Chapter 5

Loss functions

The last three chapters described linear regression, shallow neural networks, and deep neural networks. Each represents a family of functions that map input to output, where the particular member of the family is determined by the model parameters $\boldsymbol{\phi}$. When we train these models, we seek the parameters that produce the best possible mapping from input to output for the task we are considering. This chapter defines what is meant by the "best possible" mapping.

That definition requires a training dataset $\{\mathbf{x}_i, \mathbf{y}_i\}$ of input/output pairs. A *loss function* or *cost function* $L[\boldsymbol{\phi}]$ returns a single number that describes the mismatch between the model predictions $\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]$ and their corresponding ground-truth outputs $\mathbf{y}_i$. During training, we seek parameter values $\boldsymbol{\phi}$ that minimize the loss and hence map the training inputs to the outputs as closely as possible. We saw one example of a loss function in chapter 2; the least squares loss function is suitable for univariate regression problems for which the target is a real number $y \in \mathbb{R}$. It computes the sum of the squares of the deviations between the model predictions $\mathrm{f}[\mathbf{x}_i, \boldsymbol{\phi}]$ and the true values y_i.

Appendix A Sets

This chapter provides a framework that both justifies the choice of the least squares criterion for real-valued outputs and allows us to build loss functions for other prediction types. We consider *binary classification*, where the prediction $y \in \{0, 1\}$ is one of two categories, *multiclass classification*, where the prediction $y \in \{1, 2, \ldots, K\}$ is one of K categories, and more complex cases. In the following two chapters, we address model training, where the goal is to find the parameter values that minimize these loss functions.

5.1 Maximum likelihood

In this section, we develop a recipe for constructing loss functions. Consider a model $\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ with parameters $\boldsymbol{\phi}$ that computes an output from input $\mathbf{x}$. Until now, we have implied that the model directly computes a prediction $\mathbf{y}$. We now shift perspective and consider the model as computing a conditional probability distribution $Pr(\mathbf{y}|\mathbf{x})$ over possible outputs $\mathbf{y}$ given input $\mathbf{x}$. The loss encourages each training output $\mathbf{y}_i$ to have a high probability under the distribution $Pr(\mathbf{y}_i|\mathbf{x}_i)$ computed from the corresponding input $\mathbf{x}_i$ (figure 5.1).

