lecture 8|13 (LAST CLASS, no class FRI 8|15)

- word embedding
- Glove + HW6 (optional) demo

logistics: HW5 done (closed)
HW6 due FRI incl demo
PROJ: submitted, look for discussion?

Introduction to Deep Learning

22. Embeddings, Word2wec, fastText, GloVe

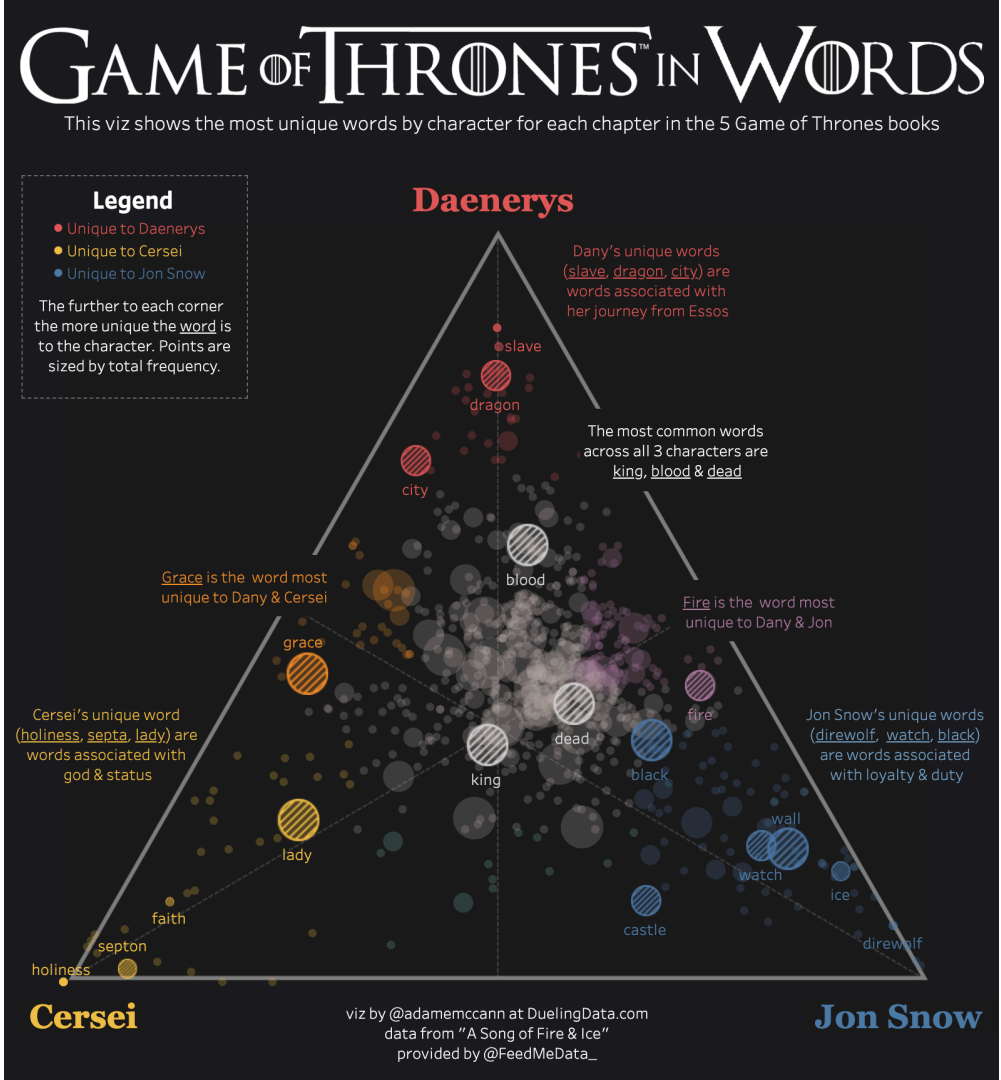
STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

