Gradientenabstieg

Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Zusammenfassung

Der Gradientenabstieg ist ein iterativer Algorithmus zur Minimierung einer differenzierbaren Funktion $f : \mathbb{R}^d \to \mathbb{R}$. Jede Iteration aktualisiert den aktuellen Schätzwert durch einen Schritt entlang des negativen Gradienten, skaliert mit einer Schrittweite η. Ein solcher Gradientenschritt lässt sich als Anwendung eines Operators interpretieren, der durch die Schrittweite und die zugrunde liegende Funktion parametrisiert ist. Der Gradientenabstieg und seine Varianten werden im Deep Learning verbreitet zum Trainieren parametrisierter Modelle eingesetzt.

Definition

Beim Training eines tiefen Netzes zur Bilderkennung ist der durchschnittliche Verlust, den das Netz auf einem Trainingsdatensatz gelabelter Bilder erleidet, eine Funktion $f(\mathbf{w})$ des Vektors $\mathbf{w} \in \mathbb{R}^d$ seiner Modellparameter, und Training bedeutet das Minimieren dieser Funktion. Viele Anwendungen des maschinellen Lernens laufen auf ein solches Optimierungsproblem hinaus: Eine Zielfunktion $f : \mathbb{R}^d \to \mathbb{R}$ misst die Güte der Modellparameter $\mathbf{w}$, welche die Optimierungsvariable bilden. Das Minimieren von f auf dem Trainingsdatensatz dient als Ersatz: Das Ziel ist ein kleiner Verlust auf Datenpunkten außerhalb davon (siehe Generalisierung).

In vielen Fällen ist die Zielfunktion differenzierbar: In jedem Punkt $\mathbf{w}$ existiert ein Gradient $\nabla f(\mathbf{w}) \in \mathbb{R}^d$, der Vektor der partiellen Ableitungen von f in $\mathbf{w}$, der eine lokale lineare Approximation von f bestimmt (siehe differenzierbar).

Der Gradientenabstieg nutzt diese lokale lineare Approximation, um die aktuelle Wahl der Modellparameter iterativ zu verbessern. Ausgehend von einer Initialisierung $\mathbf{w}^{(0)}$ erzeugt er eine Folge von Schätzwerten $\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$, die im Idealfall gegen ein Minimum von f konvergieren. In jeder Iteration t verfeinert der Gradientenabstieg den aktuellen Schätzwert $\mathbf{w}^{(t)}$ durch einen Schritt in Richtung des steilsten Abstiegs der lokalen linearen Approximation von f in $\mathbf{w}^{(t)}$. Diese Richtung ist der negative Gradient $\nabla f(\mathbf{w}^{(t)})$, woraus sich die Aktualisierung

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

ergibt, wobei $\eta > 0$ eine Schrittweite ist; die Aktualisierung ist ein Gradientenschritt mit Schrittweite η. Für ein hinreichend kleines η verringert jeder Schritt

den Funktionswert (Funktion), $f(\mathbf{w}^{(t+1)}) \leq f(\mathbf{w}^{(t)})$, mit strikter Abnahme, wann immer $\nabla f(\mathbf{w}^{(t)}) \neq \mathbf{0}$ gilt (Boyd and Vandenberghe, 2004, Abschn. 9.3). Der Gradientenabstieg und seine Varianten sind im Deep Learning die Standardlöser für das Training eines Modells (Goodfellow et al., 2016). Abb. 1 veranschaulicht einen einzelnen Gradientenabstiegsschritt. Algorithmus 1 gibt das Verfahren vollständig an.

