Repräsentationslernen

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 Repräsentationslernen ist die Aufgabe, aus einem Datensatz eine Merkmalstransformation zu lernen, welche die rohen Merkmale eines Datenpunkts in neue Merkmale überführt, die eine nachfolgende Lernaufgabe erleichtern. Ein Verfahren des Repräsentationslernens wird durch die beiden Merkmalsräume, durch einen Hypothesenraum von infrage kommenden Merkmalstransformationen und durch ein quantitatives Maß dafür festgelegt, wie nützlich eine Merkmalstransformation ist, berechnet auf einem Datensatz, der keine Labels tragen muss. Die Hauptkomponentenanalyse (PCA) lernt eine lineare Merkmalstransformation auf weniger Merkmale, bewertet anhand des linearen Rekonstruktionsfehlers, und ein Autoencoder gewinnt nach demselben Kriterium eine nichtlineare. Die verdeckten Schichten eines tiefen Netzes bilden eine Kette von Merkmalstransformationen, die gemeinsam mit der Ausgabe-Schicht gelernt werden, sodass jede Schicht ihre Merkmale aus denen der vorangehenden aufbaut. Das Vortraining eines Encoders auf ungelabelten Datenpunkte lernt Merkmale, die über mehrere Lernaufgaben hinweg wiederverwendet werden. In der Wettervorhersage werden vierzig Messwerte der fünf vorangehenden Tage in zwei gelernte Merkmale überführt, aus denen die lineare Regression die Höchsttemperatur des nächsten Tages vorhersagt.

Definition

Betrachtet sei die Aufgabe, das Wetter von morgen an einer Wetterstation aus den Messwerten der fünf vorangehenden Tage vorherzusagen. Jeder Tag ist ein Datenpunkt, und sein Label ist die Höchsttemperatur dieses Tages. Seine rohen Merkmale sind acht tägliche Messwerte: die tiefste, höchste und mittlere Lufttemperatur, der Niederschlag, die Sonnenscheindauer, die Luftfeuchtigkeit, der Luftdruck und die Windgeschwindigkeit. Über jeden der fünf vorangehenden Tage zusammengenommen ergibt das eine Liste von vierzig Zahlen. Ein Punkt mit vierzig Koordinaten lässt sich nicht in einem Streudiagramm darstellen. Eine Merkmalstransformation, die aus den vierzig rohen zwei neue Merkmale liefert, platziert jeden Tag in einem Streudiagramm (siehe Abb. 1), in dem sich die Tage, deren Label über dem Median aller Labels liegt, von den übrigen trennen, und ein lineares Modell der beiden neuen Merkmale kann das Label bereits vorhersagen. Repräsentationslernen ist die Aufgabe, eine solche Merk-

malstransformation aus einem Datensatz zu lernen, anstatt sie von Hand zu entwerfen.

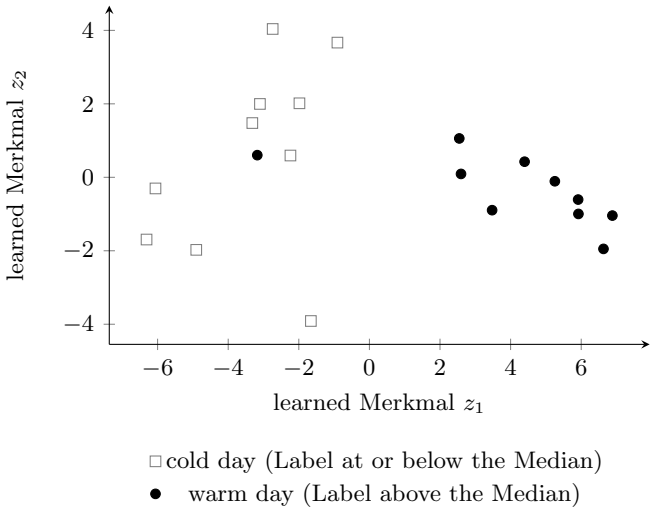

Abbildung 1: Zwanzig der 361 Tage des Jahres 2024 in Krems an der Donau, jeder 17. warme Tag (ausgefüllte Kreise) und jeder 17. kalte Tag (offene Quadrate) in kalendarischer Reihenfolge, jeweils platziert durch zwei gelernte Merkmale z_1 und z_2 anstelle seiner vierzig rohen, nämlich der acht täglichen Messwerte jedes der fünf vorangehenden Tage. Warme und kalte Tage, geteilt am Median des Labels, trennen sich entlang z_1, ebenso wie für alle 361 Tage. Daten erzeugt mit `pythondemos/featlearn.py`

