Chapter 1

Machine Learning

Deep learning belongs historically to the larger field of statistical machine learning, as it fundamentally concerns methods that are able to learn representations from data. The techniques involved come originally from artificial neural networks, and the "deep" qualifier highlights that models are long compositions of mappings, now known to achieve greater performance.

The modularity, versatility, and scalability of deep models have resulted in a plethora of specific mathematical methods and software development tools, establishing deep learning as a distinct and vast technical field.

1.1 Learning from data

The simplest use case for a model trained from data is when a signal x is accessible, for instance, the picture of a license plate, from which one wants to predict a quantity y, such as the string of characters written on the plate.

In many real-world situations where x is a high-dimensional signal captured in an uncontrolled environment, it is too complicated to come up with an analytical recipe that relates x and y.

What one can do is to collect a large training set $\mathscr{D}$ of pairs (x_n, y_n), and devise a parametric model f. This is a piece of computer code that incorporates trainable parameters w that modulate its behavior, and such that, with the proper values w^*, it is a good predictor. "Good" here means that if an x is given to this piece of code, the value $\hat{y} = f(x; w^*)$ it computes is a good estimate of the y that would have been associated with x in the training set had it been there.

This notion of goodness is usually formalized with a loss $\mathscr{L}(w)$ which is small when $f(\cdot; w)$ is good on $\mathscr{D}$. Then, training the model consists of computing a value w^* that minimizes $\mathscr{L}(w^*)$.

Most of the content of this book is about the definition of f, which, in realistic scenarios, is a complex combination of pre-defined sub-modules.

The trainable parameters that compose w are often called weights, by analogy with the synaptic weights of biological neural networks. In addition to these parameters, models usually depend on hyper-parameters, which are set according to domain prior knowledge, best practices, or resource constraints. They may also be optimized in some way, but with techniques different from those used to optimize w.

1.2 Basis function regression

We can illustrate the training of a model in a simple case where x_n and y_n are two real numbers, the loss is the mean squared error:

$$\mathscr{L}(w) = \frac{1}{N}\sum_{n=1}^{N}(y_n - f(x_n;w))^2, \qquad (1.1)$$

and $f(\cdot;w)$ is a linear combination of a predefined basis of functions $f_1,\dots,f_K$, with $w = (w_1,\dots,w_K)$:

$$f(x;w) = \sum_{k=1}^{K} w_k f_k(x).$$

Since $f(x_n;w)$ is linear with respect to the w_ks and $\mathscr{L}(w)$ is quadratic with respect to $f(x_n;w)$,

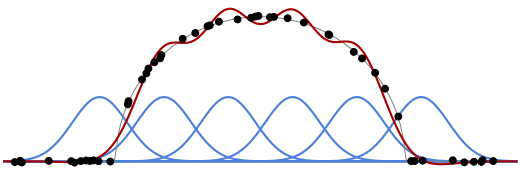

Figure 1.1: *Given a basis of functions (blue curves) and a training set (black dots), we can compute an optimal linear combination of the former (red curve) to approximate the latter for the mean squared error.*

the loss $\mathscr{L}(w)$ is quadratic with respect to the w_ks, and finding w^* that minimizes it boils down to solving a linear system. See Figure 1.1 for an example with Gaussian kernels as f_k.

1.3 Under and overfitting

A key element is the interplay between the capacity of the model, that is its flexibility and ability to fit diverse data, and the amount and quality of the training data. When the capacity is insufficient, the model cannot fit the data, resulting in a high error during training. This is referred to as underfitting.

On the contrary, when the amount of data is insufficient, as illustrated in Figure 1.2, the model will often learn characteristics specific to the training examples, resulting in excellent performance during training, at the cost of a worse

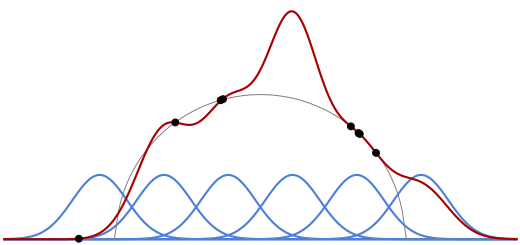

Figure 1.2: *If the amount of training data (black dots) is small compared to the capacity of the model, the empirical performance of the fitted model during training (red curve) reflects poorly its actual fit to the underlying data structure (thin black curve), and consequently its usefulness for prediction.*

fit to the global structure of the data, and poor performance on new inputs. This phenomenon is referred to as overfitting.

So, a large part of the art of applied machine learning is to design models that are not too flexible yet still able to fit the data. This is done by crafting the right inductive bias in a model, which means that its structure corresponds to the underlying structure of the data at hand.

Even though this classical perspective is relevant for reasonably-sized deep models, things get confusing with large ones that have a very large number of trainable parameters and extreme capacity yet still perform well on prediction. We will come back to this in § 3.6 and § 3.7.

1.4 Categories of models

We can organize the use of machine learning models into three broad categories:

• Regression consists of predicting a continuous-valued vector $y \in \mathbb{R}^K$, for instance, a geometrical position of an object, given an input signal X. This is a multi-dimensional generalization of the setup we saw in § 1.2. The training set is composed of pairs of an input signal and a ground-truth value.

• Classification aims at predicting a value from a finite set $\{1, \dots, C\}$, for instance, the label Y of an image X. As with regression, the training set is composed of pairs of input signal, and ground-truth quantity, here a label from that set. The standard way of tackling this is to predict one score per potential class, such that the correct class has the maximum score.

• Density modeling has as its objective to model the probability density function of the data μ_X itself, for instance, images. In that case, the training set is composed of values x_n without associated quantities to predict, and the trained model should allow for the evaluation of the probability density function, or sampling from the distribution, or both.

Both regression and classification are generally referred to as supervised learning, since the value to be predicted, which is required as a target during training, has to be provided, for instance, by human experts. On the contrary, density modeling is usually seen as unsupervised learning, since it is sufficient to take existing data without the need for producing an associated ground-truth.

These three categories are not disjoint; for instance, classification can be cast as class-score regression, or discrete sequence density modeling as iterated classification. Furthermore, they do not cover all cases. One may want to predict compounded quantities, or multiple classes, or model a density conditional on a signal.