Algorithm 1 Gradientenabstieg (Boyd and Vandenberghe, 2004, Algorithmus 9.3)

Input: Zielfunktion f, Schrittweite $\eta > 0$, initialization $\mathbf{w}^{(0)}$, Iteration count T

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: compute the Gradient $\nabla f(\mathbf{w}^{(t)})$
3: $\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)})$
4: **end for**

Output: $\mathbf{w}^{(T)}$

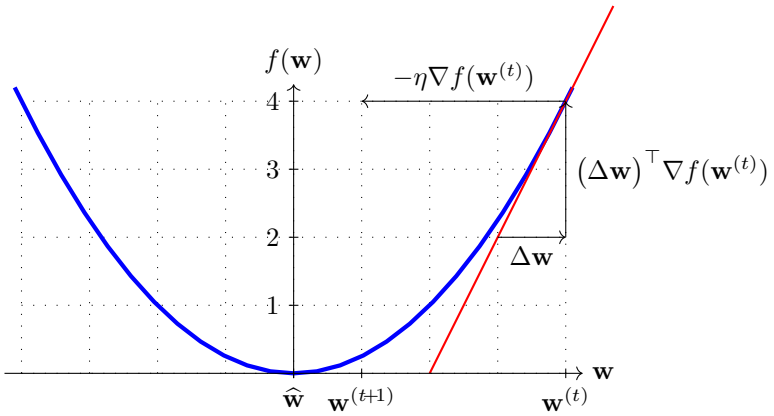

Abbildung 1: Ein einzelner Gradientenschritt (1) in Richtung des Minimierers $\widehat{\mathbf{w}}$ von $f(\mathbf{w})$. Die Kurve stellt die Funktion f dar. Die Gerade ist ihre lokale lineare Approximation am aktuellen Schätzwert $\mathbf{w}^{(t)}$. Eine Änderung $\Delta \mathbf{w}$ des Schätzwerts $\mathbf{w}^{(t)}$ verändert diese Approximation um $(\Delta \mathbf{w})^\top \nabla f(\mathbf{w}^{(t)})$. Ein Schritt entgegen dem Gradienten um $-\eta \nabla f(\mathbf{w}^{(t)})$ liefert den nächsten Schätzwert $\mathbf{w}^{(t+1)}$

Mit einer konstanten Schrittweite η, die über die Iterationen hinweg festgehalten wird, wird ein einzelner Schritt durch den Operator des Gradientenabstiegsschritts $\mathcal{F}^{(\eta)}(\mathbf{w}) = \mathbf{w} - \eta \nabla f(\mathbf{w})$ beschrieben, sodass der Gradientenabstieg die Fixpunktiteration $\mathbf{w}^{(t+1)} = \mathcal{F}^{(\eta)}(\mathbf{w}^{(t)})$ ist. Ihre Fixpunkte, in denen $\mathcal{F}^{(\eta)}(\mathbf{w}) = \mathbf{w}$ gilt, sind genau die stationären Punkte $\nabla f(\mathbf{w}) = \mathbf{0}$. Für konvexes

f sind dies nach der Nullgradientenbedingung die Minima (Boyd and Vandenberghe, 2004, Abschn. 4.2.3).

Die Konvergenz der Fixpunktiteration $\mathbf{w}^{(t+1)} = \mathcal{F}^{(\eta)}(\mathbf{w}^{(t)})$ wird durch die Eigenschaften des Operators des Gradientenabstiegsschritts $\mathcal{F}^{(\eta)}$ bestimmt. Es sei f konvex und sein Gradient L-Lipschitz-Stetigkeit-stetig, d. h.

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

Für $0 < \eta \leq 1/L$ ist der Operator $\mathcal{F}^{(\eta)}$ dann ein nichtexpansiver Operator bezüglich der euklidischen Norm $\|\cdot\|$: Er vergrößert die Abstände zwischen je zwei Punkten $\mathbf{w}, \mathbf{w}'$ nicht,

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

Nichtexpansivität allein garantiert noch keinen eindeutigen Fixpunkt (Bauschke and Combettes, 2011, Kap. 4).