aws

Word2Vec

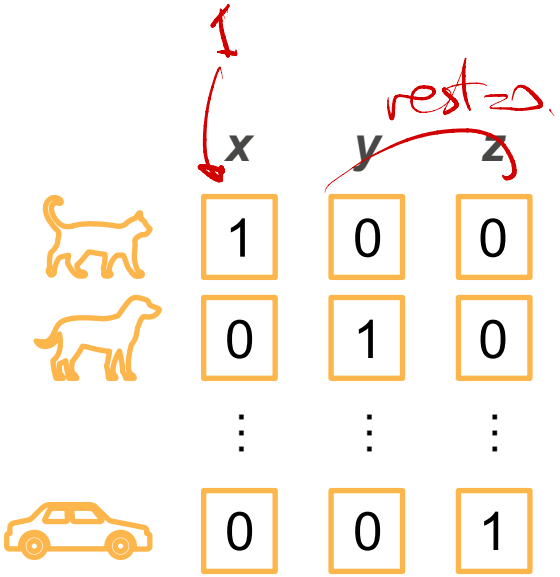


courses.d2l.ai/berkeley-stat-157

Motivation

- One-hot vectors map objects/ words into fixed-length vectors
- These vectors only contain the identity information, not semantic meaning, e.g.

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{z}, \mathbf{y} \rangle = 0$$



Word2vec

300 dim real numbers

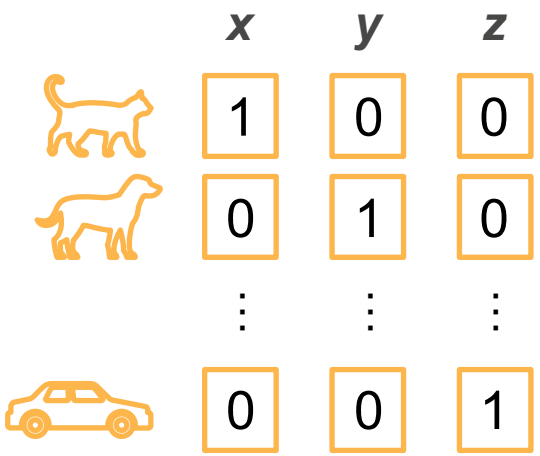
- Learn an embedding vector for each word
- Use $\langle \mathbf{x}, \mathbf{y} \rangle$ to measure the similarity

dot prod for sim

$$\text{sim}(x, y) = \langle \mathbf{x}, \mathbf{y} \rangle > \langle \mathbf{z}, \mathbf{y} \rangle$$

- Build a probability model — multiple ways
- Maximize the likelihood function to learn the model

