tiefes Netz

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

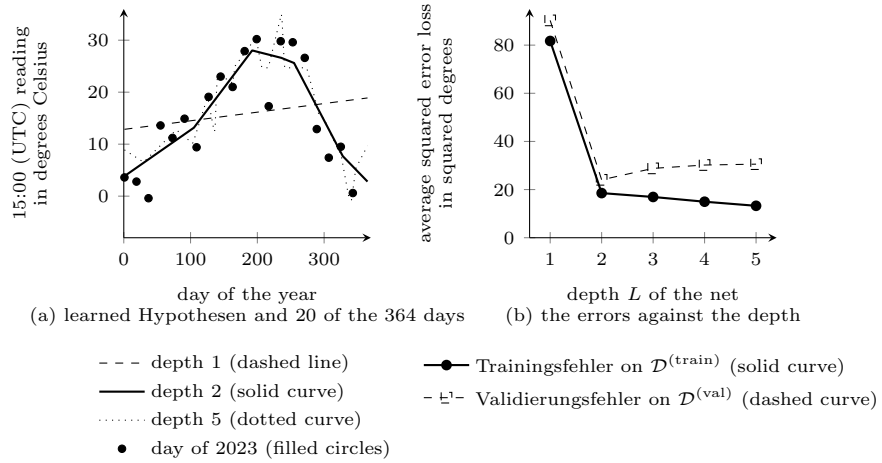
Zusammenfassung

Ein tiefes Netz ist ein künstliches neuronales Netz (KNN), dessen Hypothese Abbildungen Kompositionen mehrerer Schichten sind, wobei jede eine affine Abbildung und dann eine Aktivierungsfunktion auf jeden Eintrag des Ergebnisses anwendet. Die Anzahl der Schichten, die lernbare Modellparameter tragen, ist die Tiefe des Netzes, und deep learning ist der Name der machine learning (ML)-Methoden, die ein tiefes Netz als ihr Modell verwenden. Die Schichten vor der letzten berechnen eine aus den Daten gelernte Merkmalstransformation, und die letzte Schicht wendet eine lineare Hypothese auf die von ihnen gelieferten Aktivierungen an. Ein tiefes Netz macht daher die Wahl der Merkmale zu einem Teil des Training anstatt zu einer im Voraus getroffenen Entwurfsentscheidung. Seine Modellparameter, die Gewichte und Offset-Terme jeder Schicht, werden durch Gradientenabstieg auf dem Trainingsfehler gelernt, wobei der Gradient durch Backpropagation berechnet wird. Die Tiefe ist eine Eigenschaft des Netzes, nicht einer einzelnen Hypothese: Das Hinzufügen von Schichten vergrößert den Hypothesenraum und senkt den Trainingsfehler, während jenseits einer bestimmten Tiefe der Validierungsfehler wieder ansteigt, ein Fall von Überanpassung. Die Tiefe ist daher ein Hyperparameter, und eine im Voraus gewählte Merkmalstransformation kann einen niedrigeren Validierungsfehler mit weitaus weniger Modellparameter erreichen.

Definition

Die Lufttemperatur um 15:00 Uhr (UTC) in Krems an der Donau (Österreich) steigt von Januar bis Juli an und fällt zum Dezember hin wieder ab. Die 364 Tage des Jahres 2023, die diesen Messwert tragen, bilden einen Trainingsdatensatz $\mathcal{D}^{(\text{train})}$ und die 365 solchen Tage des Jahres 2024 einen Validierungsdatensatz $\mathcal{D}^{(\text{val})}$, wobei der Tag des Jahres das einzige Merkmal eines Tages und der Messwert das Label ist. Eine Hypothese, die eine lineare Funktion des Datums ist, kann über das Jahr hinweg nur steigen oder nur fallen. Sein Trainingsfehler, der durchschnittliche squared error loss über $\mathcal{D}^{(\text{train})}$, beträgt 81.7 Quadratgrad, kaum unter den 84.8 der konstanten Hypothese, die immer den Stichprobenmittel der Labels vorhersagt. Ein tiefes Netz ersetzt dieses einzelne rohe Merkmal durch Merkmale, die es selbst berechnet. Eine verborgene Schicht von acht künstliche Neuronen wandelt das Datum in acht Aktivierungen um,

und eine lineare Hypothese dieser acht Zahlen hat einen Trainingsfehler von 18.5 und einen Validierungsfehler von 24.1 auf $\mathcal{D}^{(\text{val})}$ (Abb. 1(a)).



