CS156 - Final Project: Learning Deep Learning via Word2Vec

Omer Ben-Ami

Minerva Schools at KGI

12/15/2017

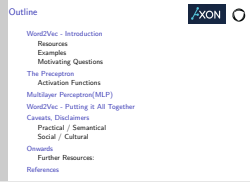
CS156 - Final Project: Learning Deep Learning via Word2Vec

Omer Ben-Ami

Minerva Schools at KGI

12/15/2017

Outline



Explain: this is not a thorough explanation about Word2Vec but rather a way to learn about the core components of DeepLearning by learning how word2vec works.

Word2Vec - Introduction

Read the following Wikipedia excerpt first:
"Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in t'he corpus are located in close proximity to one another in the space."

Now, watch the following two videos:

- **1- Quick Intro Explanation**
- **2 -Detailed Stanford Lecture (0:00 - 42:30)**

Word2Vec - Introduction
Read the following Wikipedia excerpt first:
"Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in t'he corpus are located in close proximity to one another in the space."
Now, watch the following two videos:
1- Quick Intro Explanation
2 -Detailed Stanford Lecture (0:00 - 42:30)

the videos are crucial in order to get started with jargon, terms and intuitions.

Introduction - Continued

You might won't understand some of the jargon; that's OK. The idea is to start getting familiar with the #context and the terms. By the end of these lectures you should be able to answer the following questions:

- What is a "one-hot" representation? What is its major drawback?
- What is the distribution of similarity, and why it is important?
- What is a "one-hot" representation?
- Fill in the blank, and explain: "You shall know a word by ___it ___".

Introduction - Continued

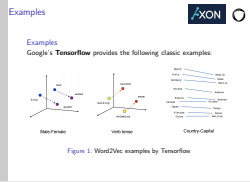
You might won't understand some of the jargon; that's OK. The idea is to start getting familiar with the #context and the terms. By the end of these lectures you should be able to answer the following questions:

- What is a "one-hot" representation? What is its major drawback?
- What is the distribution of similarity, and why it is important?
- What is a "one-hot" representation?
- Fill in the blank, and explain: "You shall know a word by ___it ___".

Introduction - Continued

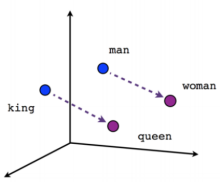
- What are the differences between CBOW and skip grams algorithms? What are the trade-offs?
- Is Skip-Gram uses Bayesian Modeling? If so, How?
- What are the hyper-parameters of the model? What are the parameters?

Introduction - Continued
- What are the differences between CBOW and skip grams algorithms? What are the trade-offs?
- Is Skip-Gram uses Bayesian Modeling? If so, How?
- What are the hyper-parameters of the model? What are the parameters?

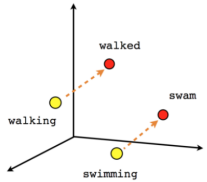


Examples

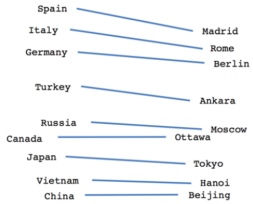
Google's **Tensorflow** provides the following classic examples:



Male-Female



Verb tense



Country-Capital

Figure 1: Word2Vec examples by Tensorflow

Examples - Continued

You can see that with Word2Vec we can map words into word embeddings using dimension-reduction, to preserve and find new semantic insights.

Motivating Questions:

- What are the benefits of this model, and in which instances will it fail?
- Recall - what are dimension-reduction techniques you have learned so far in CS156?
- What are dangerous implications of such a model, if it were to be widely accepted and implemented in the AI industry?

Examples - Continued
AXON
You can see that with Word2Vec we can map words into word embeddings using dimension-reduction, to preserve and find new semantic insights.
Motivating Questions:
- What are the benefits of this model, and in which instances will it fail?
- Recall - what are dimension-reduction techniques you have learned so far in CS156?
- What are dangerous implications of such a model, if it were to be widely accepted and implemented in the AI industry?

Customized Videos

Now proceed to **this custom-made CS156 playlist** and watch it in tandem with this slide-deck.

Customized Videos
Now proceed to **this custom-made CS156 playlist** and watch it in tandem with this slide-deck.

Activation Functions:

"...[T]he activation function of a node defines the output of that node given an input or set of inputs." (Wikipedia)

Function Types:

Sigmoid: A classic one that has been used is the Sigmoid, the same one used for logistic regression (and from there, we understand that it makes sense to use it, as it is used for classification, and fits "fire" - "don't fire" task).

Definition: $S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}.$

Derivative: $S'(x) = S(x) \cdot (1 - S(x))$

Activation Functions:

"...[T]he activation function of a node defines the output of that node given an input or set of inputs." (Wikipedia)

Function Types:

Sigmoid: A classic one that has been used is the Sigmoid, the same one used for logistic regression (and from there, we understand that it makes sense to use it, as it is used for classification, and fits "fire" - "don't fire" task).

Definition: $S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}.$

Derivative: $S'(x) = S(x) \cdot (1 - S(x))$

Activation Func. - Cont.

ReLU: Rectified Linear Unit is considered the gold standard nowadays amongst the common activation functions, nevertheless it is very simple.

