[PV 211] 06 Word and Document Embeddings

Santosh Kesiraju

March 27, 2024

About me

- Researcher in Speech@FIT, Brno University of Technology.
- Working at the intersection of speech and NLP.
  - Speech recognition, translation.
  - Spoken dialogue systems.
  - Multilingual models.
  - Interpretability of models.

Today's topics

1. Introduction to embeddings
2. Popular methods for learning word embeddings
   - continuous bag-of-words
   - skip-gram
3. Methods for learning document and word embeddings **jointly**
   - Paragraph vector and variants.
4. Objective function and gradient.
5. Interpretation of the gradient.

Where do we begin?

- Let's say you are given N number of text documents.
- **Tokenize** every document and build a set of unique words (vocabulary).
  - Vocabulary $\mathcal{V}$, where $V = |\mathcal{V}|$
  - Index the words from 1 to V.
- Example:

word	index	word	index
a :	1	elephant:	50
antelope:	2	giraffe:	61
atmosphere:	3	⋮	⋮
⋮	⋮	tv:	7289
radio:	5002		
⋮	⋮		
xerox:	12348		
⋮	⋮		

Build co-occurrence matrix

- Words co-occurring (follow) with other words.

Word index $\downarrow$	**Word index** $\rightarrow$						
	w_1	w_2	...	w_i	w_{i+1}	...	w_V
w_1	14	23	0 ...	36	0	0 ...	0
w_2	1	0	14 ...	0	3	2 ...	0
$\vdots$	...	30	0	...	0	0	1
w_i	12	...	0	0	...	30	0
$\vdots$	12	...	0	3	...	0	0
w_V	0	...	92	0	...	6	0

Build contextual co-occurrence matrix

- Words are present in a given context window.

Context index ↓	**Word index** →						
	w_1	w_2	$\dots$	w_i	w_{i+1}	$\dots$	w_V
$\mathcal{C}_1$	1	1	0 …	1	0	0 …	0
$\mathcal{C}_2$	1	0	1 …	0	1	0 …	0
$\vdots$	…	1	0	…	0	0	1
$\mathcal{C}_i$	1	…	0	0	…	1	0
$\vdots$	1	…	0	1	…	0	0
$\mathcal{C}_M$	0	…	1	0	…	1	0

Build n-gram co-occurrence matrix

- Number of times a word follows given a $n-1$ gram history.

History index $\downarrow$	**Word index** $\rightarrow$						
	w_1	w_2	$\ldots$	w_i	w_{i+1}	$\ldots$	w_V
$\mathcal{H}_1$	3	1	0 ...	15	0	0 ...	0
$\mathcal{H}_2$	5	0	21 ...	0	6	0 ...	0
$\vdots$	...	6	0	...	0	0	9
$\mathcal{H}_i$	12	...	0	0	...	12	0
$\vdots$	1	...	0	13	...	0	0
$\mathcal{H}_M$	20	...	1	0	...	11	0

Build Bag-of-words

- Represent every document n with word counts, **ignoring** the word order.

Document	**Word index** $\rightarrow$						
index $\downarrow$	w_1	w_2	$\dots$	w_i	w_{i+1}	$\dots$	w_V
1	51	4	0 …	6	0	0 …	0
2	18	0	13 …	0	3	2 …	0
⋮	…	0	0	…	0	0	1
n	7	…	0	0	…	0	0
⋮	1	…	0	3	…	0	0
N	0	…	29	0	…	6	0

Problems with the co-occurrence matrices?

- They are sparse and huge.
- Hard to find relations between multiple words.
- Not optimal to use them a feature vectors in a machine learning model.

Word embeddings

- A low-dimensional continuous vector, that captures **word** semantics.
- Continuous vectors allows us to use them in other machine learning models (for classification, retrieval, clustering).

