Training

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

Das Training ist der Prozess der Auswahl einer nützlichen Hypothese aus einem Hypothesenraum oder Modell durch Lösen eines Optimierungsproblems, dessen Zielfunktion eine Trainingsmethode aus den eingelesenen Datenpunkten aufbaut. Beim überwachtes Lernen ist die Zielfunktion der mittlere Verlust über Datenpunkte, deren Label-Werte bekannt sind, wie etwa der mittlere quadratischer Verlust. Beim unüberwachtes Lernen misst sie stattdessen die Struktur, wie etwa den Clusteringfehler, der durch eine Wahl von Clusterzentrenn entsteht. Beim bestärkendes Lernen (RL) ist sie die erwartete kumulierte Belohnung, die von einer Policy (bestärkendes Lernen) gesammelt wird. Trainingsmethoden unterscheiden sich auch darin, wie sie auf Datenpunkte zugreifen: Offline-Methoden erfordern das Trainingsdatensatz im Voraus, während Online-Methoden die Hypothese immer dann aktualisieren, wenn neue Datenpunkte eintreffen. Viele Trainingsmethoden iterieren einen Operator auf den Modellparametern eines parametrisches Modells: einen Gradientenschritt, wo die Zielfunktion differenzierbar ist, und einen Durchlauf alternierender Optimierung bei den k-Means- und EM-Algorithmus (EM)-Algorithmen. Eine Offline-Methode wendet in jedem Schritt denselben Operator an und hält an einem Fixpunkt desselben an, während eine Online-Methode einen Operator anwendet, den der eintreffende Datenpunkt verändert. Diese Operatoren machen die Zielfunktion auf den Datenpunkten klein, die der Methode gegeben sind, und die Kontrolle des Verlustes auf Datenpunkten außerhalb des Trainingsdatensatzs ist eine Herausforderung, die keine Wahl des Optimierungsverfahrens beseitigt.

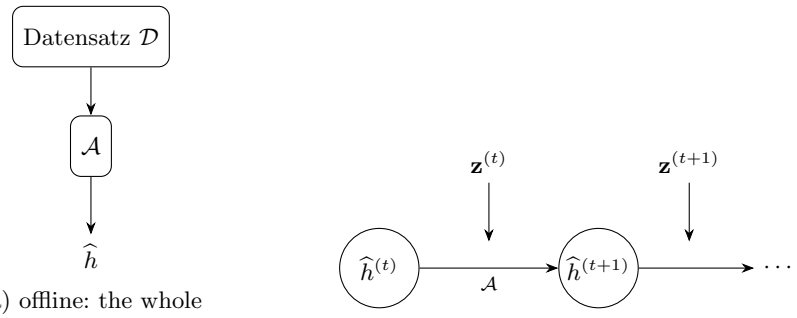
Definition

Ein Geschäft prognostiziert die tägliche Brotnachfrage aus dem Wetter mit einem lineares Modell. Es trainiert dieses lineares Modell mittels Gradientenabstieg auf den Aufzeichnungen eines vergangenen Jahres. Damit die Prognose stattdessen einer sich ändernden Kundschaft folgen kann, trainiert es dasselbe lineares Modell mittels Online-Gradientenabstieg (Online GD) auf den Verkäufen jedes neuen Tages: dasselbe Modell und dieselbe Zielfunktion, mit einer anderen Form des Zugriffs auf die Daten. Das Geschäft trainiert in beiden Fällen. Das Training ist der Prozess, durch den eine Methode des maschinelles Lernen (ML)s eine Hypothese aus einem Hypothesenraum auswählt. Der Hypothesen-

raum $\mathcal{H}$, oder das Modell, ist die Menge der Kandidatenabbildungen, aus denen die Methode wählen kann. Das Training wählt die einzelne Hypothese $\widehat{h}$ aus, die die Methode dann für die Inferenz verwendet: Die Trainingsphase lernt $\widehat{h}$, und die Phase der Inferenz wendet $\widehat{h}$ unverändert auf neu eintreffende Datenpunkte an, um Vorhersagen zu liefern. Was einen Kandidaten besser als einen anderen macht, ist eine Zielfunktion, sodass das Training auf das Lösen eines Optimierungsproblems hinausläuft.

Trainingsmethoden unterscheiden sich in zweierlei Hinsicht. Die erste ist die Zielfunktion. Beim überwachtes Lernen trägt jeder Datenpunkt ein Label, und die Zielfunktion ist der mittlere Verlust einer Hypothese über das Trainingsdatensatz, ihr empirisches Risiko, wie etwa der mittlere quadratischer Verlust bei der lineare Regression. Beim unüberwachtes Lernen ist kein Label gegeben, sodass die Zielfunktion stattdessen eine strukturelle Größe misst, wie etwa den Clusteringfehler, der durch eine Wahl von Clusterzentrenn beim hartes Clustering entsteht. Beim bestärkendes Lernen (RL) gibt es überhaupt kein festes Trainingsdatensatz: Die Zielfunktion ist die erwartete kumulierte Belohnung G, die von einer Policy (bestärkendes Lernen) gesammelt wird, welche mit einer Umgebung interagiert.

