AdaGrad [edit]

AdaGrad (for adaptive gradient algorithm) is a modified stochastic gradient descent with per-parameter learning rate, first published in 2011.[16][17] Informally, this increases the learning rate for more sparse parameters and decreases the learning rate for less sparse ones. This strategy often improves convergence performance over standard stochastic gradient descent in settings where data is sparse and sparse parameters are more informative. Examples of such applications include natural language processing and image recognition.[16] It still has a base learning rate η, but this is multiplied with the elements of a vector $\{G_{j,j}\}$ which is the diagonal of the outer product matrix.

$$G = \sum_{\tau=1}^{t} g_\tau g_\tau^\mathsf{T}$$

where $g_\tau = \nabla Q_i(w)$, the gradient, at iteration τ. The diagonal is given by

$$G_{j,j} = \sum_{\tau=1}^{t} g_{\tau,j}^2.$$

This vector is updated after every iteration. The formula for an update is now

$$w := w - \eta\, \mathrm{diag}(G)^{-\frac{1}{2}} \circ g$$[a]

or, written as per-parameter updates,

$$w_j := w_j - \frac{\eta}{\sqrt{G_{j,j}}} g_j.$$

Each $\{G_{(i,i)}\}$ gives rise to a scaling factor for the learning rate that applies to a single parameter w_i. Since the denominator in this factor, $\sqrt{G_i} = \sqrt{\sum_{\tau=1}^{t} g_\tau^2}$ is the $\ell_2$norm of previous derivatives, extreme parameter updates get dampened, while parameters that get few or small updates receive higher learning rates.[14]

While designed for convex problems, AdaGrad has been successfully applied to non-convex optimization.[18]