CS371N Lecture 9
Language Modeling

Announcements

- A1 back
- A2 due Thurs
- Bias due Thurs
- A3 out Thurs
- Heng Ji talk Friday 11am GDC 6302

Recap

Skip-gram $P(\text{context}=y \mid \text{word}=x) = \frac{e^{\bar{v}_x \cdot \bar{c}_y}}{\sum_{y'} e^{\bar{v}_x \cdot \bar{c}_{y'}}}$

Bias, multilingual, other methods

Today - Language Modeling
- N-gram LMs
- Neural LMs
- RNN LMs (brief!)

LM "Autocomplete" / predictive text

Predict the next word w_i given words that came before $w_1 \dots w_{i-1}$

$$P(w_i \mid w_1 \dots w_{i-1})$$

Prob of a sequence $\bar{w}$: (n words)

$$P(\bar{w}) = \prod_{i=1}^{n} P(w_i \mid w_1 \dots w_{i-1})$$

"What is the prob. of seeing this utterance?"

- Fill-in-the-blank (cloze)
- Maintain uncertainty over upcoming words
- Grammatical error correction:

  $\bar{w}$, $\bar{w}'$ one edit applied

  $P(\bar{w}') \stackrel{?}{>} P(\bar{w}) \Rightarrow$ more grammatical?

N-gram Language Modeling

$$P(\bar{w}) = P(w_1)\, P(w_2 \mid w_1)\, P(w_3 \mid w_1, w_2)\, P(w_4 \mid w_1, w_2, w_3) \cdots$$

↑ LM in general

Suppose we have a long book

$P(w_{100000} | w_1 \cdots w_{99999})$

n-gram LM: only look at past $n-1$ words

$$P(\bar{w}) = \prod_{i=1}^{n} P(w_i | w_{i-n+1} \cdots w_{i-1})$$

$n = 3$:

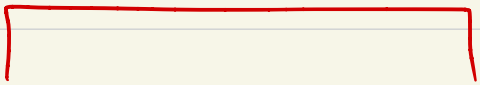
$$P(\bar{w}) = P(w_1)\, P(w_2 | w_1)\, P(w_3 | w_1 w_2)\, P(w_4 | w_2 w_3) \cdots$$

no w_1!

$n = 2$:

$$P(\bar{w}) = P(w_1)\, P(w_2 | w_1)\, P(w_3 | w_2) \cdots$$

Ex



I saw the dog ____

run swim
walk...
food

n=5

n=2 : hates

My dog hates cats

n-gram LM parameterization

Explicitly store probs for words based on counts in the data

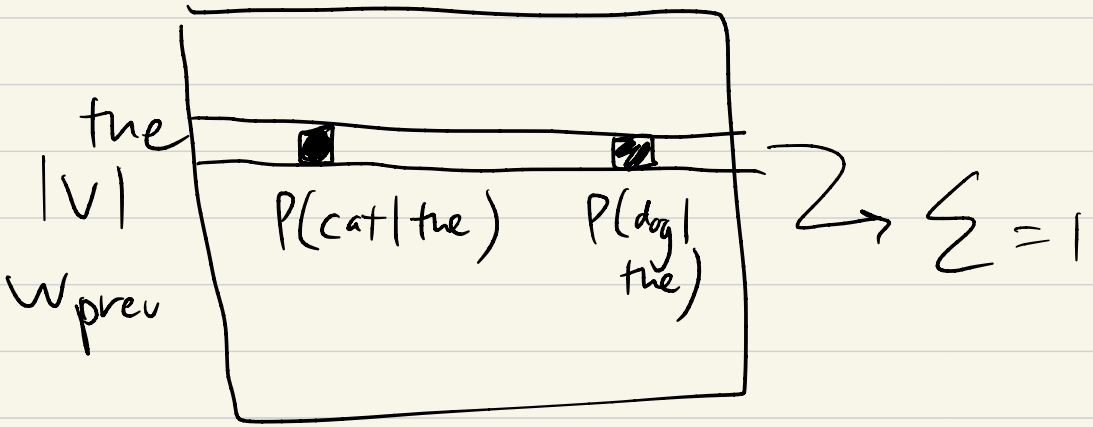
Categorical distribution

2-gram

Vocab

cat |V| w

$P(w|w_{prev})$

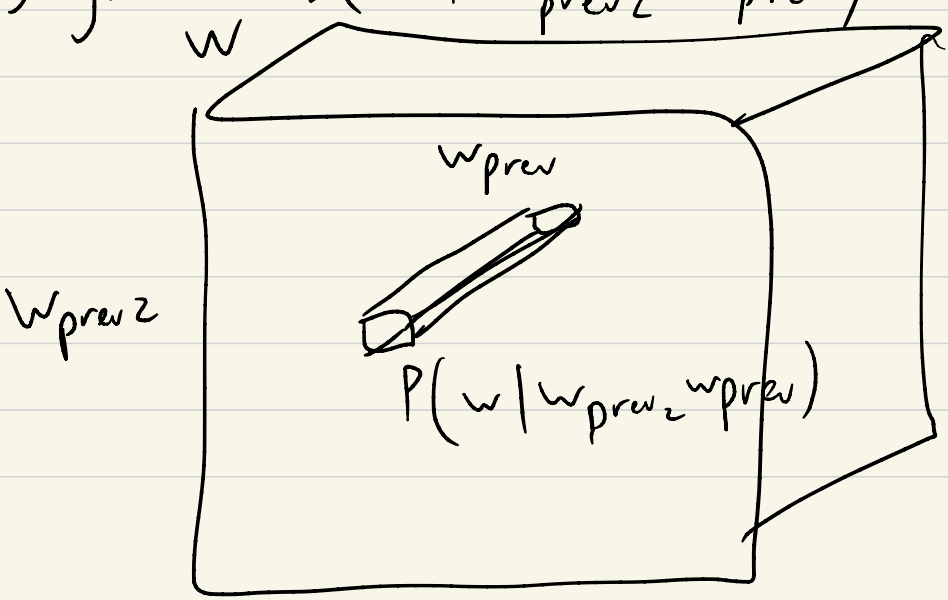


one row per word

3-gram

$P(w|w_{prev2}\ w_{prev})$

w



These probs are sparse

actual size $<< |V|^3$

$n \sim 7$ is doable

Parameter estimation

Count + normalize over a big corpus

the cat
the cat
the dog
the snake

$$P(\text{cat} \mid \text{the}) = \frac{\text{count(the cat)}}{\text{count(the)}} = \frac{2}{4}$$

$$P(\text{dog} \mid \text{the}) = \frac{1}{4}$$

$$P(\text{snake} \mid \text{the}) = \frac{1}{4}$$

Maximum-Likelihood Estimation

These probs maximize LL of this data

Smoothing

"I'd hate to go to Maui"

Suppose we have $n=5$

$$P(\text{Maui} \mid \text{hate to go to}) = \frac{\text{Count}(\text{hate to go to Maui})}{\text{Count}(\text{hate to go to})}$$

0!

Solutions

- Give every 5 gram count 1 (0.01)

"hate to go to" seen 5 times

+ |V| "fake" counts

- Getting rid of sparsity

Smoothing a 4-gram model

$$P_4(w_i \mid w_{i-3}\ w_{i-2}\ w_{i-1})$$

normalized raw counts

$$= \lambda P_4^{raw}(w_i \mid w_{i-3}\ w_{i-2}\ w_{i-1}) + (1-\lambda) P_3(w_i \mid w_{i-2}\ w_{i-1})$$

$$P_3(w_i \mid w_{i-2}\ w_{i-1}) = \lambda P_3^{raw}(w_i \mid w_{i-2}\ w_{i-1}) + (1-\lambda) P_2 \cdots$$