(a) learned Hypothesen and 20 of the 364 days (b) the errors against the depth

Abbildung 1: Netze der Tiefe 1 bis 5, trainiert auf den 364 Tagen des Jahres 2023 in Krems, wobei jede verborgene Schicht acht künstliche Neuronen breit ist. (a) Die gelernten Hypothesen der Tiefe 1 (gestrichelte Linie), Tiefe 2 (durchgezogene Kurve) und Tiefe 5 (gepunktete Kurve) über das Jahr, mit 20 der 364 Tage, jedem 18. Tag in kalendarischer Reihenfolge (gefüllte Kreise). (b) Der Trainingsfehler (durchgezogene Kurve, gefüllte Kreise) und der Validierungsfehler auf den 365 Tagen des Jahres 2024 (gestrichelte Kurve, offene Quadrate) gegen die Tiefe. Daten generiert von `pythondemos/deepnet.py`

Ein tiefes Netz ist ein künstliches neuronales Netz (KNN), dessen Hypothese Abbildungen Kompositionen von L aufeinanderfolgenden Schichten sind. Die Schicht ℓ liest die Aktivierungen $\mathbf{a}^{(\ell-1)}$ der vorherigen Schicht ein, wendet eine affine Abbildung auf sie an und dann eine Aktivierungsfunktion σ auf jeden Eintrag des Ergebnisses,

$$\mathbf{a}^{(\ell)} = \sigma\big(\mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{b}^{(\ell)}\big) \text{ für } \ell = 1, \ldots, L-1, \tag{1}$$

beginnend mit dem rohen Merkmalsvektor $\mathbf{a}^{(0)} := \mathbf{x}$ eines Datenpunkt. Die letzte Schicht wendet allein eine affine Abbildung an und gibt die Vorhersage zurück

$$h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{b}^{(L)}. \tag{2}$$

Die Anzahl L der Schichten, die lernbare Modellparameter tragen, ist die Tiefe des Netzes, und der Begriff *deep learning* leitet sich davon ab (Bishop and Bishop, 2024, Sect. 6.3; Goodfellow et al., 2016, S. 164). Deep learning umfasst die machine learning (ML)-Methoden, die ein tiefes Netz als ihr Modell verwenden. Die Aktivierungsfunktion ist vorgegeben statt gelernt, meistens die rectified linear unit (ReLU) $\sigma(z) = \max\{0, z\}$ (Goodfellow et al., 2016, Sect. 6.3). Ohne

sie wäre jede Schicht eine affine Abbildung, die Schichtrekursion (1) und die Ausgabeabbildung (2) würden sich wieder zu einer affinen Abbildung zusammensetzen, und das Netz würde denselben Hypothesenraum wie ein lineares Modell definieren (Abb. 2).

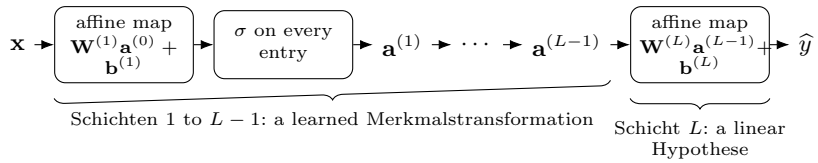


Abbildung 2: Ein tiefes Netz der Tiefe L als Komposition von Schichten. Jede der ersten $L-1$ Schichten wendet eine affine Abbildung auf die Aktivierungen der vorherigen an und dann die Aktivierungsfunktion σ auf jeden Eintrag, und die letzte Schicht wendet allein eine affine Abbildung an, so dass die Vorhersage $\widehat{y}$ eine lineare Hypothese der gelernten Merkmale $\mathbf{a}^{(L-1)}$ ist

Die Komposition teilt sich an der letzten Schicht, und diese Teilung macht ein tiefes Netz zu einer Repräsentationslernen-Methode. Die Schichten 1 bis $L-1$ bilden den rohen Merkmalsvektor $\mathbf{x}$ auf die Aktivierungen $\mathbf{a}^{(L-1)}$ ab, eine aus den Daten gelernte Merkmalstransformation, und die Ausgabeabbildung (2) wendet eine gewöhnliche lineare Hypothese auf diese gelernten Merkmale an (Goodfellow et al., 2016, Sect. 6.1). Ein tiefes Netz macht daher die Wahl der Merkmale zu einem Teil des Training anstatt zu einer im Voraus getroffenen Entwurfsentscheidung. Ein tiefes Netz, dessen letzte verborgene Schicht zwei künstliche Neuronen hat, wandelt das Datum in Krems in gelernte Merkmale um. Least squares auf ihnen erreicht einen Trainingsfehler von 16.8, gegenüber den 17.6 der letzten Schicht des Netzes selbst und den 81.7 von least squares auf dem rohen Merkmal. Zwei künstliche Neuronen müssen nicht zwei Merkmale liefern: In diesem Durchlauf ist eine davon an jedem Tag von $\mathcal{D}^{(\text{train})}$ null, eine ReLU, die kein Datenpunkt aktiviert, so dass das Netz ein Merkmal gelernt hat und der obige least squares-Wert die eigene letzte Schicht des Netzes um 5 Prozent übertrifft, wobei der Gradientenabstieg-Lauf gestoppt wurde, bevor diese Schicht die beste lineare Abbildung der Aktivierungen war. Eine gelernte Merkmalstransformation ist oft über die Lernaufgabe hinaus nützlich, für die sie trainiert wurde, so dass die Schichten vor der letzten beibehalten und nur die letzte Schicht auf einem kleineren Datensatz erneut gelernt wird (Bishop and Bishop, 2024, Sect. 6.3.4).