loss min = meaningful word vectors

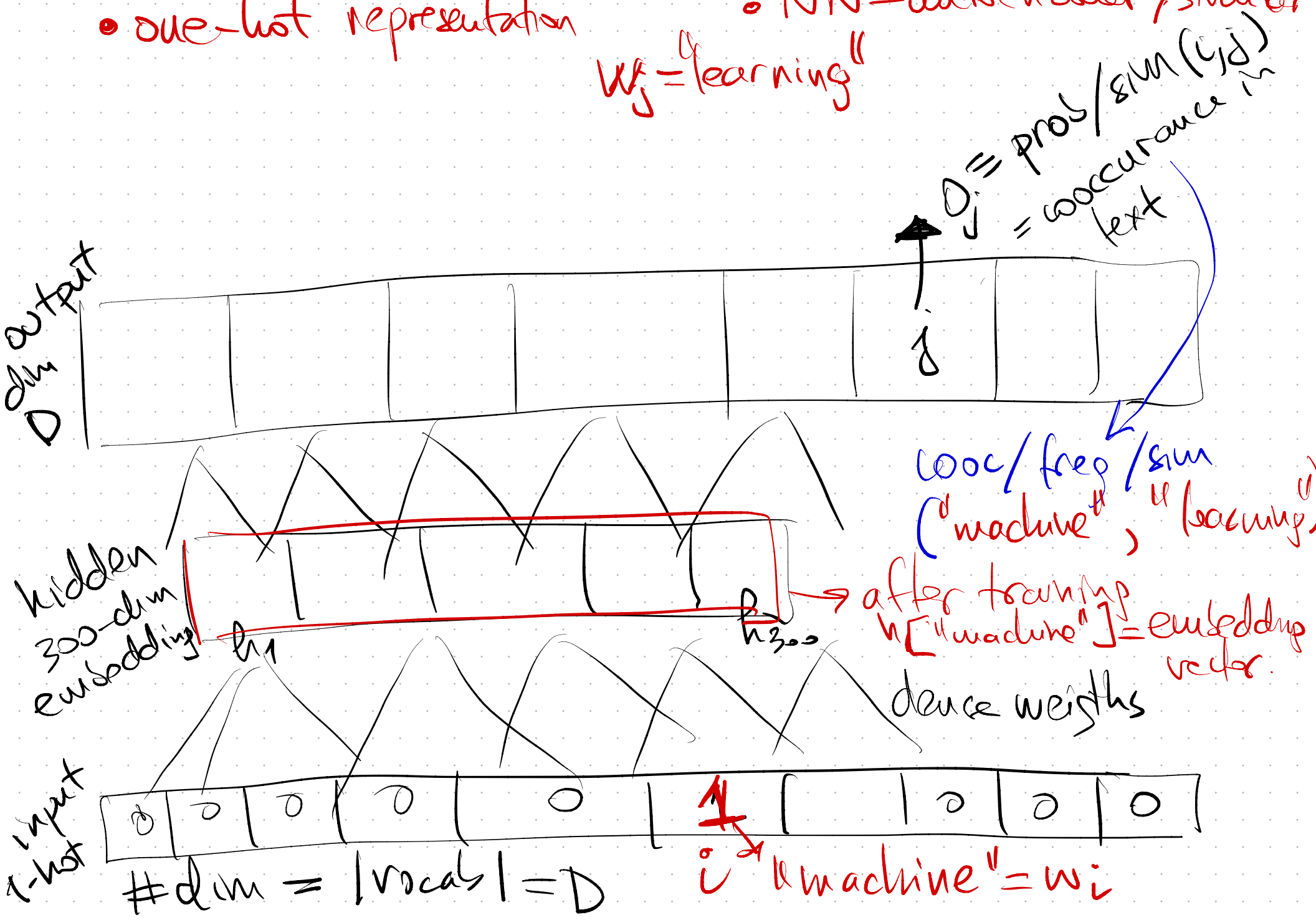


- Tricks for fast computation

simple/basic idea

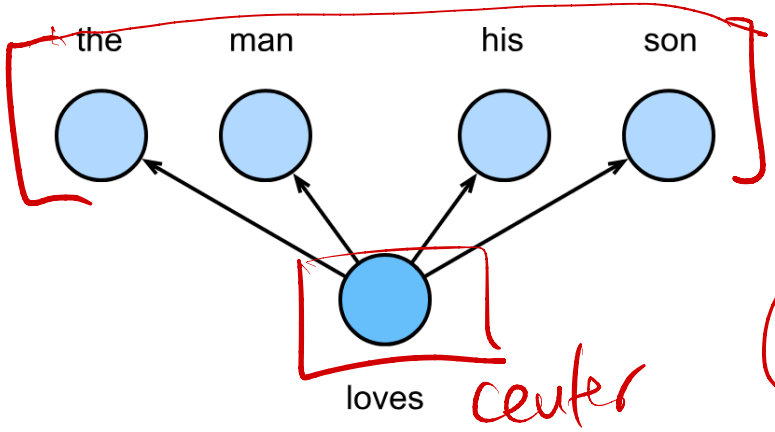
- one-hot representation
- NN-autoencoder/similar

W_j = "learning"



The Skip-Gram Model

- A word can be used to generate the words surround it
- Given the center word, the context words are generated independently



$$P[\text{context\_words} \mid \text{center}]$$

$$\mathbb{P}(\text{"the"}, \text{"man"}, \text{"his"}, \text{"son"} \mid \text{"loves"})$$

$$= \mathbb{P}(\text{"the"} \mid \text{"loves"}) \times \mathbb{P}(\text{"man"} \mid \text{"loves"}) \times \mathbb{P}(\text{"his"} \mid \text{"loves"}) \times \mathbb{P}(\text{"son"} \mid \text{"loves"})$$

naive assumption of indep

Likelihood Function

Summing over all words is too expensive

Word		Embedding
Center	w_c	$\mathbf{v}_c \in \mathbb{R}^d$
Context	w_o	$\mathbf{u}_o \in \mathbb{R}^d$

$$\mathbb{P}(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

$\mathcal{V}$: all context words

- Given length T sequence, context window m, the likelihood function:

$$\prod_{t=1}^{T} \prod_{-m \le j \le m,\ j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)})$$

TRICKS for computation

- Simplify/approx SOFTMAX; do not compute normaliz denominator $\boxed{\text{sigmoid}(u_c^T \cdot v_o)}$ instead
  - much easier to compute. prob(in window)
- $P(D=1 \mid w^t, w^{t+j})$ = prob that w^t, w^{t+j} are in same context window
- Negative sampling: include words w_n NOT in context window
  - $w_n \rightarrow$ embedd v_n
  - select some (not all)
  - use probab $\boxed{P(\quad) = 1 - \text{sigmoid}(u_c^T \cdot v_n)}$ prob(not in window)

not include all counts.

