

[Sign In](#)[PyTorch](#) [Natural Language Processing](#) [Artificial Intelligence](#) [Machine Learning](#)

## What does PyTorch Embedding do?

Ad by DatadogHQ.com

### Keep Hadoop jobs and pipelines flowing.

Get end-to-end visibility into your data infrastructure with dashboards &amp; automated alerts.

Free trial.

[Start Now](#)

### 1 Answer



Prasoon Goyal, Have been working in Machine Learning for a few years  
Answered Jan 27, 2018 · Author has **769** answers and **3.1m** answer views

I assume you are referring to [torch.nn.Embedding](#) . Every deep learning framework has such an embedding layer. Let's see why it is useful.

Suppose you are working with images. An image is represented as a matrix of RGB values. Each RGB value is a feature that is numerical, that is, values 5 and 10 are closer than values 5 and 100. This information is implicitly used by the network to identify which images are close to each other, by comparing their individual pixel values.

Now, let's say you are working with text, in particular, sentences. Each sentence is composed of words, which are categorical variables, not numerical. How would you feed a word to a NN? One way to do this is to use one-hot vectors, wherein, you decide on the set of all words you will use [the "vocabulary"]. Let's say your vocabulary has 10000 words, and you have defined an ordering over these words — "a", "the", "they", "are", "have", etc. Now, you can represent the first word in the ordering ["a"] as [1, 0, 0, 0, ...], which is a vector of size 10000 with all zeros except a 1 at position 1. Similarly, the second, third, ..., words can be defined as [0, 1, 0, 0, ...], [0, 0, 1, 0, ...], ... So, the  $i^{th}$  word will be a vector of size 10,000 with all zeros, except a 1 at the  $i^{th}$  position. Now, we have a way to feed the words into the NN. But the notion of distance that we had in case of images is not present. All words are equidistant from all other words. Secondly, the dimension of the input is huge. Your vocabulary size could easily go to 100,000 or more.

Therefore, instead of having a sparse vector for each word, you can have a dense vector for each word, that is, multiple elements of the vector are nonzero and each element of the vector can take continuous values. This immediately reduces the size of the vector. You can have infinite number of unique vectors of size, say 10, where each element can take any arbitrary value [as opposed to one-hot vectors where each element could take only values 0 or 1]. So, for instance, "a" could be represented as [0.13, 0.46, 0.85, 0.96, 0.66, 0.12, 0.01, 0.38, 0.76, 0.95], "the" could be represented as [0.73, 0.45, 0.25, 0.91, 0.06, 0.16, 0.11, 0.36, 0.76, 0.98], and so on. The size of the vectors is a hyperparameter, set using cross-validation. So, how do you feed these dense vector representations of words into the network? The answer is an embedding layer — you will have an embedding layer that is essentially a matrix of size 10,000 x 10 [or more generally,  $\text{vocab\_size} \times \text{dense\_vector\_size}$ ]. For every word, you have an index in the vocabulary, like "a" -> 0, "the" -> 1, etc., and you simply look up the corresponding row in the embedding matrix to get its 10-dimensional representation as the output.

Now, the embedding layer could be fixed, so that you don't train it when you train the NN. This could be done, for instance, when you initialize your embedding layer using pretrained word vectors for the words. Alternately, you can initialize the embedding layer randomly, and train it with the other layers. Finally, you could do both — initialize with the word vectors and finetune on the task. In any case, the embeddings of similar words are similar, solving the issue we had with one-hot vectors.

13.8k views · View 106 Upvoters

Sponsored by G-Research

### What are some advanced recursion techniques in F#?

Recursive functions can be highly beneficial. Here are some advanced recursion techniques in F#.

[Learn More](#)

### Related Questions

### Related Questions

[It's 2020. Is it safe to stick with PyTorch instead of TF2.0?](#)[Why do people use PyTorch?](#)[What are the limitations of PyTorch?](#)[I heard that for deep learning, Facebook uses PyTorch and Google uses TensorFlow. What does Amazon use?](#)[Will Pytorch overtake Tensorflow?](#)[Is PyTorch better than TensorFlow?](#)

[Sign In](#)

[What are the limitations of PyTorch?](#)

[I heard that for deep learning, Facebook uses PyTorch and Google uses TensorFlow. What does Amazon use?](#)

[Will Pytorch overtake Tensorflow?](#)

[Is PyTorch better than TensorFlow?](#)

[Is PyTorch an open source?](#)

[What is PyTorch used for?](#)

[In 2019, is PyTorch still worth learning, given that Google has released TensorFlow 2.0?](#)

[Why are people shifting from TensorFlow to PyTorch?](#)

[Is PyTorch faster than Keras?](#)

[As a beginner, which one should I go for, TensorFlow or PyTorch?](#)

[Should I go for TensorFlow or PyTorch?](#)

[How do I learn PyTorch within one week?](#)

[How does PyTorch dataloader work?](#)

[About](#) · [Careers](#) · [Privacy](#) · [Terms](#) · [Contact](#)