autoencoder (section 20.10.3) and the generative stochastic networks (section 20.12). These models naturally learn high-capacity, overcomplete encodings of the input and do not require regularization for these encodings to be useful. Their encodings are naturally useful because the models were trained to approximately maximize the probability of the training data rather than to copy the input to the output.

14.2.1 Sparse Autoencoders

A sparse autoencoder is simply an autoencoder whose training criterion involves a sparsity penalty $\Omega(\boldsymbol{h})$ on the code layer $\boldsymbol{h}$, in addition to the reconstruction error:

$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h}) \tag{14.2}$$

where $g(\boldsymbol{h})$ is the decoder output and typically we have $\boldsymbol{h} = f(\boldsymbol{x})$, the encoder output.

Sparse autoencoders are typically used to learn features for another task such as classification. An autoencoder that has been regularized to be sparse must respond to unique statistical features of the dataset it has been trained on, rather than simply acting as an identity function. In this way, training to perform the copying task with a sparsity penalty can yield a model that has learned useful features as a byproduct.

We can think of the penalty $\Omega(\boldsymbol{h})$ simply as a regularizer term added to a feedforward network whose primary task is to copy the input to the output (unsupervised learning objective) and possibly also perform some supervised task (with a supervised learning objective) that depends on these sparse features. Unlike other regularizers such as weight decay, there is not a straightforward Bayesian interpretation to this regularizer. As described in section 5.6.1, training with weight decay and other regularization penalties can be interpreted as a MAP approximation to Bayesian inference, with the added regularizing penalty corresponding to a prior probability distribution over the model parameters. In this view, regularized maximum likelihood corresponds to maximizing $p(\boldsymbol{\theta} \mid \boldsymbol{x})$, which is equivalent to maximizing $\log p(\boldsymbol{x} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$. The $\log p(\boldsymbol{x} \mid \boldsymbol{\theta})$ term is the usual data log-likelihood term and the $\log p(\boldsymbol{\theta})$ term, the log-prior over parameters, incorporates the preference over particular values of $\boldsymbol{\theta}$. This view was described in section 5.6. Regularized autoencoders defy such an interpretation because the regularizer depends on the data and is therefore by definition not a prior in the formal sense of the word. We can still think of these regularization terms as implicitly expressing a preference over functions.

Rather than thinking of the sparsity penalty as a regularizer for the copying task, we can think of the entire sparse autoencoder framework as approximating