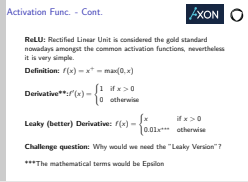
Definition: $f(x) = x^+ = \max(0, x)$

Derivative:** $f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

Leaky (better) Derivative: $f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x^{***} & \text{otherwise} \end{cases}$

Challenge question: Why would we need the "Leaky Version"?

***The mathematical terms would be Epsilon



Ask what could be problematic with this piecewise function? the answer leads to the "leaky version"

Activation Func. Cont.

Video: Activation Functions

Notebook: Activation Functions

Activation Func. Cont.

Video: Activation Functions

Notebook: Activation Functions

The Perceptron

- **Video:** Perceptron
- **Notebook:** Notebook

The Perceptron

- **Video:** Perceptron
- **Notebook:** Notebook

Multilayer Perceptron(MLP)

- **Video:**MLP Tutorial
- **Notebook:** MLP

Multilayer Perceptron(MLP)
AXON
Video:MLP Tutorial
Notebook: MLP

Word2Vec - Putting it All Together

Video: Theoretical Background
Notebook: Word2Vec-Gensim

To learn how the model learns, watch the Stanford lecture (00:46 onwards).

Word2Vec - Putting it All Together
Video: Theoretical Background
Notebook: Word2Vec-Gensim
To learn how the model learns, watch the Stanford lecture (00:46 onwards).

Very important to tie back to perceptron and activation functions, explain how SoftMax is analogous.explain how libraries work, and why we should use them in the context of ANNs

Caveats, Disclaimers

- Does language static or dynamic? (hint: consider the following relationships in the 1910's and 2010's: Banking - Crisis, LOL - laughter etc...)
- Reflection of the current state of language:
  - Gender, bigotry and discrimination - how could those factor into word embeddings? What are possible consequences?

Caveats, Disclaimers
- Does language static or dynamic? (hint: consider the following relationships in the 1910's and 2010's: Banking - Crisis, LOL - laughter etc...)
- Reflection of the current state of language:
  - Gender, bigotry and discrimination - how could those factor into word embeddings? What are possible consequences?

There have been early attempts to point at or shed light on inherent biases in language that are manifested in word embeddings. This is of a major concern, not merely harmful potential.

Onwards

- Building intuition to understand ReLU: **ReLU Tutorial**
- For further exploration, **Word2Vec original paper** is a good place to start.
- Explore the training methods of Word2Vec: **Relevant Blog Post**

Onwards
AXON
- Building intuition to understand ReLU: **ReLU Tutorial**
- For further exploration, **Word2Vec original paper** is a good place to start.
- Explore the training methods of Word2Vec: **Relevant Blog Post**

References:

- Activation funcs: ReLU and Sigmoid. (n.d.). Retrieved from https://codereview.stackexchange.com/questions/182537/activation-funcs-relu-and-sigmoid
- Barazza, L. (2017, February 18). How does Word2Vec? Skip-Gram work? Becoming Human: Artificial Intelligence Magazine. Retrieved from $https://becominghuman.ai/how-does-word2vecs-skip-gram-work-f92e0525def4$
- Deriving the Sigmoid Derivative for Neural Networks. (2017, August 6). Retrieved from https://beckernick.github.io/sigmoid-derivative-neural-network/
- How to implement the ReLU function in Numpy. (n.d.). Retrieved from https://stackoverflow.com/questions/32109319/how-to-implement-the-relu-function-in-numpy

References:
- Activation funcs: ReLU and Sigmoid. (n.d.). Retrieved from https://codereview.stackexchange.com/questions/182537/activation-funcs-relu-and-sigmoid
- Barazza, L. (2017, February 18). How does Word2Vec? Skip-Gram work? Becoming Human: Artificial Intelligence Magazine. Retrieved from $https://becominghuman.ai/how-does-word2vecs-skip-gram-work-f92e0525def4$
- Deriving the Sigmoid Derivative for Neural Networks. (2017, August 6). Retrieved from https://beckernick.github.io/sigmoid-derivative-neural-network/
- How to implement the ReLU function in Numpy. (n.d.). Retrieved from https://stackoverflow.com/questions/32109319/how-to-implement-the-relu-function-in-numpy

References - Cont.

- Lecture 2 Word Vector Representations: word2vec. (2017, April 3). Retrieved from *https://www.youtube.com/watch?v=ERibwqs9p38*
- Perceptron. (2017, November 11). Retrieved from *https://en.wikipedia.org/wiki/Perceptron*
- Word2Vec. (2016, June 6). Retrieved from *https://www.youtube.com/watch?v=xMwx2A$_o$5r4*
- What is the derivative of ReLU? - kawahara.ca. (2016, May 17). Retrieved from http://kawahara.ca/what-is-the-derivative-of-relu/

References - Cont.

- Lecture 2 Word Vector Representations: word2vec. (2017, April 3). Retrieved from *https://www.youtube.com/watch?v=ERibwqs9p38*
- Perceptron. (2017, November 11). Retrieved from *https://en.wikipedia.org/wiki/Perceptron*
- Word2Vec. (2016, June 6). Retrieved from *https://www.youtube.com/watch?v=xMwx2A$_o$5r4*
- What is the derivative of ReLU? - kawahara.ca. (2016, May 17). Retrieved from http://kawahara.ca/what-is-the-derivative-of-relu/