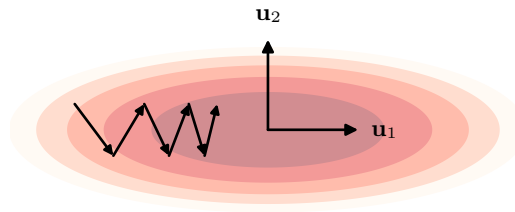

Figure 7.3 Schematic illustration of fixed-step gradient descent for an error function that has substantially different curvatures along different directions. The error surface E has the form of a long valley, as depicted by the ellipses. Note that, for most points in weight space, the local negative gradient vector $-\nabla E$ does not point towards the minimum of the error function. Successive steps of gradient descent can therefore oscillate across the valley, leading to very slow progress along the valley towards the minimum. The vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ are the eigenvectors of the Hessian matrix.

unit with M inputs:

$$\epsilon = \sqrt{\frac{2}{M}}. \tag{7.23}$$

It is also possible to treat the scale ϵ of the initialization distribution as a hyperparameter and to explore different values across multiple training runs. The bias parameters are typically set to small positive values to ensure that most pre-activations are initially active during learning. This is particularly helpful with ReLU units, where we want the pre-activations to be positive so that there is a non-zero gradient to drive learning.

Another important class of techniques for initializing the parameters of a neural network is by using the values that result from training the network on a different task or by exploiting various forms of unsupervised training. These techniques fall into the broad class of *transfer learning* techniques.

Section 6.3.4

7.3. Convergence

When applying gradient descent in practice, we need to choose a value for the learning rate parameter η. Consider the simple error surface depicted in Figure 7.3 for a hypothetical two-dimensional weight space in which the curvature of E varies significantly with direction, creating a 'valley'. At most points on the error surface, the local gradient vector for batch gradient descent, which is perpendicular to the local contour, does not point directly towards the minimum. Intuitively we might expect that increasing the value of η should lead to bigger steps through weight space and hence faster convergence. However, the successive steps oscillate back and forth across the valley, and if we increase η too much, those oscillations will become divergent. Because η must be kept sufficiently small to avoid divergent oscillations across the valley, progress along the valley is very slow. Gradient descent then takes many small steps to reach the minimum and is a very inefficient procedure.

We can gain deeper insight into the nature of this problem by considering the quadratic approximation to the error function in the neighbourhood of the minimum. From (7.7), (7.8), and (7.10), the gradient of the error function in this approximation

Section 7.1.1