- Subsampling frequent words $\left(\begin{matrix}\text{POS}\\ \& \\ \text{neg}\end{matrix}\right)$ $P(w_i) = \max\left(1 - \sqrt{\frac{t}{f(w_i)}}, 0\right)$

Negative Sampling

simplify/approx SOFTMAX

- Treat a center word and a context word appear in the same context window as an event

$$\mathbb{P}\left(D=1 \mid w_c, w_o\right) = \sigma\left(\mathbf{u}_c^T \mathbf{v}_o\right) \qquad \sigma(x) = \frac{1}{1+\exp(-x)}$$

- Change the likelihood function from $\prod_{t=1}^{T} \prod_{-m \le j \le m,\ j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)})$ to

$$\prod_{t=1}^{T} \prod_{-m \le j \le m,\ j \neq 0} \mathbb{P}(D=1 \mid w^{(t)}, w^{(t+j)})$$

Naive solution: infinity

Negative Sampling

- Sample noise word w_n that doesn't appear in the window

$$\mathbb{P}\left(D = 0 \,|\, w_c, w_n\right) = 1 - \sigma\left(\mathbf{u}_n^T \mathbf{v}_c\right)$$

- Add into the likelihood function as well
- Maximizing the likelihood equals to solve a binary classification problem with a binary logistic regression loss

The skip-gram model parameters are the center word vector and context word vector for each word in the vocabulary. In training, we learn the model parameters by maximizing the likelihood function (i.e., maximum likelihood estimation). This is equivalent to minimizing the following loss function:

$$-\sum_{t=1}^{T} \sum_{-m \leq j \leq m,\ j \neq 0} \log P(w^{(t+j)} \mid w^{(t)}). \tag{15.1.6}$$

When using stochastic gradient descent to minimize the loss, in each iteration we can randomly sample a shorter subsequence to calculate the (stochastic) gradient for this subsequence to update the model parameters. To calculate this (stochastic) gradient, we need to obtain the gradients of the log conditional probability with respect to the center word vector and the context word vector. In general, according to (15.1.4) the log conditional probability involving any pair of the center word w_c and the context word w_o is

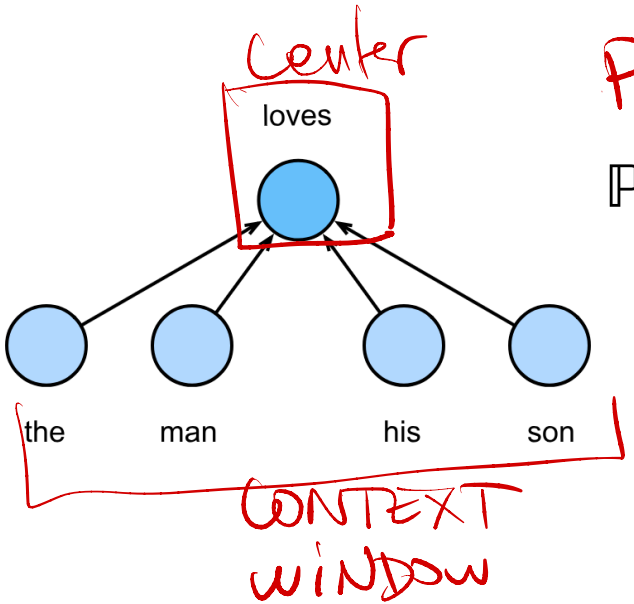
$$\log P(w_o \mid w_c) = \mathbf{u}_o^\top \mathbf{v}_c - \log \left(\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c) \right). \tag{15.1.7}$$

Through differentiation, we can obtain its gradient with respect to the center word vector $\mathbf{v}_c$ as

$$\begin{aligned} \frac{\partial \log P(w_o \mid w_c)}{\partial \mathbf{v}_c} &= \mathbf{u}_o - \frac{\sum_{j \in \mathcal{V}} \exp(\mathbf{u}_j^\top \mathbf{v}_c) \mathbf{u}_j}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)} \\ &= \mathbf{u}_o - \sum_{j \in \mathcal{V}} \left(\frac{\exp(\mathbf{u}_j^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)} \right) \mathbf{u}_j \\ &= \mathbf{u}_o - \sum_{j \in \mathcal{V}} P(w_j \mid w_c) \mathbf{u}_j. \end{aligned} \tag{15.1.8}$$

