stochastischer Gradientenabstieg (SGD)

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 stochastische Gradientenabstieg (SGD) ist eine Variante des Gradientenabstiegs, bei der der Gradient der Zielfunktion durch eine rechnerisch günstigere stochastische Approximation ersetzt wird. Seine Hauptanwendung beim maschinelles Lernen (ML) ist die empirische Risikominimierung (ERM) auf einem Trainingsdatensatz, der groß oder verteilt ist: Der Gradient des empirisches Risikos mittelt über den gesamten Trainingsdatensatz. SGD approximiert ihn durch Mittelung über einen zufällig gezogenen Batch von Datenpunkte. Die Batch-Größe wägt den Rechenaufwand einer einzelnen Aktualisierung gegen die Genauigkeit der Gradientenapproximation ab.

Definition

Betrachtet werden die Wetterstationen aus Abb. 1, von denen jede einen Datenpunkt aufzeichnet: die Lufttemperatur um 06:00 (UTC) eines Tages als Merkmal und die Temperatur um 15:00 (UTC) als Label. Eine Anwendung zur Wettervorhersage nutzt diese Datenpunkte, um eine Hypothese mittels empirische Risikominimierung (ERM) zu lernen. Für ein parametrisches Modell mit Modellparameter $\mathbf{w}$ und einer Verlustfunktion $L\left(\mathbf{z}, \mathbf{w}\right)$ lässt sich die Zielfunktion der ERM schreiben als

$$f(\mathbf{w}) := \frac{1}{m} \sum_{r=1}^{m} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right).$$

Die Zielfunktion f ist der mittlere Verlust über den Trainingsdatensatz $\mathcal{D}^{(\text{train})} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\right\}$ aus m Datenpunkte. Es ist zu beachten, dass die Minimierung von f auf dem Trainingsdatensatz nicht das Hauptziel bei Anwendungen des maschinelles Lernen (ML) ist. Das Ziel besteht darin, eine Hypothese mit einem kleinen Verlust auf Datenpunkte außerhalb des Trainingsdatensatzs zu lernen (siehe Generalisierung).

Für die lineare Regression, die eine lineare Hypothese unter dem quadratischer Verlust lernt, ist die obige Zielfunktion f differenzierbar und konvex. Der Gradientenabstieg nähert sich einem Minimum einer differenzierbaren Funktion $f(\mathbf{w})$ durch Wiederholen des Gradientenschritts

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

mit einer Schrittweite $\eta > 0$, die klein genug ist, damit jeder Schritt f verringert (siehe Gradientenabstieg) (Nesterov, 2004, Thm. 2.1.14). Der Gradient der obigen Zielfunktion ist ein Mittelwert mit einem Term pro Datenpunkt,

$$\nabla f(\mathbf{w}) = \frac{1}{m} \sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right).$$

Seine exakte Auswertung erfordert daher das Sammeln und Verarbeiten des gesamten Trainingsdatensatzs. Das Sammeln des gesamten Trainingsdatensatzs kann aufwendig oder undurchführbar sein, wenn dieser groß oder an verschiedenen Orten gespeichert ist. SGD approximiert die Berechnung des Gradienten, indem die Summe über den gesamten Trainingsdatensatz durch eine Summe über eine Teilmenge des Trainingsdatensatzs ersetzt wird. Diese Teilmenge wird zufällig gewählt und als Batch $\mathcal{B} \subseteq \{1, \ldots, m\}$ mit der Batch-Größe $B = |\mathcal{B}|$ bezeichnet (siehe Abb. 1). Die resultierende Approximation des Gradienten lautet

$$\mathbf{g}^{(\mathcal{B})}(\mathbf{w}) := \frac{1}{B} \sum_{r \in \mathcal{B}} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right).$$

Diese Approximation des Gradienten lässt sich auch als die Approximation von $f(\mathbf{w})$ selbst durch den mittleren Verlust über den Batch interpretieren (siehe Abb. 2).

