Lecture 07: Language Models (Part I)

Outline

- Introduction
- Count-based Vector Space Model
- Context-based Models: wrod2vec
  - Skip-Gram
  - Continuous Bag-of-Words
- A Hybrid Model: GloVe

What is a language model?

- A statistical language model is a probability distribution over sequences of words. Given such a sequence, say of length m, it assigns a probability $P(w_1 , \ldots , w_m)$ to the whole sequence.

- Language modeling is used in speech recognition, machine translation, part-of-speech tagging, parsing, Optical Character Recognition, handwriting recognition, information retrieval and other applications.

Featurization

- Human vocabulary comes in free text. In order to make a machine learning model understand and process the natural language, we need to transform the free-text words into numeric values, i.e., we need to featurize the words.
- One of the simplest transformation approaches is to do a **one-hot encoding** in which each distinct word stands for one dimension of the resulting vector and a binary value indicates whether the word presents (1) or not (0).

Featurization (cont'd)

- However, one-hot encoding is impractical computationally when dealing with the entire vocabulary, as the representation demands hundreds of thousands of dimensions.

N-Gram Model and Featurization

- Data sparsity is a major problem in building language models as most possible word sequences are not observed in training.

- One solution is to make the assumption that the probability of a word only depends on the previous *n* words. This is known as an ***n*-gram model** or ***unigram model*** when *n* = 1. The unigram model is also known as the bag of words model.

Word Embedding

- Word embedding represents words and phrases in vectors of (non-binary) numeric values with much lower and thus denser dimensions.
- An intuitive assumption for good word embedding is that they can approximate the similarity between words (i.e., “cat” and “kitten” are similar words, and thus they are expected to be close in the reduced vector space)
- They can also disclose hidden semantic relationships (i.e., the relationship between “cat” and “kitten” is an analogy to the one between “dog” and “puppy”).
- Word embedding can also be learned as by-product of learning some language models.

Approaches to Learn Word Embedding

All approaches rely on capturing contextual knowledge

- **Count-based**: This includes unsupervised models based on matrix factorization of a global word co-occurrence matrix.
- **Context-based**: This includes supervised models that given a local context can predict the target words, and in the meantime learn the efficient word embedding representation.

Outline

- Introduction
- Count-based Vector Space Model
- Context-based Models: wrod2vec
  - Skip-Gram
  - Continuous Bag-of-Words
- A Hybrid Model: GloVe

Count-based Vector Space Models

- These approaches rely on the word frequency and co-occurrence matrix with the assumption that words in the same contexts share similar or related semantic meanings.
- The models map count-based statistics like co-occurrences between neighboring words down to a small and dense word vectors.
- PCA and topic models are all good examples of this category.

Outline

- Introduction
- Count-based Vector Space Model
- Context-based Models: wrod2vec
  - Skip-Gram
  - Continuous Bag-of-Words
- A Hybrid Model: GloVe

Context-based Models: word2vec

- word2vec is a two-layer neural net (not a deep neural network) that processes text by “vectorizing” words.
- Its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus.
- The purpose and usefulness of word2vec is to group the vectors of similar words together in vector space.

