Part-of-Speech Tagging

+

Neural Networks 3: Word Embeddings

CS 287

Review: Neural Networks

One-layer multi-layer perceptron architecture,

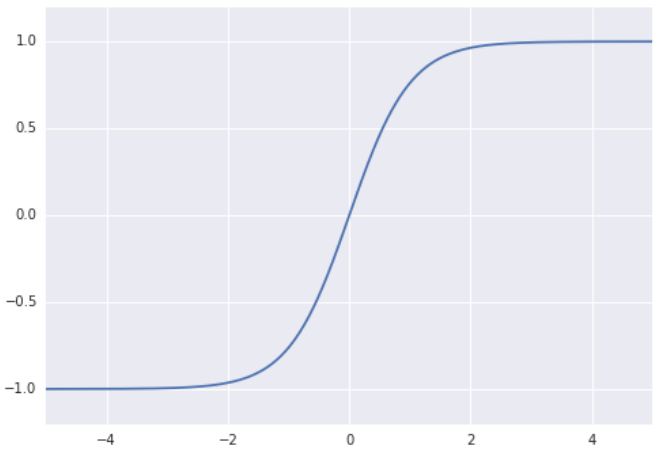
$$NN_{MLP1}(\mathbf{x}) = g(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2$$

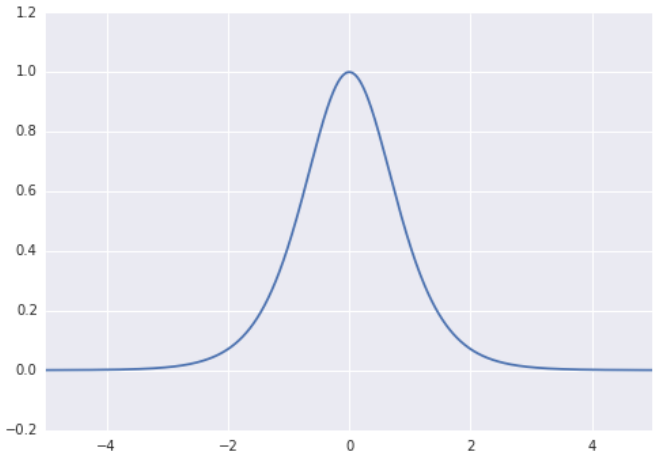
- $\mathbf{x}\mathbf{W} + \mathbf{b}$; *perceptron*
- $\mathbf{x}$ is the dense representation in $\mathbb{R}^{1 \times d_{\text{in}}}$
- $\mathbf{W}^1 \in \mathbb{R}^{d_{\text{in}} \times d_{\text{hid}}}$, $\mathbf{b}^1 \in \mathbb{R}^{1 \times d_{\text{hid}}}$; first affine transformation
- $\mathbf{W}^2 \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{out}}}$, $\mathbf{b}^2 \in \mathbb{R}^{1 \times d_{\text{out}}}$; second affine transformation
- $g : \mathbb{R}^{d_{\text{hid}} \times d_{\text{hid}}}$ is an *activation non-linearity* (often pointwise)
- $g(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1)$ is the *hidden layer*

Review: Non-Linearities Tanh

Hyperbolic Tangeant:

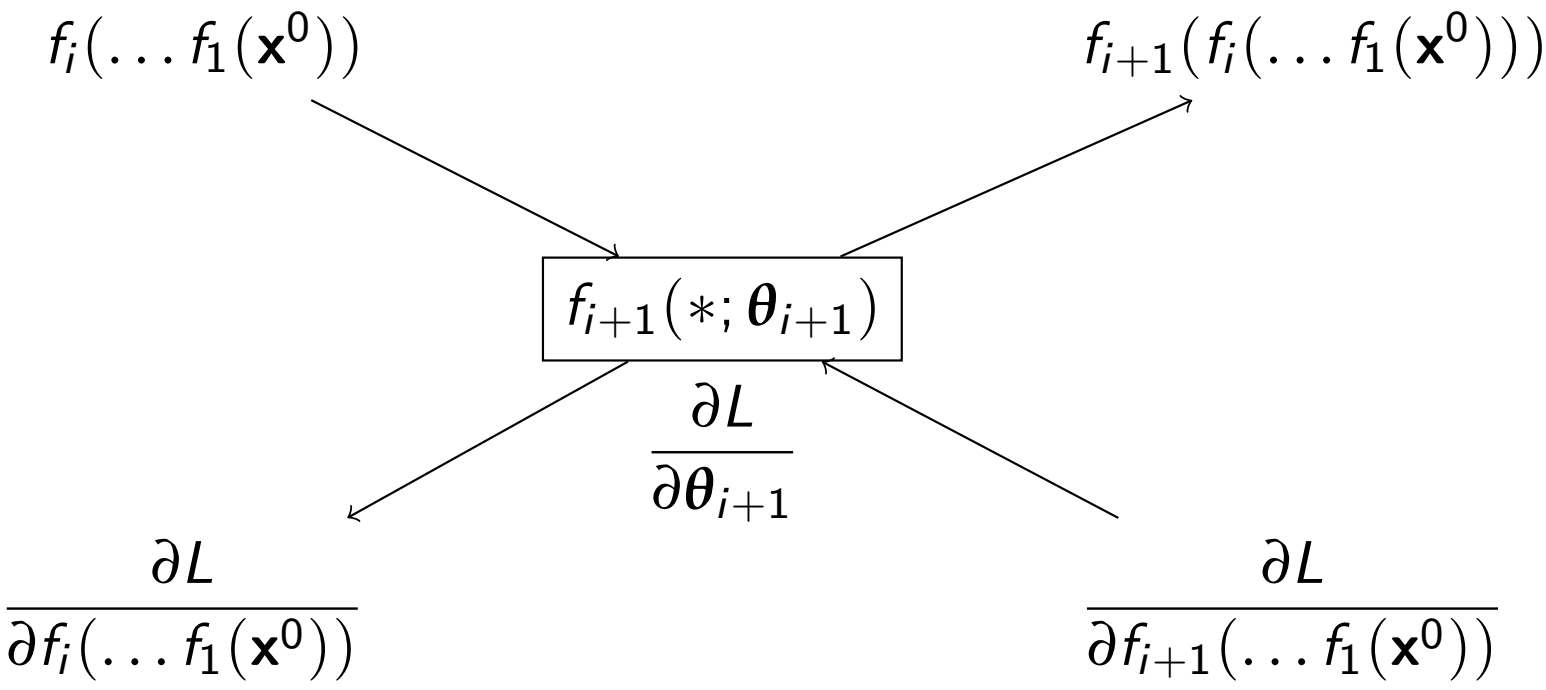
$$\tanh(t) = \frac{\exp(t) - \exp(-t)}{\exp(t) + \exp(-t)}$$





- Intuition: Similar to sigmoid, but range between 0 and -1.

Review: Backpropagation



Quiz

One common class of operations in neural network models is known as *pooling*. Informally a pooling layer consists of aggregation unit, typically unparameterized, that reduces the input to a smaller size.

Consider three pooling functions of the form $f : \mathbb{R}^n \mapsto \mathbb{R}$,

1. $f(\mathbf{x}) = \max_i x_i$
2. $f(\mathbf{x}) = \min_i x_i$
3. $f(\mathbf{x}) = \sum_i x_i / n$

What action do each of these functions have? What are their gradients? How would you implement backpropagation for these units?

Quiz

- **Max pooling**: $f(\mathbf{x}) = \max_i x_i$
  - Keeps only the most activated input
  - Fprop is simple; however must store $\arg\max$ ("switch")
  - Bprop gradient is zero except for switch, which gets gradoutput
- **Min pooling**: $f(\mathbf{x}) = \min_i x_i$
  - Keeps only the least activated input
  - Fprop is simple; however must store $\arg\min$ ("switch")
  - Bprop gradient is zero except for switch, which gets gradoutput
- **Avg pooling**: $f(\mathbf{x}) = \sum_i x_i / n$
  - Keeps the average activation input
  - Fprop is simply mean.
  - Gradoutput is averaged and passed to all inputs.

