LINEAR ALGEBRA FOR REPRESENTATION LEARNING

The Mathematics of Embeddings

How discrete symbols become vectors: the embedding lookup as a matrix product, the similarity measures used to compare vectors, how skip-gram training relates to a factorised co-occurrence matrix, and what high-dimensional geometry implies for reading similarity scores.

Intermediate · 10 min · embeddings · word2vec · cosine similarity · PMI

Sheet
by Simon-Pierre Boucher

sheet.spboucher.ai/s/the-mathematics-of-embeddings
November 17, 2026

An **embedding** maps each item of a discrete set (a word, a token, a product, a user) to a vector in $\mathbb{R}^d$. Training places the vectors so that *geometry carries meaning*: items used in similar contexts end up close together. This sheet covers the four pieces of mathematics behind that idea. It shows how a lookup is a matrix product, how closeness is measured, what the training objective factorises, and why high dimension changes how a similarity score should be read.

EMBEDDING LOOKUP

$$\mathbf{e}_i = E^\top \mathbf{x}_i = \left(E_{i,1}, E_{i,2}, \ldots, E_{i,d}\right)^\top$$

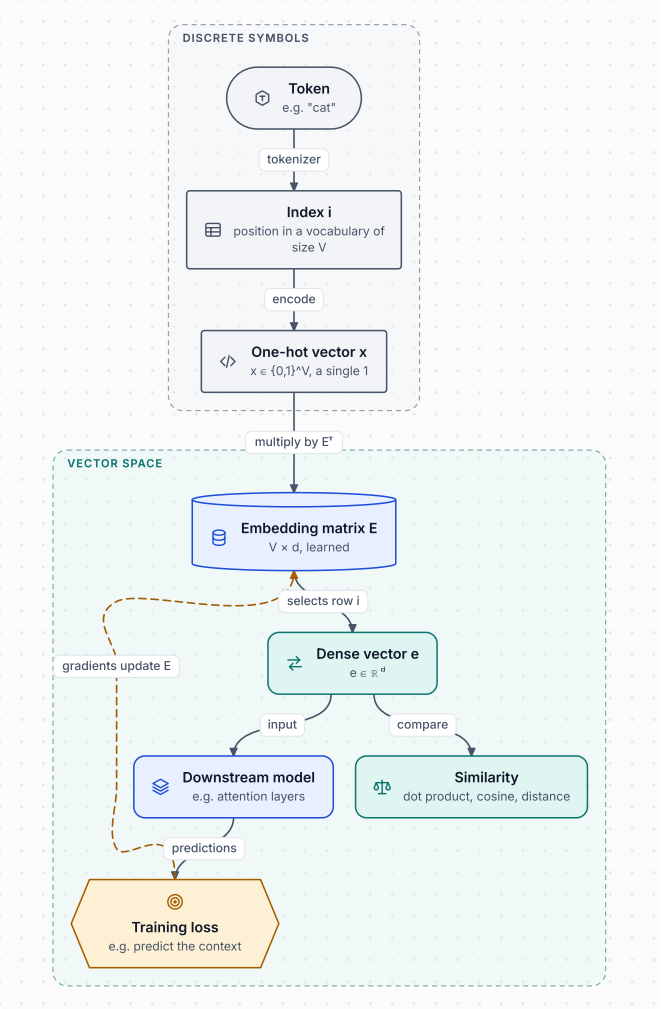
f_x Result derived in this sheet

Because $\mathbf{x}_i$ has a single 1 in position i, the product picks out row i of E. Libraries implement it as an indexed read rather than a multiplication. The product form still explains the gradient: $\partial \mathcal{L}/\partial E = \mathbf{x}_i \left(\partial \mathcal{L}/\partial \mathbf{e}_i\right)^\top$ is zero outside row i.

E	Embedding matrix, one row per vocabulary item, learned during training	$\mathbb{R}^{V \times d}$
$\mathbf{x}_i$	One-hot vector of item i: 1 in position i, 0 elsewhere	$\{0,1\}^V$
$\mathbf{e}_i$	Embedding of item i	$\mathbb{R}^d$
V	Vocabulary size (number of distinct items)	
d	Embedding dimension, typically far smaller than V	
$\mathcal{L}$	Training loss	

01 From a symbol to a vector

The life of an embedding



Discrete symbols · Learned parameters
Vectors and operations on them · Training signal

1. **Symbol to index** — The tokenizer turns text into integer indices into a fixed vocabulary.
2. **Lookup as a product** — Multiplying the one-hot vector by $E^\top$ returns exactly one row of E.
3. **Use the vector** — The vector is compared with others or passed to a model.
4. **Learn the matrix** — The loss is differentiated with respect to E. Only the rows that were looked up receive a gradient.

A token becomes an index, the index selects one row of a learned matrix, and the resulting vector is compared or fed to a model. The training loss sends gradients back into the matrix.