Continuous Bag Of Words (CBOW) REVERSE COND PROB

- The center word is generated based on the context words



$P(\text{center} \mid \text{context\_words})$

$\mathbb{P}(\text{"loves"} \mid \text{"the"}, \text{"man"}, \text{"his"}, \text{"son"})$

- no naive product-prob / indep
- instead evaluate score based on all context

Likelihood Function

- Compute the probability

$$\mathbb{P}(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{\exp\left(\frac{1}{2m}\mathbf{u}_c^\top(\mathbf{v}_{o_1} + \ldots + \mathbf{v}_{o_{2m}})\right)}{\sum_{i \in \mathcal{V}} \exp\left(\frac{1}{2m}\mathbf{u}_i^\top(\mathbf{v}_{o_1} + \ldots + \mathbf{v}_{o_{2m}})\right)}$$

dot product on whole window

normal.

SOFTMAX

- Likelihood

each word

$$\prod_{t=1}^{T} \mathbb{P}(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)})$$

min

Training

Training continuous bag of words models is almost the same as training skip-gram models. The maximum likelihood estimation of the continuous bag of words model is equivalent to minimizing the following loss function:

$$-\sum_{t=1}^{T} \log P(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}). \tag{15.1.13}$$

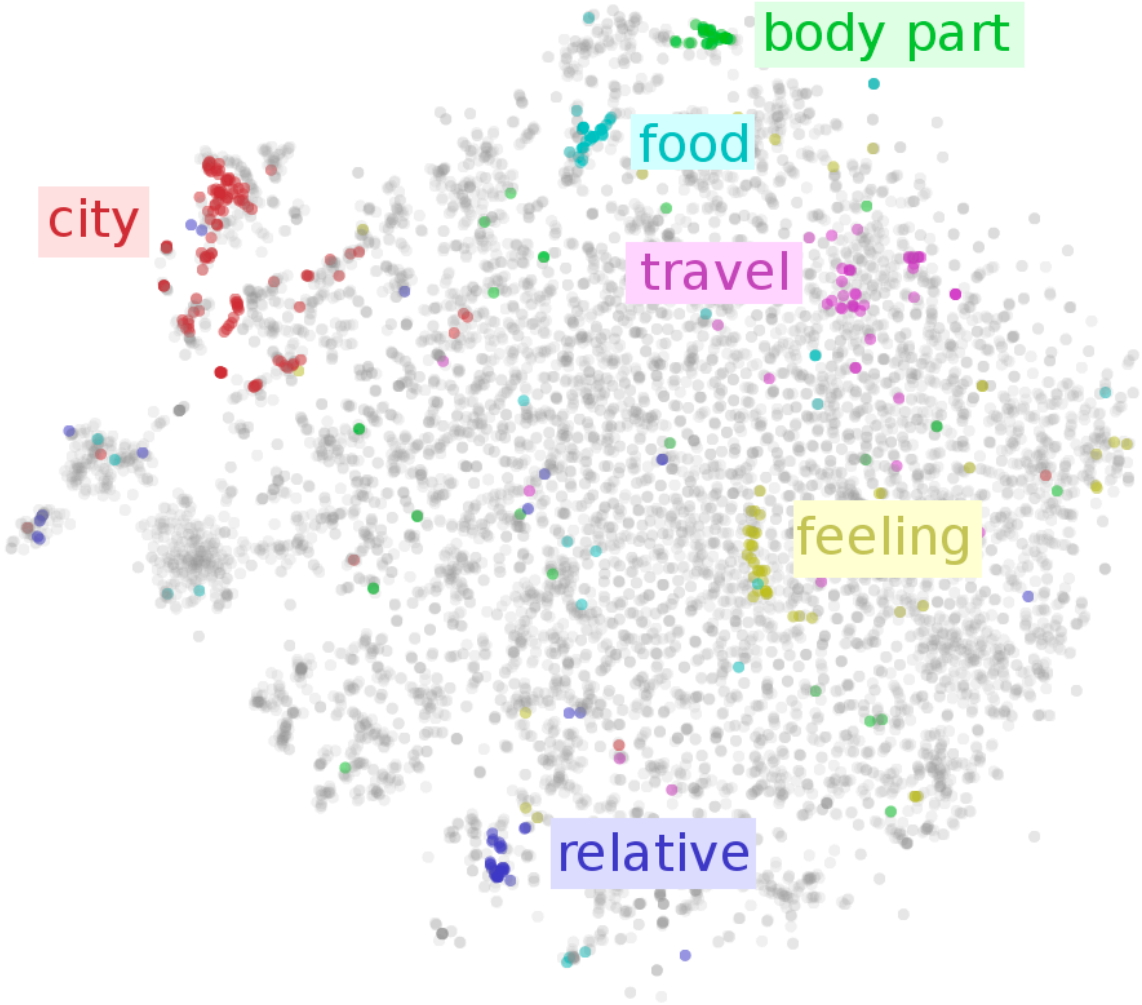
Notice that

$$\log\ P(w_c \mid \mathcal{W}_o) = \mathbf{u}_c^\top \bar{\mathbf{v}}_o - \log \left(\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^\top \bar{\mathbf{v}}_o\right) \right). \tag{15.1.14}$$

Through differentiation, we can obtain its gradient with respect to any context word vector $\mathbf{v}_{o_i} (i = 1, \ldots, 2m)$ as

$$\frac{\partial \log\ P(w_c \mid \mathcal{W}_o)}{\partial \mathbf{v}_{o_i}} = \frac{1}{2m} \left(\mathbf{u}_c - \sum_{j \in \mathcal{V}} \frac{\exp(\mathbf{u}_j^\top \bar{\mathbf{v}}_o) \mathbf{u}_j}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \bar{\mathbf{v}}_o)} \right) = \frac{1}{2m} \left(\mathbf{u}_c - \sum_{j \in \mathcal{V}} P(w_j \mid \mathcal{W}_o) \mathbf{u}_j \right).$$

More Embedding Models



