Skip-gram

Mikolov et al. 2013 "word2vec"

Learn 2 vectors for every word

Word vector

Context vector

Try to predict context given word

Inputs: a corpus of text

Output: $\vec{V}_w, \vec{C}_w$ for each word $w$ in vocab

(word $\vec{C}_w$ for $A_2$: use $\vec{V}_w$ or $\vec{V} + \vec{C}$)

Hyperparameters: $d$ (50-300)

Window size $K$
Let $K=1$  $K=2$ includes $K=1$

The film inspired me

(context) (context) (context) (context) (context)

(word) predict (Film, The) (film, inspired) (film, me)

Training examples

(skip-gram)
Model (skip-gram) $\text{V}_X$

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

distribution over context words in V

parameters: vectors $\text{V} \mid \text{U}\times \text{d}$

context vecs $\text{C} \mid \text{U}\times \text{d}$

randomly initialized

Training $(x, y)$ train exs

Minimize $\sum_{(x, y)} -\log P(\text{context} = y \mid \text{word} = x)$
\begin{align*}
\text{Ex} & \quad \text{Corpus} = \text{i} \text{saw} \quad k=1 \\
\text{vocab} & = \{i, \text{saw}\} \quad d=2 \\
\text{Assume} & \quad \overline{v}_i = [1, 0] \quad \overline{v}_{\text{saw}} = [0, 1] \\
\text{What is} & \quad P(\text{context} \mid \text{word} = \text{saw}) \\
2 \text{ outcomes} & \quad (i, \text{saw}) \\
\overline{v}_{\text{saw}} \cdot \overline{c}_i & = e \quad e \\
= 1 & \\
\text{P}(I \mid \text{saw}) & = \frac{e}{e + 1} \quad P(\text{saw} \mid \text{saw}) \\
& \approx \frac{3}{4} \quad = \frac{1}{e + 1} \approx \frac{1}{4}
\end{align*}
2. How to minimize loss further by changing $c$?

$c_i = [0, 2] \Rightarrow c_{i+} = [0, 10]$

$\frac{e^{10}}{e^{10} + 1} \approx 0.999$

3. Why is $\sqrt{c}$? Why two spaces?

(saw, saw) always be high!

Word vector selects for words that are near it

Noun $\Rightarrow$ verbs

Noun $\Rightarrow$ nouns
Problems with skip-gram

If we ran this training over 100M word corpus with $V = 30k$, what's going to be hard?

- polysemy: different word senses
  - different vector per sense?
- train on a homogeneous corpus
- context-dependent vectors ($\text{BERT, GPT}$)
- computation: $|U| \cdot d \approx 50 \sim 300$

$p(y|x) = O(|U|d)$

For training: do that $\times 100M$
Two fixes

Skip-gram w/negative sampling

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

\((word, \sim sampled\cdot context)\) as fake data

Learn a classifier

\[
P(\text{real} \mid y, x) = \frac{e^{\langle \vec{v}_x, \vec{c}_y \rangle}}{1 + e^{\langle \vec{v}_x, \vec{c}_y \rangle}}
\]

(film, buy) is this fake?
GloVe

Factorizes a matrix of (word, context) counts \((K=1)\)

\[
\begin{array}{ccc}
\text{the} & 25 & 12 \\
\text{I} & 25 & 1512 \\
\text{saw} & 12 & 1512 \\
\vdots & \vdots & \vdots \\
\end{array}
\]

\[
|U| \times |V|
\]

Matrix factorization

\[
(1\times r) \times (r \times |U|) \approx |U| \times |V|
\]

Same as SG + SGNS