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

Mikolov et al., 2013 "Efficient Estimation of Word Representations in Vector Space"

Document embeddings

- A low-dimensional continuous vector, that captures semantic relations of words present a document.
- Additionally learns word embeddings.
- Applications:
  - Discover topics from a large collection of documents.
  - Cluster documents based on similarity (one cluster - one topic).
  - Ability to compute similarity between words and documents.
  - Retrieve relevant documents given a query phrase.
  - Train classifiers on top of document embeddings for topic classification.
  - Multilingual models - cross-lingual topic discovery or classification.

Cross-lingual topic discovery

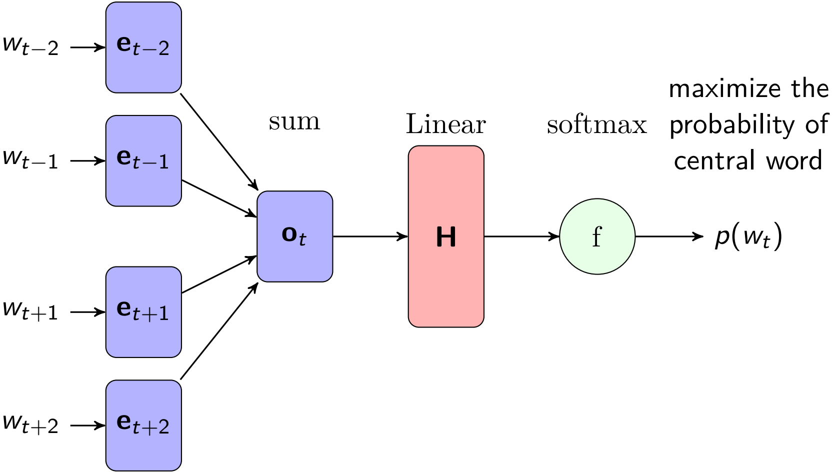
EN	resellers, dealer, stabilises, volatility
DE	uberschussen, marktpreise, preislich, anzukommen
FR	negociants, volatilite, nourrie, commercialisent
IT	responsabilizzati, concessionari, volatilita, compra
ES	subprimes, pingues, mora, abastecer
EN	inflation, inflationary, predictions, slowdown
DE	wirtschaftsindikatoren, haushaltsdefiziten, inflationsrate, wirtschaftsdaten
FR	inflationniste, inflation, inflationnistes, pronostics
IT	inflazione, inflazionistici, inflazionistiche, ciclica
ES	inflacion, inflacionistas, predicciones, coyuntural
EN	overvaluation, yen, lira, dollar
DE	dollars, yuan, wechselkurses, chinesischem
FR	surevaluation, croissent, dollar, degonflement
IT	sopravvalutazione, valutari, yen, dollaro
ES	dolar, fly, yen, redondeo

Table: An example of automatic discovery on Multilingual Reuters news.

Models for learning word embeddings

1. Continuous bag-of-words (CBoW)
2. Continuous skip-gram

CBoW



w_{t-2}
$\mathbf{e}_{t-2}$
w_{t-1}
$\mathbf{e}_{t-1}$
w_{t+1}
$\mathbf{e}_{t+1}$
w_{t+2}
$\mathbf{e}_{t+2}$
sum
$\mathbf{o}_t$
Linear
$\mathbf{H}$
softmax
f
maximize the probability of central word
$p(w_t)$

CBoW

- $\mathcal{C}_t = \{w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}\}$: context of word w_t.
- d: dimension of word embedding

$$\underbrace{\mathbf{o}_t}_{d\times 1} = \sum_{i\in\mathcal{C}_t} \underbrace{\mathbf{e}_i}_{d\times 1} \tag{1}$$

$$\underbrace{\mathbf{s}_t}_{V\times 1} = \underbrace{\mathbf{b}}_{V\times 1} + \underbrace{\mathbf{H}}_{V\times d} \underbrace{\mathbf{o}_t}_{d\times 1} \tag{2}$$

$$\underbrace{\boldsymbol{\theta}_t}_{V\times 1} = \mathrm{softmax}(\mathbf{s}_t) \tag{3}$$

$$\mathrm{softmax}(\mathbf{s}) := \frac{\exp\{s_i\}}{\sum_{j=1}^{V} s_j} \quad \forall\, i = 1 \dots V \tag{4}$$

$$p(w_t) := \theta_{w_t} := \theta_t \qquad \text{// abuse of notation for convenience}$$

