8.2 Introduction to language models

Lecture based on "Dive into Deep Learning" **http://D2L.AI** (Zhang et al., 2020)

Prof. Dr. Christoph Lippert

Digital Health & Machine Learning

- **Words** (or **characters**) are modeled as a time series of discrete observations.
  $w_1, w_2, \ldots, w_T$
- $w_t (1 \leq t \leq T)$ is the output or label of time step t.
- A **language model** estimates
  $$p(w_1, w_2, \ldots, w_T).$$
- An *ideal* language model would generate text, by drawing words
  $w_t \sim p(w_t | w_{t-1}, \ldots w_1)$
  - … would pass as natural language, e.g. English text.
  - … generate a meaningful dialog, by conditioning on previous dialog fragments.
  - Currently not possible.
    would need to *understand* the text rather than just generate grammatically sensible content.

What we *can* tackle using **language models**:

- Improve speech recognition
    - The phrases *'to recognize speech'* and *'to wreck a nice beach'* sound very similar.
- Perform document summarization
    - *'dog bites man'* is much more frequent than *'man bites dog'*,
    - *'let's eat, grandma'*.

We model a distribution over a document, or even a sequence of words.

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t | w_1, \ldots, w_{t-1}).$$

Example

The probability of a text sequence containing four tokens (words and punctuation):

$$p(\text{Statistics}, \text{is}, \text{fun}, .) = p(\text{Statistics})p(\text{is}|\text{Statistics})p(\text{fun}|\text{Statistics}, \text{is})p(.|\text{Statistics}, \text{is}, \text{fun}).$$

- Requires estimates for language model parameters
  - word probabilities $p(w)$
  - conditional word probabilities $p(w_t | w_{t-1} \ldots)$
- The training data set typically is a large text corpus
  - Wikipedia
  - Project Gutenberg
  - text posted online on the web.
- Word probabilities can be estimated from the relative word frequency in the training corpus.

$$p(\text{Statistics}, \text{is}, \text{fun}, .) = p(\text{Statistics})p(\text{is}|\text{Statistics})p(\text{fun}|\text{Statistics}, \text{is})p(.|\text{Statistics}, \text{is}, \text{fun}).$$

- $p(\text{Statistics})$ can be estimated as
  - frequency of sentences starting with 'statistics'.
  - count all occurrences of the word 'statistics' and divide it by the total number of words in the corpus.
    This works particularly well for frequent words.
- $\hat{p}(\text{is}|\text{Statistics}) = \frac{n(\text{Statistics is})}{n(\text{Statistics})}$.
  - $n(w)$ number of occurrences of **singletons**
  - $n(w, w')$ number of occurrences of **pairs** (triples, ...) of words
- Estimation gets harder, as occurrences of combinations (*'Statistics is'*) are less frequent.

In **Laplace smoothing** a small constant $\epsilon_i > 0$ is added to all counts for sequences of length i.

- m: total number of words.

$$\hat{p}(w) = \frac{n(w) + \epsilon_1/m}{n + \epsilon_1}$$
$$\hat{p}(w'|w) = \frac{n(w,w') + \epsilon_2 \hat{p}(w')}{n(w) + \epsilon_2}$$
$$\hat{p}(w''|w',w) = \frac{n(w,w',w'') + \epsilon_3 \hat{p}(w',w'')}{n(w,w') + \epsilon_3}$$

Downsides:

- need to store all counts
- ignores the meaning and context of words.

  For instance, *'cat'* and *'feline'* should occur in related contexts.
- frequencies for long word sequences are hard/impossible to estimate.

n-grams

A distribution over sequences satisfies the **Markov property** of **first order** if

$$p(w_{t+1}|w_t, \dots w_1) = p(w_{t+1}|w_t)$$

Higher orders correspond to longer dependencies.

This leads to a number of approximations:

$$p(w_1, w_2, w_3, w_4) = p(w_1)p(w_2)p(w_3)p(w_4) \qquad \text{(}0^{th}\text{ order / \textbf{unigram} model)}$$

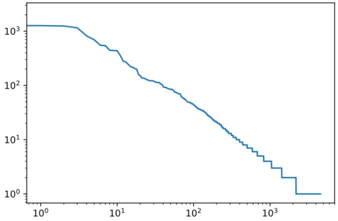
$$p(w_1, w_2, w_3, w_4) = p(w_1)p(w_2|w_1)p(w_3|w_2)p(w_4|w_3) \qquad \text{(}1^{st}\text{ order / \textbf{bigram} model)}$$

$$p(w_1, w_2, w_3, w_4) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)p(w_4|w_2, w_3) \qquad \text{(}2^{nd}\text{ order / \textbf{trigram} model)}$$

Example

H.G. Wells' Time Machine.

- Small corpus: 30,000 words.
- The word frequencies decay rapidly in a well defined way.
- After dealing with the first four words as exceptions ('the', 'i', 'and', 'of'), all remaining words follow a straight line on a log-log plot.



Word frequencies satisfy Zipf's law given by

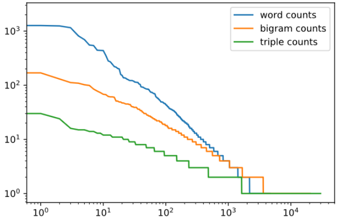
$$n(x) \propto (x+c)^{-\alpha} \quad \Leftrightarrow \quad \log n(x) = -\alpha \log(x+c) + \text{const.}$$

Modeling words by count statistics and smoothing will significantly overestimate the frequency of infrequent words.

Example

H.G. Wells' Time Machine.

- Small corpus: 30,000 words.
- After the first four words ('the', 'i', 'and', 'of'), all remaining words follow Zipf's law.



- Sequences of words follow Zipf's law.
  - with a lower exponent, depending on sequence length.
- the number of distinct n-grams not large.
  - non-uniform distribution of sequences (language structure).
- *many* n-grams occur very rarely
  - makes Laplace smoothing rather unsuitable for language modeling.

Summary

- n-grams model long sequences by truncating the dependences.
- Long sequences occur very rarely or never.

  This requires smoothing.
- Word and n-gram distributions
  - follow Zipf's law.
  - Have lot of structure but not enough frequency for infrequent word combinations efficiently via smoothing.