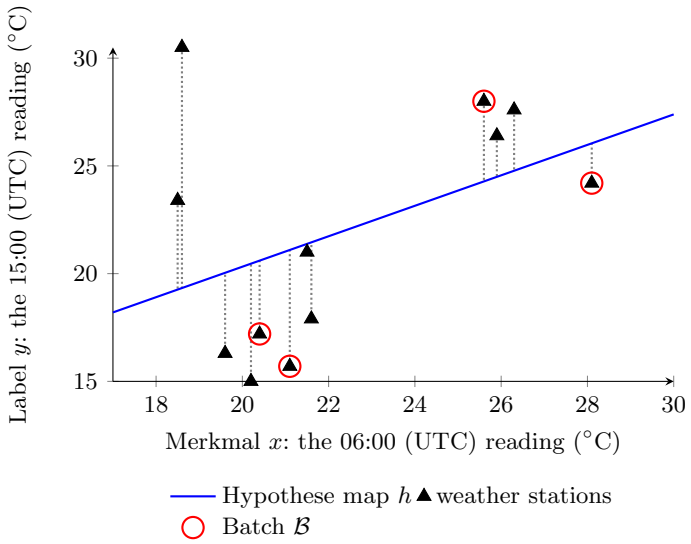

Abbildung 1: Ein Trainingsdatensatz aus $m = 12$ Datenpunkte: Jeder ist die am 1. Juli 2026 an einer anderen Wetterstation (Dreiecke) beobachtete Lufttemperatur um 06:00 (UTC) (Merkmal x) und um 15:00 (UTC) (Label y), wobei jede Station ein eigenes Gerät ist, das den einzelnen von ihr aufgezeichneten Datenpunkt hält. Die gepunkteten vertikalen Segmente sind die Vorhersagefehler einer linearen Hypotheseabbildung h (durchgezogene Kurve). Eine Aktualisierung der Modellparameter $\mathbf{w}$ mittels Gradientenabstieg nutzt die Vorhersagefehler aller m Datenpunkte; eine SGD-Aktualisierung nutzt nur diejenigen eines zufällig gewählten Batches $\mathcal{B}$ (innerhalb roter Ringe). Daten erzeugt mit `pythondemos/stochGD.py`, welches die Beobachtungen vom Open-Data-Dienst von GeoSphere Austria herunterlädt

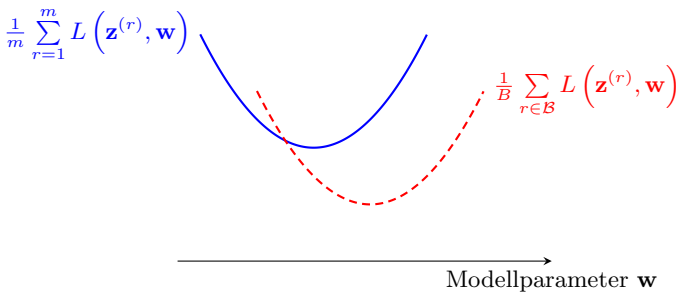

Abbildung 2: SGD als Gradientenabstieg auf einer Approximation der Zielfunktion: das empirisches Risiko $f(\mathbf{w})$, der mittlere Verlust über alle m Datenpunkte des Trainingsdatensatzs (durchgezogene Kurve) und der Mittelwert über einen zufällig gezogenen Batch $\mathcal{B}$ (gestrichelte Kurve). Der Gradient der gestrichelten Kurve ist der stochastische Schätzwert $\mathbf{g}(\mathbf{w})$, den SGD anstelle von $\nabla f(\mathbf{w})$ verwendet

Jede SGD-Iteration erzeugt einen neuen Batch $\mathcal{B}^{(t)}$ und führt die Aktualisierung des Gradientenabstiegs mit $\mathbf{g}(\mathbf{w})$ anstelle des Gradienten $\nabla f(\mathbf{w})$ durch. Die Aktualisierung lautet

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta^{(t)} \underbrace{\frac{1}{B}\sum_{r\in\mathcal{B}^{(t)}} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}^{(t)}\right)}_{=\mathbf{g}^{(\mathcal{B}^{(t)})}\left(\mathbf{w}^{(t)}\right)},$$

mit einer Schrittweite $\eta^{(t)} > 0$. Bedingt auf $\mathbf{w}^{(t)}$ wird der Batch $\mathcal{B}^{(t)}$ unabhängig davon gezogen, sodass jeder Datenpunkt mit gleicher Wahrscheinlichkeit eingeht. Der Erwartungswert der Approximation des Gradienten $\mathbf{g}^{(\mathcal{B}^{(t)})}\left(\mathbf{w}^{(t)}\right)$ ist daher der exakte Gradient $\nabla f(\mathbf{w}^{(t)})$: Wie jeder Mittelwert über gleichverteilt gezogene Datenpunkte ist die Approximation ein unverzerrter Schätzer (Bertsekas and Tsitsiklis, 2008, Sect. 9.1). Die Fixpunkte des Gradientenschritts sind die Punkte, welche die Nullgradientenbedingung $\nabla f(\mathbf{w}) = \mathbf{0}$ erfüllen. Ein klassischer Satz hinreichender Bedingungen für die Konvergenz solcher zufälliger Iterationen gegen ein Minimum von f ist $\sum_{t=1}^{\infty} \eta^{(t)} = \infty$ und $\sum_{t=1}^{\infty} \left(\eta^{(t)}\right)^2 < \infty$ (Robbins and Monro, 1951, Eqs. (6) and (26)).

