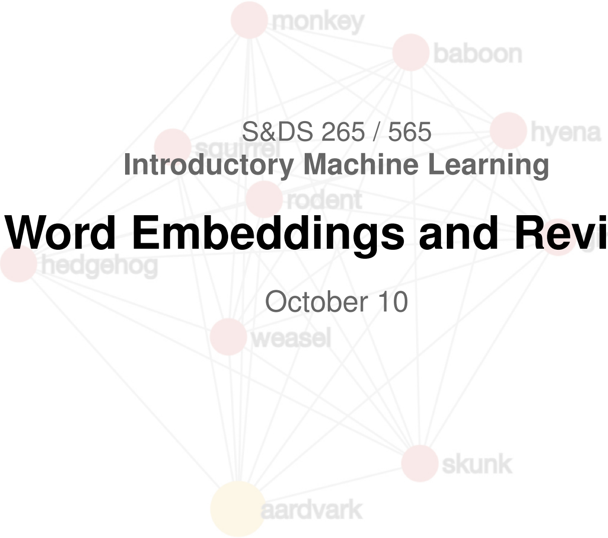


S&DS 265 / 565

Introductory Machine Learning

Word Embeddings and Review

October 10

Yale

Plan for today

- Reminders
- Recap of word embeddings
- Demo notebook
- Review–AMA

Reminders

- Assn 3 is out; due October 24
- Quiz 3 posted today; open at 1pm; discriminitive vs. generative models, trees, bias/variance, SGD – good review for midterm!
- Midterm Tuesday, October 15, in class
- “Closed book, notes, computer...”
- $8\frac{1}{2} \times 11$ sheet of notes, handwritten double-sided
- Practice midterms posted on Canvas (with solutions)

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1})$$

Modern language models

Suppose a computer program assigns a "score" to possible next words v:

$$s(\underset{\substack{\uparrow \\ \text{possible next word}}}{v};\ \overbrace{w_1, \ldots, w_n}^{\text{previous words}}\ )$$

---

https://en.wikipedia.org/wiki/Softmax_function

Modern language models

Suppose a computer program assigns a "score" to possible next words v:

$$s(\underset{\substack{\uparrow \\ \text{possible next word}}}{v};\ \overbrace{w_1, \ldots, w_n}^{\text{previous words}}\ )$$

Can convert this to a language model by the "softmax" operation:

$$p(w \mid w_1, \ldots, w_n) = \frac{\exp(s(w; w_1, \ldots, w_n))}{\sum_{v \in V} \exp(s(v; w_1, \ldots, w_n))}$$

https://en.wikipedia.org/wiki/Softmax_function

Modern language models

Suppose a computer program assigns a "score" to possible next words v:

$$s(v; \overbrace{w_1, \dots, w_n}^{\text{previous words}})$$
$$\underset{\text{possible next word}}{\uparrow}$$

Can convert this to a language model by the "softmax" operation:

$$p(w \mid w_1, \dots, w_n) = \frac{\exp(s(w; w_1, \dots, w_n))}{\sum_{v \in V} \exp(s(v; w_1, \dots, w_n))}$$

In ChatGPT, the function $s(v; w_{1:n})$ is learned on large amounts of text (unsupervised) using a type of deep neural network called a *transformer*.

https://en.wikipedia.org/wiki/Softmax_function

Modern language models

A language model assigns a “score” to possible next words v:

$$s(v;\ \underbrace{w_1, \ldots, w_n}_{\text{word history}}\ )$$

Modern language models

A language model assigns a "score" to possible next words v:

$$s(v;\ \underbrace{w_1, \ldots, w_n}_{\text{word history}}\ )$$

Today, we'll be working with a simple case where

$$\begin{aligned} s(v; w_1, \ldots, w_n) &= \beta_v^T \phi(w_1, \ldots, w_n) \\ &= \beta_v^T \phi(w_n) \\ &= \phi(v)^T \phi(w_n) \end{aligned}$$

