

---

**Algorithm 8.7** The Adam algorithm
 

---

**Require:** Step size  $\epsilon$  (Suggested default: 0.001)

**Require:** Exponential decay rates for moment estimates,  $\rho_1$  and  $\rho_2$  in  $[0, 1)$ .  
(Suggested defaults: 0.9 and 0.999 respectively)

**Require:** Small constant  $\delta$  used for numerical stabilization. (Suggested default:  $10^{-8}$ )

**Require:** Initial parameters  $\theta$

Initialize 1st and 2nd moment variables  $\mathbf{s} = \mathbf{0}$ ,  $\mathbf{r} = \mathbf{0}$

Initialize time step  $t = 0$

**while** stopping criterion not met **do**

    Sample a minibatch of  $m$  examples from the training set  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$  with corresponding targets  $\mathbf{y}^{(i)}$ .

    Compute gradient:  $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)})$

$t \leftarrow t + 1$

    Update biased first moment estimate:  $\mathbf{s} \leftarrow \rho_1 \mathbf{s} + (1 - \rho_1) \mathbf{g}$

    Update biased second moment estimate:  $\mathbf{r} \leftarrow \rho_2 \mathbf{r} + (1 - \rho_2) \mathbf{g} \odot \mathbf{g}$

    Correct bias in first moment:  $\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1 - \rho_1^t}$

    Correct bias in second moment:  $\hat{\mathbf{r}} \leftarrow \frac{\mathbf{r}}{1 - \rho_2^t}$

    Compute update:  $\Delta \theta = -\epsilon \frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}} + \delta}}$  (operations applied element-wise)

    Apply update:  $\theta \leftarrow \theta + \Delta \theta$

**end while**

---

Newton's method is an optimization scheme based on using a second-order Taylor series expansion to approximate  $J(\theta)$  near some point  $\theta_0$ , ignoring derivatives of higher order:

$$J(\theta) \approx J(\theta_0) + (\theta - \theta_0)^\top \nabla_{\theta} J(\theta_0) + \frac{1}{2} (\theta - \theta_0)^\top \mathbf{H} (\theta - \theta_0), \quad (8.26)$$

where  $\mathbf{H}$  is the Hessian of  $J$  with respect to  $\theta$  evaluated at  $\theta_0$ . If we then solve for the critical point of this function, we obtain the Newton parameter update rule:

$$\theta^* = \theta_0 - \mathbf{H}^{-1} \nabla_{\theta} J(\theta_0) \quad (8.27)$$

Thus for a locally quadratic function (with positive definite  $\mathbf{H}$ ), by rescaling the gradient by  $\mathbf{H}^{-1}$ , Newton's method jumps directly to the minimum. If the objective function is convex but not quadratic (there are higher-order terms), this update can be iterated, yielding the training algorithm associated with Newton's method, given in algorithm 8.8.