

Filter

Linear regression (70 min)

Logistic regression (35 min)

Classification (70 min)

Data

Working with numerical data (85 min)

Working with categorical data (50 min)

Datasets, generalization, and overfitting (105 min)

Advanced ML models

Neural networks (75 min)

Embeddings (45 min)

Introduction (5 min)

Embedding space and static embeddings (10 min)

Interactive exercises (15 min)

Obtaining embeddings (15 min)

Test your knowledge (10 min)

What's next

Large language models (LLMs) (45 min)

Real-world ML



Embeddings: Interactive exercises

Send feedback

- On this page
- Task 1
- Task 2
- Task 3

The following widget, based on TensorFlow's [Embedding Projector](#), flattens 10,000 `word2vec` static vectors into a 3D space. This collapse of dimensions can be misleading, because the points closest to each other in the original high-dimensional space may appear farther apart in the 3D projection. The closest n points are highlighted in purple, with n chosen by the user in **Isolate __ points**. The sidebar on the right identifies those nearest neighbors.

Search for a point

Isolate

100 points

Embedding visualizer

Nearest points

Try to find the 20 nearest neighbors for the following, and see where the groups fall in the cloud.

- `iii`, `third`, and `three`
- `tao` and `way`
- `orange`, `yellow`, and `juice`

What do you notice about these results?

Click here for our answer

Even though `iii`, `third`, and `three` are semantically similar, they appear in different contexts in text and don't appear to be close together in this embedding space. In `word2vec`, `iii` is closer to `iv` than to `third`.

Similarly, while `way` is a direct translation of `tao`, these words most frequently occur with completely different groups of words in the dataset used, and so the two vectors are very far apart.

The first several nearest neighbors of `orange` are colors, but `juice` and `peel`, related to the meaning of `orange` as fruit, show up as the 14th and 18th nearest neighbors. `prince`, meanwhile, as in the Prince of Orange, is 17th. In the projection, the words closest to `orange` are `yellow` and other colors, while the closest words to `juice` don't include `orange`.

Task 2

Try to figure out some characteristics of the training data. For example, try to find the 100 nearest neighbors for the following, and see where the groups are in the cloud:

- `boston`, `paris`, `tokyo`, `delhi`, `moscow`, and `seoul` (this is a trick question)
- `jane`, `sarah`, `john`, `peter`, `rosa`, and `juan`

Click here for our answer

Many of the nearest neighbors to `boston` are other cities in the US. Many of the nearest neighbors to `paris` are other cities in Europe. `tokyo` and `delhi` don't seem to have similar results: one is associated with cities around the world that are travel hubs, while the other is associated with `india` and related words. `seoul` doesn't appear in this trimmed-down set of word vectors at all.

It seems that this dataset contains many documents related to US national geography, some documents relate to European regional geography, and not much fine-grained coverage of other countries or regions.

Similarly, this dataset seems to contain many male English names, some female English names, and far fewer names from other languages. Note that Don Rosa wrote and illustrated Scrooge McDuck comics for Disney, which is the likely reason that ``scrooge`` and ``mcduck`` are among the nearest neighbors for ``rosa``.

The pre-trained word vectors offered by `word2vec` were in fact trained on [Google News articles up to 2013](#).

Task 3

Embeddings aren't limited to words. Images, audio, and other data can also be embedded. For this task:

- Open TensorFlow's [Embedding Projector](#).
- In the left sidebar titled **Data**, choose **Mnist with images**. This brings up a projection of the embeddings of the [MNIST](#) database of handwritten digits.
- Click to stop the rotation and choose a single image. Zoom in and out as needed.
- Look in the right sidebar for nearest neighbors. Are there any surprises?
 - Why do some `7` s have `1` s as their nearest neighbor? Why do some `8` s have `9` as their nearest neighbor?
 - Is there anything about the images on the edges of the projection space that seem different from the images in the center of the projection space?

Keep in mind that the model that created these embeddings is receiving image data, which is to say, pixels, and choosing a numerical vector representation for each image. The model doesn't make an automatic mental association between the image of the handwritten digit and the numerical digit itself.

Click here for our answer

Due to similarities in shape, the vector representations of some of the skinnier, narrower `7` s are placed closer to the vectors for handwritten `1` s. The same thing happens for some `8` s and `9` s, and even some of the `5` s and `3` s.

The handwritten digits on the outside of the projection space appear more strongly definable as one of the nine digits and strongly differentiated from other possible digits.

- A

Key terms:

 - [Embedding vector](#)
 - [Embedding space](#)

Help Center

Previous

← Embedding space and static embeddings (10 min)

Next

Obtaining embeddings (15 min) →

Was this helpful?

👍👎

Send feedback

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see the [Google Developers Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-04-15 UTC.

- Connect

Blog

Instagram

LinkedIn

X (Twitter)

YouTube
- Programs

Google Developer Groups

Google Developer Experts

Accelerators

Women Techmakers

Google Cloud & NVIDIA
- Developer consoles

Google API Console

Google Cloud Platform Console

Google Play Console

Firebase Console

Actions on Google Console

Cast SDK Developer Console

Chrome Web Store Dashboard

Google Home Developer Console