Modern language models

A language model assigns a “score” to possible next words v:

$$s(v;\ \underbrace{w_1, \ldots, w_n}_{\text{word history}}\ )$$

Today, we'll be working with a simple case where

$$\begin{aligned} s(v; w_1, \ldots, w_n) &= \beta_v^T \phi(w_1, \ldots, w_n) \\ &= \beta_v^T \phi(w_n) \\ &= \phi(v)^T \phi(w_n) \end{aligned}$$

Key intuition

- Similar words will appear with similar words
- Self-referential notion of similarity

Constructing embeddings

Language model is

$$p(w_2 \mid w_1) = \frac{\exp(\phi(w_2)^T \phi(w_1))}{\sum_w \exp(\phi(w)^T \phi(w_1))}.$$

Carry out stochastic gradient descent over the embedding vectors $\phi \in \mathbb{R}^d$ (where $d \approx 50$–500 is chosen by hand)

This is what Mikolov et al. (2014, 2015) did at Google.

"Distributed representations of words," (2014) "Efficient representations of words in vector space" (2015)

Constructing embeddings

`word2vec`:

- Skip-gram: predict surrounding words from current word, rather than the next word.

"Distributed representations of words," (2014) "Efficient representations of words in vector space" (2015)

Constructing embeddings

`word2vec`:

- Skip-gram: predict surrounding words from current word, rather than the next word.
- This leads to a model of nearby words $p_{\text{near}}(w_2 \mid w_1)$.

"Distributed representations of words," (2014) "Efficient representations of words in vector space" (2015)

GloVe

Shortly after, a group at Stanford group introduced a variant called "GloVe"

- Based on a type of regression model
- More scalable with SGD

Pennington et al., "GloVe: Global vectors for word representation," (2015)

Using PCA

A closely related approach is to use PCA of pointwise mutual information (PMI):

- Form $V \times V$ matrix of pointwise mutual information values

$$\log \left(\frac{p_{\text{near}}(w_1, w_2)}{p(w_1)p(w_2)} \right)$$

- Compute top k eigenvectors $\phi_1, \ldots, \phi_k$
- For each word w, define embedding as

$$\phi(w) \equiv (\phi_{1w}, \phi_{2w}, \ldots, \phi_{kw})^T$$

Analogies

Leads to vector representations of words with interesting properties.

For example, analogies:

`king` is to `man` as ? is to `woman`

Analogies

Leads to vector representations of words with interesting properties.

For example, analogies:

`king` is to `man` as ? is to `woman`

`Paris` is to `France` as ? is to `Germany`

Analogies

Leads to vector representations of words with interesting properties.

For example, analogies:

$\texttt{king}$ is to $\texttt{man}$ as ? is to $\texttt{woman}$

$\texttt{Paris}$ is to $\texttt{France}$ as ? is to $\texttt{Germany}$

$$\phi(\texttt{king}) - \phi(\texttt{man}) \stackrel{?}{\approx} \phi(\texttt{queen}) - \phi(\texttt{woman})$$

$$\widehat{w} = \arg\min_{w} \|\phi(\texttt{king}) - \phi(\texttt{man}) + \phi(\texttt{woman}) - \phi(w)\|^2$$

Does $\widehat{w} = \texttt{queen}$?

Learned Analogies

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Mikolov et al., "Distributed representations of words," (2014); "Efficient representations of words in vector space" (2015)

Evaluation Analogies

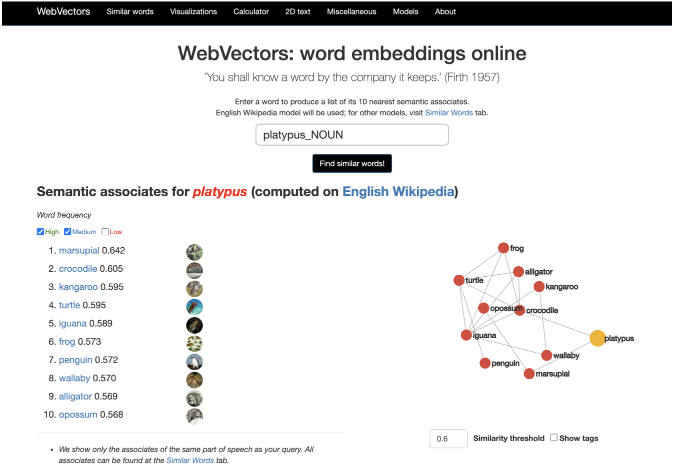
Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Mikolov et al., "Distributed representations of words," (2014); "Efficient representations of words in vector space" (2015)

Notebook

Let’s go to the Python notebook!