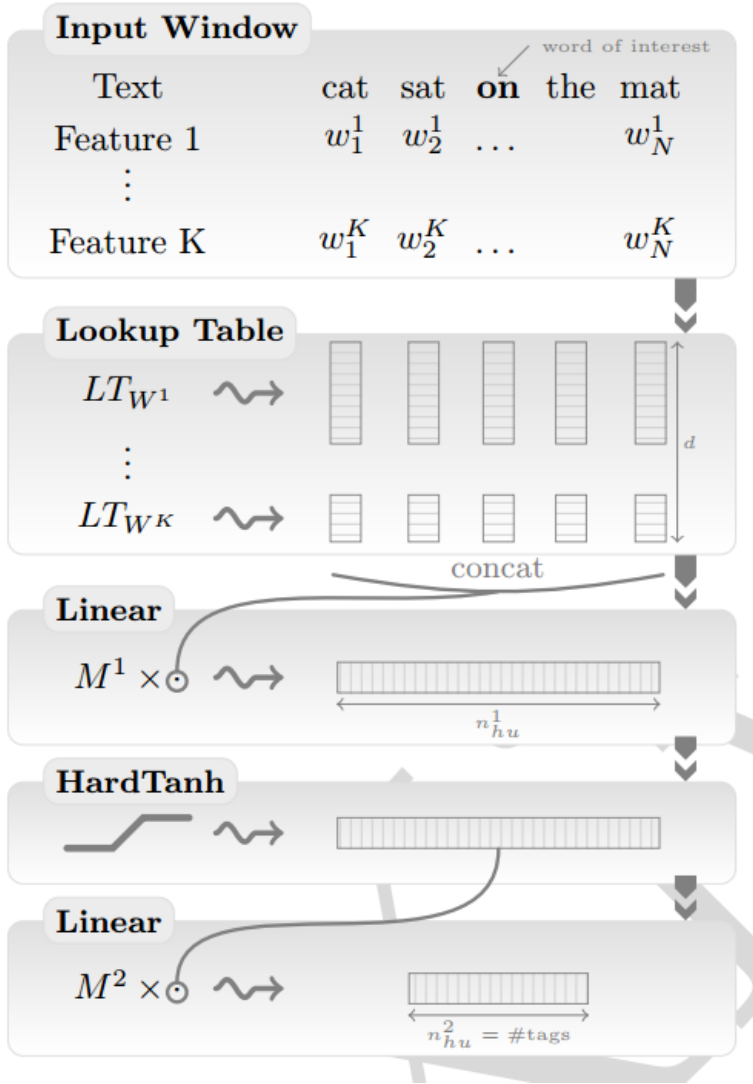
Quiz

- **Max pooling**: $f(\mathbf{x}) = \max_i x_i$
  - Keeps only the most activated input
  - Fprop is simple; however must store $\arg\max$ ("switch")
  - Bprop gradient is zero except for switch, which gets gradoutput
- **Min pooling**: $f(\mathbf{x}) = \min_i x_i$
  - Keeps only the least activated input
  - Fprop is simple; however must store $\arg\min$ ("switch")
  - Bprop gradient is zero except for switch, which gets gradoutput
- **Avg pooling**: $f(\mathbf{x}) = \sum_i x_i / n$
  - Keeps the average activation input
  - Fprop is simply mean.
  - Gradoutput is averaged and passed to all inputs.

