

Introduction to Deep Learning

Embeddings, Word2vec, fastText, GloVe

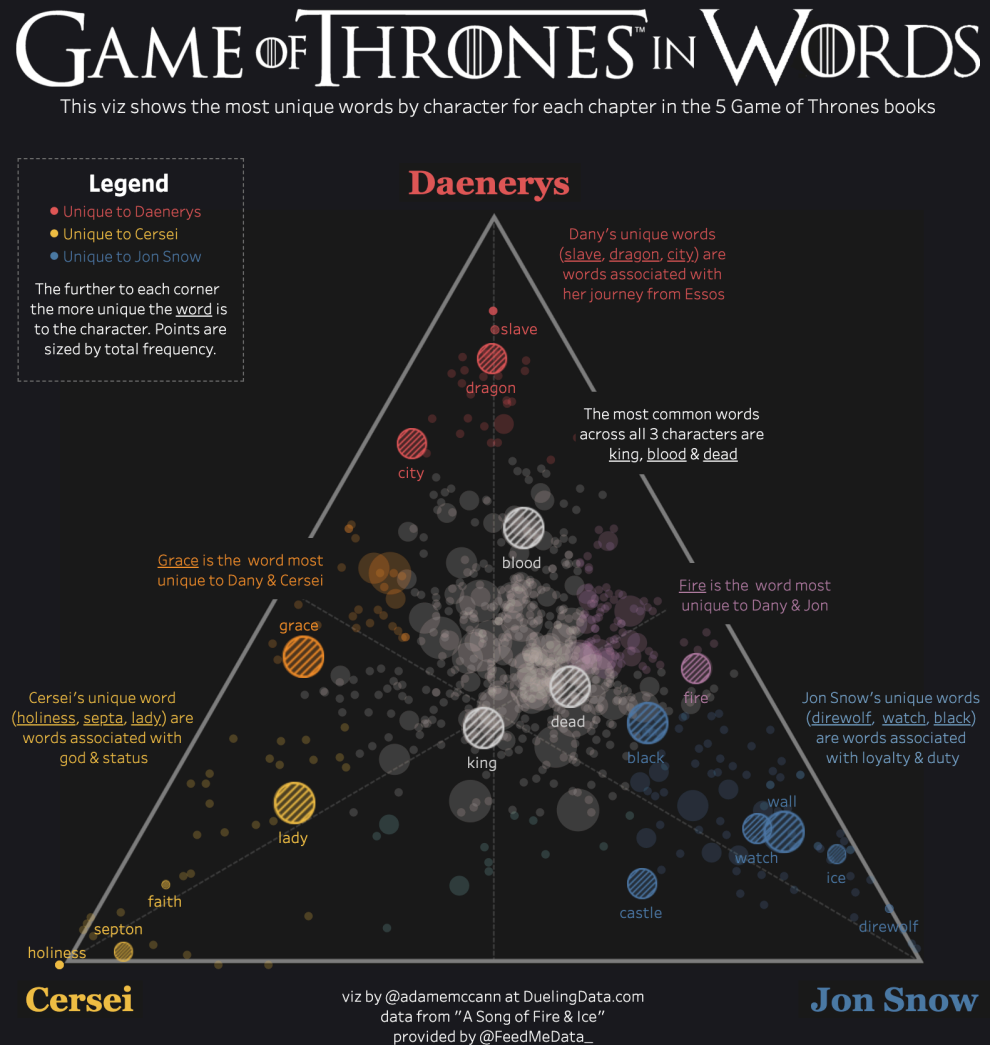
Haibin Lin and Leonard Lausen

gluon-nlp.mxnet.io



8:30-9:00	Continental Breakfast
9:00-9:45	Introduction and Setup
9:45-10:30	Neural Networks 101
10:30-10:45	Break
10:45-11:15	Machine Learning Basics
11:15-11:45	Context-free Representations for Language
11:45-12:15	Convolutional Neural Networks
12:15-13:15	Lunch Break
13:15-14:00	Recurrent Neural Networks
14:00-14:45	Attention Mechanism and Transformer
14:45-15:00	Coffee Break
15:00-16:15	Contextual Representations for Language
16:15-17:00	Language Generation




Word2Vec



Motivation

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

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




	\mathbf{x}	\mathbf{y}	\mathbf{z}
	1	0	0
	0	1	0
\vdots	\vdots	\vdots	\vdots
	0	0	1

Word2vec

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

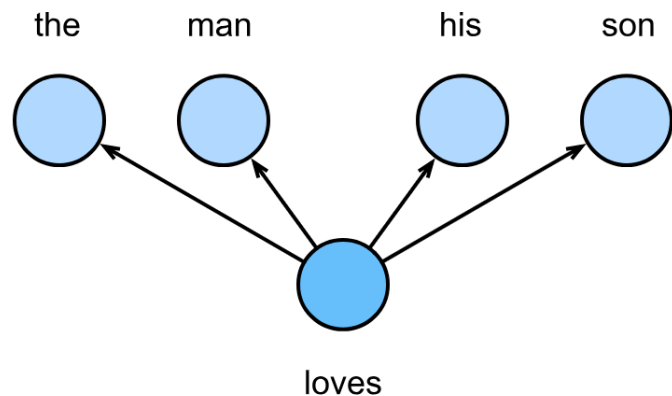
$$\langle \mathbf{x}, \mathbf{y} \rangle > \langle \mathbf{z}, \mathbf{y} \rangle$$

- Build a probability model
- Maximize the likelihood function to learn the model

	\mathbf{x}	\mathbf{y}	\mathbf{z}
	1	0	0
	0	1	0
	\vdots	\vdots	\vdots
	0	0	1

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



$$\begin{aligned} & \mathbb{P}(\text{"the", "man", "his", "son"} \mid \text{"loves"}) \\ &= \mathbb{P}(\text{"the"} \mid \text{"loves"}) \cdot \mathbb{P}(\text{"man"} \mid \text{"loves"}) \\ & \quad \cdot \mathbb{P}(\text{"his"} \mid \text{"loves"}) \cdot \mathbb{P}(\text{"son"} \mid \text{"loves"}) \end{aligned}$$

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:

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

Negative Sampling

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

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

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

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

Naive solution: infinity

Negative Sampling

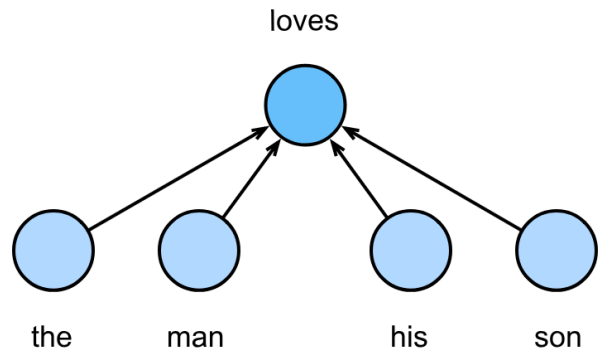
- Sample noise word w_n that doesn't appear in the window

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

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

Continuous Bag Of Words (CBOW)

- The center word is generated based on the context words



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

Likelihood Function

- Compute the probability

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

- Likelihood

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

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

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

- The rest model is same as skip-gram

Word Embedding with Global Vectors (GloVe)

- Denote by $q_{ij} = \frac{\exp(\mathbf{u}_j^\top \mathbf{v}_i)}{\sum_{k \in \mathcal{V}} \exp(\mathbf{u}_k^\top \mathbf{v}_i)}$
- Rewrite the negative log-likelihood function of skip-gram

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

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

Glove

- Further rewrite

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

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

Cross entropy

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 $[0,1]$

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