rithm, meaning that it optimizes each piece of the solution independently, one piece at a time, rather than jointly optimizing all pieces. It is called **layer-wise** because these independent pieces are the layers of the network. Specifically, greedy layer-wise pretraining proceeds one layer at a time, training the k-th layer while keeping the previous ones fixed. In particular, the lower layers (which are trained first) are not adapted after the upper layers are introduced. It is called **unsupervised** because each layer is trained with an unsupervised representation learning algorithm. However it is also called **pretraining**, because it is supposed to be only a first step before a joint training algorithm is applied to **fine-tune** all the layers together. In the context of a supervised learning task, it can be viewed as a regularizer (in some experiments, pretraining decreases test error without decreasing training error) and a form of parameter initialization.

It is common to use the word "pretraining" to refer not only to the pretraining stage itself but to the entire two phase protocol that combines the pretraining phase and a supervised learning phase. The supervised learning phase may involve training a simple classifier on top of the features learned in the pretraining phase, or it may involve supervised fine-tuning of the entire network learned in the pretraining phase. No matter what kind of unsupervised learning algorithm or what model type is employed, in the vast majority of cases, the overall training scheme is nearly the same. While the choice of unsupervised learning algorithm will obviously impact the details, most applications of unsupervised pretraining follow this basic protocol.

Greedy layer-wise unsupervised pretraining can also be used as initialization for other unsupervised learning algorithms, such as deep autoencoders (Hinton and Salakhutdinov, 2006) and probabilistic models with many layers of latent variables. Such models include deep belief networks (Hinton *et al.*, 2006) and deep Boltzmann machines (Salakhutdinov and Hinton, 2009a). These deep generative models will be described in chapter 20.

As discussed in section 8.7.4, it is also possible to have greedy layer-wise *supervised* pretraining. This builds on the premise that training a shallow network is easier than training a deep one, which seems to have been validated in several contexts (Erhan *et al.*, 2010).

15.1.1 When and Why Does Unsupervised Pretraining Work?

On many tasks, greedy layer-wise unsupervised pretraining can yield substantial improvements in test error for classification tasks. This observation was responsible for the renewed interested in deep neural networks starting in 2006 (Hinton *et al.*,