Algorithm 7.3: Stochastic gradient descent with momentum

Input: Training set of data points indexed by $n \in \{1, \ldots, N\}$
Batch size B
Error function per mini-batch $E_{n:n+B-1}(\mathbf{w})$
Learning rate parameter η
Momentum parameter μ
Initial weight vector $\mathbf{w}$
Output: Final weight vector $\mathbf{w}$

$n \leftarrow 1$
$\Delta\mathbf{w} \leftarrow \mathbf{0}$
repeat
  $\Delta\mathbf{w} \leftarrow -\eta\nabla E_{n:n+B-1}(\mathbf{w}) + \mu\Delta\mathbf{w}$ `// calculate update term`
  $\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{w}$ `// weight vector update`
  $n \leftarrow n + B$
  if $n > N$ **then**
    shuffle data
    $n \leftarrow 1$
  end if
until convergence
return $\mathbf{w}$

new location to find the update, so that

$$\Delta\mathbf{w}^{(\tau-1)} = -\eta\nabla E\left(\mathbf{w}^{(\tau-1)} + \mu\Delta\mathbf{w}^{(\tau-2)}\right) + \mu\Delta\mathbf{w}^{(\tau-2)}. \tag{7.34}$$

For batch gradient descent, Nesterov momentum can improve the rate of convergence, although for stochastic gradient descent it can be less effective.

7.3.2 Learning rate schedule

Section 7.3.1

In the stochastic gradient descent learning algorithm (7.18), we need to specify a value for the learning rate parameter η. If η is very small then learning will proceed slowly. However, if η is increased too much it can lead to instability. Although some oscillation can be tolerated, it should not be divergent. In practice, the best results are obtained by using a larger value for η at the start of training and then reducing the learning rate over time, so that the value of η becomes a function of the step index τ:

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta^{(\tau-1)}\nabla E_n(\mathbf{w}^{(\tau-1)}). \tag{7.35}$$