Embedding / Visualization Examples



http://vectors.nlpl.eu/explore/embeddings/en/

Many uses

species2vec: A novel method for species representation

Boyan Angelov

doi: https://doi.org/10.1101/461996

Abstract

Word embeddings are omnipresent in Natural Language Processing (NLP) tasks. The same technology which defines words by their context can also define biological species. This study showcases this new method - species embedding (species2vec). By proximity sorting of 6761594 mammal observations from the whole world (2862 different species), we are able to create a training corpus for the skip-gram model. The resulting species embeddings are tested in an environmental classification task. The classifier performance confirms the utility of those embeddings in preserving the relationships between species, and also being representative of species consortia in an environment.

Visualisation

In [10]:
```
m = gensim.models.KeyedVectors.load_word2vec_format('reptilia.vec')
```

In [11]:
```
len(m.vocab)
```

Out[11]:
```
7397
```

In [15]:
```
m.most_similar(u'Alligator_mississippiensis')
```

Out[15]:
```
[(u'Sternotherus_bonevalleyensis', 0.8425856828689575),
 (u'Apalone_ferox', 0.8147842884063721),
 (u'Macrochelys_suwanniensis', 0.8063992261886597),
 (u'Deirochelys_reticularia', 0.7871163487434387),
 (u'Terrapene_putnami', 0.7841686010360718),
 (u'Chelydra_floridana', 0.7829421758651733),
 (u'Alligator_mefferdi', 0.7742743492126465),
 (u'Macrochelys_temminckii', 0.7682404518127441),
 (u'Trachemys_inflata', 0.7563525438308716),
 (u'Deirochelys_carri', 0.755811333656311)]
```

In [16]:
```
%matplotlib inline

def tsne_plot(model):
    "Creates and TSNE model and plots it"
    labels = []
```

Summary: Word embeddings

- Word embeddings are vector representations of words, learned from cooccurrence statistics
- The models can be built using language modeling (or regression or PCA)
- Surprising semantic relations are encoded in linear relations—for example, analogies
- Embeddings are the “ground floor” representations in ChatGPT

1	Aug 29	Course overview		Thu: Course overview		
2	Sept 3, 5	Python and background concepts	Python elements Covid trends	Tue: Python elements Thu: Pandas and linear regression	Data8 Chapters 3, 4, 5	Quiz 1 Assn 1 out
3	Sept 10, 12	Linear regression and classification	Covid trends (revisited) Classification examples	Tue: Regression concepts Thu: Classification	ISL Sections 3.1, 3.2, 3.5 Notes on regression ISL Sections 4.3, 4.4 Notes on classification	
4	Sept 17, 19	Stochastic gradient descent	SGD examples	Tue: Classification (continued) Thu: Stochastic gradient descent	ISL Section 6.2.2 ISL Section 10.7.2	Assn 1 in Assn 2 out
5	Sept 24, 26	Bias and variance, cross-validation	Bias-variance tradeoff Covid trends (revisited) California housing	Tue: Bias and variance Thu: Cross-validation	ISL Section 2.2 ISL Section 5.1	Quiz 2
6	Oct 1, 3	Tree-based methods and principal components	Trees and forests Visualizing trees PCA examples	Tue: Trees (and Forests) Thu: Forests and PCA	ISL Sections 8.1, 8.2 ISL Section 12.2	Assn 2 in Assn 3 out
7	Oct 8, 10	PCA and dimension reduction	PCA revisited Used for dimension reduction Word	Tue: PCA and word embeddings Thu: Embeddings and review	ISL Section 12.2	Quiz 3

“Ask Me Anything” (AMA)