CBoW

- Training data $\mathcal{D} = \{(w_t, \mathcal{C}_t) \ldots\}$
  set of words and corresponding contexts.
- $\mathbf{E} \in \mathbb{R}^{V \times d}$ word embeddings : *latent variables*.
- $\mathbf{H}, \mathbf{b} \in \mathbb{R}^{d \times V}$ linear projection matrix, and bias: *model parameters*.
- Training objective:

$$\arg\max_{\mathbf{H},\mathbf{b},\mathbf{E}} \sum_{w_t, \mathcal{C}_t \in \mathcal{D}} p(w_t \mid \mathcal{C}_t)$$

$$\arg\max_{\mathbf{H},\mathbf{b},\mathbf{E}} \sum_{w_t, \mathcal{C}_t \in \mathcal{D}} \log p(w_t \mid \mathcal{C}_t) \tag{5}$$

CBoW: training objective

$$
\begin{aligned}
\mathcal{L} &= \sum_{\mathcal{D}} \log p(w_t \mid \mathcal{C}_t) \\
&= \sum_{\mathcal{D}} \log \theta_t \\
&= \sum_{\mathcal{D}} \log \left(\frac{\exp\{s_t\}}{\sum_{j=1}^{V} \exp\{s_j\}} \right) \\
&= \sum_{\mathcal{D}} s_t - \log \left(\sum_{j=1}^{V} \exp\{s_j\} \right) \\
&= \sum_{\mathcal{D}} (b_t + \mathbf{h}_t^{\top} \mathbf{o}_t) - \log \left(\sum_{j=1}^{V} \exp\Big\{ (b_j + \mathbf{h}_j^{\top} \mathbf{o}_t) \Big\} \right) \\
&= \sum_{\mathcal{D}} (b_t + \mathbf{h}_t^{\top} \sum_{i \in \mathcal{C}_t} \mathbf{e}_i)_t - \log \left(\sum_{j=1}^{V} \exp \left\{ (b_j + \mathbf{h}_j^{\top} \sum_{i \in \mathcal{C}_t} \mathbf{e}_i) \right\} \right) \quad (6)
\end{aligned}
$$

CBoW: derivative of the objective

$$\mathcal{L} = \sum_{\mathcal{D}} (b_t + \mathbf{h}_t^\top \sum_{i \in \mathcal{C}_t} \mathbf{e}_i)_t - \log\Big(\sum_{j=1}^{V} \exp\Big\{(b_j + \mathbf{h}_j^\top \sum_{i \in \mathcal{C}_t} \mathbf{e}_i)\Big\}\Big)$$

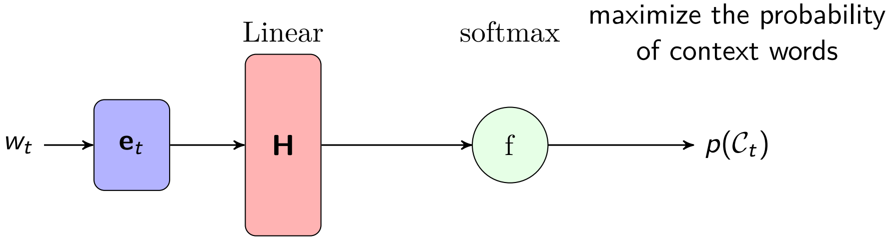
taking the derivative with respect to word embedding $\mathbf{e}_i$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{e}_i} = \sum_{\mathcal{D}} \frac{\partial\big(b_t + \mathbf{h}_t^\top \sum_{i \in \mathcal{C}_t} \mathbf{e}_i\big)}{\partial \mathbf{e}_i} - \frac{\partial \log\Big(\sum_{j=1}^{V} \exp\Big\{\big(b_j + \mathbf{h}_j^\top \sum_{i \in \mathcal{C}_t} \mathbf{e}_i\big)\Big\}\Big)}{\partial \mathbf{e}_i}$$

Remember the pattern in this expression.