Quiz

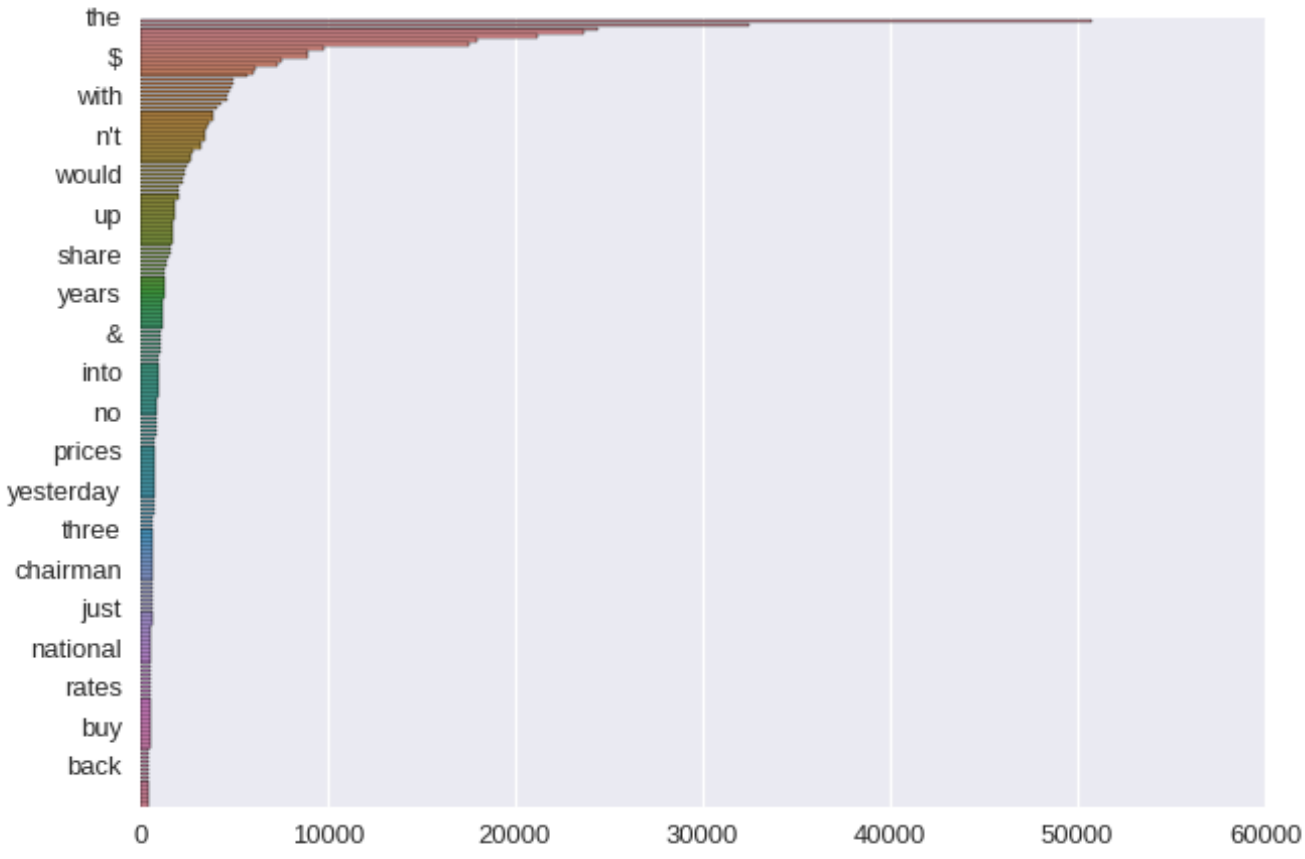
- **Max pooling**: $f(\mathbf{x}) = \max_i x_i$
  - Keeps only the most activated input
  - Fprop is simple; however must store $\arg\max$ ("switch")
  - Bprop gradient is zero except for switch, which gets gradoutput
- **Min pooling**: $f(\mathbf{x}) = \min_i x_i$
  - Keeps only the least activated input
  - Fprop is simple; however must store $\arg\min$ ("switch")
  - Bprop gradient is zero except for switch, which gets gradoutput
- **Avg pooling**: $f(\mathbf{x}) = \sum_i x_i / n$
  - Keeps the average activation input
  - Fprop is simply mean.
  - Gradoutput is averaged and passed to all inputs.

Contents



1. Use dense representations instead of sparse
2. Use windowed area instead of sequence models
3. Use neural networks to model windowed interactions

What about rare words?



