



MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning[Course](#)[Progress](#)[Dates](#)[Discussion](#)[Resources](#)[🏠](#) [Course](#) / [Unit 3. Neural networks \(2.5 weeks\)](#) / [Homework 3](#)[← Previous](#)

4. Word Embeddings

[🔖](#) Bookmark this page

Homework due Mar 29, 2023 08:59 -03 Completed

Word Embeddings

0/1 point (graded)

Training a neural network using back-propagation and SGD moves the network weights to minimize the loss function. If the network contains a *bottleneck*, a layer in which many inputs are reduced to only a few outputs, training will adjust the weights to maximize the useful information that can be passed to the output. In this way, a sparse input representation can be embedded in a lower-dimensional, dense, *distributed* representation. Embeddings often have interesting properties like transferring visual similarity into geometric proximity.

For example, imagine you have the words "cat", "lion", "car", "bridge". You could have a vector representation like: "cat" : [0.9, 0.2], "lion" : [0.9, 0.5], "car" : [0.01, 0.8], "bridge" : [0.01, 0.5]. In this representation, the first component gives (not exactly) a measure of "animalness" and the second component a measure of "man-made-ness". In addition, the vectors for similar or related words may be close together in space. In this problem, we will examine the utility of (the highly popular) word embeddings.

Consider two neural networks for classifying sequences of words that differ only in the input representation.

- The first of which uses a sparse *one-hot* encoding of each word in which word i is represented by a vector that contains a **1** in position i and **0**s elsewhere. For instance, a dictionary containing the words "cat" and "lion" might be represented as $\begin{bmatrix} 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \end{bmatrix}$, respectively. You may assume the dictionary contains all words in both the training and testing sets.
- The second neural network, instead, uses a pre-trained embedding of the dictionary where each vector represents every word in the dictionary.



Assuming that both networks use **tanh** activations and have randomly initialized weights, which of the following statement(s) is/are true about the output of the network for this sequence?



Now, at test time, each network is presented with a sequence of words not seen during training. Which of the following statement(s) is/are true about the output of the network for this sequence?

[About](#)
[Affiliates](#)
[edX for Business](#)
[Open edX](#)
[Careers](#)
[News](#)


The second network has a fighting chance at classifying the sequence

Legal



The first network has a fighting chance at classifying the sequence

[Terms of Service & Honor Code](#)
[Privacy Policy](#)
[Accessibility Policy](#)

[Previous](#)[Next >](#)

© 2023 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)