CS 378 Lecture 15:
Language Modeling, RNNs

Today

- Intro to language modeling
- N-gram LMs
- Neural LMs
- (Start) RNN LMs (A4)

Announcements

- Midterm, A3 grading
- A4
- FP custom proposals (optional!) due Monday (changed)

Recap (so far)

text →
- classify (sentiment)
- POS
- NER
- syntactic parse trees

text → label or structure

Next few weeks: text → text
(example: machine translation) seq^{2seq} models
dialogue, summarization
... everything?

Today: Language modeling
"autocomplete", predictive text
predict the next word given words
that come before it

Technique: recurrent neural networks
(RNNs) (+ Transformers)
sequence models

Language Modeling

Distribution $P(\overline{w})$ over (grammatical, well-formed, natural) sentences in a language

$\overline{w}$ is a sequence of words

Why LM?

Grammatical error correction.
You give me $\overline{w}$
I fix some errors by finding
$\overline{w}'$ s.t. $P(\overline{w}') > P(\overline{w})$

Machine translation:

Sentence $\overline{w}_s$ in source language

$\rightarrow$ $\overline{w}_{t,1}$ two candidates.
$\overline{w}_{t,2}$
Check if $P(\overline{w}_{t,1}) > P(\overline{w}_{t,2})$
return higher

N-gram language modeling

By the chain rule of probability:

$$P(\bar{w}) = P(w_1)\, P(w_2 | w_1)\, P(w_3 | w_1 w_2)\, P(w_4 | w_1 w_2 w_3) \cdots P(w_n | w_1 \ldots w_{n-1})$$

(not Markov assumption)

If we make an assumption:
only depend on past $n-1$ words

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

2-gram model: the cat ran

$$P_2(\bar{w}) = P(\text{the} | \langle s \rangle)\, P(\text{cat} | \text{the})\, P(\text{ran} | \text{cat})\, P(\text{STOP} | \cancel{\text{ran}})$$

3-gram: $P(\text{the} \mid \langle s \rangle \langle s \rangle)\, P(\text{cat} \mid \langle s \rangle\ \text{the})$
$P(\text{ran} \mid \text{the cat})$

