def}}{=\!=} -\frac{\partial}{\partial p}\int_{-\infty}^{\infty} F(p)\mathrm{d}x = 0$$

由于 F 是 $p(x), x$ 的泛函，因此：

$$\frac{\partial F}{\partial p} = 0 = \ln p(x) + 1 - \lambda_1 - \lambda_2 x - \lambda_3 (x-\mu)^2$$

所以：

$$\begin{aligned} p(x) &= \exp\{-1 + \lambda_1 + \lambda_2 x + \lambda_3 (x-\mu)^2\} \\ &= C_1 \exp\{\lambda_2 x + \lambda_3 (x^2 - 2x\mu + \mu^2)\} \\ &= C_1 \exp \lambda_3 \{x^2 - 2x\mu + \mu^2 + \frac{\lambda_2}{\lambda_3} x\} \\ &= C_2 \exp \lambda_3 \left(x - (\mu - \frac{\lambda_2}{2\lambda_3})\right)^2 \end{aligned}$$

得到 $p(x)$ 关于直线 $x = \mu - \frac{\lambda_2}{2\lambda_3}$ 对称，有：

$$\mathbb{E}[x] = \mu - \frac{\lambda_2}{2\lambda_3} = \mu \Rightarrow \lambda_2 = 0 \Rightarrow p(x) = C_2 \exp\{\lambda_3 (x-\mu)^2\}$$

设 $\lambda_3 = -a^2$ 再代入其余二式：

$$\begin{cases} \int_{-\infty}^{\infty} C_2 e^{\lambda_3 (x-\mu)^2} = 1 \Rightarrow \frac{a}{C_2} = \sqrt{\pi} \\ \int_{-\infty}^{\infty} (x-\mu)^2 C_2 e^{\lambda_3 (x-\mu)^2} = \sigma^2 \Rightarrow \frac{C_2}{a^3} \cdot \frac{\sqrt{\pi}}{2} = \sigma^2 \end{cases} \Rightarrow \lambda_3 = -\frac{1}{2\sigma^2}, C_2 = \frac{1}{\sqrt{2\pi}\sigma}$$

最后可得：

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$

于是微分熵最大的分布就是高斯分布，计算高斯分布的微分熵:

$$\mathrm{H}[x] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\} \cdot \left[\frac{(x-\mu)^2}{2\sigma^2} + \frac{1}{2}\ln(2\pi\sigma^2)\right] \mathrm{d}x = \frac{1}{2}\left[1 + \ln(2\pi\sigma^2)\right]$$

2.4.5 相对熵 (Relative entropy)

考虑未知分布 $p(\mathbf{x})$，假设用已知分布 $q(\mathbf{x})$ 来对其建模，如果使用 $q(\mathbf{x})$ 构造将 $\mathbf{x}$ 的值发送到接收器的编码方案，则由于 $q(\mathbf{x})$ 并不是真正的分布 $p(\mathbf{x})$，所以指定 x 的平均额外信息量 (单位为 nats) 由下列式子给出，称为是 $p(\mathbf{x})$、$q(\mathbf{x})$ 之间的相对熵(relative entropy) 或KL 散度(Kullback–Leibler divergence)

$$\begin{aligned} \mathrm{KL}(p||q) &= \int p(\mathbf{x}) \ln q(\mathbf{x}) \mathrm{d}\mathbf{x} - \left(-\int p(\mathbf{x}) \ln p(\mathbf{x}) \mathrm{d}\mathbf{x}\right) \\ &= -\int p(\mathbf{x}) \ln \left\{\frac{q(\mathbf{x})}{p(\mathbf{x})}\right\} \mathrm{d}\mathbf{x} \end{aligned}$$

注意到上式不是关于 $p(\mathbf{x})$、$q(\mathbf{x})$ 对称的，也即 $\mathrm{KL}(p||q) \not\equiv \mathrm{KL}(q||p)$

下面证明，实际上有 $\mathrm{KL}(q||p) \geqslant 0 \Leftrightarrow p(\mathbf{x}) = q(\mathbf{x})$