Der Approximationsfehler des Batch-Gradienten wird durch die Batch-Größe bestimmt. Für gegebenes $\mathbf{w}$ ist der Batch-Gradient das Stichprobenmittel von B gleichverteilt aus dem Trainingsdatensatz gezogenen Gradienten einzelner Datenpunkte, sodass sein mittlerer quadratischer Fehler höchstens die Varianz einer einzelnen Ziehung geteilt durch B beträgt (Bertsekas and Tsitsiklis, 2008, Sect. 9.1),

$$\mathbb{E}\left\{ \left\| \mathbf{g}^{(\mathcal{B})}(\mathbf{w}) - \nabla f(\mathbf{w}) \right\|^2 \,\middle|\, \mathbf{w} \right\} \leq \frac{\sigma^2(\mathbf{w})}{B}, \quad \sigma^2(\mathbf{w}) := \frac{1}{m}\sum_{r=1}^{m} \left\| \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right) - \nabla f(\mathbf{w}) \right\|^2, \tag{1}$$

mit Gleichheit für einen mit Zurücklegen gezogenen Batch. Für ein β-glattes f beschränkt das Abstiegslemma (Nesterov, 2004, Lemma 1.2.3) f nach einer Aktualisierung durch seine lineare Approximation bei $\mathbf{w}^{(t)}$ plus das $\beta/2$-Fache der quadrierten Länge der Aktualisierung. Bildet man den bedingter Erwartungswert für gegebenes $\mathbf{w}^{(t)}$, liefert der lineare Term aufgrund der Erwartungstreue den exakten Gradienten und die quadrierte Länge liefert $\|\nabla f(\mathbf{w}^{(t)})\|^2$ plus den Fehler (1), was den Fortschritt einer einzelnen SGD-Aktualisierung ergibt:

$$\mathbb{E}\big\{ f(\mathbf{w}^{(t+1)}) \,\big|\, \mathbf{w}^{(t)} \big\} \leq f(\mathbf{w}^{(t)}) - \eta^{(t)} \Big(1 - \frac{\beta \eta^{(t)}}{2}\Big) \|\nabla f(\mathbf{w}^{(t)})\|^2 + \frac{\beta \big(\eta^{(t)}\big)^2}{2} \, \frac{\sigma^2(\mathbf{w}^{(t)})}{B}. \tag{2}$$

Die ersten beiden Terme von (2) sind der Fortschritt eines Gradientenschritts mit dem exakten Gradienten; der letzte Term ist der Preis der Approximation des Gradienten, der über (1) mit der Batch-Größe und über ihr Quadrat mit der Schrittweite schrumpft. Die beiden obigen Bedingungen an die Schrittweite machen diesen Preis über die Iterationen summierbar, während der Fortschritt dies nicht ist.

Die Batch-Größe B ist der Parameter von SGD, der den Rechenaufwand seiner Iteration gegen den bei $\mathbf{g}^{(\mathcal{B})}\big(\mathbf{w}\big) \approx \nabla f(\mathbf{w})$ entstehenden Approximationsfehler abwägt. Ein Extremfall ist $B = m$, d. h., der Batch enthält den gesamten Trainingsdatensatz und die Approximation des Gradienten wird exakt: $\mathbf{g}^{(\mathcal{B})}\big(\mathbf{w}\big) = \nabla f(\mathbf{w})$. Bei $B = 1$ wird der Gradient von f anhand eines einzigen Datenpunkts approximiert, was zu einem erheblichen Approximationsfehler führt (siehe Abb. 3).

SGD mit $1 < B < m$ wird als Mini-Batch-SGD bezeichnet (Bottou, 1999). Jede Aktualisierung kostet B Auswertungen des Gradienten $\nabla L\left(\mathbf{z}^{(r)}, \mathbf{w}^{(t)}\right)$ für $r \in \mathcal{B}$, anstelle der m Auswertungen, die für die Berechnung des exakten Gradienten erforderlich sind. Allerdings führt die Aktualisierung einen Approximationsfehler ein, der mit wachsender Batch-Größe abnimmt (siehe Abb. 3).