1. derivative of a linear term.
2. derivative of a log-sum-exp.

Skip-gram



maximize the probability of context words
Linear
softmax
w_t
$\mathbf{e}_t$
$\mathbf{H}$
f
$p(\mathcal{C}_t)$

Skip-gram: training objective

- Training data $\mathcal{D} = \{(w_t, \mathcal{C}_t) \ldots\}$
  set of words and corresponding contexts.
- $\mathbf{E} \in \mathbb{R}^{V \times d}$ word embeddings.
- $\mathbf{H}, \mathbf{b} \in \mathbb{R}^{d \times V}$ linear projection matrix, and bias.
- Training objective:

$$
\begin{aligned}
\operatorname*{arg\,max}_{\mathbf{H},\mathbf{b},\mathbf{E}} & \sum_{\mathcal{D}} p(\mathcal{C}_t \mid w_t) \\
&= \sum_{\mathcal{D}} p(w_{t-2} \mid w_t) p(w_{t-1} \mid w_t) p(w_{t+1} \mid w_t) p(w_{t+2} \mid w_t) \\
&= \sum_{\mathcal{D}} \prod_{w_k \in \mathcal{C}_t} p(w_k \mid w_t) \\
&= \sum_{\mathcal{D}} \log \Big(\prod_{w_k \in \mathcal{C}_t} p(w_k \mid w_t) \Big) \\
&= \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} \log p(w_k \mid w_t)
\end{aligned}
$$

Skip-gram: training objective

$$
\begin{aligned}
\mathcal{L} &= \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} \log p(w_k \mid w_t) \\
&= \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} \log(\theta_k) \qquad \text{simplified notation: } \theta_{w_k} \rightarrow \theta_k \\
&= \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} \log \frac{\exp\{s_k\}}{\sum_{j=1}^{V} \exp\{s_j\}} \\
&= \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} s_k - \log\left(\sum_{j=1}^{V} \exp\{s_j\}\right) \\
&= \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} (b_k + \mathbf{h}_k^{\top} \mathbf{e}_t) - \log\left(\sum_{j} \exp\left\{b_j + \mathbf{h}_j^{\top} \mathbf{e}_t\right\}\right) \qquad (7)
\end{aligned}
$$

$$\mathcal{L} = \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} (b_k + \mathbf{h}_k^\top \mathbf{e}_t) - \log \left(\sum_j \exp\Big\{ b_j + \mathbf{h}_j^\top \mathbf{e}_t \Big\} \right)$$

taking the derivative with respect to word embedding $\mathbf{e}_i$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{e}_i} = \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} \frac{\partial (b_k + \mathbf{h}_k^\top \mathbf{e}_t)}{\partial \mathbf{e}_i} - \frac{\partial \log \left(\sum_{j=1}^{V} \exp\Big\{ (b_j + \mathbf{h}_j^\top \mathbf{e}_t) \Big\} \right)}{\partial \mathbf{e}_i}$$

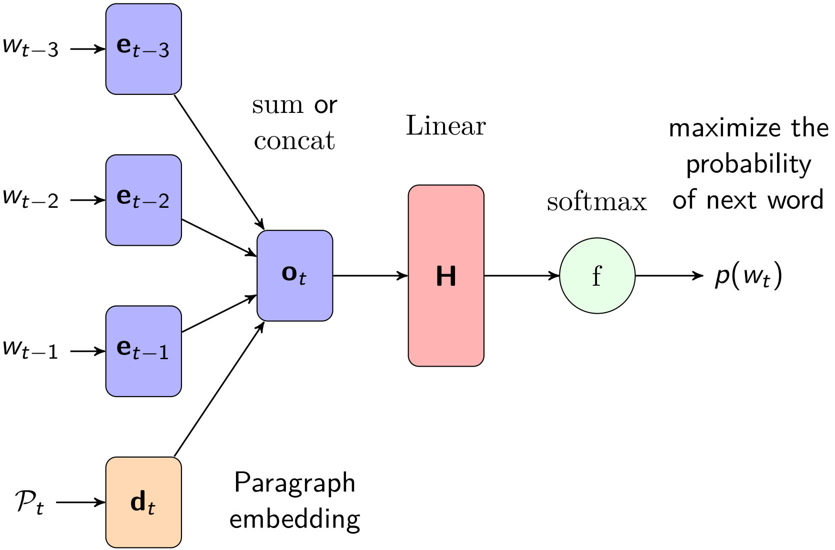
Similar pattern as earlier.