Die zweite Hinsicht ist die Form des Zugriffs auf die Daten, offline oder online. Eine Trainingsmethode ist ein Algorithmus $\mathcal{A}$, der einen Datensatz $\mathcal{D}$ von Datenpunkten einliest, daraus zusammen mit dem gewählten Modell und dem Verlust die Zielfunktion konstruiert und die Hypothese liefert, die diese Zielfunktion klein macht, $\widehat{h} = \mathcal{A}(\mathcal{D})$. Ein Offline-Algorithmus liest alle Datenpunkte auf einmal ein, rechnet und liefert $\widehat{h}$. Ein Online-Algorithmus liest neue Datenpunkte ein, sobald sie eintreffen, und aktualisiert die aktuelle Hypothese zu jedem Zeitpunkt $t = 1, 2, \ldots$, anstatt auf ein vollständiges Trainingsdatensatz zu warten. Die Hypothesen, die er erzeugt, bilden eine Markow-Kette, da jede einzelne vom Datensatz nur über ihren Vorgänger und den gerade eingetroffenen Datenpunkt abhängt (Abb. 1).



(a) offline: the whole Datensatz at once

(b) online: one Datenpunkt per time instant

Abbildung 1: Zwei Formen des Zugriffs auf Daten. In (a) liest ein Offline-Algorithmus $\mathcal{A}$ den gesamten Datensatz $\mathcal{D}$ ein und liefert eine gelernte Hypothese $\widehat{h}$. In (b) hält ein Online-Algorithmus eine aktuelle Hypothese $\widehat{h}^{(t)}$, die durch den zum Zeitpunkt t eintreffenden Datenpunkt $\mathbf{z}^{(t)}$ in $\widehat{h}^{(t+1)}$ überführt wird. Die Hypothesen in (b) bilden daher eine Markow-Kette: Die nächste hängt vom Datensatz nur über die aktuelle und den eintreffenden Datenpunkt ab

Sowohl die Inferenz als auch das Online-Training wirken auf einen neu eintreffenden Datenpunkt, und dieser Datenpunkt spielt jeweils eine andere Rolle. Die Inferenz wendet eine Hypothese an, die das Training bereits geliefert hat, und belässt sie unverändert, sodass der eintreffende Datenpunkt nur die Merkmale liefert, aus denen vorhergesagt wird. Das Online-Training verändert die Hypothese fortlaufend, sodass der eintreffende Datenpunkt den Verlust liefert, der über die nächste Hypothese entscheidet. Eine Online-Methode tut in jeder Runde beides: Sie legt sich auf eine Hypothese fest und trifft damit eine Vorhersage, der Verlust dieser Hypothese wird dann aufgedeckt, und die nächste Hypothese wird aus den bisher aufgedeckten Verlusteen gewählt (Hazan, 2022, Abschn. 1.1).

Für ein parametrisches Modell wird jede Hypothese $h^{(\mathbf{w})}$ durch eine Wahl von Modellparametern $\mathbf{w}$ festgelegt, sodass das Training über $\mathbf{w}$ sucht. Viele Trainingsmethoden laufen darauf hinaus, einen Operator auf diesen Modellparametern zu iterieren,

$$\mathbf{w}^{(t+1)} = T(\mathbf{w}^{(t)}), \text{ für } t = 0, 1, \ldots,$$

und die zwei Formen des Zugriffs auf die Daten erscheinen hier als zwei Formen des Operators. Eine Offline-Methode baut ihre Zielfunktion einmalig aus dem gesamten Datensatz auf, sodass T fest ist: In jedem Schritt wird dieselbe Abbildung angewendet, und das Training hält an oder nahe einem Fixpunkt von T an, wo eine weitere Iteration die Modellparameter unverändert lässt. Eine Online-Methode hat keine feste Zielfunktion. Ihr Operator liest den eintreffenden Datenpunkt ein,

$$\mathbf{w}^{(t+1)} = T\big(\mathbf{w}^{(t)}, \mathbf{z}^{(t)}\big),$$

sodass sich die Abbildung von Schritt zu Schritt ändert und es im Allgemeinen keinen einzelnen Fixpunkt gibt, an dem angehalten werden kann. Die Modellpa-

rameter ändern sich weiter, solange neue Datenpunkte eintreffen. Die gelernte Hypothese $\widehat{h}$ ist diejenige, die durch die abschließenden Modellparameter definiert wird.