由 Jensen 不等式，有

$$f(\mathbb{E}[x]) \leqslant \mathbb{E}[f(x)]$$

于是，

$$f(\int \mathbf{x} p(\mathbf{x})) \mathrm{d}\mathbf{x} \leqslant \int f(\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

因此

$$\mathrm{KL}(p||q) = \int p(\mathbf{x}) \ln \left\{\frac{q(\mathbf{x})}{p(\mathbf{x})}\right\} \mathrm{d}\mathbf{x} \geqslant -\ln \int q(\mathbf{x}) \mathrm{d}\mathbf{x} = 0$$

当 $p = q$ 时，上述所有等号均成立。

以上关于密度估计的结果与数据压缩之间存在密切关系。因为我们事先不知道真实分布，所以传递的信息效率一定偏低，传输的平均额外信息至少为两分布之间的 KL 散度，因此我们可以将 KL 散度解释为 $p(\mathbf{x})$、$q(\mathbf{x})$ 不相似程度的度量。

假设数据是从我们希望建模的未知分布 $p(\mathbf{x})$ 生成的，则可使用由一组可调参数 $\boldsymbol{\theta}$ 控制的参数分布 $q(\mathbf{x}|\boldsymbol{\theta})$ 来近似此分布。一种确定 $\boldsymbol{\theta}$ 的方法是最小化 $p(\mathbf{x})$ 和 $q(\mathbf{x}|\boldsymbol{\theta})$ 之间相对于 $\boldsymbol{\theta}$ 的 KL 散度。但因为我们事先不清楚 $p(\mathbf{x})$，因此无法直接做到。现假设有一个训练集 $\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}$。那么，相对于 $p(\mathbf{x})$ 的期望可以通过这些点来近似，因此

$$\mathrm{KL}(p||q) \simeq \frac{1}{N}\sum_{n=1}^{N}\{-\ln q(\mathbf{x_n}|\boldsymbol{\theta}) + \ln p(\mathbf{x_n})\}$$

公式右侧的第二项与 $\boldsymbol{\theta}$ 无关，第一项是在使用训练集评估下 $q(\mathbf{x}|\boldsymbol{\theta})$ 的 $\boldsymbol{\theta}$ 的负对数似然函数。因此，得到：最小化 KL 散度等价于最大化对数似然函数。

2.4.6 条件熵 (Conditional entropy)

现在考虑由 $p(\mathbf{x}, \mathbf{y})$ 给出的两个变量集合 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的联合分布，如果 $\mathbf{x}$ 的值已经知道，则指定对应的 $\mathbf{y}$ 值所需的附加信息由 $-\ln p(\mathbf{y}|\mathbf{x})$ 给出。因此，指定 $\mathbf{y}$ 所需的平均附加信息可以写为

$$\mathrm{H}[\mathbf{y}|\mathbf{x}] = -\iint p(\mathbf{y}, \mathbf{x}) \ln p(\mathbf{y}|\mathbf{x}) \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x}$$

它被称为 $\mathbf{y}$ 在给定 $\mathbf{x}$ 条件下的条件熵(conditional entropy)，条件熵满足以下关系：

$$\begin{aligned}
\mathrm{H}[\mathbf{x}, \mathbf{y}] &= -\int p(\mathbf{y}, \mathbf{x}) \ln p(\mathbf{y}, \mathbf{x}) \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x} \\
&= -\int p(\mathbf{y}, \mathbf{x}) \left[\ln p(\mathbf{y}|\mathbf{x}) + \ln p(\mathbf{x})\right] \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x} \\
&= -\int p(\mathbf{y}, \mathbf{x}) \ln p(\mathbf{y}|\mathbf{x}) \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x} - \int p(\mathbf{x}, \mathbf{y}) \ln p(\mathbf{x}) \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x} \\
&= \mathrm{H}[\mathbf{y}|\mathbf{x}] + \mathrm{H}[\mathbf{x}]
\end{aligned}$$

其中 $\mathrm{H}[\mathbf{x}, \mathbf{y}]$ 是 $p(\mathbf{x}, \mathbf{y})$ 的微分熵，$\mathrm{H}[\mathbf{x}]$ 是边缘分布 $p(\mathbf{x})$ 的微分熵。因此，描述 $\mathbf{x}$ 和 $\mathbf{y}$ 所需的信息由描述 $\mathbf{x}$ 本身所需的信息与给定 $\mathbf{x}$ 指定 $\mathbf{y}$ 所需的附加信息之和给出。

2.4.7 互信息 (Mutual information)

当两个变量 $\mathbf{x}$ 和 $\mathbf{y}$ 是独立的时，它们的联合分布将分解为边缘分布的乘积 $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$。如果变量不是独立的，我们可以通过考虑联合分布与边缘分布乘积之间的 KL 散度，来计算它们是否“接近”独立，该散度由以下公式给出

$$\begin{aligned}
\mathrm{I}[\mathbf{x}, \mathbf{y}] &\equiv \mathrm{KL}(p(\mathbf{x}, \mathbf{y})||p(\mathbf{x})p(\mathbf{y})) \\
&= -\iint p(\mathbf{x}, \mathbf{y}) \ln\left(\frac{p(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x}, \mathbf{y})}\right) d\mathbf{x}\, d\mathbf{y}
\end{aligned}$$

这被称为变量 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的互信息(mutual information)。从 KL 散度的性质来看，我们知道 $\mathrm{I}[\mathbf{x}, \mathbf{y}] \geq 0$ 当且仅当 $\mathbf{x}$ 和 $\mathbf{y}$ 是独立的，利用定义，有：

$$\mathrm{I}[\mathbf{x}, \mathbf{y}] = \mathrm{H}[\mathbf{x}] - \mathrm{H}[\mathbf{x}|\mathbf{y}] = \mathrm{H}[\mathbf{y}] - \mathrm{H}[\mathbf{y}|\mathbf{x}].$$

因此，互信息表示通过知道 $\mathbf{y}$（或反之亦然）值而减少的不确定性。

从贝叶斯的角度来看，我们可以将 $p(\mathbf{x})$ 视为 $\mathbf{x}$ 的先验分布，并将 $p(\mathbf{x}|\mathbf{y})$ 视为在观察到新数据 $\mathbf{y}$ 后的后验分布。因此，互信息可以看作是由于观察到新的 $\mathbf{y}$ 而减少的 $\mathbf{x}$ 的不确定性。

2.5 贝叶斯概率 (Bayesian Probability)

古典的概率用随机、可重复试验的频率来定义。而贝叶斯概率(Bayesian probability) 是由贝叶斯理论所提供的一种对概率的解释，它采用将概率定义为某人对一个命题信任的程度。贝叶斯概率的核心思想就是不断根据

新的证据，将先验概率调整为后验概率，更新现有的置信度，使之更接近客观事实。

2.5.1 模型参数

以使用训练集 $\mathcal{D}$ 的回归模型为例来说明，贝叶斯观点为机器学习的几个方面提供了有价值的见解。我们已经看到，参数可以通过最大似然估计来选择，即选择使似然函数 $p(\mathcal{D}|\mathbf{w})$ 最大化的 $\mathbf{w}$ 值。在机器学习文献中，常将似然函数的负对数称为是误差函数(error function)。从贝叶斯理论来看，我们可以捕捉关于 $\mathbf{w}$ 的先验假设，然后通过似然函数 $p(\mathcal{D}|\mathbf{w})$ 将其与观测数据结合，得到参数 $\mathbf{w}$ 的后验分布 $p(\mathbf{w}|\mathcal{D})$:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} \tag{2.1}$$

这使我们能够评估在观察到 $\mathcal{D}$ 后,$\mathbf{w}$ 的不确定性。需要强调的是, 似然函数 $p(\mathcal{D}|\mathbf{w})$ 不是 $\mathbf{w}$ 的概率分布，而是表示对于不同 $\mathbf{w}$ 值，观测数据集 $\mathcal{D}$ 出现的可能性，因此其积分不一定为 1。可以用一句话概括贝叶斯定理:

$$\text{后验 (posterior)} \propto \text{似然 (likelihood)} \times \text{先验 (prior)}$$