Die Modellparameter $\mathbf{w}$ sammeln die Gewichte in jeder Matrix $\mathbf{W}^{(\ell)}$ und die Offset-Terme in jedem Vektor $\mathbf{b}^{(\ell)}$, und eine Wahl von $\mathbf{w}$ wählt eine Hypothese $h^{(\mathbf{w})}$ des Netzes aus. Sie werden durch empirische Risikominimierung (ERM) gelernt: Gradientenabstieg mit einer festen Lernrate senkt den Trainingsfehler $\widehat{L}\big(h^{(\mathbf{w})}|\mathcal{D}^{(\text{train})}\big)$ Schritt für Schritt um einen Gradientenschritt nach dem anderen. Jeder von ihnen benötigt den Gradient des Trainingsfehler in Bezug auf $\mathbf{w}$, und Backpropagation berechnet ihn Schicht für Schicht (Rumelhart et al., 1986).

Die Tiefe ist eine Eigenschaft des Netzes, nicht einer einzelnen Hypothese darin, und was eine größere Tiefe ändert, ist die Größe des Hypothesenraum. Abb. 1(b) zeigt den Trainingsfehler in Krems, der von 81.7 bei Tiefe 1 auf 13.3 bei Tiefe 5 fällt, wobei jede verborgene Schicht acht künstliche Neuronen breit ist. Der Validierungsfehler folgt ihm nicht nach unten. Er ist am kleinsten bei Tiefe 2, bei 24.1, und steigt auf 30.6 bei Tiefe 5 an, ein Fall von Überanpassung. Die Tiefe ist daher ein Hyperparameter, der auf $\mathcal{D}^{(\text{val})}$ gewählt wird, anstatt im Voraus festgelegt zu werden.

Das Netz der Tiefe 2 senkt den Validierungsfehler auf 24.1 von den 90.3 der least squares auf dem rohen Merkmal. Eine Merkmalstransformation im Voraus gewählt erreicht einen niedrigeren Validierungsfehler mit einem Fünftel der Modellparameter. Least squares auf dem Sinus und Kosinus des Datums zur jährlichen Periode und zur Hälfte davon benötigt fünf Modellparameter und hat einen Validierungsfehler von 22.9, gegenüber den 24.1 des Netzes mit seinen 25. Zwei weitere Kosten entstehen mit dem Netz. Sein Trainingsfehler ist keine konvex Funktion von $\mathbf{w}$ und hat viele lokale Minima (Goodfellow et al., 2016, Sect. 8.2.2). Gradientenabstieg liefert daher aus jeder Initialisierung eine andere Hypothese: Fünf Initialisierungen des Netzes der Tiefe 2 ergeben Trainingsfehler zwischen 17.1 und 20.6, und vier der fünf enden über dem kleinsten davon. Die gelernten Merkmale tragen auch keine eigene Bedeutung, wo der Sinus und Kosinus den Jahreszyklus benennen. Eine erklärbare künstliche Intelligenz (XAI)-Methode erklärt das trainierte Netz nur durch eine Annäherung, die ihm nicht vollständig treu sein kann (Rudin, 2019, S. 207). Ein tiefes Netz ist die Antwort, wo keine solche Merkmalstransformation im Voraus aufgeschrieben werden kann: Bei der Bild-Klassifikation liefert die erste verborgene Schicht Kanten, die zweite Ecken und Konturen und die dritte Teile von Objekten (Goodfellow et al., 2016, Abb. 1.2).

Synonyme: tiefes neuronales Netz.

Siehe auch: KNN, Schicht, Aktivierung, Aktivierungsfunktion, ReLU, Backpropagation, deep learning, Repräsentationslernen, Merkmalstransformation, Überanpassung, Hyperparameter, interpretierbares maschinelles Lernen, Convolutional Neural Network (CNN), großes Sprachmodell (LLM).

Querverweise

Literatur

1. Bishop and Bishop (2024). Deep Learning: Foundations and Concepts. Springer Nature, Cham, Switzerland.

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

3. Rumelhart et al. (1986). Learning Representations by Back-Propagating Errors. Nature, 323, pp. 533–536.

4. Rudin (2019). Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead. Nature Mach. Intell., 1(5), pp. 206–215.