1. derivative of a linear term.
2. derivative of a log-sum-exp.

Paragraph vector (PV)

- Learn word and document embeddings jointly
  1. PV - distributed memory (PV-DM)
  2. PV - distributed bag-of-words (PV-DBOW)

PV-DM


w_{t-3}
$\mathbf{e}_{t-3}$
w_{t-2}
$\mathbf{e}_{t-2}$
w_{t-1}
$\mathbf{e}_{t-1}$
$\mathcal{P}_t$
$\mathbf{d}_t$
sum or concat
$\mathbf{o}_t$
Paragraph embedding
Linear
$\mathbf{H}$
softmax
f
maximize the probability of next word
$p(w_t)$

PV-DM

- $\mathcal{C}_t$: a set with n number of context (history) words for w_t

option 1: sum

$$\underbrace{\mathbf{o}_t}_{d \times 1} = \mathbf{d}_t + \sum_{w_k \in \mathcal{C}_t} \mathbf{e}_k \qquad // \text{ notation: } \mathbf{e}_{w_k} \rightarrow \mathbf{e}_k$$

$$\mathbf{s}_t = \mathbf{b} + \underbrace{\mathbf{H}}_{V \times d} \mathbf{o}_t$$

option 2: concat

$$\underbrace{\mathbf{o}_t}_{(dn+1) \times 1} = [\mathbf{d}_t; \mathbf{e}_k; \ldots] \qquad \forall w_k \in \mathcal{C}_t$$

$$\mathbf{s}_t = \mathbf{b} + \underbrace{\mathbf{H}}_{V \times (dn+1)} \mathbf{o}_t$$

generic notation for both options:

$$\mathbf{o}_t = f(\mathbf{d}_t, \mathbf{E}, \mathcal{C}_t)$$

PV-DM: training objective

- Training data $\mathcal{D} = \{(w_t, \mathcal{C}_t, \mathcal{P}_t) \ldots\}$
- $\mathbf{E} \in \mathbb{R}^{V \times d}$ word embeddings.
- $\mathbf{H}, \mathbf{b} \in \mathbb{R}^{d \times V}$ linear projection matrix, and bias.
- $\mathbf{D} \in \mathbb{R}^{d \times M}$ M number of paragraph (document) embeddings.

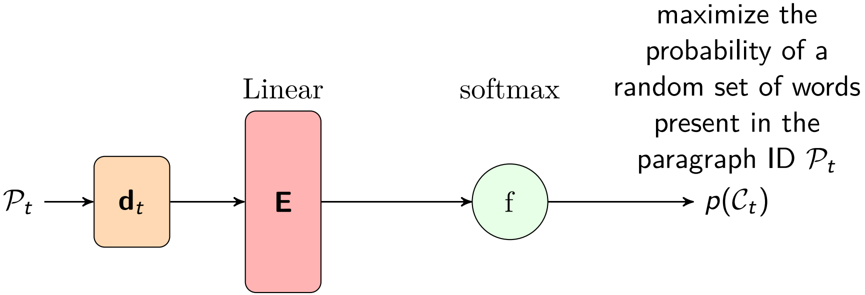
$$
\begin{aligned}
\underset{\mathbf{H},\mathbf{b},\mathbf{E},\mathbf{D}}{\arg\max} \sum_{\mathcal{D}} p(w_t \mid \mathcal{C}_t, \mathcal{P}_t) \\
= \sum_{\mathcal{D}} \log p(w_t \mid \mathcal{C}_t, \mathcal{P}_t) \qquad (8)
\end{aligned}
$$

