gradient descent (GD)

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

Gradient descent (GD) is an iterative algorithm for minimizing a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. Each iteration updates the current estimate by a step along the negative gradient, scaled by a step size η. Such a gradient step can be interpreted as the application of an operator that is parameterized by the step size and the underlying function. GD and its variants are widely used to train parametric models in deep learning.

Definition

Many machine learning (ML) applications amount to solving an optimization problem: an objective function $f : \mathbb{R}^d \to \mathbb{R}$ measures the quality of the vector $\mathbf{w} \in \mathbb{R}^d$ of model parameters, which is the optimization variable. When training a deep net for image recognition, for example, $f(\mathbf{w})$ is the average loss incurred by the network on a training set of labeled images.

In many important cases, the objective function is differentiable: at every point $\mathbf{w}$ there exists a gradient $\nabla f(\mathbf{w}) \in \mathbb{R}^d$, the vector of partial derivatives of f at $\mathbf{w}$, which determines a local linear approximation of f (see differentiable).

GD uses this local linear approximation to iteratively improve the current choice of the model parameters: starting from an initialization $\mathbf{w}^{(0)}$, GD generates a sequence of estimates $\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$ that ideally converge to a minimum of f. GD and its variants are the standard solvers for model training in deep learning (Goodfellow et al., 2016).

At each iteration t, GD refines the current estimate $\mathbf{w}^{(t)}$ by stepping in the direction of steepest descent of the local linear approximation to f at $\mathbf{w}^{(t)}$. This direction is the negative gradient $\nabla f(\mathbf{w}^{(t)})$, giving the update

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}), \tag{1}$$

where $\eta > 0$ is a step size. For a sufficiently small η, each step decreases the function value, $f(\mathbf{w}^{(t+1)}) \leq f(\mathbf{w}^{(t)})$, with strict decrease whenever $\nabla f(\mathbf{w}^{(t)}) \neq \mathbf{0}$ (Boyd and Vandenberghe, 2004, Sect. 9.3). Fig. 1 illustrates a single GD step.

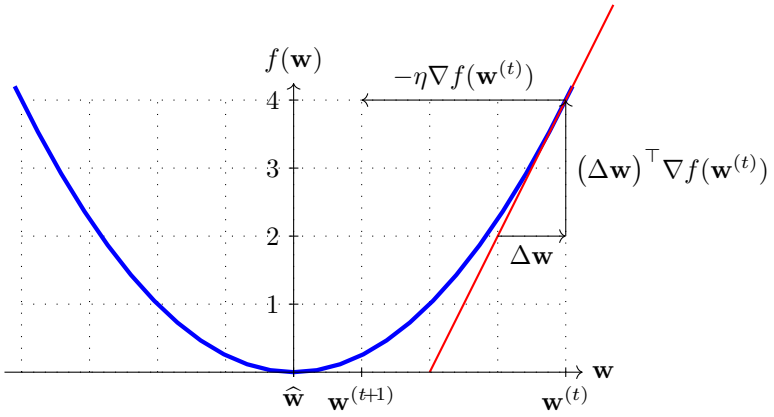

Figure 1: A single gradient step (1) toward the minimizer $\widehat{\mathbf{w}}$ of $f(\mathbf{w})$. The curve depicts the function f. The straight line is its local linear approximation at the current estimate $\mathbf{w}^{(t)}$. A change $\Delta\mathbf{w}$ of the estimate $\mathbf{w}^{(t)}$ changes this approximation by $(\Delta\mathbf{w})^\top\nabla f(\mathbf{w}^{(t)})$. Stepping against the gradient by $-\eta\nabla f(\mathbf{w}^{(t)})$ yields the next estimate $\mathbf{w}^{(t+1)}$

With a constant step size η, held fixed across iterations, a single step is described by the GD step operator $\mathcal{F}^{(\eta)}(\mathbf{w}) = \mathbf{w} - \eta\nabla f(\mathbf{w})$, so GD is the fixed-point iteration $\mathbf{w}^{(t+1)} = \mathcal{F}^{(\eta)}(\mathbf{w}^{(t)})$. Its fixed points, where $\mathcal{F}^{(\eta)}(\mathbf{w}) = \mathbf{w}$, are exactly the stationary points $\nabla f(\mathbf{w}) = \mathbf{0}$. For convex f, these are the minima according to the zero-gradient condition (Boyd and Vandenberghe, 2004, Sect. 4.2.3).

Consider shrinking the step size η in the GD step (1) toward zero. This results in iterates $\mathbf{w}^{(t)}$ becoming approximately the values of a continuous-time trajectory $\mathbf{w}(\tau)$. This trajectory satisfies the ordinary differential equation $\dot{\mathbf{w}}(\tau) = -\nabla f(\mathbf{w}(\tau))$, which is known as gradient flow. In particular, GD is the explicit Euler discretization of this gradient flow with step η (Helmke and Moore, 1994). Indeed, replacing the time derivative $\dot{\mathbf{w}}(\tau)$ by the difference quotient $\big(\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\big)/\eta$ recovers the update (1). Fig. 2 illustrates how, for a small step size, the GD iterates closely follow the gradient flow trajectory.