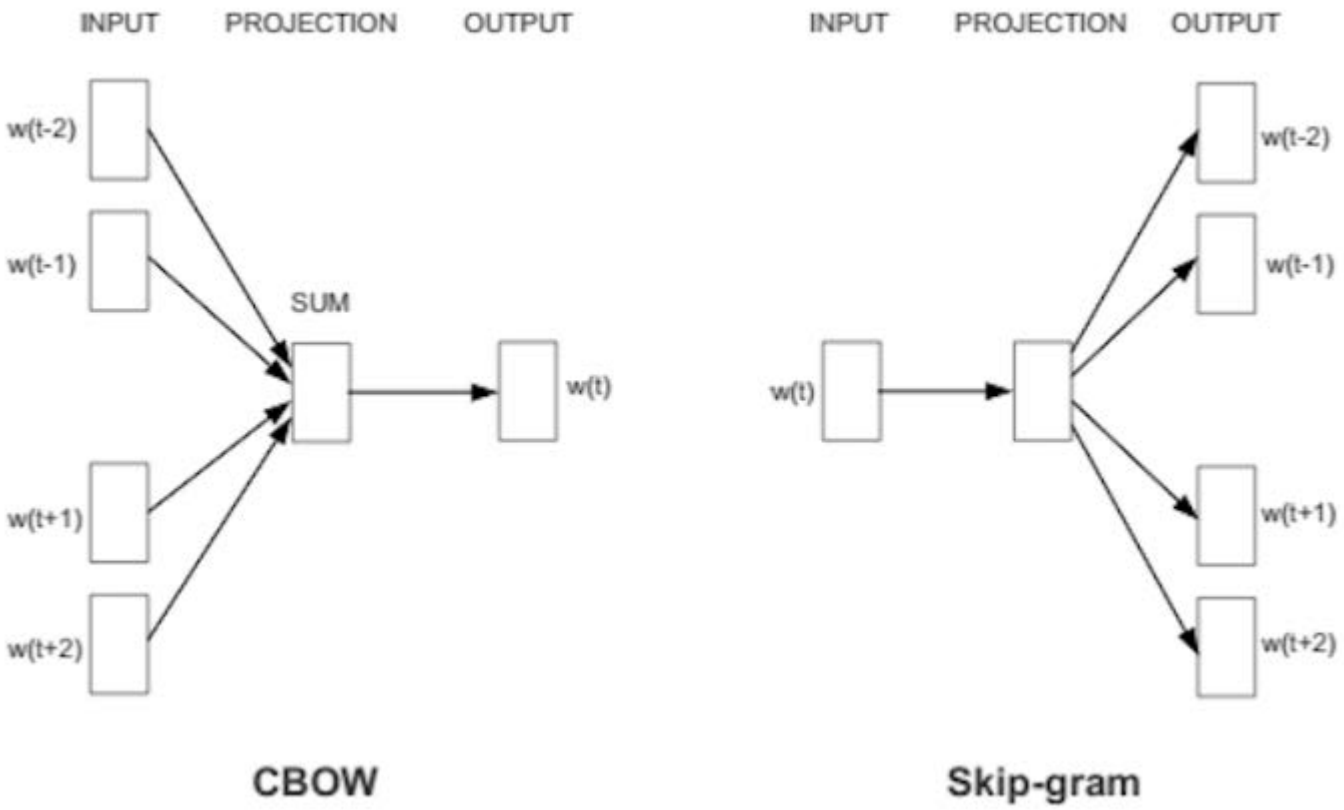
Context-based Models: word2vec (cont'd)

- Word2vec is similar to an autoencoder, encoding each word in a vector, but rather than training against the input words through reconstruction, word2vec trains words against other words that neighbor them in the input corpus.

Context-based Models: word2vec (cont'd)

- It does so in one of two ways, either using context to predict a target word (a method known as continuous bag of words, or CBOW), or using a word to predict a target context, which is called skip-gram.

Context-based Models: word2vec (cont'd)



Mikolov et al., 2013

Skip-Gram

- Suppose that you have a sliding window of a fixed size moving along a sentence: the word in the middle is the “target” and those on its left and right within the sliding window are the context words.

- The skip-gram model is trained to predict the probabilities of a word being a context word for the given target.

