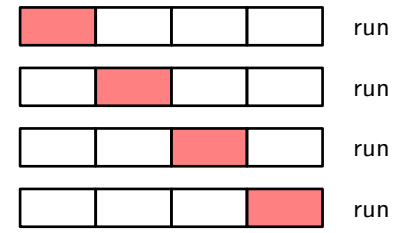

Figure 1.11 The technique of S-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into S groups of equal size. Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure 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.

validation, which is illustrated in Figure 1.11. This allows a proportion $(S-1)/S$ of the available data to be used for training while making use of all of the data to assess performance. When data is particularly scarce, it may be appropriate to consider the case $S = N$, where N is the total number of data points, which gives the *leave-one-out* technique.

The main drawback of cross-validation is that the number of training runs that must be performed is increased by a factor of S, and this can prove problematic for models in which the training is itself computationally expensive. A further problem with techniques such as cross-validation that use separate data to assess performance is that we might have multiple complexity hyperparameters for a single model (for instance, there might be several regularization hyperparameters). Exploring combinations of settings for such hyperparameters could, in the worst case, require a number of training runs that is exponential in the number of hyperparameters. The state of the art in modern machine learning involves extremely large models, trained on commensurately large data sets. Consequently, there is limited scope for exploration of hyperparameter settings, and heavy reliance is placed on experience obtained with smaller models and on heuristics.

This simple example of fitting a polynomial to a synthetic data set generated from a sinusoidal function has illustrated many key ideas from machine learning, and we will make further use of this example in future chapters. However, real-world applications of machine learning differ in several important respects. The size of the data sets used for training can be many orders of magnitude larger, and there will generally be many more input variables, perhaps numbering in the millions for image analysis, for example, as well as multiple output variables. The learnable function that relates outputs to inputs is governed by a class of models known as neural networks, and these may have a large number of parameters perhaps numbering in the hundreds of billions, and the error function will be a highly nonlinear function of those parameters. The error function can no longer be minimized through a closed-form solution and instead must be minimized through iterative optimization techniques based on evaluation of the derivatives of the error function with respect to the parameters, all of which may require specialist computational hardware and incur substantial computational cost.