Decoder

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 Decoder ist eine Abbildung, die den Code eines Datenpunkt, den kurzen Vektor, den ein encoder daraus berechnet, zurück in den Raum überführt, in dem der Datenpunkt selbst liegt. Der encoder und der Decoder werden gemeinsam durch empirische Risikominimierung (ERM) gelernt, mit einer Verlustfunktion, die den Rekonstruktionsfehler zwischen einem Datenpunkt und dem, was der Decoder für dessen Code zurückgibt, misst. In der Hauptkomponentenanalyse (PCA) sind beide Abbildungen linear und der Decoder ist die transpose des encoder, sodass die Rekonstruktion die orthogonale Projektion des Merkmalsvektor auf den von den beibehaltenen Eigenvektoren aufgespannten Unterraum ist, und der durchschnittliche quadrierte Rekonstruktionsfehler ist die Summe der verworfenen Eigenwerte. Ein Autoencoder ersetzt beide lineare Abbildungen durch künstliche neuronale Netze (KNN), und sein Decoder wird oft verworfen, sobald der Code für eine spätere Lernaufgabe benötigt wird. Ein Decoder kann anstelle eines Punktes eine Wahrscheinlichkeitsverteilung zurückgeben: Der Decoder eines variationalen Autoencoders (VAE) ist eine conditional probability distribution über Merkmalsvektoren gegeben eine latent variable. Der Ausgabeschritt eines großen Sprachmodells (LLM) ist ebenfalls ein Decoder, der einen Vektor in eine Wahrscheinlichkeitsverteilung über das vocabulary umwandelt, was die Generierung von Text Token für Token ermöglicht.

Definition

Die Wetterstation in Krems an der Donau (Österreich) zeichnet die niedrigste und höchste Lufttemperatur jedes Tages auf, sodass ein Tag ein Punkt der Ebene mit diesen beiden Merkmale als Koordinaten ist. Das Speichern einer Zahl pro Tag anstelle von zwei ist möglich, aber diese Zahl ist nutzlos, es sei denn, eine Abbildung wandelt sie zurück in ein Temperaturpaar. Ein Decoder ist eine solche Abbildung.

Ein Decoder ist eine Abbildung $\mathbf{\Phi}^* : \mathcal{X}' \to \mathcal{X}$ von einem Raum $\mathcal{X}'$ von Codes zurück in den Merkmalsraum $\mathcal{X}$, der den Merkmalsvektor $\mathbf{x}$ eines Datenpunkt enthält. Sein Gegenstück ist der encoder $\mathbf{\Phi} : \mathcal{X} \to \mathcal{X}'$, der den Code $\mathbf{z} = \mathbf{\Phi}(\mathbf{x})$ berechnet, einen Vektor von d' Zahlen im Vergleich zu den d Zahlen des Merkmalsvektor, mit $d' < d$. Die beiden Abbildungen bilden einen Autoencoder, und sie werden gemeinsam durch empirische Risikominimierung (ERM)

mit einer Verlustfunktion gelernt, die den Rekonstruktionsfehler $\mathbf{x} - \mathbf{\Phi}^*(\mathbf{\Phi}(\mathbf{x}))$ misst (Goodfellow et al., 2016, Sect. 14.1). Kein Label fließt in diese Lernaufgabe ein: Was der Decoder reproduzieren muss, ist der Datenpunkt selbst. Was der Decoder zurückgibt, muss kein Punkt von $\mathcal{X}$ sein: Es kann eine Wahrscheinlichkeitsverteilung sein, über $\mathcal{X}$ oder über ein vocabulary von Tokens. Abb. 1 zeigt die drei Formen, die der Decoder in der Hauptkomponentenanalyse (PCA), in einem variational autoencoder (VAE) und in einem großes Sprachmodell (LLM) annimmt.

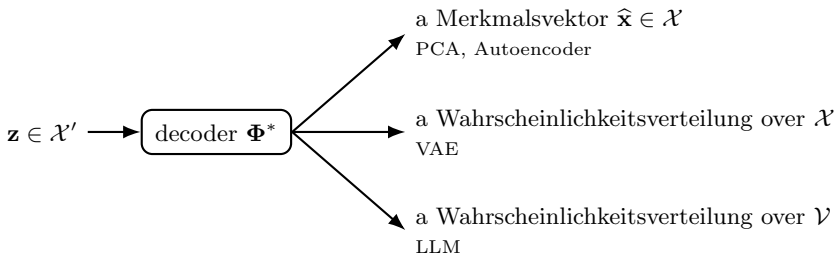


Abbildung 1: Die drei Formen, die ein Decoder annimmt. Die Pfeile verlassen den Decoder für einen Merkmalsvektor $\widehat{\mathbf{x}}$, was die PCA und ein Autoencoder zurückgeben, für eine Wahrscheinlichkeitsverteilung über den Merkmalsraum, was der Decoder eines VAE zurückgibt, und für eine Wahrscheinlichkeitsverteilung über das vocabulary $\mathcal{V}$, was der Ausgabeschritt eines LLM ist.