Appendix C.1.3 Conditional probability

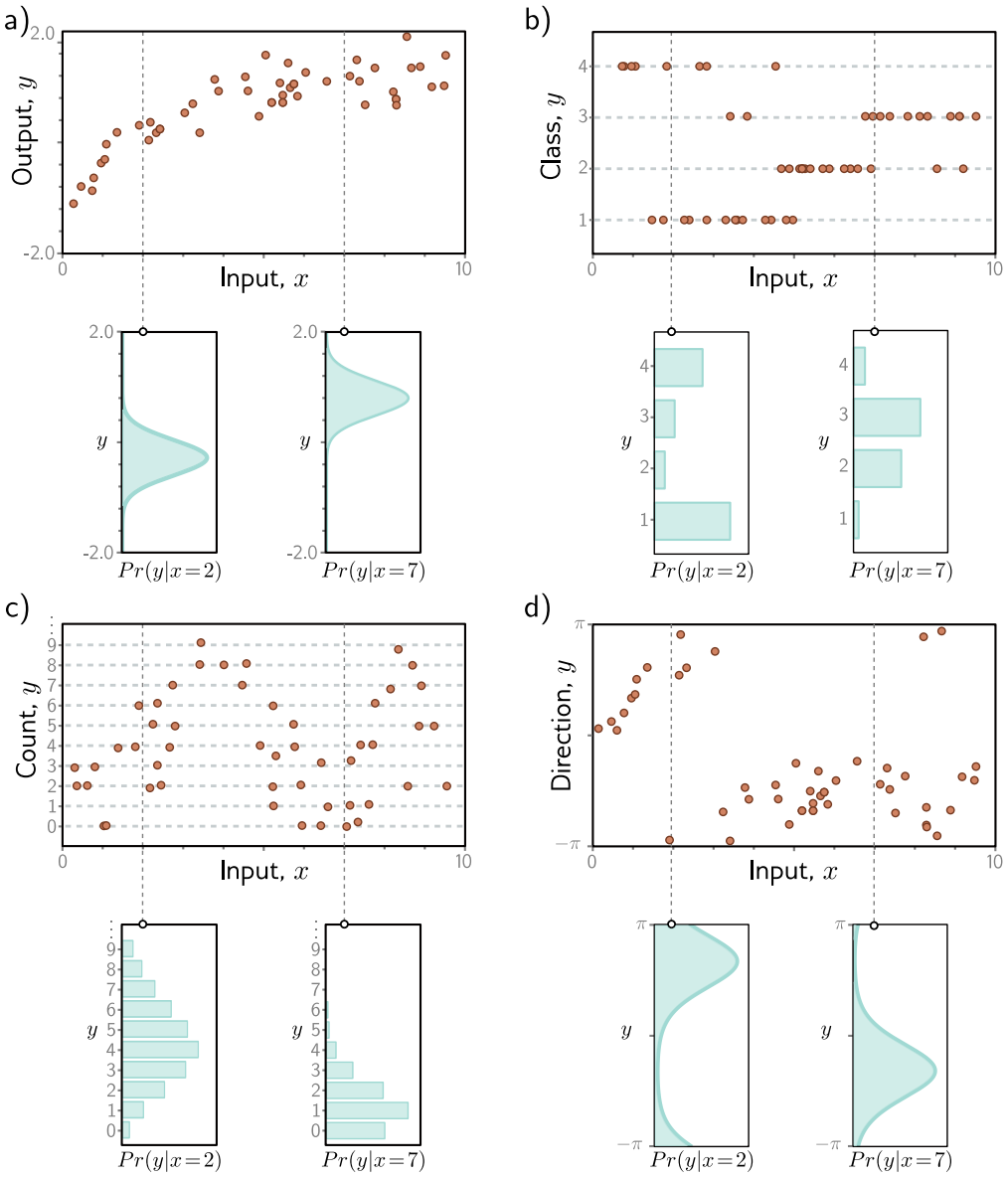

Figure 5.1 Predicting distributions over outputs. a) Regression task, where the goal is to predict a real-valued output y from the input x based on training data $\{x_i, y_i\}$ (orange points). For each input value x, the machine learning model predicts a distribution $Pr(y|x)$ over the output $y \in \mathbb{R}$ (cyan curves show distributions for $x=2.0$ and $x=7.0$). Minimizing the loss function corresponds to maximizing the probability of the training outputs y_i under the distribution predicted from the corresponding inputs x_i. b) To predict discrete classes $y \in \{1, 2, 3, 4\}$ in a classification task, we use a discrete probability distribution, so the model predicts a different histogram over the four possible values of y_i for each value of x_i. c) To predict counts $y \in \{0, 1, 2, \ldots\}$ and d) direction $y \in (-\pi, \pi]$, we use distributions defined over positive integers and circular domains, respectively.

5.1.1 Computing a distribution over outputs

This shift in perspective raises the question of exactly how a model $\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ can be adapted to compute a probability distribution. The solution is simple. First, we choose a parametric distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ defined on the output domain $\mathbf{y}$. Then we use the network to compute one or more of the parameters $\boldsymbol{\theta}$ of this distribution.

For example, suppose the prediction domain is the set of real numbers, so $y \in \mathbb{R}$. Here, we might choose the univariate normal distribution, which is defined on $\mathbb{R}$. This distribution is defined by the mean μ and variance σ^2, so $\boldsymbol{\theta} = \{\mu, \sigma^2\}$. The machine learning model might predict the mean μ, and the variance σ^2 could be treated as an unknown constant.

5.1.2 Maximum likelihood criterion

The model now computes different distribution parameters $\boldsymbol{\theta}_i = \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]$ for each training input $\mathbf{x}_i$. Each observed training output $\mathbf{y}_i$ should have high probability under its corresponding distribution $Pr(\mathbf{y}_i|\boldsymbol{\theta}_i)$. Hence, we choose the model parameters $\boldsymbol{\phi}$ so that they maximize the combined probability across all I training examples:

$$\begin{aligned}\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i)\right] \\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\boldsymbol{\theta}_i)\right] \\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}])\right]. \end{aligned} \tag{5.1}$$

The combined probability term is the *likelihood* of the parameters, and hence equation 5.1 is known as the *maximum likelihood* criterion.[1]

Here we are implicitly making two assumptions. First, we assume that the data are identically distributed (the form of the probability distribution over the outputs $\mathbf{y}_i$ is the same for each data point). Second, we assume that the conditional distributions $Pr(\mathbf{y}_i|\mathbf{x}_i)$ of the output given the input are independent, so the total likelihood of the training data decomposes as:

Appendix C.1.5 Independence

$$Pr(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I) = \prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i). \tag{5.2}$$

In other words, we assume the data are *independent and identically distributed* (*i.i.d.*).

[1] A conditional probability $Pr(z|\psi)$ can be considered in two ways. As a function of z, it is a probability distribution that sums to one. As a function of ψ, it is known as a *likelihood* and does not generally sum to one.