Das Repräsentationslernen ist die Aufgabe, eine Merkmalstransformation

$$\mathbf{\Phi} : \mathcal{X} \to \mathcal{X}', \qquad \mathbf{x} \mapsto \mathbf{z} := \mathbf{\Phi}(\mathbf{x})$$

zu lernen. Die Transformation bildet den rohen Merkmalsvektor $\mathbf{x} \in \mathcal{X}$ eines Datenpunkts auf einen neuen Merkmalsvektor $\mathbf{z}$ aus einem neuen Merkmalsraum $\mathcal{X}'$ ab. Ein Verfahren des Repräsentationslernens wird durch drei Entwurfsentscheidungen festgelegt: die beiden Merkmalsräume $\mathcal{X}$ und $\mathcal{X}'$, einen Hypothesenraum $\mathcal{H}$ von infrage kommenden Merkmalstransformationen und ein quantitatives Maß dafür, wie nützlich ein bestimmtes $\mathbf{\Phi} \in \mathcal{H}$ ist. Das Maß wird aus einem Datensatz berechnet, der keine Labels tragen muss und nicht der Datensatz der vorliegenden Lernaufgabe sein muss. Eine nützliche Merkmalstransformation ist eine, die eine nachfolgende Lernaufgabe erleichtert (Goodfellow et al., 2016, Ch. 15): Die Datenpunkte werden durch ein lineares Modell trennbar, weniger Merkmale reichen aus, oder die einmal gelernten Merkmale dienen mehreren Lernaufgaben. Im Wetterbeispiel werden die gelernten Merkmale anschließend anhand des Validierungsfehlers der linearen Regression auf ihnen bewertet.

Ein Maß für die Nützlichkeit lässt sich mithilfe einer zweiten Abbildung konstruieren. Die Merkmalstransformation wird dann als Encoder aufgefasst, und

ein Decoder $\mathbf{\Phi}^* : \mathcal{X}' \to \mathcal{X}$ rekonstruiert den rohen Merkmalsvektor aus dem neuen (Goodfellow et al., 2016, Ch. 14). Was der Encoder bewahrt hat, wird dann daran gemessen, wie gut der Decoder den Datenpunkt daraus wiederherstellt, also am Rekonstruktionsfehler

$$\frac{1}{m}\sum_{r=1}^{m} L\Big(\mathbf{x}^{(r)},\, \mathbf{\Phi}^*\big(\mathbf{\Phi}\big(\mathbf{x}^{(r)}\big)\big)\Big).$$

Die Verlustfunktion L bestraft einen rekonstruierten Merkmalsvektor dafür, dass er vom Original abweicht. Eine gängige Wahl ist der quadratische Verlust (Goodfellow et al., 2016, Sect. 14.1). Der Datensatz, auf dem er berechnet wird, benötigt keine Labels: Was der Decoder reproduzieren muss, ist der Datenpunkt selbst.

Der Decoder kann gemeinsam mit dem Encoder gelernt werden. Das Maß ist dann das Minimum des Rekonstruktionsfehlers über einem zweiten Hypothesenraum von Decoder. Er kann stattdessen fest vorgegeben sein, und der Encoder allein muss dann liefern, was der Decoder nicht leisten kann. Die Einschränkung des Decoders entscheidet daher darüber, was die neuen Merkmale tragen müssen.