Die PCA ist der lineare Fall (Hotelling, 1933). Sein encoder ist eine Matrix $\mathbf{W} \in \mathbb{R}^{d' \times d}$ mit orthonormalen Zeilen, sodass der Code eines Merkmalsvektor $\mathbf{z} = \mathbf{W}\mathbf{x}$ ist, und sein Decoder ist die transpose $\mathbf{W}^\top$ dieser Matrix (Hastie et al., 2009, Sect. 14.5.1, Gl. (14.52)). Die Rekonstruktion $\mathbf{W}^\top\mathbf{W}\mathbf{x}$ unterscheidet sich von $\mathbf{x}$ durch einen Vektor, der orthogonal zu den Zeilen von $\mathbf{W}$ ist, und kein anderer Punkt des von ihnen aufgespannten Unterraum liegt näher an $\mathbf{x}$: Die Rekonstruktion ist die orthogonale Projektion von $\mathbf{x}$ auf diesen Unterraum (Hastie et al., 2009, Sect. 14.5.1). Die Zeilen von $\mathbf{W}$ sind die Eigenvektoren der sample covariance matrix $\widehat{\mathbf{Q}}$, die zu ihren d' größten Eigenwerte gehören (Bishop, 2006, Sect. 12.1.2).

Die Eigenwerte von $\widehat{\mathbf{Q}}$, in absteigender Reihenfolge $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$, geben an, was jede Codelänge kostet. Für einen Datensatz $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ von Merkmalsvektoren mit Stichprobenmittel Null ist der durchschnittliche quadrierte Rekonstruktionsfehler die Summe der Eigenwerte, die der Code verwirft (Bishop, 2006, Sect. 12.1.2, Gl. (12.18)),

$$\frac{1}{m}\sum_{r=1}^{m}\left\|\mathbf{x}^{(r)} - \mathbf{W}^\top\mathbf{W}\mathbf{x}^{(r)}\right\|_2^2 = \sum_{j=d'+1}^{d}\lambda_j. \tag{1}$$

Die Identität (1) gilt für jede Codelänge $d' = 1, \ldots, d$. Das Archiv in Krems enthält acht Messungen pro Tag: die niedrigste, höchste und mittlere Lufttemperatur, den Niederschlag, die Sonnenscheindauer, die Luftfeuchtigkeit, den

Luftdruck und die Windgeschwindigkeit. Jede der acht Messungen wird auf eine Einheits-Stichprobenvarianz skaliert, da sie in verschiedenen Einheiten gemessen werden. Ein Code aus zwei Zahlen lässt dann 2.91 von insgesamt 8 übrig, sodass der Decoder 64 Prozent der gesamten Stichprobenvarianz wiederherstellt. Abb. 2 zeigt den Decoder eines Codes aus einer Zahl für die beiden Temperaturen und beide Seiten der Identität (1) für jede Codelänge.