02 Measuring closeness

ƒ COSINE SIMILARITY

$$\cos(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \, \|\mathbf{v}\|_2} = \frac{\sum_{k=1}^{d} u_k v_k}{\sqrt{\sum_{k=1}^{d} u_k^2}\,\sqrt{\sum_{k=1}^{d} v_k^2}}$$

The cosine keeps only the angle between two vectors and discards their lengths. It equals 1 for vectors pointing the same way, 0 for orthogonal vectors and −1 for opposite ones. It is undefined if either vector is zero.

$\mathbf{u}, \mathbf{v}$	Two embedding vectors	$\mathbb{R}^d$
u_k, v_k	Their k-th coordinates	
$\|\cdot\|_2$	Euclidean norm (length) of a vector	

Three ways to compare two embeddings

	Dot product	Cosine similarity	Euclidean distance
FORMULA	$\mathbf{u}^\top \mathbf{v}$	$\mathbf{u}^\top \mathbf{v} \,/\, (\|\mathbf{u}\|\,\|\mathbf{v}\|)$	$\|\mathbf{u}-\mathbf{v}\|_2$
RANGE	$(-\infty, \infty)$	$[-1, 1]$	$[0, \infty)$
A LARGER VALUE MEANS	More similar	More similar	Less similar
DEPENDS ON VECTOR LENGTHS	Yes: long vectors score high	No: direction only	Yes
TYPICAL USE	Attention scores, output logits	Semantic search when norms vary	Clustering and nearest neighbours on normalised vectors

VERDICT

Once every vector is scaled to unit length, all three give the **same ranking of neighbours** (see the identity below). They differ only on unnormalised vectors.

ƒ DISTANCE AND COSINE ON UNIT VECTORS

$$\|\mathbf{u}-\mathbf{v}\|_2^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\,\mathbf{u}^\top \mathbf{v} \;=\; 2 - 2\cos(\mathbf{u},\mathbf{v}) \quad \text{if } \|\mathbf{u}\| = \|\mathbf{v}\| = 1$$

ƒx Result derived in this sheet

Expand the squared norm. With unit vectors the squared distance is a decreasing function of the cosine. Normalising first therefore makes nearest-neighbour search by distance, by cosine and by dot product interchangeable. This is why vector databases often store normalised embeddings.

$\mathbf{u}, \mathbf{v}$	Two embeddings scaled to unit length	$\mathbb{R}^d$

Cosine similarities of four toy vectors

SYNTHETIC DATA

	cat	dog	car	truck
cat	1.00	0.96	0.10	0.18
dog	0.96	1.00	0.18	0.31
car	0.10	0.18	1.00	0.96
truck	0.18	0.31	0.96	1.00

0.10 — 1.00

Synthetic: computed exactly from hand-picked 3-d vectors cat = (3, 1, 0), dog = (3, 2, 0), car = (0, 1, 3), truck = (0, 2, 3). The vectors are not from any trained model.

03 Where the vectors come from

ƒ SKIP-GRAM WITH NEGATIVE SAMPLING (WORD2VEC)

$$\ell(w,c) = \log \sigma\left(\mathbf{u}_c^\top \mathbf{v}_w\right) + \sum_{j=1}^{k} \mathbb{E}_{c_j \sim P_n}\left[\log \sigma\left(-\mathbf{u}_{c_j}^\top \mathbf{v}_w\right)\right]$$

This quantity is maximised over all (word, context) pairs observed within a sliding window. The first term pushes the dot product of a real pair up. The second pushes down the dot products with k random *negative* contexts. In the original paper P_n is the unigram distribution raised to the power 3/4 [2]. Two matrices are learned: word vectors $\mathbf{v}$ and context vectors $\mathbf{u}$.

w, c	A centre word and a word observed in its context window	
$\mathbf{v}_w$	Word (input) embedding of w	$\mathbb{R}^d$
$\mathbf{u}_c$	Context (output) embedding of c	$\mathbb{R}^d$
$\sigma(z)$	Logistic function $1/(1+e^{-z})$	
k	Number of negative samples per observed pair	
P_n	Noise distribution from which negative contexts are drawn	

Sources: 2

THEOREM

Embeddings as a factorised co-occurrence matrix

Levy and Goldberg showed that, if the dimension is large enough for every dot product to be set independently, the skip-gram negative-sampling objective is optimised when

$$\mathbf{v}_w^\top \mathbf{u}_c = \mathrm{PMI}(w,c) - \log k, \text{ with } \mathrm{PMI}(w,c) = \log \frac{P(w,c)}{P(w)\,P(c)}.$$

Word2vec therefore implicitly factorises a shifted pointwise-mutual-information matrix into two low-rank factors [3]. GloVe makes the factorisation explicit. It fits $\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j \approx \log X_{ij}$ by weighted least squares on co-occurrence counts X_{ij} [4]. Either way, a dot product between embeddings estimates how much more often two items co-occur than chance would predict.

