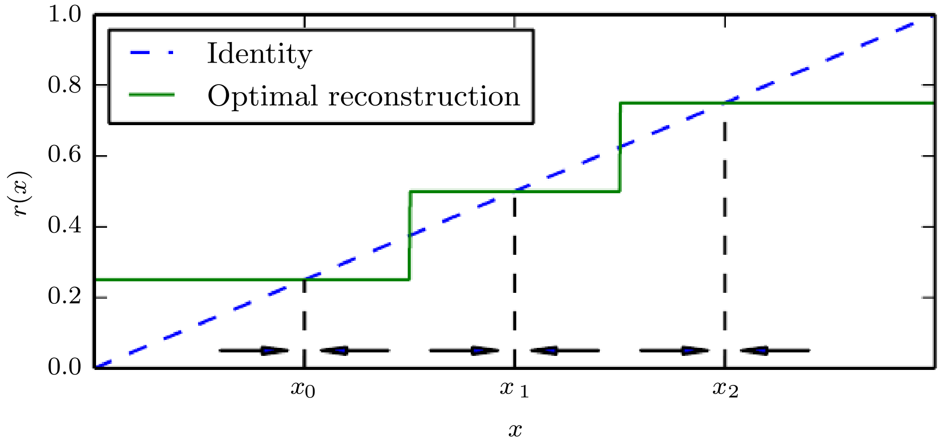

Figure 14.7: If the autoencoder learns a reconstruction function that is invariant to small perturbations near the data points, it captures the manifold structure of the data. Here the manifold structure is a collection of 0-dimensional manifolds. The dashed diagonal line indicates the identity function target for reconstruction. The optimal reconstruction function crosses the identity function wherever there is a data point. The horizontal arrows at the bottom of the plot indicate the $r(\boldsymbol{x}) - \boldsymbol{x}$ reconstruction direction vector at the base of the arrow, in input space, always pointing towards the nearest "manifold" (a single datapoint, in the 1-D case). The denoising autoencoder explicitly tries to make the derivative of the reconstruction function $r(\boldsymbol{x})$ small around the data points. The contractive autoencoder does the same for the encoder. Although the derivative of $r(\boldsymbol{x})$ is asked to be small around the data points, it can be large between the data points. The space between the data points corresponds to the region between the manifolds, where the reconstruction function must have a large derivative in order to map corrupted points back onto the manifold.

the manifold. Such a representation for a particular example is also called its embedding. It is typically given by a low-dimensional vector, with less dimensions than the "ambient" space of which the manifold is a low-dimensional subset. Some algorithms (non-parametric manifold learning algorithms, discussed below) directly learn an embedding for each training example, while others learn a more general mapping, sometimes called an encoder, or representation function, that maps any point in the ambient space (the input space) to its embedding.

Manifold learning has mostly focused on unsupervised learning procedures that attempt to capture these manifolds. Most of the initial machine learning research on learning nonlinear manifolds has focused on **non-parametric** methods based on the **nearest-neighbor graph**. This graph has one node per training example and edges connecting near neighbors to each other. These methods (Schölkopf *et al.*, 1998; Roweis and Saul, 2000; Tenenbaum *et al.*, 2000; Brand, 2003; Belkin