PV-DM: training objective

$$
\begin{aligned}
\mathcal{L} &= \sum_{\mathcal{D}} \log p(w_t \mid \mathcal{C}_t, \mathcal{P}_t) \\
&= \sum_{\mathcal{D}} \log(\theta_t) \qquad \text{simplified notation: } \theta_{w_t} \rightarrow \theta_t \\
&= \sum_{\mathcal{D}} \log \frac{\exp\{s_t\}}{\sum_{j=1}^{V} \exp\{s_j\}} \\
&= \sum_{\mathcal{D}} s_t - \log\left(\sum_{j=1}^{V} \exp\{s_j\}\right) \\
&= \sum_{\mathcal{D}} (b_t + \mathbf{h}_t^{\intercal} \mathbf{o}_t) - \log\left(\sum_{j=1}^{V} (b_j + \mathbf{h}_j^{\intercal} \mathbf{o}_t)\right)
\end{aligned}
$$

We ended up with a similar expression as before.

PV-DBoW



$\mathcal{P}_t$
$\mathbf{d}_t$
Linear
$\mathbf{E}$
softmax
f
maximize the probability of a random set of words present in the paragraph ID $\mathcal{P}_t$
$p(\mathcal{C}_t)$

PV-DBoW

- $\mathcal{C}_t$: a random set of words in paragraph $\mathcal{P}_t$.
- Training data $\mathcal{D} = \{(\mathcal{C}_t, \mathcal{P}_t) \ldots\}$
- $\mathbf{E} \in \mathbb{R}^{V \times d}, \mathbf{b} \in \mathbb{R}^{V \times 1}$ linear projection matrix, and bias.
- $\mathbf{D} \in \mathbb{R}^{d \times M}$ M number of paragraph (document) embeddings.

$$\begin{aligned} \arg\max_{\mathbf{E},\mathbf{b},\mathbf{D}} \sum_{\mathcal{D}} p(\mathcal{C}_t \mid \mathcal{P}_t) \\ = \sum_{\mathcal{D}} \prod_{w_k \in \mathcal{C}_t} p(w_k \mid \mathcal{P}_t) \\ = \sum_{\mathcal{D}} \sum_{w_k \in \mathcal{C}_t} \log p(w_k \mid \mathcal{C}_t, \mathcal{P}_t) \end{aligned} \tag{9}$$

Document model

- Generalize PV-DBoW to all the words present in the document.
- Let x_{ni} denote the number of occurrences of word w_i in document n.

$$
\begin{aligned}
\mathcal{L} &= \sum_n \prod_{i=1}^{V} p(w_{ni})^{x_{ni}} \\
&= \sum_n \sum_{i=1}^{V} \log p(w_{ni})^{x_{ni}} \\
&= \sum_n \sum_{i=1}^{V} x_{ni} \log \theta_{ni} \\
&= \sum_n \sum_{i=1}^{V} x_{ni} \log \frac{\exp\left\{ b_i + \mathbf{e}_i^{\mathsf{T}} \mathbf{d}_n \right\}}{\sum_j \exp\left\{ b_j + \mathbf{h}_j^{\mathsf{T}} \mathbf{d}_n \right\}}
\end{aligned}
$$