Word Embeddings

Embedding layer,

$$\mathbf{x}^0\mathbf{W}^0$$

- $\mathbf{x}^0 \in \mathbb{R}^{1\times d_0}$ one-hot word.
- $\mathbf{W}^0 \in \mathbb{R}^{d_0\times d_{\text{in}}}$, $d_0 = |\mathcal{V}|$

Notes:

- $d_0 >> d_{\text{in}}$, e.g. $d_0 = 10000, d_{\text{in}} = 50$

Pretraining Representations

- We would strong shared representations of words
- However, PTB only 1M labeled words, relatively small
- Collobert et al. (2008, 2011) use semi-supervised method.
- (Close connection to Bengio et al (2003), next topic)

Semi-Supervised Training

Idea: Train representations separately on more data

1. Pretrain word embeddings $\mathbf{W}^0$ first.
2. Substitute them in as first NN layer
3. Fine-tune embeddings for final task
   - Modify the first layer based on supervised gradients
   - Optional, some work skips this step

Large Corpora

To learn rare word embeddings, need many more tokens,

- C&W
  - English Wikipedia (631 million words tokens)
  - Reuters Corpus (221 million word tokens)
  - Total vocabulary size: 130,000 word types
- word2vec
  - Google News (6 billion word tokens)
  - Total vocabulary size: $\approx$ 1M word types

But this data has no labels...

Contents

C&W Embeddings

- All text in Wikipedia in some sense coherent.
- Good embeddings should be able to help distinguish coherent text from nonsense
- We can train on this task.

C&W Setup

Let $\mathcal{V}$ be the vocabulary of English and let s score any window of size d_{win}, if we see the phrase

to the

It should score higher by s than

to the

to the

to the

- ...

C&W Setup

Can estimate s as a windowed neural network.

$$s(w_1, \ldots, w_{d_{\text{win}}}) = \text{hardtanh}(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}$$

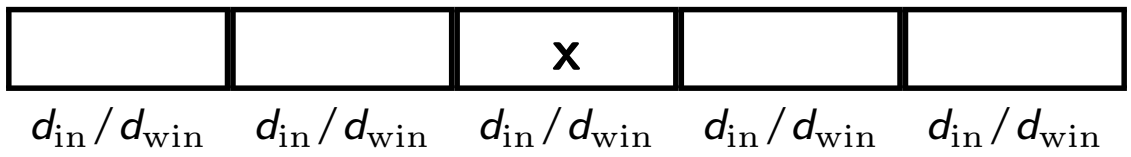
with

$$\mathbf{x} = [v(w_1)\ v(w_2)\ \ldots\ v(w_{d_{\text{win}}})]$$

- $d_{\text{in}} = d_{\text{win}} \times 50$, $d_{\text{hid}} = 100$, $d_{\text{win}} = 11$, $d_{\text{out}} = 1$!

Example: Function s

$$\mathbf{x} = [v(w_3)\ v(w_4)\ v(w_5)\ v(w_6)\ v(w_7)]$$



Training?

- Different setup than previous experiments.
-

Ranking Loss

Given only example $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and for each example have set $\mathcal{D}(\mathbf{x})$ of alternatives.

$$\mathcal{L} = \sum_i \sum_{\mathbf{x}' \in \mathcal{D}(\mathbf{x})} L_{ranking}(s(\mathbf{x}_i), s(\mathbf{x}')$$

$$L_{ranking}(y, \hat{y}) = \max\{0, 1 - (y - \hat{y})\}$$

Example: C&W ranking

$\mathbf{x}$ = [the dog walks to the]

$\mathcal{D}(\mathbf{x}) = \{$ [the dog skips to the], [the dog in to the], $\ldots \}$

terion

- Note: slightly different setup, both $s(\mathbf{x}_i)$ and $s(\mathbf{x}')$ receive gradient.

- Vocabulary size $|\mathcal{D}(\mathbf{x})| > 100,000$
- Training time for 4 weeks
- (guy who invented torch)

WSABIE

Observation: in many contexts

$$L_{ranking}(y, \hat{y}) = \max\{0, 1 - (y - \hat{y})\} = 0$$

k But for difficult contexts, may be easy to find

$$L_{ranking}(y, \hat{y}) = \max\{0, 1 - (y - \hat{y})\} \neq 0$$

We can therefore sample from $\mathcal{D}(\mathbf{x})$ to find an update.

