

the latter. More specifically, $\mathbf{XW} + \mathbf{b}$ should be replaced by a normalized version of \mathbf{XW} . The bias term should be omitted because it becomes redundant with the β parameter applied by the batch normalization reparametrization. The input to a layer is usually the output of a nonlinear activation function such as the rectified linear function in a previous layer. The statistics of the input are thus more non-Gaussian and less amenable to standardization by linear operations.

In convolutional networks, described in chapter 9, it is important to apply the same normalizing μ and σ at every spatial location within a feature map, so that the statistics of the feature map remain the same regardless of spatial location.

8.7.2 Coordinate Descent

In some cases, it may be possible to solve an optimization problem quickly by breaking it into separate pieces. If we minimize $f(\mathbf{x})$ with respect to a single variable x_i , then minimize it with respect to another variable x_j and so on, repeatedly cycling through all variables, we are guaranteed to arrive at a (local) minimum. This practice is known as **coordinate descent**, because we optimize one coordinate at a time. More generally, **block coordinate descent** refers to minimizing with respect to a subset of the variables simultaneously. The term “coordinate descent” is often used to refer to block coordinate descent as well as the strictly individual coordinate descent.

Coordinate descent makes the most sense when the different variables in the optimization problem can be clearly separated into groups that play relatively isolated roles, or when optimization with respect to one group of variables is significantly more efficient than optimization with respect to all of the variables. For example, consider the cost function

$$J(\mathbf{H}, \mathbf{W}) = \sum_{i,j} |H_{i,j}| + \sum_{i,j} \left(\mathbf{X} - \mathbf{W}^\top \mathbf{H} \right)_{i,j}^2. \quad (8.38)$$

This function describes a learning problem called sparse coding, where the goal is to find a weight matrix \mathbf{W} that can linearly decode a matrix of activation values \mathbf{H} to reconstruct the training set \mathbf{X} . Most applications of sparse coding also involve weight decay or a constraint on the norms of the columns of \mathbf{W} , in order to prevent the pathological solution with extremely small \mathbf{H} and large \mathbf{W} .

The function J is not convex. However, we can divide the inputs to the training algorithm into two sets: the dictionary parameters \mathbf{W} and the code representations \mathbf{H} . Minimizing the objective function with respect to either one of these sets of variables is a convex problem. Block coordinate descent thus gives