$P_1(w) > 0$ for all $w \in V$

λ : hyperparameter $\in [0, 1]$

Key ideas

We can learn from a lot of data

We can model a distribution over next words given (unlimited) context

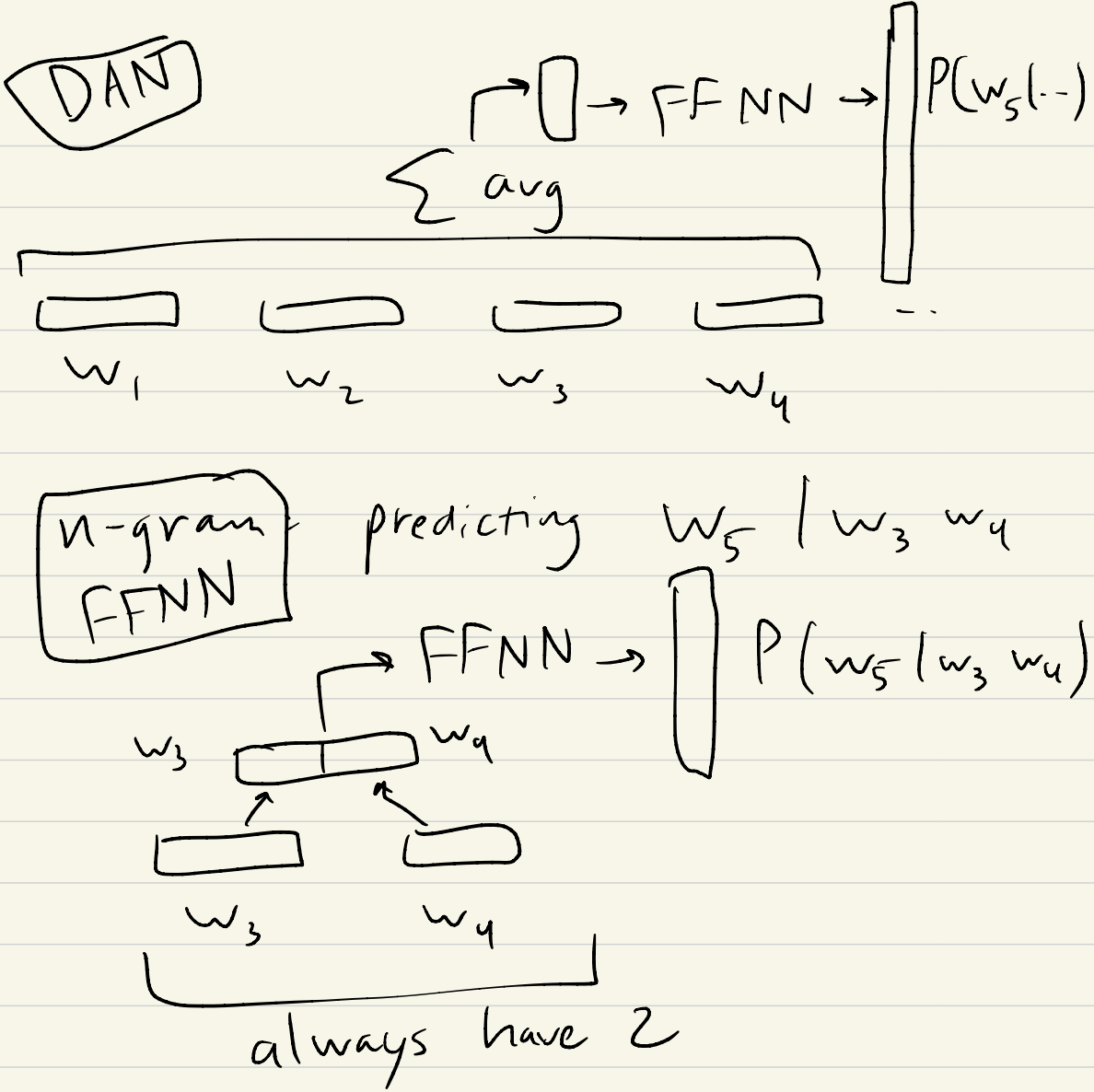
Neural Language Modeling

$P(w_i | w_1 \dots w_{i-1})$ → model w/a neural net

Two choices: ① Make the n-gram approximation; ② don't

Predict w_i based on $\underbrace{w_1 \dots w_{i-1}}$

"sentence"


DAN
FFNN
P(w5|...)
avg
w1
w2
w3
w4
n-gram FFNN
predicting w5 | w3 w4
FFNN
P(w5 | w3 w4)
w3
w4
w3
w4
always have 2

DAN Advantages:

① Fewer params ($n=2$)

② Look at all the context

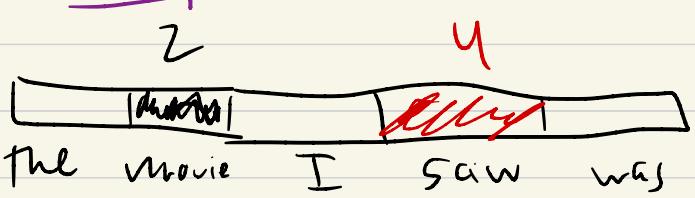
Disadvantages

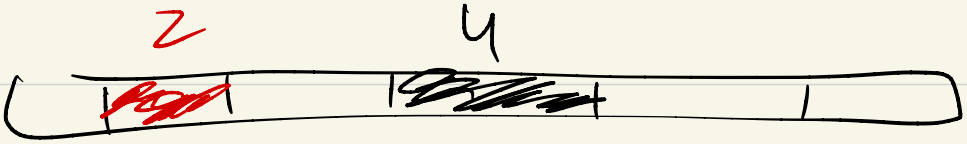
① No notion of order

FFNN ① Keeps order (over a shorter context)

↳ negation!

Parameters: it's inefficient to completely model order





I saw the movie which was

FF-NN distinguishes these

Solutions

(1) RNNs (recurrent neural nets): process sequences "uniformly"

(2) Transformers: DAN + ordering approach

RNN: maintains a "hidden state" $\bar{h}$ after processing n words

$$\bar{h}_n = NN(w_1, \cdots, w_n)$$

$$P(w_{n+1} \mid w_1 \cdots w_n) = \text{softmax}(U \bar{h}_n)$$

U: $|V| \times d$ matrix

Key RNN idea:

$$\bar{h}_{n+1} = \underline{NN}(\bar{h}_n, w_{n+1})$$

easy to update