C&W Results

Approach	POS (PWA)	CHUNK (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99
NN+WLL+LM1	97.05	91.91	85.68	58.18
NN+SLL+LM1	97.10	93.65	87.58	73.84
NN+WLL+LM2	97.14	92.04	86.96	58.34
NN+SLL+LM2	97.20	93.63	88.67	74.15

1. Use dense representations instead of sparse
2. Use windowed area instead of sequence models
3. Use neural networks to model windowed interactions
4. Use semi-supervised learning to pretrain representations.

Contents

Other Embedding Method: word2vec

- Instead of MLP uses Bilinear model ("linear" in paper)
- Instead of ranking model, directly predict word (cross-entropy)
- Two different models
  1. Continuous Bag-of-Words (CBOW)
  2. Continuous Skip-gram
- Various other ideas.

Bilinear Model

Bilinear model,

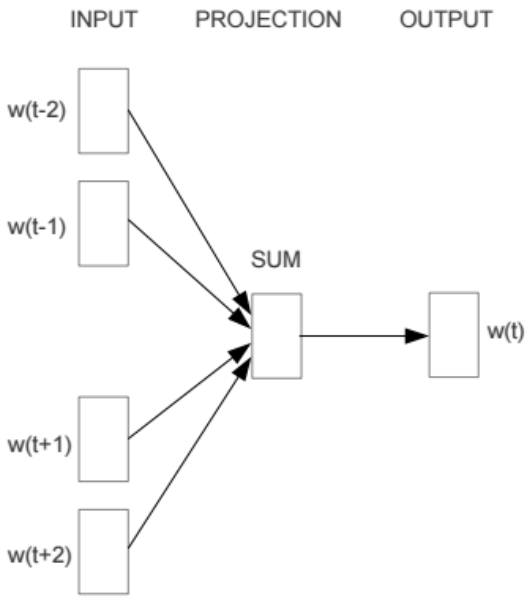
$$\hat{\mathbf{y}} = f((\mathbf{x}^0\mathbf{W}^0)\mathbf{W}^1 + \mathbf{b})$$

- $\mathbf{x}^0 \in \mathbb{R}^{1\times d_0}$ start with one-hot.
- $\mathbf{W}^0 \in \mathbb{R}^{d_0\times d_{\text{in}}}$, $d_0 = |\mathcal{F}|$
- $\mathbf{W}^1 \in \mathbb{R}^{d_{\text{in}}\times d_{\text{out}}}, \mathbf{b} \in \mathbb{R}^{1\times d_{\text{out}}}$; model parameters

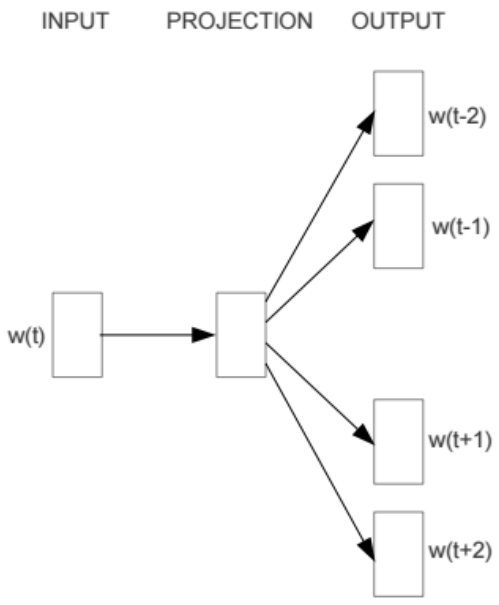
Notes:

- Bilinear parameter interaction.
- $d_0 >> d_{\text{in}}$, e.g. $d_0 = 10000, d_{\text{in}} = 50$

word2vec (Mikolov, 2013)



CBOW



Skip-gram

Continuous Bag-of-Words (CBOW)

- Bag-of-words bilinear model, after embedding
- Attempt to predict the middle word

Example: CBOW

$\mathbf{W}^1$ is no longer partitioned by row

Skip-gram

- Also a bilinear model, after embedding
- Attempt to predict outer word from middle

Example: Skip-gram

Softmax

Use a softmax to force a distribution,

$$\text{softmax}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\sum_{c \in \mathcal{C}} \exp(z_c)}$$

$$\log \text{softmax}(\mathbf{z}) = \mathbf{z} - \log \sum_{c \in \mathcal{C}} \exp(z_c)$$

- Issue: class $\mathcal{C}$ is huge.
- For C&W, 100,000, for word2vec 1,000,000
- Note datasets, 6 billion words

First run hard clustering.