Es sei darüber hinaus f μ-stark konvex, d. h., die Funktion $f(\mathbf{w}) - (\mu/2)\|\mathbf{w}\|^2$ ist weiterhin konvex, sodass die Krümmung von f in jeder Richtung mindestens $\mu > 0$ beträgt. Dann ist $\mathcal{F}^{(\eta)}$ ein kontraktiver Operator bezüglich derselben Norm,

$$\|\mathcal{F}^{(\eta)}(\mathbf{w}) - \mathcal{F}^{(\eta)}(\mathbf{w}')\| \leq (1 - \mu\eta) \|\mathbf{w} - \mathbf{w}'\| \quad \text{für } 0 < \eta \leq 1/L, \tag{2}$$

für beliebige $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$ (Boyd and Vandenberghe, 2004, Abschn. 9.3; Bubeck, 2015, Satz 3.10). In diesem Fall liefern die Kontraktion (2) und der Banachscher Fixpunktsatz einen eindeutigen Fixpunkt $\widehat{\mathbf{w}}$ (den Minimierer von f), gegen den der Gradientenabstieg mit geometrischer Rate konvergiert. Für die lineare Regression lassen sich diese Eigenschaften explizit angeben: Der Gradient des durchschnittlichen quadratischen Fehlerverlusts ist eine affine Funktion von $\mathbf{w}$, und der Kontraktionsfaktor von $\mathcal{F}^{(\eta)}$ wird durch die Eigenwerte der Matrix $\mathbf{X}^\top\mathbf{X}$ bestimmt. Hierbei bezeichnet $\mathbf{X}$ die Merkmalsmatrix, deren Zeilen die Merkmalsvektoren der Datenpunkte im Trainingsdatensatz sind (siehe lineare Regression).

Wie schnell der Gradientenabstieg konvergiert, hängt von der Krümmung von f ab, wie sie durch die obigen Konstanten L und μ quantifiziert wird. Es sei $\widehat{\mathbf{w}}$ ein Minimierer und $f^\star = f(\widehat{\mathbf{w}})$. Für konvexes f mit L-Lipschitz-Stetigkeitstetigem Gradienten und der konstanten Schrittweite $\eta = 1/L$ gilt (Beck, 2017, Satz 10.21)

$$f\big(\mathbf{w}^{(T)}\big) - f^\star \leq \frac{L \|\mathbf{w}^{(0)} - \widehat{\mathbf{w}}\|^2}{2\,T}, \tag{3}$$

sodass nach der Schranke (3) die Suboptimalität nach T Schritten proportional zu $1/T$ fällt: Eine Halbierung der Suboptimalität erfordert eine Verdopplung der Anzahl der Iterationen. Ist f zusätzlich μ-stark konvex, verbessert sich die Schranke auf die geometrische Rate

$$f\big(\mathbf{w}^{(T)}\big) - f^\star \leq \big(1 - \tfrac{\mu}{L}\big)^T \big(f(\mathbf{w}^{(0)}) - f^\star\big). \tag{4}$$

Die geometrische Rate (4) steht im Einklang mit dem Faktor $1-\mu\eta$ des Kontraktionsoperators oben (Boyd and Vandenberghe, 2004, Abschn. 9.3; Bubeck, 2015, Satz 3.10). Eine geometrische Rate bedeutet, dass die Suboptimalität in jeder einzelnen Iteration um mindestens den konstanten Faktor $1 - \mu/L$ schrumpft. Damit erfordert eine Halbierung der Suboptimalität nun eine feste Anzahl von Iterationen, unabhängig davon, wie klein die aktuelle Suboptimalität bereits ist. Momentum (Momentum)-Verfahren, die weiter unten besprochen werden, verbessern beide Schranken.