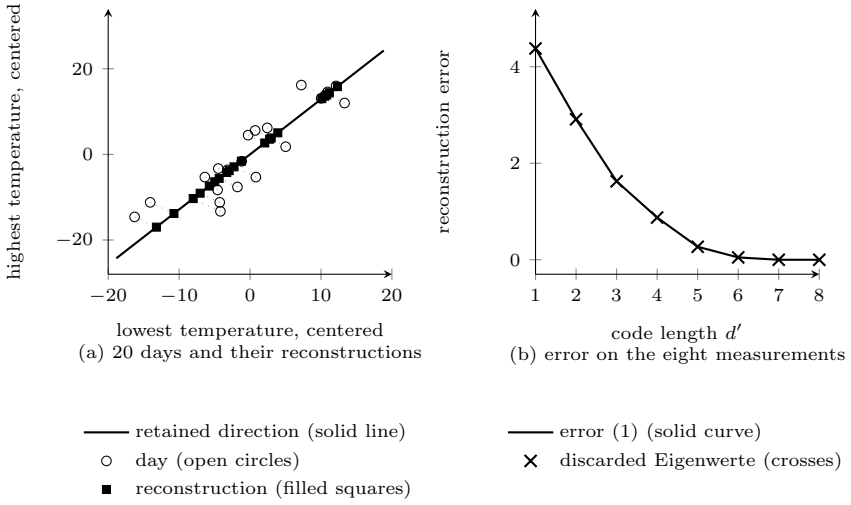


Abbildung 2: (a) Zwanzig der 366 Tage des Jahres 2024 in Krems, jeder 19, jeweils ein Datenpunkt mit der niedrigsten und der höchsten Tagestemperatur als seine zwei Merkmale, zentriert (offene Kreise), die Rekonstruktion jedes einzelnen aus einem Code einer Zahl (gefüllte Quadrate) und die beibehaltene Richtung (durchgezogene Linie); die gepunkteten Segmente sind der Fehler, den der Decoder hinterlässt, der im Durchschnitt 7.94 Quadratgrad über alle 366 Tage beträgt und dem verworfenen Eigenwert entspricht. (b) Der mittlere quadratische Rekonstruktionsfehler (1) bei den acht Messungen eines Tages, jeweils auf eine Einheits-Stichprobenvarianz skaliert, gegen die Codelänge d' (durchgezogene Kurve), zusammen mit der Summe $\sum_{j=d'+1}^{d} \lambda_j$ der verworfenen Eigenwerte (Kreuze). Daten generiert von `pythondemos/decoder.py`

Ein Autoencoder behält den Rekonstruktionsfehler bei und ersetzt beide lineare Abbildungen durch künstliche neuronale Netze (KNN), sodass sein Decoder ein Netzwerk ist, das den Code zurück in einen Merkmalsvektor umwandelt (Goodfellow et al., 2016, Sect. 14.1). Ein denoising autoencoder ändert, was vom Decoder verlangt wird: Der encoder liest einen korrumpierten Merkmalsvektor und der Decoder muss den sauberen zurückgeben (Goodfellow et al., 2016, Sect. 14.5). Nach dem Training wird der Decoder oft verworfen und der Code beibehalten, sodass eine spätere Lernaufgabe eine Hypothese auf dem Code lernt, anstatt auf den rohen Merkmale.

Ein Decoder muss keinen Punkt zurückgeben. Der Decoder eines VAE ist eine conditional probability distribution $p_{\mathbf{w}}(\mathbf{x} \mid \mathbf{z})$ über Merkmalsvektoren gegeben eine latent variable $\mathbf{z}$, mit Modellparameter $\mathbf{w}$: gegeben einen Code gibt er eine Wahrscheinlichkeitsverteilung über die Merkmalsvektoren zurück, die ihn erzeugt haben könnten (Kingma and Welling, 2014, Sect. 2.1). Das Ziehen von $\mathbf{z}$ aus einer festen Wahrscheinlichkeitsverteilung und dann eines Merkmalsvektor aus dem Decoder macht das Paar zu einem generative model. Die beiden Lesarten treffen sich dort, wo diese conditional probability distribution Gaußsch ist mit Mittelwert $\mathbf{Rz}$, für eine Matrix $\mathbf{R} \in \mathbb{R}^{d \times d'}$, und einer Streuung, die nicht von $\mathbf{z}$ abhängt. Seine Log-likelihood function ist dann eine abnehmende affine Funktion des durchschnittlichen quadrierten Rekonstruktionsfehlers, sodass die Matrix $\mathbf{R}$, die das eine maximiert, das andere minimiert.

Der Schritt, der einen Vektor in Text umwandelt, ist ebenfalls ein Decoder. Ein LLM berechnet einen Vektor $\mathbf{z} \in \mathbb{R}^d$ aus den bisher gelesenen Tokens. Der Decoder bildet ihn auf eine Wahrscheinlichkeitsverteilung über das vocabulary $\mathcal{V}$ ab, mittels einer lineare Abbildung gefolgt von der softmax function (Vaswani et al., 2017, Sect. 3.4). Die Zeilen dieser lineare Abbildung sind $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(|\mathcal{V}|)} \in \mathbb{R}^d$, und der Decoder gibt zurück

$$\big(\mathbf{\Phi}^*(\mathbf{z})\big)_v = \frac{\exp\big(\langle \mathbf{a}^{(v)}, \mathbf{z}\rangle\big)}{\sum_{v'=1}^{|\mathcal{V}|} \exp\big(\langle \mathbf{a}^{(v')}, \mathbf{z}\rangle\big)} \quad \text{for } v = 1, \ldots, |\mathcal{V}|. \tag{2}$$

Jeder Wert, den der Quotient (2) zuweist, ist nicht-negativ, und die Werte summieren sich über $v = 1, \ldots, |\mathcal{V}|$ zu eins, sodass der Decoder einen Punkt des probability simplex $\Delta^{|\mathcal{V}|-1}$ zurückgibt. Das Ziehen eines Token aus dieser Wahrscheinlichkeitsverteilung, das Anhängen an die bisher gelesenen Tokens und das Wiederholen ist die Art und Weise, wie Text Token für Token generiert wird (Vaswani et al., 2017, Sect. 3): Ohne den Decoder benennt der Vektor $\mathbf{z}$ überhaupt kein Token.