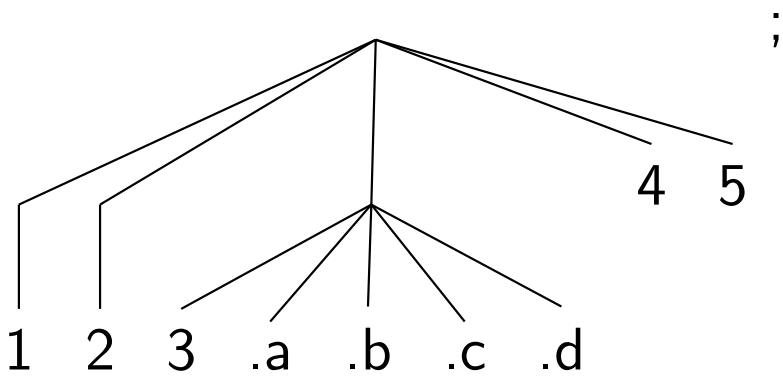
$$p(Y|X) = P(Y|C, X; \theta)P(C|X; \theta)$$

$P(C|X; \theta)$

$$\hat{\mathbf{y}} = \text{softmax}((\mathbf{x}^0\mathbf{W}^0)\mathbf{W}^1 + \mathbf{b})$$

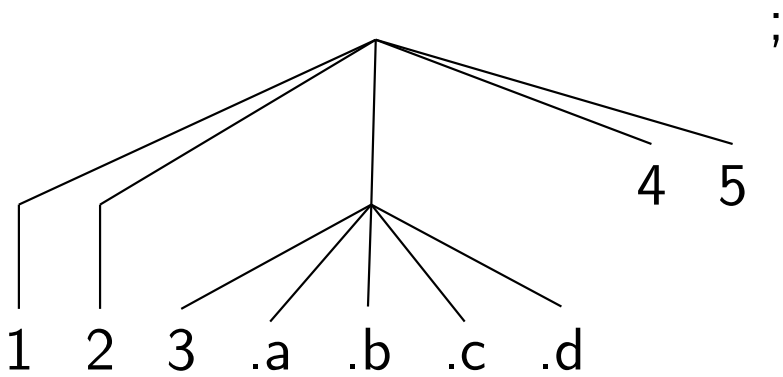
$P(Y|C, X; \theta)$

$$\hat{\mathbf{y}} = \text{softmax}((\mathbf{x}^0\mathbf{W}^0)\mathbf{W}^C + \mathbf{b}))$$


;
4 5
1 2 3 .a .b .c .d

Computing full $p(Y|X)$ still requires $O(|\mathcal{V}|)$ time.

