encoder

Author and editors

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

Abstract

An encoder is a map that sends the raw feature vector of a data point to a shorter vector, a code of d' numbers with d' below the number d of raw features. Its counterpart is the decoder, which maps a code back to a vector of the original length, and the two are judged together: the decoder applied to the code of a data point should return approximately the feature vector the encoder started from. The average loss that remains, the reconstruction error, is what training minimizes. Principal component analysis (PCA) uses a linear map as the encoder, a matrix whose rows are the eigenvectors of the sample covariance matrix belonging to its largest eigenvalues. An autoencoder learns the encoder jointly with the decoder, so it can be a deep net rather than a linear map. A large language model (LLM) begins with the token-to-vector encoder, which carries every token of a vocabulary to one row of an embedding matrix. The code length decides what the decoder can rebuild, and the code kept in place of the raw feature vectors is what makes an encoder a feature transformation for a later learning task. What a shorter code drops is beyond every decoder, so a code learned to reconstruct well need not serve that learning task well.

Definition

The weather station at Krems an der Donau (Austria) records eight numbers for each day: the lowest, the highest, and the mean air temperature, the precipitation, the sunshine duration, the humidity, the air pressure, and the wind speed. Two numbers computed from the eight are enough to tell a day in January from a day in July, and a second map rebuilds the eight from the two with a small error. The map that delivers the two numbers is an encoder. It sends the raw feature vector $\mathbf{x} \in \mathbb{R}^d$ of a data point to a shorter vector $\mathbf{z} \in \mathbb{R}^{d'}$, its *code*,

$$\boldsymbol{\Phi} : \mathbb{R}^d \to \mathbb{R}^{d'} : \mathbf{x} \mapsto \mathbf{z} := \boldsymbol{\Phi}(\mathbf{x}),$$

with a code length d' below the number d of raw features. Its counterpart is the decoder $\boldsymbol{\Phi}^* : \mathbb{R}^{d'} \to \mathbb{R}^d$, which maps a code back to a vector with d entries (Goodfellow et al., 2016, Ch. 14).

Encoders differ in the hypothesis space they are picked from. Principal component analysis (PCA) picks a linear map: the encoder is a matrix $\mathbf{W} \in \mathbb{R}^{d' \times d}$ with $\mathbf{z} = \mathbf{W}\mathbf{x}$, whose rows are the eigenvectors of the sample covariance

matrix of the feature vectors belonging to its d' largest eigenvalues, and the decoder is the transpose $\mathbf{W}^\top$ (Bishop, 2006, Sect. 12.1.2; Hotelling, 1933). An autoencoder picks the encoder from a hypothesis space of deep nets and learns it jointly with the decoder, so that neither map need be linear (Goodfellow et al., 2016, Sect. 14.1). A large language model (LLM) begins with the token-to-vector step, an encoder on text. Every token of a vocabulary $\mathcal{V}$ carries one row of an embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d'}$. Picking the row of the token with index v is the linear map $\mathbf{z} = \mathbf{E}^\top \mathbf{e}^{(v)}$ applied to the one-hot vector $\mathbf{e}^{(v)} \in \mathbb{R}^{|\mathcal{V}|}$ of that token (Vaswani et al., 2017, Sect. 3.4).

What makes a code useful is that the decoder can rebuild the data point from it. The reconstruction error of an encoder on a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ of sample size m is the average loss between a feature vector and what the decoder returns for its code,

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

A common loss function is the squared error loss $\|\mathbf{x} - \boldsymbol{\Phi}^*(\boldsymbol{\Phi}(\mathbf{x}))\|_2^2$, and minimizing the reconstruction error (1) over a hypothesis space of encoders and a second one of decoders is empirical risk minimization (ERM). No labels enter it: what the decoder has to reproduce is the data point itself, so the training of an encoder is a form of self-supervised learning.

Each of the eight measurements of a day at Krems is scaled to zero sample mean and unit sample variance, so that they carry comparable weight. Over the 366 days of 2024, the PCA encoder with $d' = 2$ then leaves a reconstruction error of 2.91 out of the total variance 8.00, and the eight reconstructed measurements of 1 January 2024 differ from the scaled originals by at most 0.80. Fig. 1(a) places 21 of the days at their two code numbers, split at the median of the maximum temperature, and Fig. 1(b) compares the reconstruction error of this encoder with what a random encoder with orthonormal rows leaves as the code grows.