Document model: training objective

$$\mathcal{L} = \sum_n \sum_{i=1}^{V} x_{ni} \log \frac{\exp\Big\{ b_i + \mathbf{e}_i^\intercal \mathbf{d}_n \Big\}}{\sum_j \exp\Big\{ b_j + \mathbf{e}_j^\intercal \mathbf{d}_n \Big\}}$$

$$= \sum_n \sum_{i=1}^{V} x_{ni} \Big[(b_i + \mathbf{e}_i^\intercal \mathbf{d}_n) - \log \big(\sum_j \exp\Big\{ b_j + \mathbf{e}_j^\intercal \mathbf{d}_n \Big\} \big) \Big]$$

taking the derivative w.r.t. word embedding $\mathbf{e}_k$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{e}_k} = \nabla_{e_k} \mathcal{L}$$

derivative

$$\nabla_{\mathbf{e}_k}\mathcal{L} = \frac{\partial \sum_{n=1}^{N}\sum_{i=1}^{V} x_{ni}\Big[(b_i + \mathbf{e}_i^T\mathbf{d}_n) - \log\big(\sum_j \exp\{b_j + \mathbf{e}_j^T\mathbf{d}_n\}\big)\Big]}{\partial \mathbf{e}_k}$$

$$= \sum_n \sum_i x_{ni}\left[\frac{\partial(b_i + \mathbf{e}_i^T\mathbf{d}_n)}{\partial \mathbf{e}_k}\right] - \sum_i x_{ni}\left[\frac{\partial \log\big(\sum_j \exp\{b_j + \mathbf{e}_j^T\mathbf{d}_n\}\big)}{\partial \mathbf{e}_k}\right]$$

$$= \sum_n \left[x_{nk}(\mathbf{d}_n^T)\right] - \sum_i x_{ni}\left[\frac{1}{\sum_j \exp\{b_j + \mathbf{e}_j^T\mathbf{d}_n\}}\Big(\frac{\partial \sum_j \exp\{b_j + \mathbf{e}_j^T\mathbf{d}_n\}}{\partial \mathbf{e}_k}\Big)\right]$$

$$= \sum_n \left[x_{nk}\mathbf{d}_n^T\right] - \sum_i x_{ni}\left[\frac{1}{\sum_j \exp\{b_j + \mathbf{e}_j^T\mathbf{d}_n\}}(0 + \ldots + \exp\{b_k + \mathbf{e}_k^T\mathbf{d}_n\}\mathbf{d}_n^T)\right]$$

$$= \sum_n \left[x_{nk}\mathbf{d}_n^T\right] - \sum_i x_{ni}\left[\frac{\exp\{b_i + \mathbf{e}_k^T\mathbf{d}_n^T\}}{\sum_j \exp\{b_j + \mathbf{e}_j^T\mathbf{d}_n\}}\mathbf{d}_n\right]$$

$$= \sum_n \left[x_{nk}\mathbf{d}_n^T\right] - \sum_i x_{in}\left[\theta_{nk}\mathbf{d}_n^T\right]$$

$$= \sum_{n=1}^{N}\left[x_{nk} - (\sum_{i=1}^{V} x_{ni})\theta_{nk}\right]\mathbf{d}_n^T$$

derivative w.r.t word embeddings

$$\nabla_{\mathbf{e}_k}\mathcal{L} = \sum_{n=1}^{N}\Big[x_{nk} - (\sum_{i=1}^{V} x_{ni})\theta_{nk}\Big]\mathbf{d}_n^T \qquad (10)$$

- x_{nk}: number of times word k appeared in document n.
- θ_{nk}: the estimated probability of word k in document n.
- $\sum_i x_{ni}$: sum of all the word counts in document n.
- $(\sum_i x_{ni})\theta_{nk}$: relative count of word k in document n.
- $\Big[x_{nk} - (\sum_i x_{ni})\theta_{nk}\Big]$: the difference of absolute word count to the relative word count.
- $\Big[x_{nk} - (\sum_i x_{ni})\theta_{nk}\Big]\mathbf{d}_n$, and weight this along the direction of document embedding.
- The final gradient is the sum of all weighted document embeddings.

Why this gradient

- The $\mathrm{softmax}$ function appears nearly everywhere in current neural architectures.
- All forms of expressions involving $\mathrm{softmax}$ have the same interpretation.

- In the seminar, you will have hands on experience training the document model.
- Along with some application in classification, retrieval.