

such as saddle points, the application of Newton’s method for training large neural networks is limited by the significant computational burden it imposes. The number of elements in the Hessian is squared in the number of parameters, so with  $k$  parameters (and for even very small neural networks the number of parameters  $k$  can be in the millions), Newton’s method would require the inversion of a  $k \times k$  matrix—with computational complexity of  $O(k^3)$ . Also, since the parameters will change with every update, the inverse Hessian has to be computed *at every training iteration*. As a consequence, only networks with a very small number of parameters can be practically trained via Newton’s method. In the remainder of this section, we will discuss alternatives that attempt to gain some of the advantages of Newton’s method while side-stepping the computational hurdles.

## 8.6.2 Conjugate Gradients

Conjugate gradients is a method to efficiently avoid the calculation of the inverse Hessian by iteratively descending **conjugate directions**. The inspiration for this approach follows from a careful study of the weakness of the method of steepest descent (see section 4.3 for details), where line searches are applied iteratively in the direction associated with the gradient. Figure 8.6 illustrates how the method of steepest descent, when applied in a quadratic bowl, progresses in a rather ineffective back-and-forth, zig-zag pattern. This happens because each line search direction, when given by the gradient, is guaranteed to be orthogonal to the previous line search direction.

Let the previous search direction be  $\mathbf{d}_{t-1}$ . At the minimum, where the line search terminates, the directional derivative is zero in direction  $\mathbf{d}_{t-1}$ :  $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \cdot \mathbf{d}_{t-1} = 0$ . Since the gradient at this point defines the current search direction,  $\mathbf{d}_t = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$  will have no contribution in the direction  $\mathbf{d}_{t-1}$ . Thus  $\mathbf{d}_t$  is orthogonal to  $\mathbf{d}_{t-1}$ . This relationship between  $\mathbf{d}_{t-1}$  and  $\mathbf{d}_t$  is illustrated in figure 8.6 for multiple iterations of steepest descent. As demonstrated in the figure, the choice of orthogonal directions of descent do not preserve the minimum along the previous search directions. This gives rise to the zig-zag pattern of progress, where by descending to the minimum in the current gradient direction, we must re-minimize the objective in the previous gradient direction. Thus, by following the gradient at the end of each line search we are, in a sense, undoing progress we have already made in the direction of the previous line search. The method of conjugate gradients seeks to address this problem.

In the method of conjugate gradients, we seek to find a search direction that is **conjugate** to the previous line search direction, i.e. it will not undo progress made in that direction. At training iteration  $t$ , the next search direction  $\mathbf{d}_t$  takes