CS 378 Lecture 7
Word Embeddings

Announcements

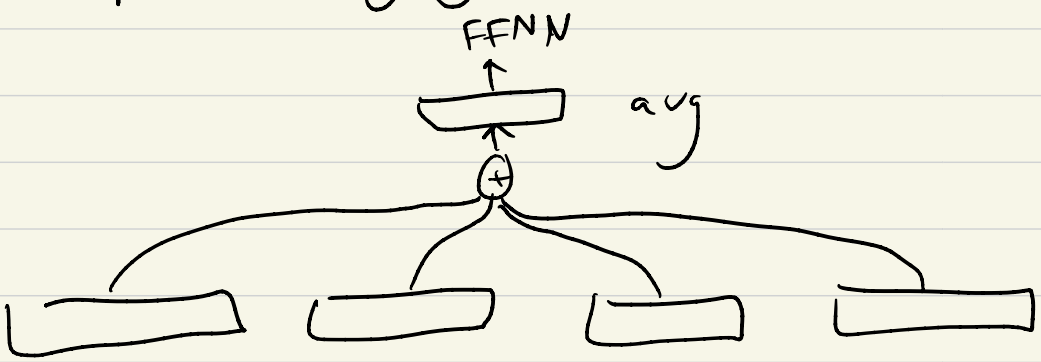
- Fairness response due today
- Friday: talk by Nanyun Peng (UCLA) 11 am 6.302

Recap Last time: FFNNs

$$P(\bar{y}|\bar{x}) = \text{softmax}(Wg(Vf(\bar{x})))$$

Pytorch basics

Deep Averaging Network (A2)



How does this relate to BoW?

Word Embeddings

So far: one-hot representations of words

movie was good →

[1 (movie) 1 (was) 1 (good)]

= movie [0000 1 0000]
style 1

+ was [00 ... 100]

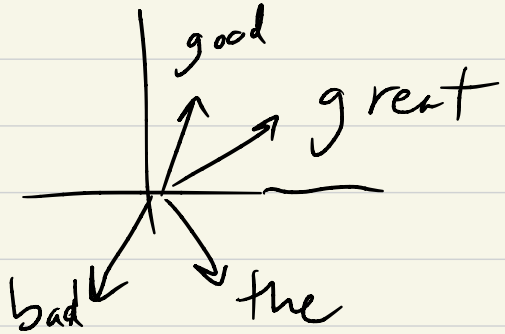
+ good [00 ... 1]

Problem: ① Long vectors

② good vs. great, not similar

film was great | movie is good

Instead of ~10k dims, how about ~100?



sim(good, great)
> sim(good, bad)

Distributional hypothesis

JR Firth 1957 "You shall know a word by the company it keeps"

I watched the movie
I watched the film
The film inspired me
The movie inspired me

I took a picture with film

There was a film on the liquid

Polysemous: word has multiple senses / meanings

Mikolov et al. 2013 word2vec

Learn 2 vectors for every word

word vec + Context vec.

Attempt to predict context given word

Skip-gram

Input: a corpus of text

Output: $\vec{v}_w$, $\vec{c}_w$ for each w in vocab
(word, context)

(for applications: use either $\vec{v}$ or $\vec{c}$ or $\vec{v}+\vec{c}$)

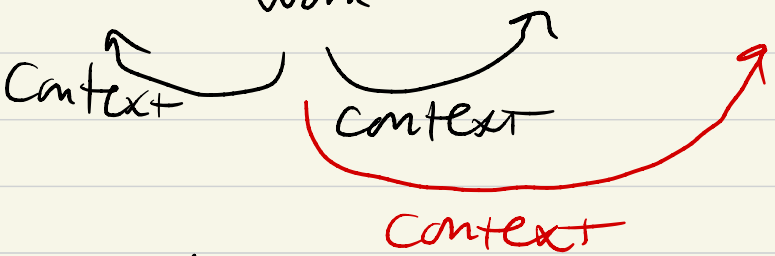
Hyperparameters: d ($50 \sim 300$)
window size K

look in both directions

Let $k=1$ 2

The film inspired me

word



word, context

(film, The)

(film, inspired)

(film, me)

} Training examples

Other pairs (The, film)

(inspired ...)