FastText

- English words usually have internal structures and formation methods
  - dog, dogs, dogcatcher
- Each center word is represented as a set of subwords
  - “where” -> “<where>” -> *n*-gram
  - n=3: “<wh", "whe", "her", "ere", “re>”
- Useful for long but infrequent words
  - e.g. pneumonoultramicroscopicsilicovolcanoconiosis

FastText

- For word *w*, $\mathcal{G}_w$ is the union of subwords with length from 3 to 6
- The center vector is then

$$\mathbf{u}_w = \sum_{g \in \mathcal{G}_w} \mathbf{u}_g$$

sum (grams)

- The rest model is same as skip-gram

Word Embedding with Global Vectors (GloVe)

general word2vec pretrained (large corpus)

- Denote by $$q_{ij} = \frac{\exp(\mathbf{u}_j^\top \mathbf{v}_i)}{\sum_{k \in \mathcal{V}} \exp(\mathbf{u}_k^\top \mathbf{v}_i)}$$ (SOFTMAX)

- Rewrite the negative log-likelihood function of skip-gram

$$-\log \left[\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)}) \right]$$

tune for your specific corpus.

- as $$-\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log q_{ij}$$ with proper counts x_{ij}

MULTI-COUNT: count (i, j) in context window

Glove download various sizes

not approx, same as original.

- Further rewrite

$$-\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} x_{ij}\log q_{ij} = -\sum_{i\in\mathcal{V}} x_i \sum_{j\in\mathcal{V}} p_{ij}\log q_{ij}$$

Cross entropy

with $x_i = \sum_j x_{ij}$, $p_{ij} = x_i/x_{ij}$

x_i / ratio - normalized

Glove

- Replace the cross entropy with a log square loss

$$\sum_{j \in \mathcal{V}} p_{ij} \log q_{ij} \rightarrow \sum_{j \in \mathcal{V}} (\log p_{ij} - \log q'_{ij})^2$$

  with an easy to compute $q'_{ij} = \exp(\mathbf{u}_j^\top \mathbf{v}_i)$

- Add bias term for center and context words
- Replace the weights x_i with a monotone increasing function in [0,1]

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} h(x_{ij}) \left(\mathbf{u}_j^\top \mathbf{v}_i + b_i + c_j - \log x_{ij} \right)^2$$