$L_{2SM}() = -\log p(Y|X, C) - \log p(C|X)$

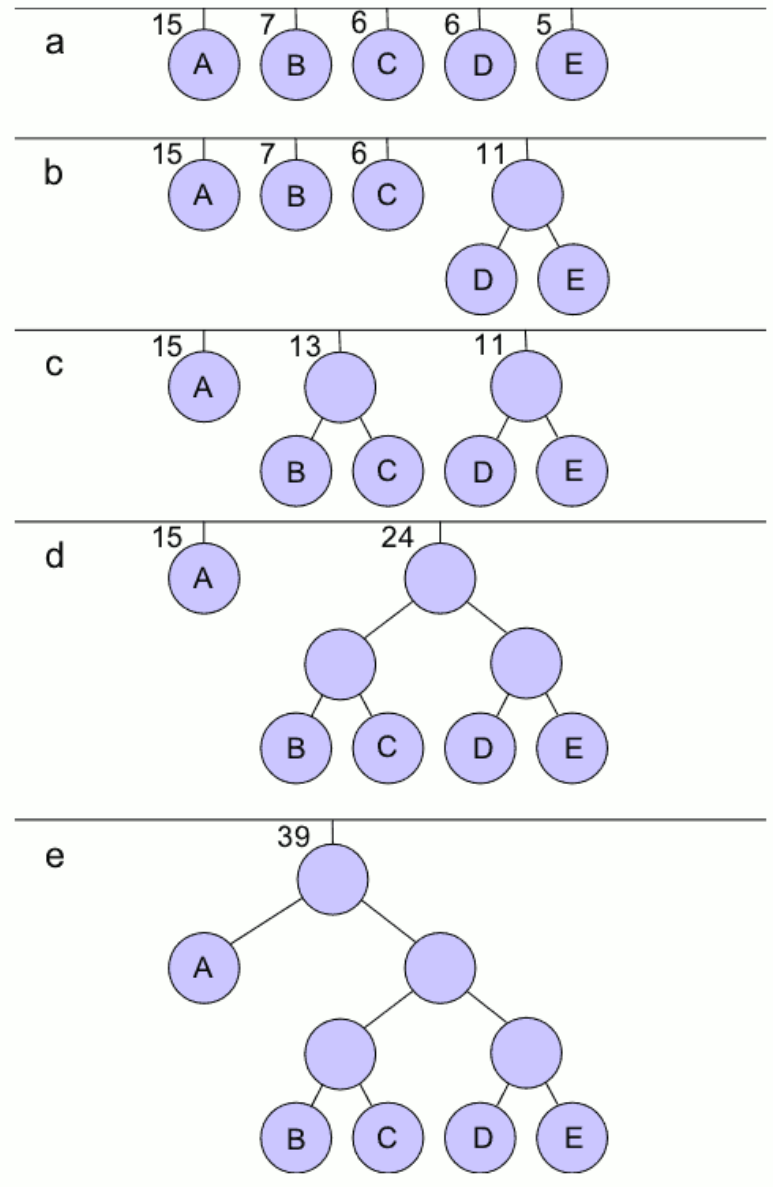

;
4 5
1 2 3 .a .b .c .d

- Best two-layer is with a balanced tree.
- Requires $O(\sqrt{|\mathcal{V}|})$

Hierarchical Softmax

- Requires $O(\log_2 |\mathcal{V}|)$
  $L_{HSM}() = -\sum_{l=0}^{L(c)} \log p(\hat{\mathbf{y}} = \delta(c)|X, C_1, \ldots, C_l)$
- One paper on website (Mnih and Hinton, 2008)

Huffman Encoding



Hierarchical Softmax

- Requires $O(\log_2 perp(unigram))$

Other Elements

- At each SGD step d_{win} is sampled
- Frequent words are used less

Contents

How good are embeddings?

- Anecdotal Metric-based
- Analogy task
- Extrinsic Metrics

Dot-product

$$\mathbf{x}_{cat}\mathbf{x}_{dog}^{\top}$$

Cosine Similarity

$$\frac{\mathbf{x}_{cat}\mathbf{x}_{dog}^{\top}}{||\mathbf{x}_{cat}||||\mathbf{x}_{dog}||}$$

dog	cat	0.921800527377
	dogs	0.851315870426
	horse	0.790758298322
	puppy	0.775492121034
	pet	0.772470734611
	rabbit	0.772081457265
	pig	0.749006160038
	snake	0.73991884888

Analogy questions:

`A:B::C:__`

- 5 types of semantic questions, 9 types of syntactic

$$\mathbf{x}' = \mathbf{x}_B - \mathbf{x}_A + \mathbf{x}_C$$

$$\arg\max_{\mathbf{x}_D} \frac{\mathbf{x}_D \mathbf{x}'^{\top}}{||\mathbf{x}_D|| ||\mathbf{x}'||}$$

Embedding Tasks

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
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Extrinsic Tasks

-

Visualizations

- Projections (PCA)
- T-SNE