Model: (skip-gram)

vocab V

$$P(\text{context}=y \mid \text{word}=x) = \frac{e^{\bar{v}_x \cdot \bar{c}_y}}{\sum_{y' \in V} e^{\bar{v}_x \cdot \bar{c}_{y'}}}$$

distribution over all context words in V

parameters: vectors $\bar{v}$ $|V| \times d$

randomly initialized $\bar{c}$ $|V| \times d$

Training (x, y) ↑ train examples

Maximize $\sum_{(x,y) \text{ in data}} \log P(\text{context}=y \mid \text{word}=x)$

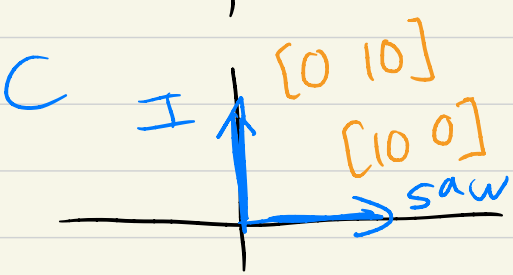
Ex Corpus = I saw $k=1$ $d=2$

vocab = {I, saw}

Assume: $\bar{v}_I = [1\ 0]$ $\bar{v}_{saw} = [0\ 1]$

① Let $\bar{c}_{saw} = [1\ 0]$

$\bar{c}_I = [0\ 1]$



What is P(context | word=saw)

$= I = \frac{3}{4}$

$$P(\text{context} = \text{saw} \mid \text{word} = \text{saw}) = \frac{e^{\bar{v}_{saw} \cdot \bar{c}_{saw}}}{e^{\bar{v}_{saw} \cdot \bar{c}_{saw}} + e^{\bar{v}_{saw} \cdot \bar{c}_I}} = \frac{1}{1 + e^{\bar{v}_{saw} \cdot \bar{c}_I - \bar{v}_{saw} \cdot \bar{c}_{saw}}}$$

$\bar{v}_{saw} \cdot \bar{c}_I = 1$, $\bar{v}_{saw} \cdot \bar{c}_{saw} = 0$, $1 - 0 = 1$

(2) What are the values of $\bar{c}$ that maximize likelihood?

see prev page

(3) Why do we have $\bar{v} \neq \bar{c}$?
Why two spaces?

dot product of word w/self would be high

(4) "we saw"
if we add this, should get $\bar{v}_{we} = \bar{v}_I$

Other methods

FastText: each word embedding
= Sum of char n grams

Problem with skip-gram:

For each example, how expensive is it to compute $P(c|w)$?

$|V|$ vocab d-dim vectors

$O(|V|d)$ one evaluation

Corpus: $|C| \cdot k$

whole training: $O(|V|d|C|k)$

Alternative:

Skip-gram with negative sampling (SGNS)

Take (word, context) pairs as "real" data

Sample "fake" data

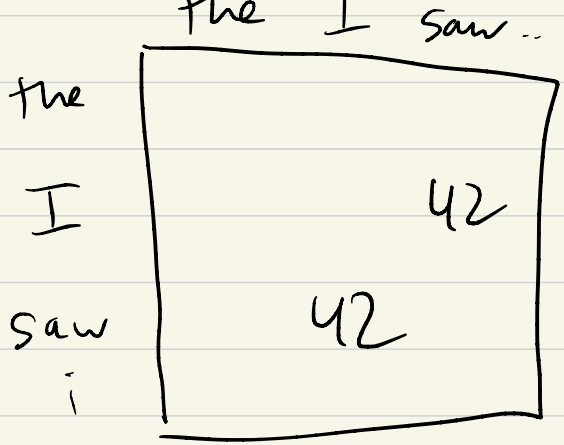
Learn a binary classifier

$$P(\text{Real} | x, y) = \frac{e^{\vec{v}_x \cdot \vec{c}_y}}{1 + e^{\vec{v}_x \cdot \vec{c}_y}}$$

(2014)

GloVe Global Vectors

Factorizes a matrix of (word, context) counts



$|V| \times |V|$

$SG \approx SGNS \approx GloVe$

GloVe has no dependence on corpus size