<u>Poll</u>

I saw the dog _____

wagging, bark, in, on, jump

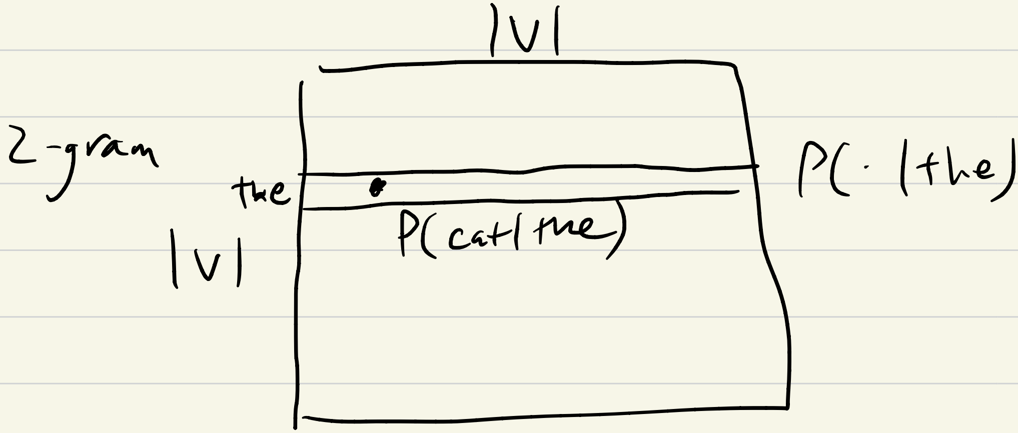
I saw the dog. _____

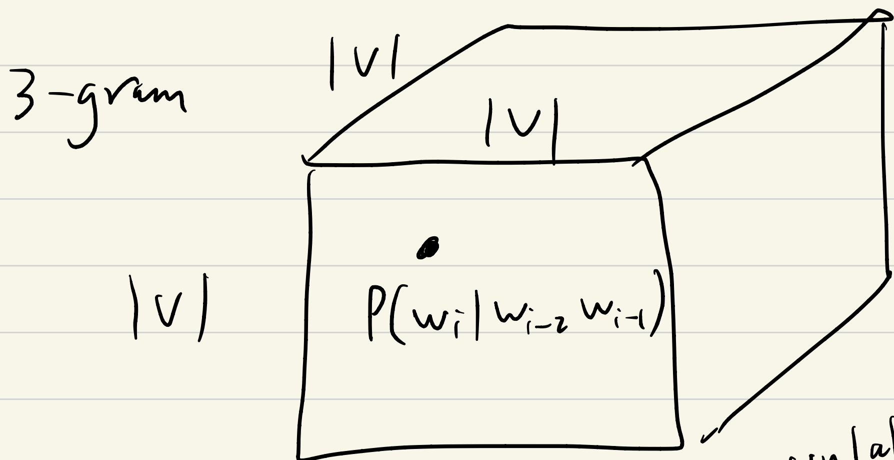
It, any word that starts a sentence

<u>less</u> <u>restrictive</u> <u>context</u>

N-gram parameterization   V = vocabulary

Big lookup table (like HMM transitions)





Count + normalize over a big unlabeled! corpus

To do well, we need $n \geq 5$ (5-gram)

$$\text{I} \underbrace{\text{hate to go to Maui}}_{\text{5 words}}$$

Count (hate to go to Maui) on the web? May be 0!

$\Rightarrow P(\text{Maui} \mid \text{hate to go to}) = 0$

$\Rightarrow$ incorrect, should be > 0

<u>Smoothing</u> (in n-gram LMs)

$$P_5(w_i \mid w_{i-4}\, w_{i-3}\, w_{i-2}\, w_{i-1})$$

raw counts

$$\approx \lambda\, P_5^{raw}(w_i \mid w_{i-4} \cdots w_{i-1})$$
$$+ (1-\lambda)\, P_4(w_i \mid w_{i-3}\, w_{i-2}\, w_{i-1})$$

$$P_4 = \lambda P_4^{raw} + (1-\lambda) P_3$$

$$P_5 = \lambda_1 P_5^{raw} + \lambda_2 P_4^{raw} + \lambda_3 P_3^{raw} + \lambda_4 P_2^{raw} + \lambda_5 P_1^{raw}$$

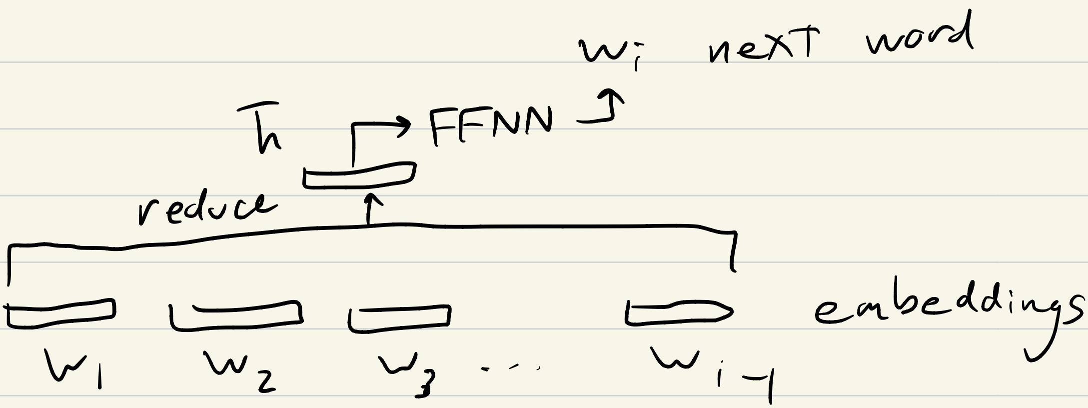
$$P_1^{raw} = \frac{\text{count}(\text{Mauri})}{\text{size of corpus}} > 0$$

$$P_5(\text{Mauril} \cdots) > 0$$

Lots of very complex tricks here!

Neural Language Models

$P(w_i | \underbrace{w_1, \ldots, w_{i-1}}_{\text{all prev words, not just } n-1}) \Rightarrow$ model w/a NN



