14.7 Contractive Autoencoders

The contractive autoencoder (Rifai *et al.*, 2011a,b) introduces an explicit regularizer on the code $\boldsymbol{h} = f(\boldsymbol{x})$, encouraging the derivatives of f to be as small as possible:

$$\Omega(\boldsymbol{h}) = \lambda \left\| \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_F^2 . \tag{14.18}$$

The penalty $\Omega(\boldsymbol{h})$ is the squared Frobenius norm (sum of squared elements) of the Jacobian matrix of partial derivatives associated with the encoder function.

There is a connection between the denoising autoencoder and the contractive autoencoder: Alain and Bengio (2013) showed that in the limit of small Gaussian input noise, the denoising reconstruction error is equivalent to a contractive penalty on the reconstruction function that maps $\boldsymbol{x}$ to $\boldsymbol{r} = g(f(\boldsymbol{x}))$. In other words, denoising autoencoders make the reconstruction function resist small but finite-sized perturbations of the input, while contractive autoencoders make the feature extraction function resist infinitesimal perturbations of the input. When using the Jacobian-based contractive penalty to pretrain features $f(\boldsymbol{x})$ for use with a classifier, the best classification accuracy usually results from applying the contractive penalty to $f(\boldsymbol{x})$ rather than to $g(f(\boldsymbol{x}))$. A contractive penalty on $f(\boldsymbol{x})$ also has close connections to score matching, as discussed in section 14.5.1.

The name **contractive** arises from the way that the CAE warps space. Specifically, because the CAE is trained to resist perturbations of its input, it is encouraged to map a neighborhood of input points to a smaller neighborhood of output points. We can think of this as contracting the input neighborhood to a smaller output neighborhood.

To clarify, the CAE is contractive only locally—all perturbations of a training point $\boldsymbol{x}$ are mapped near to $f(\boldsymbol{x})$. Globally, two different points $\boldsymbol{x}$ and $\boldsymbol{x}'$ may be mapped to $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ points that are farther apart than the original points. It is plausible that f be expanding in-between or far from the data manifolds (see for example what happens in the 1-D toy example of figure 14.7). When the $\Omega(\boldsymbol{h})$ penalty is applied to sigmoidal units, one easy way to shrink the Jacobian is to make the sigmoid units saturate to 0 or 1. This encourages the CAE to encode input points with extreme values of the sigmoid that may be interpreted as a binary code. It also ensures that the CAE will spread its code values throughout most of the hypercube that its sigmoidal hidden units can span.

We can think of the Jacobian matrix $\boldsymbol{J}$ at a point $\boldsymbol{x}$ as approximating the nonlinear encoder $f(\boldsymbol{x})$ as being a linear operator. This allows us to use the word "contractive" more formally. In the theory of linear operators, a linear operator