Was T tut, wird durch die Gestalt der Zielfunktion entschieden. Wenn die Zielfunktion in allen Modellparametern gleichzeitig differenzierbar ist, ist T ein Gradientenschritt, $T(\mathbf{w}) = \mathbf{w} - \eta \nabla f(\mathbf{w})$ mit Schrittweite η (Boyd and Vandenberghe, 2004, Abschn. 9.3). Die Form des Zugriffs auf die Daten entscheidet dann, von welcher Zielfunktion der Gradient gebildet wird, sodass ein einziger Operator drei verschiedene Trainingsmethoden liefert. Die Offline-empirische Risikominimierung (ERM) bildet den Gradienten des mittleren Verlustes $f(\mathbf{w})$ über das gesamte Trainingsdatensatz. Wenn dieses Trainingsdatensatz zu groß ist, um in jedem Schritt vollständig durchlaufen zu werden, ersetzt der stochastischer Gradientenabstieg (SGD) $\nabla f(\mathbf{w})$ durch den Gradienten des mittleren Verlustes über einige wenige Datenpunkte (Bottou, 2010). Sein Operator wird in jedem Schritt zufällig aus einer Verteilung gezogen, die das Trainingsdatensatz festlegt, sodass er im Erwartungswert fest ist und nicht absolut fest. Beim Online-Lernen gibt es überhaupt kein Trainingsdatensatz, und der Online-Gradientenabstieg führt einen Gradientenschritt allein auf dem Verlust des eintreffenden Datenpunkts aus (Zinkevich, 2003).

Einige Zielfunktionen, die in Trainingsmethoden auftreten, lassen sich leicht minimieren, sobald eine Teilmenge der Modellparameter festgehalten wird. Dann ist T ein Durchlauf alternierender Optimierung, eine Minimierung über jede Teilmenge nacheinander (Beck, 2017, Kap. 14). Der k-Means-Algorithmus wechselt zwischen der Zuweisung jedes Datenpunkts zum nächstgelegenen Cluster und der Neuberechnung der Cluster-Mittelwerte ab (Lloyd, 1982). Der EM-Algorithmus (EM)-Algorithmus ist das probabilistische Gegenstück: Beim Training eines Gaußsches Mischmodell (GMM) wechselt er zwischen den Zuständigkeiten der Mischungskomponenten für jeden Datenpunkt und den Parametern dieser Komponenten ab (Dempster et al., 1977). Was EM von einer einfachen Alternierung unterscheidet, ist, dass jeder Schritt ein Surrogat minimiert, eine Schranke für die Zielfunktion anstelle der Zielfunktion selbst, weshalb die Likelihood-Funktion nicht abnehmen kann (Bishop, 2006, Abschn. 9.4). Beide arbeiten offline. Der k-Means-Algorithmus erreicht nach endlich vielen Schritten einen Fixpunkt seines Operators, da jeder Schritt die Zielfunktion verringert und es nur endlich viele Zuweisungen gibt (Bishop, 2006, Abschn. 9.1); bei EM konvergiert unter einer schwachen Stetigkeitsbedingung die Likelihood-Funktion, und jeder Häufungspunkt der Iterierten ist ein stationärer Punkt derselben (Wu, 1983, Thm. 2). Für keinen von beiden ist garantiert, dass er den kleinsten in $\mathcal{H}$ verfügbaren Verlust erreicht: k-Means kann an einem lokalen Minimum seiner Zielfunktion anhalten (Bishop, 2006, Abschn. 9.1), und ein stationärer Punkt der Likelihood-Funktion muss diese nicht maximieren.

Sowohl ein Gradientenschritt als auch ein Durchlauf alternierender Optimierung machen die Zielfunktion auf den Datenpunkten klein, die der Trainingsmethode gegeben sind, und mehr leistet keines von beiden. Die Kontrolle der Diskrepanz zwischen dem Verlust auf dem Trainingsdatensatz und dem Verlust

auf Datenpunkten außerhalb desselben ist eine der zentralen Herausforderungen im ML, und keine Wahl des Optimierungsverfahrens beseitigt sie (Goodfellow et al., 2016).

Synonyme: Modellanpassung, Anpassung.

Siehe auch: Optimierungsproblem, Inferenz, überwachtes Lernen, unüberwachtes Lernen, RL, ERM, Gradientenabstieg, SGD, Online-Lernen, EM, GMM, k-Means, Fixpunkt, Modell, Modellparameter, Verlust.

Querverweise

Literatur

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

2. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

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

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

5. Hazan (2022). Introduction to Online Convex Optimization. 2nd ed., MIT Press, Cambridge, MA, USA.

6. Lloyd (1982). Least squares quantization in PCM. IEEE Trans. Inf. Theory, 28(2), pp. 129–137.

7. Wu (1983). On the convergence properties of the EM algorithm. Ann. Statist., 11(1), pp. 95–103.

8. Zinkevich (2003). Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In: Proc. 20th Int. Conf. Mach. Learn., pp. 928–935.

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

10. Dempster et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc.: Ser. B (Methodological), 39(1), pp. 1–22.