Mikolov et al., 2013

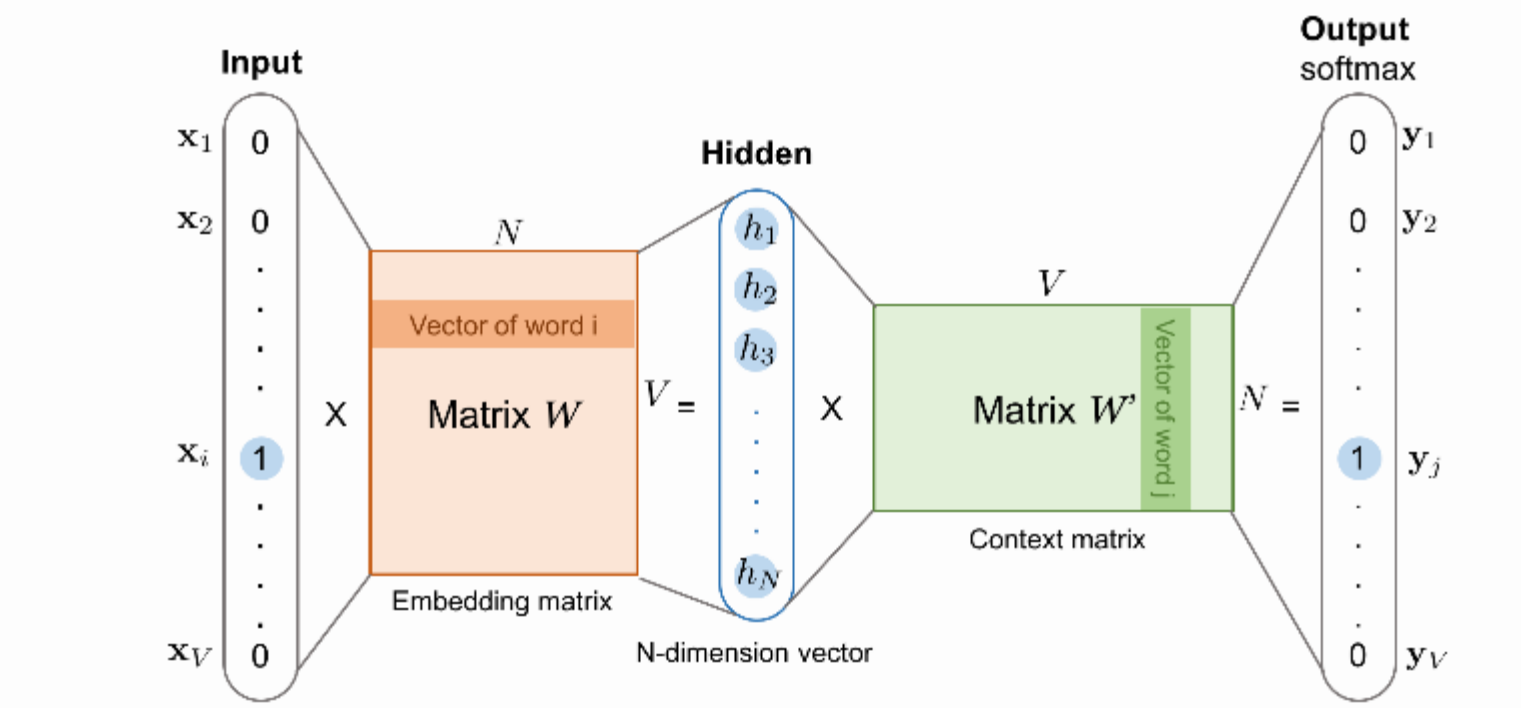
Skip-Gram (cont'd)

> "The man who passes the sentence should swing the sword." – Ned Stark

Sliding window (size = 5)	Target word	Context
[The man who]	the	man, who
[The man who passes]	man	the, who, passes
[The man who passes the]	who	the, man, passes, the
[man who passes the sentence]	passes	man, who, the, sentence
...	...	...
[sentence should swing the sword]	swing	sentence, should, the, sword
[should swing the sword]	the	should, swing, sword
[swing the sword]	sword	swing, the

Each context-target pair is treated as a new observation in the data. For example, the target word "swing" in the above case produces four training samples: ("swing", "sentence"), ("swing", "should"), ("swing", "the"), and ("swing", "sword").

Skip-Gram Model Architecture



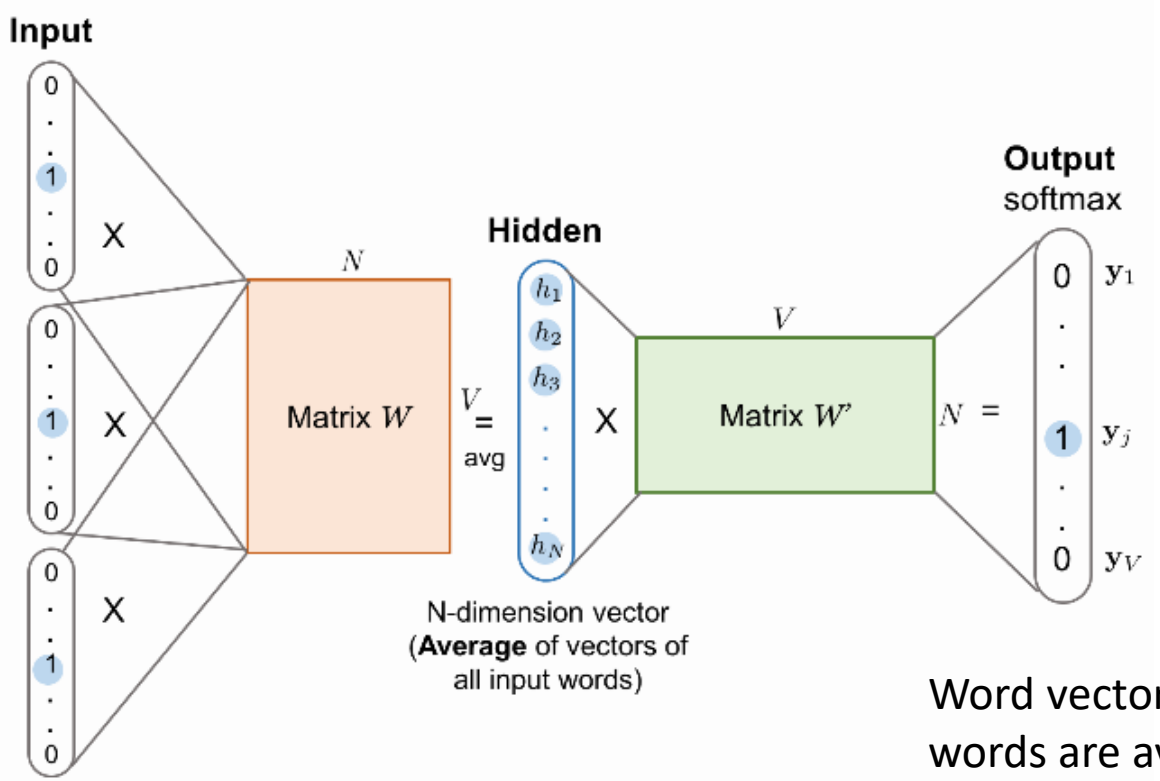
Both the input vector x and the output y are one-hot encoded word representations. The hidden layer is the word embedding of size N. Given the vocabulary size V, we learn word embedding vectors of size N. The model learns to predict one context word (output) using one target word (input) at a time.