Die Iteration besitzt auch ein zeitkontinuierliches Gegenstück. Betrachtet wird der Grenzübergang, bei dem die Schrittweite η im Gradientenabstiegsschritt (1) gegen null geht. Für $\eta \to 0$ konvergiert die Iterierte $\mathbf{w}^{(t)}$ gegen $\mathbf{w}(t\eta)$, den Wert zum Zeitpunkt $t\eta$ der zeitkontinuierlichen Trajektorie $\mathbf{w}(\tau)$, welche die gewöhnliche Differentialgleichung $\dot{\mathbf{w}}(\tau) = -\nabla f(\mathbf{w}(\tau))$ erfüllt, die als Gradientenfluss bezeichnet wird. Insbesondere ist der Gradientenabstieg die explizite Euler-Diskretisierung dieses Gradientenflusses mit Schritt η (Helmke and Moore, 1994). Ersetzt man die zeitliche Ableitung $\dot{\mathbf{w}}(\tau)$ durch den Differenzenquotienten $\big(\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\big)/\eta$, erhält man wieder die Aktualisierung (1). Abb. 2 veranschaulicht, wie die Iterierten des Gradientenabstiegs für eine kleine Schrittweite eng der Trajektorie des Gradientenflusses folgen.

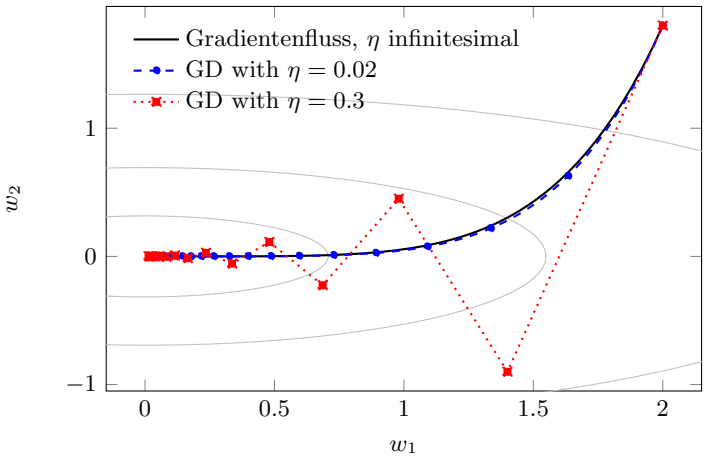

Abbildung 2: Der Gradientenabstieg als explizite Euler-Diskretisierung des Gradientenflusses für die quadratische Funktion $f(\mathbf{w}) = \big(w_1^2 + 5\,w_2^2\big)/2$ mit der Initialisierung $\mathbf{w}^{(0)} = (2, 1.8)^\top$. Die grauen Ellipsen sind Niveaumengen von f; die durchgezogene Kurve ist die exakte Trajektorie des Gradientenflusses. Für die kleine Schrittweite $\eta = 0.02$ folgen die Iterierten des Gradientenabstiegs (gestrichelt, Kreise) der Trajektorie eng. Für die große Schrittweite $\eta = 0.3$ oszillieren die Iterierten (gepunktet, Quadrate) quer zur schmalen Richtung der Niveaumengen. Daten erzeugt mit `pythondemos/gd.py`

Zurück bei der diskreten Iteration sind in zwei verbreiteten Szenarien, die Va-

rianten des Gradientenabstiegs motivieren, die Annahmen hinter diesen Garantien verletzt oder die Auswertung des Gradienten wird zu aufwendig. Erstens: Ist f nichtglatt, existiert sein Gradient möglicherweise nicht, und der Gradientenabstieg wird durch den Subgradientenabstieg ersetzt, der einen Schritt entlang eines Subgradienten anstelle des Gradienten ausführt. Ein konkretes Beispiel ist die Zielfunktion beim Lasso (Lasso), deren Strafterm $\alpha\|\mathbf{w}\|_1$ nichtglatt ist. Zweitens: Bei Verfahren auf Basis der empirischen Risikominimierung ist f ein Durchschnitt von m Verlustfunktionen, einer für jeden Datenpunkt, sodass die Auswertung des vollständigen Gradienten m Gradienten aufsummiert und ihr Aufwand proportional zum Umfang m des Trainingsdatensatzes wächst. Der Gradientenabstieg wird daher durch den stochastischen Gradientenabstieg ersetzt, der in jedem Schritt eine Schätzung des Gradienten aus einer zufälligen Teilmenge des Trainingsdatensatzes verwendet.

Momentum-Verfahren bieten eine weitere Verallgemeinerung des einfachen Gradientenabstiegs: Jede Aktualisierung kombiniert Gradienten aus mehreren vergangenen Schritten statt nur den aktuellen. Der Name entstammt einer physikalischen Analogie: Die Iterierten beschreiben die Bahn eines Teilchens, dessen Impuls (engl. *momentum*) die durch den negativen Gradienten der Zielfunktion ausgeübte Kraft akkumuliert. Ein prototypisches Beispiel für ein Momentum-Verfahren ist Polyaks Heavy-Ball-Verfahren. Es addiert einen Anteil $\beta \in [0, 1)$ des vorangegangenen Zuwachses,

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

Für ein stark konvexes quadratisches f konvergiert diese Iteration schneller als der einfache Gradientenabstieg (Polyak, 1987, Abschn. 3.2.1). Abb. 3 vergleicht den einfachen Gradientenabstieg mit dem Heavy-Ball-Verfahren an einer stark konvexen quadratischen Funktion.

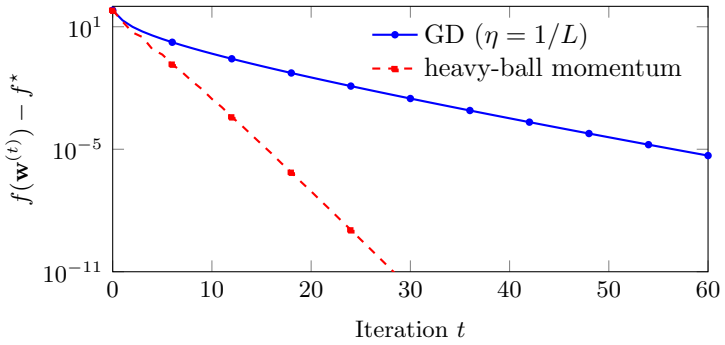

Abbildung 3: Suboptimalität $f(\mathbf{w}^{(t)}) - f^\star$ des einfachen Gradientenabstiegs (durchgezogen, Kreise) und des Heavy-Ball-Momentums (gestrichelt, Quadrate) auf einer stark konvexen quadratischen Funktion, dargestellt auf einer logarithmischen Skala. Beide Folgen fallen geometrisch ab, was auf dieser Skala als Gerade erscheint. Das Momentum-Verfahren verringert die Suboptimalität pro Iteration um einen kleineren Faktor und erreicht daher jede vorgegebene Genauigkeit in weniger Iterationen. Daten erzeugt mit `pythondemos/gd.py`

Ein verwandter Momentum-Algorithmus ist Nesterovs beschleunigte Variante des Gradientenabstiegs, welche die Schranke (3) für konvexes f mit L-Lipschitz-Stetigkeit-stetigem Gradienten verbessert auf

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

Im Gegensatz zur Schranke für den einfachen Gradientenabstieg, die proportional zu $1/T$ fällt, fällt diese Schranke proportional zu $1/T^2$ (Beck and Teboulle, 2009, Satz 4.4; Bubeck, 2015, Satz 3.19). Für μ-stark konvexes f verbessert die Beschleunigung den Kontraktionsfaktor von $1 - \mu/L$ auf $1 - \sqrt{\mu/L}$ (Nesterov, 2004, Abschn. 2.2; Bubeck, 2015, Satz 3.18).

Siehe auch: Gradient, Schrittweite, SGD, Subgradientenabstieg, Subgradient, Gradientenfluss, konvex, stark konvex, Fixpunkt.

Querverweise

Literatur

1. Bauschke and Combettes (2011). Convex Analysis and Monotone Operator Theory in Hilbert Spaces. 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.