Die Hauptkomponentenanalyse (PCA) ist diese Konstruktion mit $\mathcal{X} := \mathbb{R}^d$, $\mathcal{X}' := \mathbb{R}^{d'}$ für ein $d' < d$ und dem Hypothesenraum der linearen Abbildungen,

$$\mathcal{H} := \big\{\mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^{d'} : \mathbf{x} \mapsto \mathbf{W}\mathbf{x} \text{ mit einem } \mathbf{W} \in \mathbb{R}^{d'\times d}\big\}.$$

Der Decoder ist ebenfalls linear und wird gemeinsam mit dem Encoder $\mathbf{W}$ gelernt. Die Nützlichkeit eines bestimmten $\mathbf{\Phi}(\mathbf{x}) = \mathbf{W}\mathbf{x}$ ist daher das Minimum des Rekonstruktionsfehlers über den Rekonstruktions-Matrizen $\mathbf{R}$. Auf dem Datensatz $\mathcal{D} = \big\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\big\}$ beträgt dieses Minimum

$$\min_{\mathbf{R}\in\mathbb{R}^{d\times d'}} \sum_{r=1}^{m} \Big\|\mathbf{x}^{(r)} - \mathbf{R}\mathbf{W}\mathbf{x}^{(r)}\Big\|_2^2.$$

Für Merkmalsvektoren mit einem Stichprobenmittel von null hat ein bestes $\mathbf{W}$ als Zeilen die d' Eigenvektoren der Stichproben-Kovarianzmatrix zu den größten Eigenwerte (Hastie et al., 2009, Sect. 14.5.1). Nur der Zeilenraum von $\mathbf{W}$ ist entscheidend: Ersetzt man $\mathbf{W}$ durch $\mathbf{T}\mathbf{W}$ mit einer invertierbaren Matrix $\mathbf{T} \in \mathbb{R}^{d'\times d'}$ und $\mathbf{R}$ durch $\mathbf{R}\mathbf{T}^{-1}$, so bleibt das Produkt $\mathbf{R}\mathbf{W}$ und damit auch der Rekonstruktionsfehler unverändert. Ein Autoencoder behält dasselbe Kriterium bei und ersetzt beide lineare Abbildungen durch künstliche neuronale Netze (KNN), sodass die neuen Merkmale aus einer nichtlinearen Merkmalstransformation stammen (Hinton and Salakhutdinov, 2006). Mit einem linearen Decoder und dem quadratischen Verlust spannt er denselben Unterraum auf, den auch die PCA findet (Goodfellow et al., 2016, Sect. 14.1).

Angewendet auf das Wetterbeispiel liefert die PCA die beiden gelernten Merkmale aus Abb. 1. Die vierzig rohen Merkmale der 361 Tage des Jahres 2024 in Krems an der Donau werden in zwei gelernte Merkmale überführt. Jedes rohe Merkmal wird zunächst auf einen Stichprobenmittel von null und eine

Stichprobenvarianz von eins skaliert, da Temperaturen, Niederschlag und Luftdruck unterschiedliche Einheiten tragen. Warme und kalte Tage trennen sich entlang des ersten gelernten Merkmals: 89 Prozent der warmen Tage haben ein positives z_1 und 92 Prozent der kalten Tage ein nichtpositives. Eine lineare Regression auf den beiden gelernten Merkmale wird auf den Tagen von Januar bis August trainiert. Sie sagt dann die Höchsttemperatur der Tage von September bis Dezember mit einem Validierungsfehler von 13,5 Quadratgrad Celsius voraus. Die lineare Regression auf allen vierzig rohen Merkmale erreicht 10,2, und die Vorhersage des Stichprobenmittels der Training-Labels erreicht 110,9. Die beiden gelernten Merkmale bewahren folglich den Großteil des Vorhersagegehalts der vierzig rohen.