Skip-Gram Algorithm

- Both input word w_i and the output word w_j are one-hot encoded into binary vectors $\mathbf{x}$ and $\mathbf{y}$ of size V.
- First, the multiplication of the binary vector $\mathbf{x}$ and the word embedding matrix W of size $V \times N$ gives us the embedding vector of the input word w_i: the i-th row of the matrix W.
- This newly discovered embedding vector of dimension N forms the hidden layer.
- The multiplication of the hidden layer and the word context matrix W' of size $N \times V$ produces the output one-hot encoded vector $\mathbf{y}$.
- The output context matrix W' encodes the meanings of words as context, different from the embedding matrix W. NOTE: Despite the name, W' is independent of W, not a transpose or inverse or whatsoever.

Continuous BOW (CBOW)

The Continuous Bag-of-Words (CBOW) is another similar model for learning word vectors. It predicts the target word (i.e. “swing”) from source context words (i.e., “sentence should the sword”).



Word vectors of multiple context words are averaged to get a fixed-length vector as in the hidden layer.

Loss Functions

- Both the skip-gram model and the CBOW model should be trained to minimize a well-designed loss/objective function.
- There are several loss functions we can incorporate to train these language models.
- Possible choices:
  - Full Softmax
  - Hierarchical Softmax
  - Cross Entropy
  - Noise Constructive Estimation (NCE)
  - Negative Sampling (NEG)

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^\top v_{w_I})}{\sum_{i=1}^{V} \exp({v'_{w_i}}^\top v_{w_I})}$$

Tips for Learning Word Embedding

- **Soft sliding window**. When pairing the words within the sliding window, we could assign less weight to more distant words. One heuristic is — given a maximum window size parameter defined, $s_{\max}$, the actual window size is randomly sampled between 1 and $s_{\max}$ for every training sample. Thus, each context word has the probability of 1/(its distance to the target word) being observed, while the adjacent words are always observed.

- **Subsampling frequent words**. Extremely frequent words might be too general to differentiate the context (i.e. think about stopwords). While on the other hand, rare words are more likely to carry distinct information. To balance the frequent and rare words, Mikolov et al. proposed to discard words w with probability $1 - \sqrt{t/f(w)}$ during sampling. Here $f(w)$ is the word frequency and t is an adjustable threshold.

- **Learning phrases first**. A phrase often stands as a conceptual unit, rather than a simple composition of individual words. For example, we cannot really tell "New York" is a city name even we know the meanings of "new" and "york". Learning such phrases first and treating them as word units before training the word embedding model improves the outcome quality.

