(Hinton, 2012), giving

$$r_i^{(\tau)} = \beta r_i^{(\tau-1)} + (1-\beta)\left(\frac{\partial E(\mathbf{w})}{\partial w_i}\right)^2 \tag{7.41}$$

$$w_i^{(\tau)} = w_i^{(\tau-1)} - \frac{\eta}{\sqrt{r_i^\tau} + \delta}\left(\frac{\partial E(\mathbf{w})}{\partial w_i}\right) \tag{7.42}$$

where $0 < \beta < 1$ and a typical value is $\beta = 0.9$.

If we combine RMSProp with momentum, we obtain the *Adam* optimization method (Kingma and Ba, 2014) where the name is derived from 'adaptive moments'. Adam stores the momentum for each parameter separately using update equations that consist of exponentially weighted moving averages for both the gradients and the squared gradients in the form

$$s_i^{(\tau)} = \beta_1 s_i^{(\tau-1)} + (1-\beta_1)\left(\frac{\partial E(\mathbf{w})}{\partial w_i}\right) \tag{7.43}$$

$$r_i^{(\tau)} = \beta_2 r_i^{(\tau-1)} + (1-\beta_2)\left(\frac{\partial E(\mathbf{w})}{\partial w_i}\right)^2 \tag{7.44}$$

$$\widehat{s}_i^{(\tau)} = \frac{s_i^{(\tau)}}{1-\beta_1^\tau} \tag{7.45}$$

$$\widehat{r}_i^{\,\tau} = \frac{r_i^\tau}{1-\beta_2^\tau} \tag{7.46}$$

$$w_i^{(\tau)} = w_i^{(\tau-1)} - \eta \frac{\widehat{s}_i^{\,\tau}}{\sqrt{\widehat{r}_i^{\,\tau}} + \delta}. \tag{7.47}$$

Exercise 7.12

Here the factors $1/(1-\beta_1^\tau)$ and $1/(1-\beta_2^\tau)$ correct for a bias introduced by initializing $s_i^{(0)}$ and $r_i^{(0)}$ to zero. Note that the bias goes to zero as τ becomes large, since $\beta_i < 1$, and so in practice this bias correction is sometimes omitted. Typical values for the weighting parameters are $\beta_1 = 0.9$ and $\beta_2 = 0.99$. Adam is the most widely adopted learning algorithm in deep learning and is summarized in Algorithm 7.4.

7.4. Normalization

Normalization of the variables computed during the forward pass through a neural network removes the need for the network to deal with extremely large or extremely small values. Although in principle the weights and biases in a neural network can adapt to whatever values the input and hidden variables take, in practice normalization can be crucial for ensuring effective training. Here we consider three kinds of normalization according to whether we are normalizing across the input data, across mini-batches, or across layers.