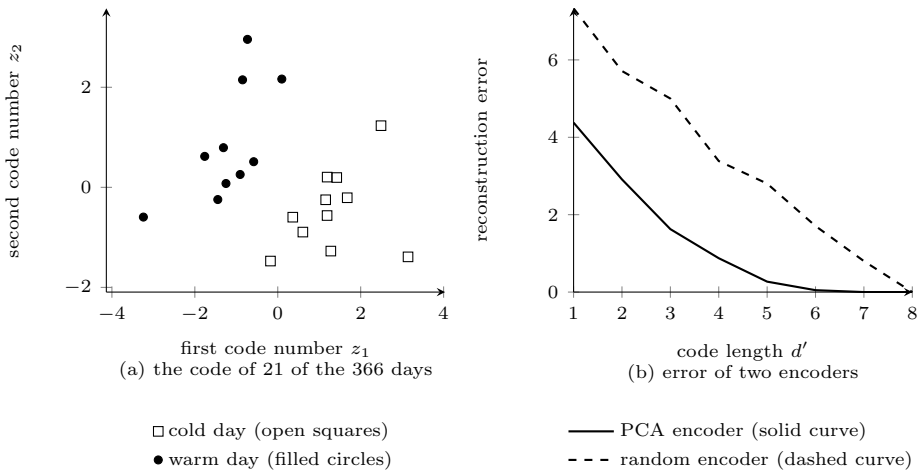


Figure 1: The PCA encoder of the 366 days of 2024 at Krems, each day a data point with eight scaled measurements. (a) The two code numbers of every 18th day, 21 in all, drawn as a cold day (open squares) or a warm day (filled circles) by whether the maximum temperature lies below or above its median. (b) The reconstruction error against the code length d' for the PCA encoder (solid curve) and for a random encoder with orthonormal rows (dashed curve). Data generated by `pythondemos/encoder.py`

The code length decides what survives. On the days at Krems, the reconstruction error of the PCA encoder falls from 4.38 at one number to 2.91 at two and 0.88 at four, and at every code length it equals the sum of the eigenvalues the code drops (Bishop, 2006, Sect. 12.1.2, Eq. (12.18)). A random encoder with orthonormal rows leaves 5.71 at two numbers, so which encoder is used matters beside how long its code is. Losing something is not inevitable: the token-to-vector encoder shortens a one-hot vector of length $|\mathcal{V}|$ to d' numbers and still loses nothing, since distinct tokens carry distinct rows of $\mathbf{E}$ and the nearest row therefore recovers the token. What training keeps is usually the encoder alone: the decoder is dropped and the code serves as the feature transformation of a later learning task, which makes an encoder a feature learning and a dimensionality reduction method (Bengio et al., 2013).

The code of two numbers leaves a reconstruction error of 2.91, and that loss is not spread evenly over the eight measurements. Of the unit sample variance of each measurement, the code keeps 96 percent of the mean temperature and 3 percent of the wind speed. What the dropped eigenvectors carried is beyond every decoder, not only the transpose $\mathbf{W}^\top$. Of the 80 pairs of days whose codes lie closer than 0.1 together, 14 February and 22 November 2024 differ most in wind speed, 1.0 against 4.5 meters per second, and the decoder returns 1.76 meters per second for both. Whether the loss matters is decided by the later learning task and not by the reconstruction error alone. A day counts as windy when its mean wind speed lies above the median of the 366 values, which holds

for 130 of them. A linear classifier learned by least squares has an accuracy of 98 percent from all eight scaled measurements and 66 percent from the two code numbers, against the 64 percent reached by calling every day calm. One new feature chosen from the labels instead of from the sample covariance matrix, the linear discriminant analysis (LDA) direction, reaches 98 percent as well. A code learned to reconstruct well is therefore not a code learned to predict well, which is the difference LDA draws against PCA. An encoder is also fixed once learned, and the dataset it was learned from decides how well it codes a later day. The PCA encoder of the 183 days of January to March and of October to December leaves 1.93 on those days and 10.87 on the other 183, against 2.48 for the encoder of the whole year. What the code buys is its shortness: fewer numbers per data point to store and to hand to a later learning task, and at $d' = 2$ a plot of the dataset on two axes.

Cross References

decoder, autoencoder, VAE, feature learning, feature transformation, dimensionality reduction, PCA, LDA, embedding, token, latent variable, ERM, squared error loss

References

1. Bengio et al. (2013). Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell., 35(8), pp. 1798–1828.

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

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

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

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