这些量都被视为 $\mathbf{w}$ 的函数。$p(\mathcal{D})$ 是归一化常数，确保了左侧的后验分布是有效的概率密度，并且对 $\mathbf{w}$ 积分为 1。通过对 (2.1) 式的两侧对 $\mathbf{w}$ 积分，我们可以用先验分布和似然函数来表示贝叶斯定理:

$$p(D) = \int p(D|\mathbf{w})p(\mathbf{w})d\mathbf{w}$$

在贝叶斯理论和频率主义理论中，似然函数 $p(D|\mathbf{w})$ 都起着核心作用，但在两种方法中的用法却不同。在频率主义理论中，$\mathbf{w}$ 被视为固定参数，其值由某种“估计量”决定，并且对该估计值的误差区间是通过考虑可能的数据集 $\mathcal{D}$ 的分布来确定的。相比之下，从贝叶斯理论来看，只有一个观察到的数据集 $\mathcal{D}$，参数 $\mathbf{w}$ 的不确定性通过 $\mathbf{w}$ 上的概率分布来表示。

2.5.2 正则化

我们可以使用贝叶斯观点来了解用于减少过拟合的正则化技术。我们可以不通过最大化似然函数 $p(D|\mathbf{w})$ 来选择模型参数 $\mathbf{w}$，转而最大化后验概率，这种技术称为最大后验估计(maximum a posteriori, MAP)。等效地，我们可以最小化后验概率的负对数。对 (2.1) 两侧同时取对数，我们可得:

$$-\ln p(\mathbf{w}|D) = -\ln p(D|\mathbf{w}) - \ln p(\mathbf{w}) + \ln p(D) \tag{2.2}$$

右侧第一项是通常的对数似然。第三项可以省略，因为它不依赖于 $\mathbf{w}$。第二项采用了 $\mathbf{w}$ 的函数形式，我们可以将其视为一种正则化形式。假设我们将先验 $p(\mathbf{w})$ 选为相互独立的高斯分布 $\mathcal{N}(0, s^2)$, 则有:

$$p(\mathbf{w}) = \prod_{i=0}^{M} \mathcal{N}(w_i|0, s^2) = \prod_{i=0}^{M} \frac{1}{\sqrt{2\pi}s} \exp\left\{-\frac{w_i^2}{2s^2}\right\}$$

代入 (2.2) 中，可得：

$$-\ln p(\mathbf{w}|D) = -\ln p(D|\mathbf{w}) + \frac{1}{2s^2}\sum_{i=0}^{M} w_i^2 + \text{const} \tag{2.3}$$

如果是线性回归模型，则最大化后验分布就等同于最小化下面函数：

$$E(\mathbf{w}) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{1}{2s^2}\mathbf{w}^T\mathbf{w}$$

2.5.3 贝叶斯机器学习

贝叶斯观点让我们理解了正则化的动机，并推导出正则化项的某一特定形式。然而，单独使用贝叶斯定理并不是真正的贝叶斯机器学习方法，因为它仍是寻找 $\mathbf{w}$ 的某一确定解，而没有考虑参数 $\mathbf{w}$ 的不确定性。现假设有一组数据集 D，目标是预测新输入值 x 对应的目标变量 t。从贝叶斯角度来看，我们感兴趣的是 t 的分布，这

需要考虑 $\mathbf{w}$ 的分布:

$$p(t|x,D)=\int p(t|x,\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}$$

我们看到，此处的预测是通过对所有可能的 $\mathbf{w}$ 值进行加权平均 $p(t|x,\mathbf{w})$ 而获得的，其中权重由后验概率分布 $p(\mathbf{w}|D)$ 给出，这区别于常规的频率主义方法（频率主义仅使用通过优化损失函数获得的点估计参数）。

这种完全贝叶斯式的机器学习方法为我们提供了一种全新的强大的观点。例如，多项式回归中遇到的过拟合问题，就是一个由最大似然引起病态的例子，而在贝叶斯方法中不会出现这种情况。同样，我们可以有多个潜在模型来解决给定的问题，如不同阶次的多项式，每个模型都有其后验概率。最大似然只是简单地选择数据概率最高的模型，容易导致过拟合；而贝叶斯方法则是对所有模型由其后验概率进行加权平均。