GPT-3 reduce: Transformer

For us reduce: RNN

Also possible reduce: DAN

FFNN over $n-1$ words

d-dim vector

$\bar{h} = \text{neural net}(w_1, \ldots, w_{i-1})$

$P(w_i \mid w_1, \ldots, w_{i-1}) = \text{softmax}(W\bar{h})$

$W: |V| \times d$ matrix

weight matrix

multiclass w/ $|V|$ classes

A simple ex: DAN no ordering

$\bar{h} = \text{avg}(\bar{w}_1, \ldots, \bar{w}_{i-1})$ d-dim vector

$W\bar{h}: |V|$-dim vector

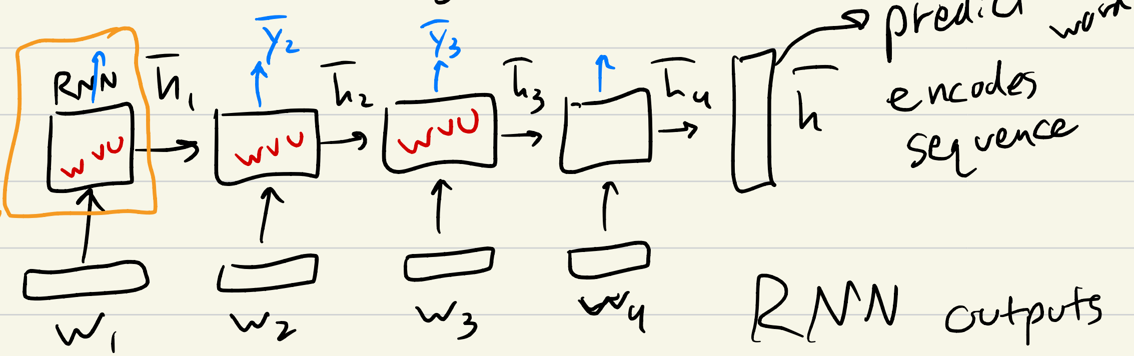
softmax: $P(w_i \mid \ldots)$ prob dist over $|V|$

FFNN doesn't scale to large n

$\bar{h} = \text{concat}(\bar{w}_{i-n+1}, \ldots, \bar{w}_{i-1})$ concat n-1 word embs

$W: |V| \times (d \cdot (n-1))$

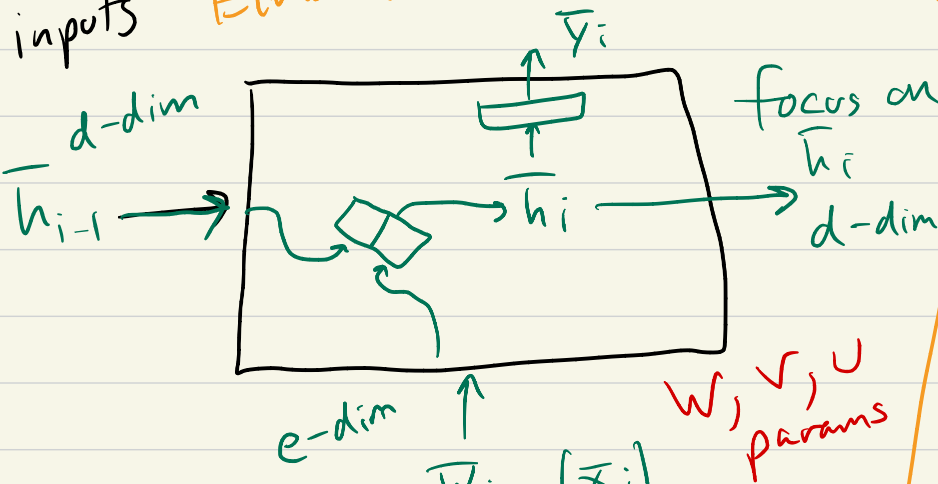
RNN encodes a sequence of vectors into a single vector "in order"



RNN outputs $\bar{y}_i, \bar{h}_i$ at each step

$\bar{h}_{\text{end of seq}} = \bar{h}$

inputs Elman network



$$\bar{h}_i = \tanh(W\bar{x}_i + V\bar{h}_{i-1})$$

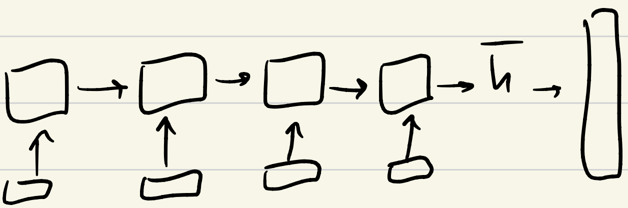
$$\bar{y}_i = \tanh(U\bar{h}_i)$$

Properties

Only params are W, V, U

Params don't depend on seq. len!

Copy-paste single RNN cell



$P(w_i \mid \cdots)$

differentiable!

Compute gradients for W, V, U
w/ backprop

Form loss $(-\log P(w_i \mid \cdots))$

Compute gradients