Ein vocabulary von acht Tokens ist bereits ausreichend, um den Decoder (2) numerisch zu überprüfen. Das Aufteilen der höchsten Temperatur jedes Tages in Krems in acht Bins von jeweils etwa 45 Tagen und die Wahl der Zeilen $\mathbf{a}^{(v)}$ mittels maximum likelihood reproduziert die beobachteten Häufigkeiten des nächsten Token auf 0.0014 genau. Eine Folge von 20,001 Tokens, die einzeln generiert werden, stimmt dann mit den beobachteten Anteilen der acht Tokens auf 0.009 genau überein.

Die Rekonstruktionen eines linearen Decoders liegen in einem Unterraum der Dimension d', unabhängig davon, wie das Datensatz aussieht. Die Identität (1) besagt, was das im Durchschnitt kostet, und nichts darüber, wo die Rekonstruktionen landen. Zurückgelesen in den aufgezeichneten Einheiten, ergibt die Rekonstruktion eines Tages aus einem Code von zwei Zahlen für 128 der 366 Tage einen negativen Niederschlag, bis zu -7.3 Millimeter, und für 17 Tage eine negative Sonnenscheindauer. Kein Tag der Aufzeichnung nimmt in keiner der Messungen einen negativen Wert an. Ein linearer Decoder, der mit der squared error loss gelernt wurde, stellt den Unterraum wieder her, den die PCA findet, sodass nur ein nichtlinearer Decoder eine Rekonstruktion außerhalb eines Unter-

raum platziert (Goodfellow et al., 2016, Sect. 14.1), weshalb der Decoder eines Autoencoder ein KNN und keine Matrix ist.

Was der lineare Decoder bietet, ist eine geschlossene Form: Seine Eigenvektoren stammen aus der sample covariance matrix ohne Trainingslauf, und seine Eigenwerte geben den Fehler jeder Codelänge im Voraus an. Das Krümmen der Rekonstruktionsmenge senkt den Fehler nicht immer: Mit demselben Code aus einer Zahl lässt der Durchschnitt der Tage in jedem der acht Bins 5.10 an Tagen, von denen nicht gelernt wurde, übrig, gegenüber den 5.07 des linearen Decoders.

Ein Decoder, der eine Wahrscheinlichkeitsverteilung anstelle eines Punktes zurückgibt, erkauft sich eine Wahrscheinlichkeit für jede Rekonstruktion und die Fähigkeit zu generieren, da ein Merkmalsvektor oder ein Token aus dem gezogen werden kann, was der Decoder zurückgibt. Der Preis ist ein Wahrscheinlichkeitsmodell, das gelernt werden muss und falsch sein kann, sowie ein im Voraus festgelegter Support. Der oben genannte Gaußsche Decoder verpflichtet sich zu einer Streuung, die nicht vom Code abhängt. Das vocabulary von acht Tokens löst die maximale Tagestemperatur eines Tages nur bis zu seinem Bin auf: 1.4 Grad im quadratischen Mittel und 10.3 Grad für das kälteste Token, das jeden Tag bei oder unter 6.7 Grad enthält. Keine Folge dieser acht Tokens unterscheidet zwei Tage eines Bins voneinander.

Siehe auch: encoder, Autoencoder, denoising autoencoder, VAE, latent variable, conditional probability distribution, Wahrscheinlichkeitsmodell, PCA, Repräsentationslernen, Dimensionsreduktion, Merkmalstransformation, ERM, squared error loss, orthogonale Projektion, Unterraum, Eigenvektor, generative model, softmax function, probability simplex, vocabulary, LLM.

Querverweise

Literatur

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

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

3. Hotelling (1933). Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology, 24(6), pp. 417–441.

4. Kingma and Welling (2014). Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations (ICLR).

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

6. Vaswani et al. (2017). Attention is all you need. In: Adv. Neural Inf. Process. Syst., pp. 5998–6008.