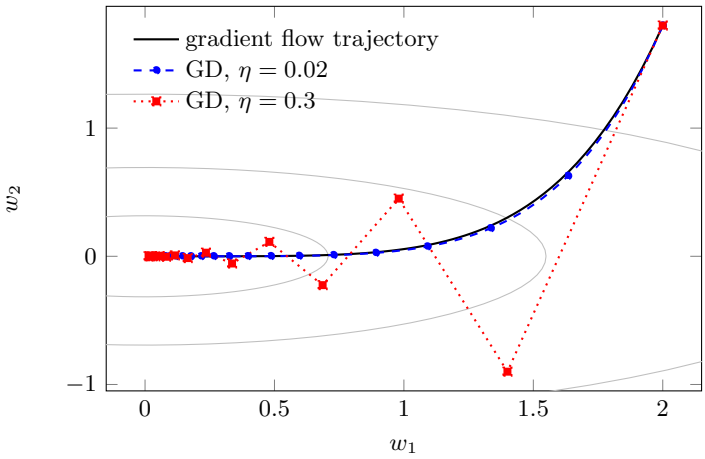

Figure 2: GD as the explicit Euler discretization of gradient flow, for the quadratic $f(\mathbf{w}) = \left(w_1^2 + 5\,w_2^2\right)/2$ with initialization $\mathbf{w}^{(0)} = (2, 1.8)^\top$. Gray ellipses are level sets of f; the solid curve is the exact gradient flow trajectory. For the small step size $\eta = 0.02$, the GD iterates (dashed, circular marks) follow the trajectory closely. For the large step size $\eta = 0.3$, the iterates (dotted, square marks) oscillate across the narrow direction of the level sets. Data generated by `pythondemos/gd.py`

Returning to the fixed-point iteration $\mathbf{w}^{(t+1)} = \mathcal{F}^{(\eta)}(\mathbf{w}^{(t)})$, its convergence is governed by the properties of the GD step operator $\mathcal{F}^{(\eta)}$. Assume f is convex and its gradient is L-Lipschitz continuous, i.e.,

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\| \leq L\,\|\mathbf{w} - \mathbf{w}'\| \quad \text{for all } \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d.$$

For $0 < \eta \leq 1/L$, the operator $\mathcal{F}^{(\eta)}$ is then a non-expansive operator with respect to the Euclidean norm $\|\cdot\|$: it does not increase distances between any two points $\mathbf{w}, \mathbf{w}'$,

$$\|\mathcal{F}^{(\eta)}(\mathbf{w}) - \mathcal{F}^{(\eta)}(\mathbf{w}')\| \leq \|\mathbf{w} - \mathbf{w}'\|.$$

Non-expansiveness alone does not guarantee a unique fixed point (Bauschke and Combettes, 2011, Ch. 4).

Assume, moreover, that f is μ-strongly convex, i.e., the function $f(\mathbf{w}) - (\mu/2)\|\mathbf{w}\|^2$ is still convex, so the curvature of f is at least $\mu > 0$ in every direction. Then $\mathcal{F}^{(\eta)}$ is a contractive operator with respect to the same norm,

$$\|\mathcal{F}^{(\eta)}(\mathbf{w}) - \mathcal{F}^{(\eta)}(\mathbf{w}')\| \leq (1 - \mu\eta)\,\|\mathbf{w} - \mathbf{w}'\|.$$

In this case, Banach's fixed-point theorem gives a unique fixed point $\widehat{\mathbf{w}}$ (the minimizer of f) to which GD converges at a geometric rate (Boyd and Vandenberghe, 2004; Nesterov, 2004). For linear regression, these properties are explicit: the gradient of the average squared error loss is an affine function of

$\mathbf{w}$, and the contraction factor of $\mathcal{F}^{(\eta)}$ is governed by the eigenvalues of the matrix $\mathbf{X}^\top\mathbf{X}$. Here, $\mathbf{X}$ denotes the feature matrix, whose rows are the feature vectors of the data points in the training set (see linear regression).

How fast GD converges depends on the curvature of f, as quantified by the constants L and μ above. Let $\widehat{\mathbf{w}}$ be a minimizer and $f^\star = f(\widehat{\mathbf{w}})$. For convex f with L-Lipschitz gradient and the constant step size $\eta = 1/L$ (Beck, 2017, Ch. 10; Bubeck, 2015, Thm. 3.3),

$$f\big(\mathbf{w}^{(T)}\big) - f^\star \leq \frac{L\,\|\mathbf{w}^{(0)} - \widehat{\mathbf{w}}\|^2}{2\,T},$$

so the suboptimality after T steps falls in proportion to $1/T$: halving the suboptimality requires doubling the number of iterations. If f is in addition μ-strongly convex, the bound improves to the geometric rate

$$f\big(\mathbf{w}^{(T)}\big) - f^\star \leq \big(1 - \tfrac{\mu}{L}\big)^T \big(f(\mathbf{w}^{(0)}) - f^\star\big),$$

consistent with the contractive operator factor $1 - \mu\eta$ above (Boyd and Vandenberghe, 2004, Sect. 9.3; Bubeck, 2015, Thm. 3.10). A geometric rate means that the suboptimality shrinks by at least the constant factor $1 - \mu/L$ in every single iteration. Thus, halving the suboptimality now costs a fixed number of iterations, however small the current suboptimality already is. Momentum methods, discussed below, sharpen both bounds.