Outline

- Introduction
- Count-based Vector Space Model
- Context-based Models: wrod2vec
  - Skip-Gram
  - Continuous Bag-of-Words
- A Hybrid Model: GloVe

Pennington et al. (2014)

Global Vectors (GloVe)

The Global Vector (GloVe) model proposed by aims to combine the count-based matrix factorization and the context-based skip-gram model together.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Only in the ratio of probabilities does noise from non-discriminative words like water and fashion cancel out, so that large values (much greater than 1) correlate well with properties specific to ice, and small values (much less than 1) correlate well with properties specific of steam. In this way, the ratio of probabilities encodes some crude form of meaning associated with the abstract concept of thermodynamic phase.

https://nlp.stanford.edu/projects/glove/

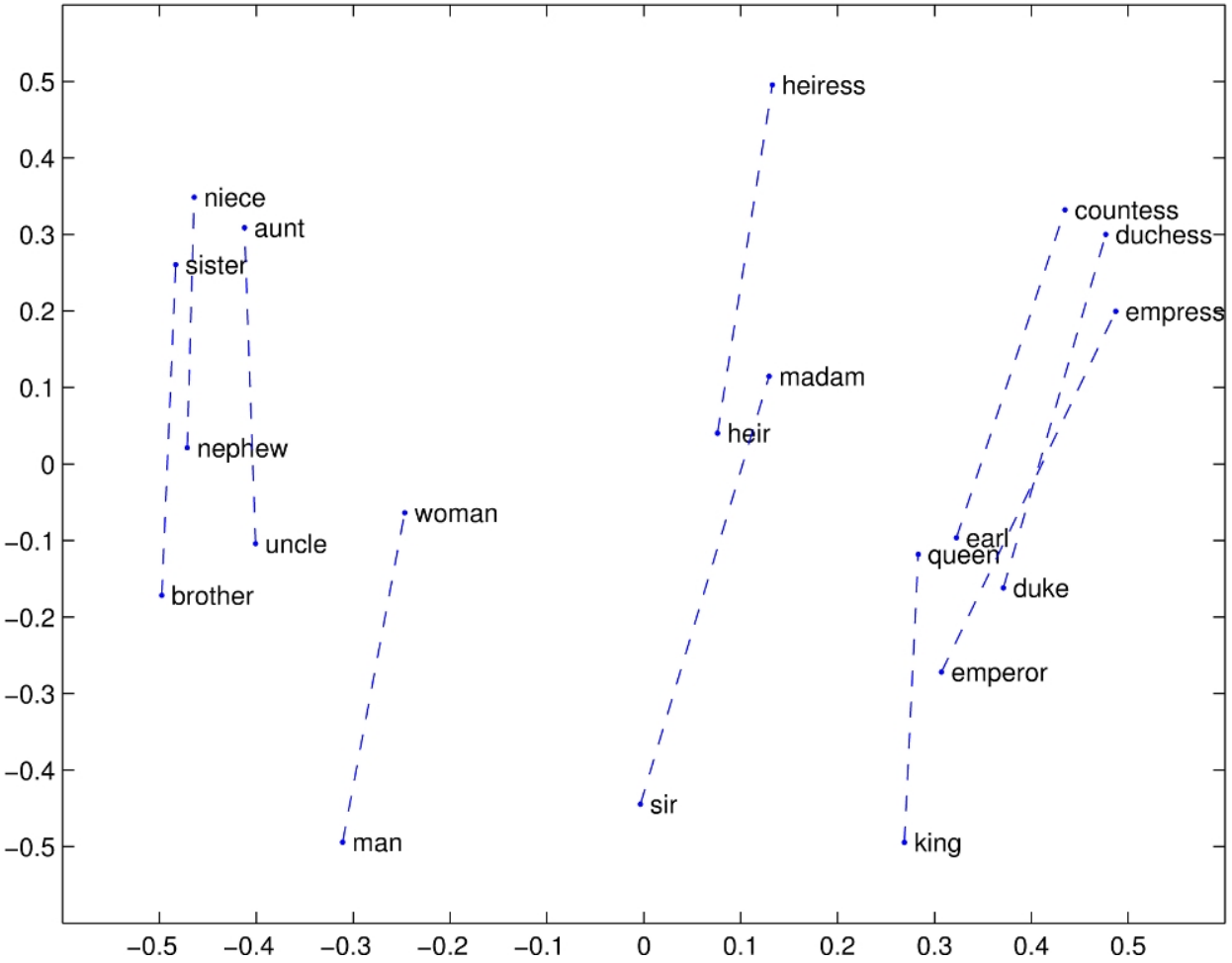
Global Vectors (GloVe) (cont'd)

- The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence.

- Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associates (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space.

Embedding Space

We might expect that the vector differences "man – woman", "king – queen", and "brother – sister" might all be roughly equal.



Embedding Space (cont'd)