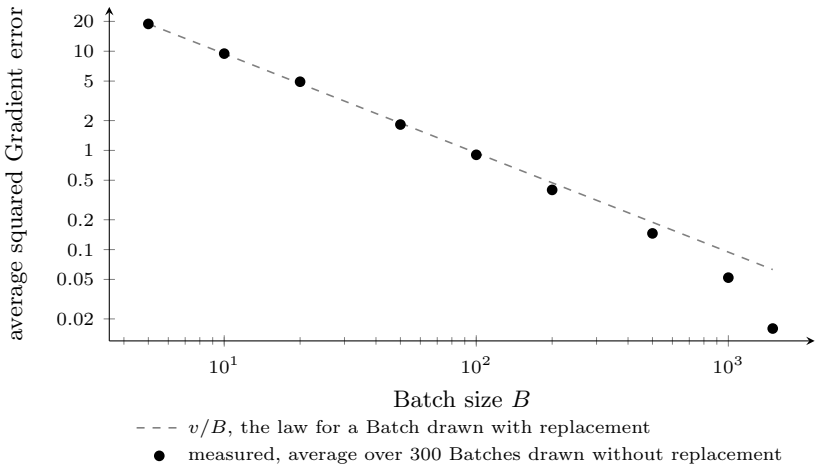

Abbildung 3: Der Approximationsfehler des Gradienten $\|\mathbf{g}^{(\mathcal{B})}(\mathbf{w}) - \nabla f(\mathbf{w})\|^2$ für die Zielfunktion f der lineare Regression: der durchschnittliche quadratischer Verlust eines lineares Modells über $m = 2000$ Datenpunkte mit $d = 5$ Merkmale. Der Approximationsfehler (gefüllte Kreise) ist über der Größe B des Batches aufgetragen und über 300 zufällige Wahlen des Batches gemittelt. Der Gradient $\nabla f(\mathbf{w})$ und seine Approximation $\mathbf{g}^{(\mathcal{B})}(\mathbf{w})$ werden bei denselben festen Modellparameter $\mathbf{w} = \mathbf{0}$ ausgewertet. Die gestrichelte Referenzlinie zeigt v/B, den verbleibenden quadratischen Fehler bei der Mittelung von B unabhängigen Termen der Varianz v, also der Varianz des Gradienten $\nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ für einen zufällig gewählten Datenpunkt $\mathbf{z}^{(r)}$ (Bertsekas and Tsitsiklis, 2008, Abschn. 9.1). Die Messung folgt dieser Referenzlinie, solange B klein gegenüber m ist, und fällt darunter, wenn sich B dem Wert m nähert, da der Batch ohne Zurücklegen gezogen wird: Ein Batch, der einen großen Teil des Trainingsdatensatzs abdeckt, lässt nur wenige Datenpunkte übrig, die sich zwischen den Ziehungen unterscheiden können. Die Darstellung endet bei $B = 1500$; bei $B = m$ würde der Approximationsfehler verschwinden, was auf der logarithmischen Achse nicht darstellbar ist. Daten erzeugt mit `pythondemos/stochGD.py`

Algorithmus 1 gibt das Verfahren vollständig an.

Algorithm 1 SGD (Bottou, 2010, Abschn. 2.2)

Input: Trainingsdatensatz $\mathcal{D}^{(\text{train})}$, Verlustfunktion, Batch size B, Iteration count T, Schrittweiten $\eta^{(t)} > 0$ for $t = 0, \ldots, T-1$, initialization $\mathbf{w}^{(0)}$

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: draw a Batch $\mathcal{B}^{(t)}$ of B indices uniformly at random
3: compute $\mathbf{g}^{(\mathcal{B}^{(t)})}\left(\mathbf{w}^{(t)}\right)$
4: $\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \eta^{(t)}\mathbf{g}^{(\mathcal{B}^{(t)})}\left(\mathbf{w}^{(t)}\right)$
5: **end for**

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

Schritt 3 von Algorithmus 1 wertet B Gradienten der Verlustfunktion aus, von denen jeder bei einem lineares Modell aus d Zahlen besteht, sodass eine Iteration in der Größenordnung von Bd arithmetischen Operationen kostet, gegenüber md für einen exakten Gradienten. Dieser geringere Rechenaufwand von SGD ermöglicht das Training hochdimensionaler Modelle aus großen, dezentralen Sammlungen von Datenpunkte (Bottou, 2010). Ein Paradebeispiel für eine Anwendung des ML, die große dezentrale Daten umfasst, ist die Wettervorhersage: Die Stationen aus Abb. 1 zu Beginn dieses Eintrags halten jeweils einen einzigen Datenpunkt, und ein Batch ist eine Handvoll Stationen, die für einen Gradientenschritt abgefragt werden.

Siehe auch: Gradientenabstieg, gradientenbasiertes Verfahren, Gradient, ERM, Batch, Schrittweite, Zielfunktion, Nullgradientenbedingung, Online-Gradientenabstieg (Online GD).

Querverweise

Literatur

1. Bertsekas and Tsitsiklis (2008). Introduction to Probability. 2nd ed., Athena Scientific, Belmont, MA, USA.

2. Bottou (1999). On-line learning and stochastic approximations. In: On-Line Learning in Neural Networks, pp. 9–42. Cambridge Univ. Press.

3. Bottou (2010). Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer.

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

5. Robbins and Monro (1951). A Stochastic Approximation Method. The Annals of Mathematical Statistics, 22(3), pp. 400–407.