不过，贝叶斯框架 (framework) 也存在一个缺点，那就是需要对参数空间进行积分。因为现代深度学习模型可能有数百万乃至数十亿个参数，所以直接采用贝叶斯方法通常是不可行的。事实上，在受限的计算预算和丰富的训练数据下，通常更好的方法是使用最大似然技术配合一种或多种正则化形式应用于大型神经网络，而不是在小型模型应用贝叶斯方法。

2.6 课后题选做

✍ **练习 2.2 课后题 2.29** 考虑变量 $\mathbf{x}$、$\mathbf{y}$ 的联合分布 $p(\mathbf{x},\mathbf{y})$，证明其微分熵满足:$\mathrm{H}[\mathbf{x},\mathbf{y}]\leqslant\mathrm{H}[\mathbf{x}]+\mathrm{H}[\mathbf{y}]$

证明 回顾定义与结论：

$$\mathrm{H}[\mathbf{x},\mathbf{y}]=-\int p(\mathbf{x},\mathbf{y})\ln p(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}$$

$$\mathrm{H}[\mathbf{x}]=-\int p(\mathbf{x})\ln p(\mathbf{x})\mathrm{d}\mathbf{x}$$

$$\mathrm{H}[\mathbf{y}|\mathbf{x}]=-\iint p(\mathbf{y},\mathbf{x})\ln p(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x}$$

$$\mathrm{H}[\mathbf{x},\mathbf{y}]=\mathrm{H}[\mathbf{x}]+\mathrm{H}[\mathbf{y}|\mathbf{x}]=\mathrm{H}[\mathbf{y}]+\mathrm{H}[\mathbf{y}|\mathbf{x}]$$

又，

$$p(\mathbf{y}|\mathbf{x})\geqslant p(\mathbf{y})$$

所以，

$$\begin{aligned}\mathrm{H}[\mathbf{x},\mathbf{y}]=\mathrm{H}[\mathbf{x}]+\mathrm{H}[\mathbf{y}|\mathbf{x}]&=\mathrm{H}[\mathbf{x}]-\iint p(\mathbf{y},\mathbf{x})\ln p(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x}\\&\leqslant\mathrm{H}[\mathbf{x}]-\int p(\mathbf{y},\mathbf{x})\ln p(\mathbf{y})\mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x}=\mathrm{H}[\mathbf{x}]-\int p(\mathbf{y})\ln p(\mathbf{y})\mathrm{d}\mathbf{y}\\&=\mathrm{H}[\mathbf{x}]+\mathrm{H}[\mathbf{y}]\end{aligned}$$

✍ **练习 2.3 课后题 2.30** 考虑一个连续变量 $\mathbf{x}$，其密度函数为 $p(x)$，对应的熵为 $H[x]$。假设我们对 $\mathbf{x}$ 做非奇异线性变换得到新变量 $y=Ax$，求证：$\mathrm{H}[\mathbf{y}]=\mathrm{H}[\mathbf{x}]+\ln|\det A|$

证明 由

$$p(\mathbf{y})=p(\mathbf{x})\left|\det A^{-1}\right|=p(\mathbf{x})\frac{1}{|\det A|},\quad \mathrm{d}\mathbf{y}=|\det A|\,\mathrm{d}\mathbf{x}$$

因此，

$$\begin{aligned}\mathrm{H}[\mathbf{y}]&=-\int p(\mathbf{y})\ln p(\mathbf{y})\mathrm{d}\mathbf{y}=-\int p(\mathbf{x})\frac{1}{|\det A|}\ln\left(p(\mathbf{x})\frac{1}{|\det A|}\right)|\det A|\mathrm{d}\mathbf{x}\\&=-\int p(\mathbf{x})\ln\left(p(\mathbf{x})-\ln|\det A|\right)\mathrm{d}\mathbf{x}=-\int p(\mathbf{x})\ln p(\mathbf{x})\mathrm{d}\mathbf{x}+\ln|\det A|\int p(\mathbf{x})\mathrm{d}\mathbf{x}\\&=\mathrm{H}[\mathbf{x}]+\ln|\det A|\end{aligned}$$