Ein tiefes Netz lernt eine Kette von Merkmalstransformationen. Jede verdeckte Schicht $\ell = 1, \ldots, L$ ist eine Merkmalstransformation $\mathbf{\Phi}^{(\ell)}$, welche die Aktivierungen $\mathbf{z}^{(\ell-1)}$ der vorangehenden Schicht einliest, mit $\mathbf{z}^{(0)} := \mathbf{x}$. Die an die Ausgabe-Schicht gelieferten Merkmale sind die Verkettung $\mathbf{z}^{(L)} = \mathbf{\Phi}^{(L)} \circ \cdots \circ \mathbf{\Phi}^{(1)}(\mathbf{x})$ (siehe Abb. 2). Die Merkmalstransformationen werden gemeinsam mit der Ausgabe-Schicht durch empirische Risikominimierung (ERM) mit dem Verlust der endgültigen Vorhersage gelernt, sodass die Nützlichkeit der gelernten Merkmale daran gemessen wird, wie gut die Ausgabe-Schicht, typischerweise ein lineares Modell, das Label aus $\mathbf{z}^{(L)}$ vorhersagt (Goodfellow et al., 2016, Ch. 15). Die Kette ist nützlich, weil jede Schicht ihre Merkmale aus denen der vorangehenden aufbaut, sodass die Merkmalstransformationen $\mathbf{\Phi}^{(\ell)} \circ \cdots \circ \mathbf{\Phi}^{(1)}$ mit ℓ tiefer werden. In der Bild-Klassifikation liefern frühe Schichten Kanten, spätere Ecken und Objektteile (Goodfellow et al., 2016, Fig. 1.2), und die Merkmale an der letzten verdeckten Schicht trennen die Klassen durch eine lineare Hyperfläche (Bishop and Bishop, 2024, Sect. 6.3.3).

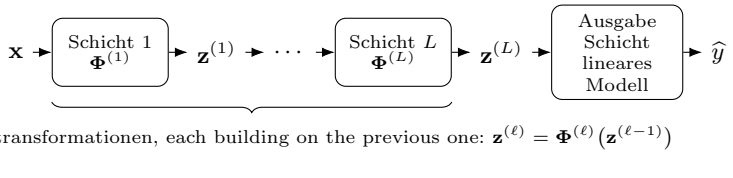

Abbildung 2: Ein tiefes Netz als Kette von Merkmalstransformationen: Die verdeckte Schicht ℓ überführt die Aktivierungen $\mathbf{z}^{(\ell-1)}$ der vorangehenden Schicht in $\mathbf{z}^{(\ell)}$, wobei $\mathbf{z}^{(0)}$ der rohe Merkmalsvektor ist, und die Ausgabe-Schicht wendet ein lineares Modell auf $\mathbf{z}^{(L)}$ an. Alle Schichten werden gemeinsam mit dem linearen Modell durch ERM mit dem Verlust der endgültigen Vorhersage gelernt

Das Vortraining eines Encoders auf einem großen Datensatz ungelabelter Datenpunkte, mit einem aus den Datenpunkte selbst konstruierten Verlust (siehe selbstüberwachtes Lernen), lernt Merkmale, die über mehrere Lernaufgaben hinweg wiederverwendet werden (Goodfellow et al., 2016, Sect. 15.1). Eine neue Lernaufgabe hängt eine aufgabenspezifische Hypothese mit wenigen Modellparameter an das gelernte $\mathbf{\Phi}$ an und passt sie durch Feinabstimmung auf ihrem eigenen Datensatz an. Der zum Lernen von $\mathbf{\Phi}$ verwendete Datensatz

kann dann wesentlich größer sein als jeder gelabelte Datensatz, da ungelabelte Datenpunkte kostengünstig zu sammeln sind (Bishop and Bishop, 2024, Sect. 6.3.3).
Alternative Konstruktionen einer Merkmalstransformation lernen $\mathbf{\Phi}$ nicht (Goodfellow et al., 2016, Ch. 6). Eine Kernel-Methode legt $\mathbf{\Phi}$ durch die Wahl eines Kernels fest, wobei $\mathcal{X}'$ der RKHS (RKHS) $\mathcal{H}_k$ ist und $\mathbf{z} = k\big(\mathbf{x}, \cdot\big)$, der Kernel mit festgehaltenem erstem Argument, niemals explizit ausgewertet wird, und lernt lediglich eine lineare Hypothese auf $\mathcal{X}'$. Ein von Hand entworfenes $\mathbf{\Phi}$ kodiert Fachwissen anstelle eines Datensatzes.

Siehe auch: Merkmalstransformation, Encoder, Decoder, Merkmal, Merkmalsraum, Hypothesenraum, PCA, Autoencoder, Dimensionsreduktion, tiefes Netz, Schicht, Vortraining.

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. Hinton and Salakhutdinov (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), pp. 504–507.

4. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.