Sources: 3 4

04 Geometry in high dimension

ƒ RANDOM DIRECTIONS ARE NEARLY ORTHOGONAL

$$\mathbb{E}\left[\cos(\mathbf{u},\mathbf{v})\right] = 0, \qquad \mathrm{Var}\left[\cos(\mathbf{u},\mathbf{v})\right] = \frac{1}{d}$$

ƒx Result derived in this sheet

For $\mathbf{u}, \mathbf{v}$ independent and uniform on the unit sphere, rotation invariance lets us fix $\mathbf{v} = (1, 0, \ldots, 0)$, so $\cos = u_1$. Since $\sum_k u_k^2 = 1$ and all coordinates play the same role, $\mathbb{E}[u_1^2] = 1/d$. In $d = 300$ the typical chance cosine is about $1/\sqrt{300} \approx 0.06$, so most pairs of random directions are close to orthogonal. Trained embeddings are **not** uniform on the sphere: contextual embeddings in particular have been found to be strongly anisotropic, crowding into a narrow cone [6]. The chance baseline for a given model must therefore be measured, not assumed.

$\mathbf{u}, \mathbf{v}$	Independent random unit vectors, uniformly distributed on the sphere	$\mathbb{R}^d$
d	Dimension of the space	
u_1	First coordinate of $\mathbf{u}$	

Sources: 6

WARNING

Handle analogy arithmetic with care

The well-known result $\mathbf{v}_{\text{king}} - \mathbf{v}_{\text{man}} + \mathbf{v}_{\text{woman}} \approx \mathbf{v}_{\text{queen}}$ [1] means that *queen* is the nearest vector to that sum **once the three query words are excluded** from the candidates. Without that exclusion the nearest vector is often one of the inputs. Simple baselines that ignore the offset also do surprisingly well on standard analogy sets [5]. Treat vector offsets as a tendency of the geometry, not as exact algebra.

Sources: 1 5

05 Check and explore

Dot products turned into attention weights

Synthetic data — seed 7

Attention weights $\mathrm{softmax}\left((q_i^\top k_j/\sqrt{d} - \lambda|i-j|)/\tau\right)$ of a single head. Embeddings (one vector per word plus a positional term) and the Q and K projections are drawn from the seed: the map shows the mechanism, not what a trained model has learned. λ is a simplified relative-position bias; $\tau = 1$ and $\lambda = 0$ give the standard formula.

query ↓ key →	the	cat	sleeps	on	the	mat
the	0.17	0.17	0.13	0.26	0.11	0.17
cat	0.21	0.09	0.06	0.31	0.19	0.14
sleeps	0.09	0.17	0.16	0.32	0.18	0.08
on	0.24	0.04	0.02	0.52	0.11	0.09
the	0.19	0.08	0.26	0.19	0.11	0.16
mat	0.15	0.17	0.25	0.08	0.15	0.20

Mean row entropy (normalised)	Mean weight on itself	Causal mask
0.92	0.208	no

Parameters: Temperature τ = 1; Dimension d = 16; Locality bias λ = 0

In a transformer, embeddings are projected to queries and keys and compared by a scaled dot product $q_i^\top k_j/\sqrt{d}$ before a softmax . The vectors here are random, not trained. Change the dimension and temperature to see how sharp the weights become.

Sources: 7

QUESTION

Two unit-length embeddings are at Euclidean distance $\sqrt{2}$ from each other. What is their cosine similarity?

A 1

B 0

C -1

D $\sqrt{2}/2$

SHOW THE ANSWER

A — Incorrect. A cosine of 1 means identical directions, hence distance 0.

B — Correct. From $\|\mathbf{u}-\mathbf{v}\|^2 = 2 - 2\cos$: $2 = 2 - 2\cos$, so $\cos = 0$. The vectors are orthogonal.

C — Incorrect. Opposite unit vectors are at distance 2, not $\sqrt{2}$.

D — Incorrect. That would give a squared distance of $2 - \sqrt{2} \approx 0.59$, i.e. a distance of about 0.77.

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean (2013). *Efficient Estimation of Word Representations in Vector Space.* arXiv preprint. arxiv.org/abs/1301.3781 UNVERIFIED

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean (2013). *Distributed Representations of Words and Phrases and their Compositionality.* Advances in Neural Information Processing Systems 26 (NIPS 2013). UNVERIFIED

[3] Omer Levy, Yoav Goldberg (2014). *Neural Word Embedding as Implicit Matrix Factorization.* Advances in Neural Information Processing Systems 27 (NIPS 2014). UNVERIFIED

[4] Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014). *GloVe: Global Vectors for Word Representation.* Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). UNVERIFIED

[5] Tal Linzen (2016). *Issues in evaluating semantic spaces using word analogies.* Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP. UNVERIFIED

[6] Kawin Ethayarajh (2019). *How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings.* Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP). UNVERIFIED

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems 30 (NIPS 2017). arxiv.org/abs/1706.03762 UNVERIFIED