Stochastic Gradient Descent (SGD) und Beispiel

Dr. Hermann Völlinger

17.09.2024

Stochastic Gradient Descent (SGD) - Verfahren

Das Stochastic Gradient Descent (SGD) ist eine Optimierungsmethode, die häufig in maschinellem Lernen und Deep Learning verwendet wird. Sie basiert auf der Idee, schrittweise den Gradient der zu minimierenden Funktion zu berechnen und die Parameter entsprechend anzupassen. Der Prozess funktioniert, indem er in kleinen Schritten in die Richtung der größten Abnahme des Fehlergradienten geht.
Ein gutes Verständnis von Gradient Descent ist entscheidend, um tiefe neuronale Netze effektiv zu trainieren.
Anders als beim klassischen Gradientenabstieg, wo der Gradient der Funktion über den gesamten Datensatz berechnet wird, verwendet SGD nur einen zufälligen Datenpunkt (oder eine kleine Gruppe von Datenpunkten, einen sogenannten *Mini-Batch*) für jede Aktualisierung.

Grundlegender Algorithmus

Für eine Funktion $f(x)$, die wir minimieren wollen, berechnen wir den Gradienten $\nabla f(x)$ und aktualisieren den Parameter x in die Richtung des negativen Gradienten:

$$x_{\text{neu}} = x_{\text{alt}} - \eta \nabla f(x_{\text{alt}})$$

wobei η die Lernrate ist, die bestimmt, wie groß der Schritt in Richtung des Minimums sein soll.

Unterschied zum klassischen Gradientenabstieg

Im klassischen Gradientenabstieg wird der Gradient über den gesamten Datensatz berechnet, was bei großen Datenmengen teuer ist. Beim SGD wird stattdessen ein zufälliger Datenpunkt ξ ausgewählt, und die Aktualisierung erfolgt basierend auf dem Gradient dieses Punktes:

$$x_{\text{neu}} = x_{\text{alt}} - \eta \nabla f(x_{\text{alt}}, \xi)$$

Dadurch entsteht ein stochastisches Element im Algorithmus, was dazu führt, dass die Gradientenaktualisierungen *rauschen* und zufällig variieren.

Vorteile von SGD

Ein wesentlicher Vorteil von SGD ist seine Fähigkeit, lokale Minima zu vermeiden. Durch die zufälligen Schwankungen in den Gradientenaktualisierungen kann SGD aus einem lokalen Minimum *herausspringen* und sich weiter in Richtung eines tieferen Minimums bewegen. Darüber hinaus ist es effizienter bei der Verarbeitung großer Datenmengen, da nicht der gesamte Datensatz auf einmal betrachtet werden muss.

Abbruchkriterium

Das SGD-Verfahren wird iterativ durchgeführt, bis eine Konvergenz erreicht ist. Das Abbruchkriterium kann beispielsweise eine ausreichend kleine Differenz zwischen den Funktionswerten in aufeinanderfolgenden Iterationen sein. In diesem Beispiel verwenden wir als Abbruchkriterium, dass der Unterschied $|f(x_{n+1}) - f(x_n)|$ kleiner als ein Schwellwert $\epsilon = 0.0001$ wird.

Beispiel

Die zu minimierende Funktion lautet:

$$f(x) = x^4 - 4x^2 + x + 3$$

Die Ableitung der Funktion, also der Gradient, ist:

$$f'(x) = 4x^3 - 8x + 1$$

Wir setzen die folgenden Parameter:

- Startwert: $x_0 = 0$
- Lernrate: $\eta = 0.01$
- Maximale Anzahl der Iterationen: 1000
- Konvergenzschwelle: 10^{-6}

Iterationsprozess

In jeder Iteration wird der neue Wert von x mit folgender Formel berechnet:

$$x_{\text{neu}} = x_{\text{alt}} - \eta \cdot f'(x_{\text{alt}})$$

wobei die Ableitung $f'(x)$ wie folgt lautet:

$$f'(x) = 4x^3 - 8x + 1$$

Die Iterationen werden fortgesetzt, bis die Änderung in x kleiner als die Konvergenzschwelle von 10^{-6} ist.

Ergebnisse

Nach ungefähr 150 Iterationen ergibt sich das folgende Minimum:

- Das minimale x:
$$x_{\min} \approx -1.473$$
- Der minimale Funktionswert:
$$f(x_{\min}) \approx -2.444$$

Die Iterationen enden, wenn der Betrag der Änderung in x kleiner als die Toleranzschwelle wird.
Dies hängt in diesem Fall aber auch stark am Startpunkt ab. Zudem ist die Steigung an dieser Stelle in Richtung globales Minimum größer als in die andere Richtung Hätte man eine Startpunkt etwa bei $x = 1.5$ gewählt wäre man beim lokalen Minimum bei $x = 1.36$ gelandet.
Siehe dies auch genauer im Diagramm des Lösungsansatzes mit Python: