true distribution $p(\mathbf{h} \mid \boldsymbol{v})$ by seeking an approximate distribution $q(\mathbf{h}|\mathbf{v})$ that is as close to the true one as possible. This and other techniques are described in depth in chapter 19.

16.7 The Deep Learning Approach to Structured Probabilistic Models

Deep learning practitioners generally use the same basic computational tools as other machine learning practitioners who work with structured probabilistic models. However, in the context of deep learning, we usually make different design decisions about how to combine these tools, resulting in overall algorithms and models that have a very different flavor from more traditional graphical models.

Deep learning does not always involve especially deep graphical models. In the context of graphical models, we can define the depth of a model in terms of the graphical model graph rather than the computational graph. We can think of a latent variable h_i as being at depth j if the shortest path from h_i to an observed variable is j steps. We usually describe the depth of the model as being the greatest depth of any such h_i. This kind of depth is different from the depth induced by the computational graph. Many generative models used for deep learning have no latent variables or only one layer of latent variables, but use deep computational graphs to define the conditional distributions within a model.

Deep learning essentially always makes use of the idea of distributed representations. Even shallow models used for deep learning purposes (such as pretraining shallow models that will later be composed to form deep ones) nearly always have a single, large layer of latent variables. Deep learning models typically have more latent variables than observed variables. Complicated nonlinear interactions between variables are accomplished via indirect connections that flow through multiple latent variables.

By contrast, traditional graphical models usually contain mostly variables that are at least occasionally observed, even if many of the variables are missing at random from some training examples. Traditional models mostly use higher-order terms and structure learning to capture complicated nonlinear interactions between variables. If there are latent variables, they are usually few in number.

The way that latent variables are designed also differs in deep learning. The deep learning practitioner typically does not intend for the latent variables to take on any specific semantics ahead of time—the training algorithm is free to invent the concepts it needs to model a particular dataset. The latent variables are