In two common settings that motivate variants of GD, the assumptions behind these guarantees fail or the gradient becomes too costly. First, when f is non-smooth, its gradient may fail to exist and GD is replaced by subgradient descent, which steps along a subgradient in place of the gradient. A concrete case is the least absolute shrinkage and selection operator (Lasso) objective, whose penalty term $\alpha\|\mathbf{w}\|_1$ is non-smooth. Second, in empirical risk minimization (ERM)-based methods, f is an average of m loss functions, one for each data point, so evaluating the full gradient sums m gradients and its cost grows in proportion to the training set size m. GD is thus replaced by stochastic gradient descent (SGD), which uses a cheap gradient estimate from a random subset of the training set at each step.

Momentum methods provide another generalization of plain GD: each update combines gradients from several past steps rather than the current one alone. The name stems from a physical analogy: the iterates trace a particle whose momentum accumulates the force exerted by the negative gradient of the objective function. One prototypical example of a momentum method is Polyak's heavy-ball method. It adds a fraction $\beta \in [0, 1)$ of the previous increment,

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla f(\mathbf{w}^{(t)}) + \beta\big(\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\big), \text{ for } t = 1, \ldots.$$

For a strongly convex quadratic f, this iteration converges faster than plain GD (Polyak, 1987). Fig. 3 compares plain GD with the heavy-ball method on a strongly convex quadratic function.

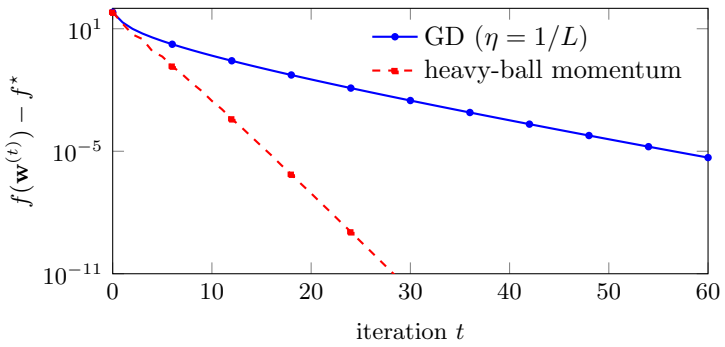

Figure 3: Suboptimality $f(\mathbf{w}^{(t)}) - f^\star$ of plain GD (solid, circular marks) and heavy-ball momentum (dashed, square marks) on a strongly convex quadratic, shown on a logarithmic scale. Both sequences decay geometrically, which appears as a straight line on this scale. The momentum method shrinks the suboptimality by a smaller factor per iteration and therefore reaches any prescribed accuracy in fewer iterations. Data generated by `pythondemos/gd.py`

A related momentum algorithm is Nesterov's accelerated variant of GD, which improves the $1/T$ bound above for convex f with L-Lipschitz gradient to

$$f\big(\mathbf{w}^{(T)}\big) - f^\star \le \frac{2\,L\,\|\mathbf{w}^{(0)} - \widehat{\mathbf{w}}\|^2}{(T+1)^2}.$$

In contrast to the bound for plain gradient descent (GD), which falls in proportion to $1/T$, this bound falls in proportion to $1/T^2$ (Beck and Teboulle, 2009, Thm. 4.4; Bubeck, 2015, Thm. 3.19). For μ-strongly convex f, acceleration improves the contraction factor from $1 - \mu/L$ to $1 - \sqrt{\mu/L}$ (Nesterov, 2004, Sect. 2.2; Bubeck, 2015, Thm. 3.18).

Cross References

gradient step, gradient, step size, SGD, subgradient descent, subgradient, gradient flow, convex, strongly convex, minimum, differentiable, ERM, fixed point, Banach's fixed-point theorem

References

1. Bauschke and Combettes (2011). Convex Analysis and Monotone Operator Theory in Hilbert Spaces. 1st ed., Springer Science+Business Media, New York, NY, USA.

2. Beck (2017). First-Order Methods in Optimization. SIAM-Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

3. Beck and Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci., 2(1), pp. 183–202.

4. Boyd and Vandenberghe (2004). Convex Optimization. Cambridge Univ. Press, Cambridge, U.K.

5. Bubeck (2015). Convex Optimization: Algorithms and Complexity. Found. Trends Mach. Learn., 8(3–4), pp. 231–357.

6. Goodfellow et al. (2016). Deep Learning. MIT Press, Cambridge, MA, USA.

7. Helmke and Moore (1994). Optimization and Dynamical Systems. Springer, London, U.K.

8. Polyak (1987). Introduction to optimization. Optimization Software, Inc., Publications Division, New York.

9. Nesterov (2004). Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic, Boston, MA, USA.