Introduction to NLP

Gokul Swamy · ML@B

Basic Concepts 1

- Document: Ordered collection of words (i.e. reddit post)
- Distributional Hypothesis: words that occur in the same document probably have similar meanings
  - Very logical assumption if documents are not very long
- Term-document matrix: $X_{i,j}$ contains number of occurrences of word i in document j
- Bigram model: First-order Markov Property (next word only depends on word directly preceding it) - usually works well

Basic Concepts 2

- Embedding: representation of a word in some vector space
- Bag of words: Characterize document by number of word occurrences, ignoring order and other documents
- Zipf's Law: frequency of word is inversely proportional to position to frequency list
- tf-idf: Stands for term-frequency * inverse-document frequency
- TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
- IDF(t) = ln(Total number of documents / Number of documents with term t in it)

Named Entity Recognition 1

- In philosophy, a named entity is what is known as a “rigid designator” - something that in all conceivable concepts refers to the same object (e.g. “Matthew Trepte”)
- This includes words that are relatively rare and almost always are used in the same way (like tax terms)
- When we recognize an entity, we tag it with a label (e.g. “Person”)
- This is a supervised learning problem - need data of usage

Named Entity Recognition 2

- Input: (word, label) pairs (x_i, y_i)
- Bayesian Inference:

$$\begin{aligned} f(x) &= \arg\max_y p(y|x) \\ &= \arg\max_y \frac{p(y)p(x|y)}{p(x)} \\ &= \arg\max_y p(y)p(x|y) \end{aligned}$$

- Prior over labels corrupted by noisy channel (cond. dist.)
- Viterbi (decoding HMM) Extension of this to phrases in implemented in NLTK

Word Vectors

- Term-document matrixes can be very tall (many words) - can we perform some kind of intelligent dimensionality reduction?
- word2vec: tried and true method to reduce words to 300 dimensional embedding (where an embedding is a representation of a word in some vector space)
- Combination of skip-gram model and continuous bag of words

Continuous Bag of Words 1

- Assume we had 2 matrixes V and U
- ith column of V is input representation of ith word
- jth row of U is output representation of jth word
- Assuming we had U and V we would
  - One-hot encode words [-m, m] around center word (not including center word)
  - Left-multiply vectors by V and take average
  - Left-multiply average by U and take softmax

Continuous Bag of Words 2

- This gives us a distribution over possible output representations
- Minimize the KL-Divergence between distributions (label is one-hot so we only have one term)

$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log(\hat{y}_j) \; = -y_i \log(\hat{y}_i)$$

Continuous Bag of Words 3

- Thus, our optimization objective is:

$$\begin{aligned}\text{minimize } J &= -\log P(w_c|w_{c-m},\ldots,w_{c-1},w_{c+1},\ldots,w_{c+m}) \\ &= -\log P(u_c|\hat{v}) \\ &= -\log \frac{\exp(u_c^T\hat{v})}{\sum_{j=1}^{|V|}\exp(u_j^T\hat{v})} \\ &= -u_c^T\hat{v} + \log\sum_{j=1}^{|V|}\exp(u_j^T\hat{v})\end{aligned}$$

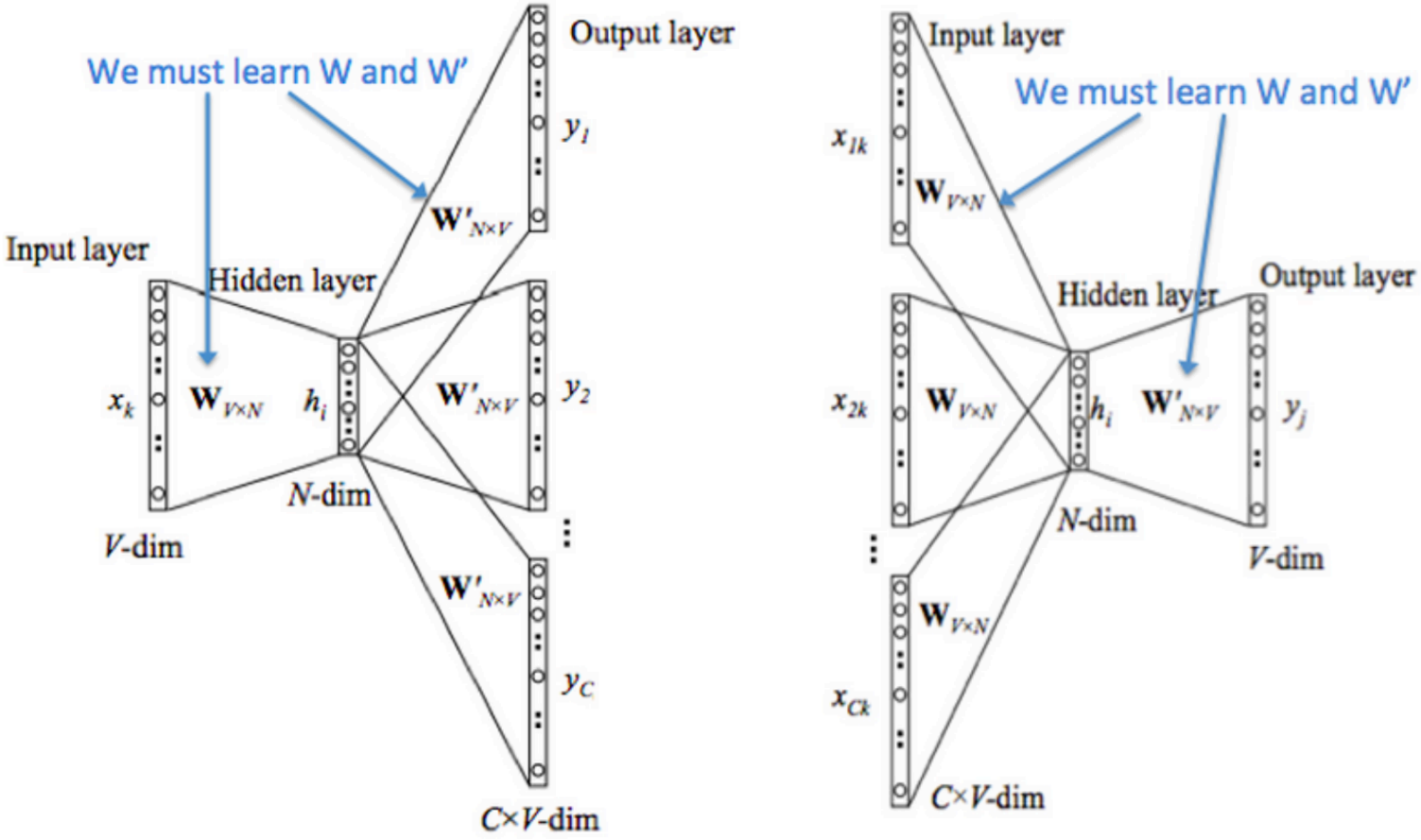
- Use gradient descent to optimize U and V

Skip-Gram Model

- Effectively the inverse of continuous bag of words
- Now input is single center word and output is set of surrounding words
- We assume Naive Bayes (that all words around center word are independent given center word)

word2vec Architecture

Skip-gram + CBOW = Autoencoder



Other Resources

- https://arxiv.org/pdf/1301.3781.pdf
- http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
- http://www.cs.columbia.edu/~mcollins/cs4705-fall2017/
  - For named entity recognition
- https://cs224d.stanford.edu/
  - For word2vec