CS 378 Lecture 15: Language Modeling, RNNs

Today

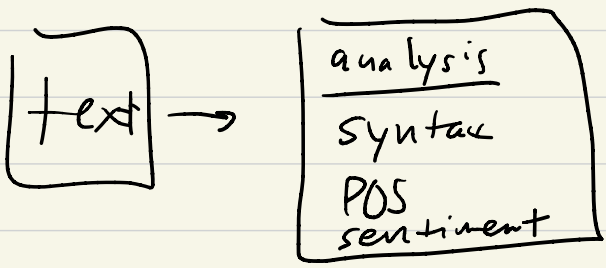
- Language modeling
- N-gram LMs
- Neural LMs
- (Start) RNNs

Announcements

- Midterm
- A4 + A5
- FP custom proposals

Where we are

First half of the course: analysis of text



input: text
output: structured analysis

same models! (or are they?)

Next ~3 weeks:

Task: text generation ☆

→ dialogue
→ summarization

Machine translation

input: text
output: text

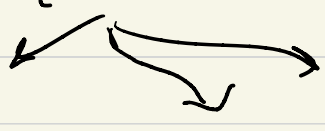
Language modeling: "autocomplete"
next word prediction

☆ Technique: recurrent neural networks (RNNs)
(and Transformers)

Final ~3 weeks: Question answering] BERT, Transformers

End of the course: miscellaneous applications + topics

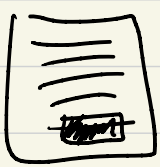
QA: question (text)



DB

freeform generation

Greg | Nov 9 :



Language Modeling

Imagine we want a distribution over grammatical sents. in a language

$P(\overline{w})$

PCFGs give us this

the cat ran ← higher prob
the the cat ran

Why?

Grammatical error correction:
I give you a sent $\overline{w}$ has a grammatical error
Then: $P(\overline{w})$ is low

Correct the error by finding $\overline{w}'$ close to $\overline{w}$ with $P(\overline{w}') >> P(\overline{w})$

Another application: machine translation

Translate $\overline{w}_1$ in lang $1 \Rightarrow \overline{w}_2$ in lang 2. Maybe we want to make $\overline{w}_2$ more natural
Find $\overline{w}_2'$ s.t. $P(\overline{w}_2') > P(\overline{w}_2)$

N-gram language models

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

True by chain rule of prob.

N-gram model: only look at the past n-1 words

$$P(\bar{w}) = P(w_1)\,P(w_2 \mid w_1) \cdots P(w_i \mid w_{i-n+1}, \ldots, w_{i-1})$$

2-gram model: $\bar{w}$ = the cat ran

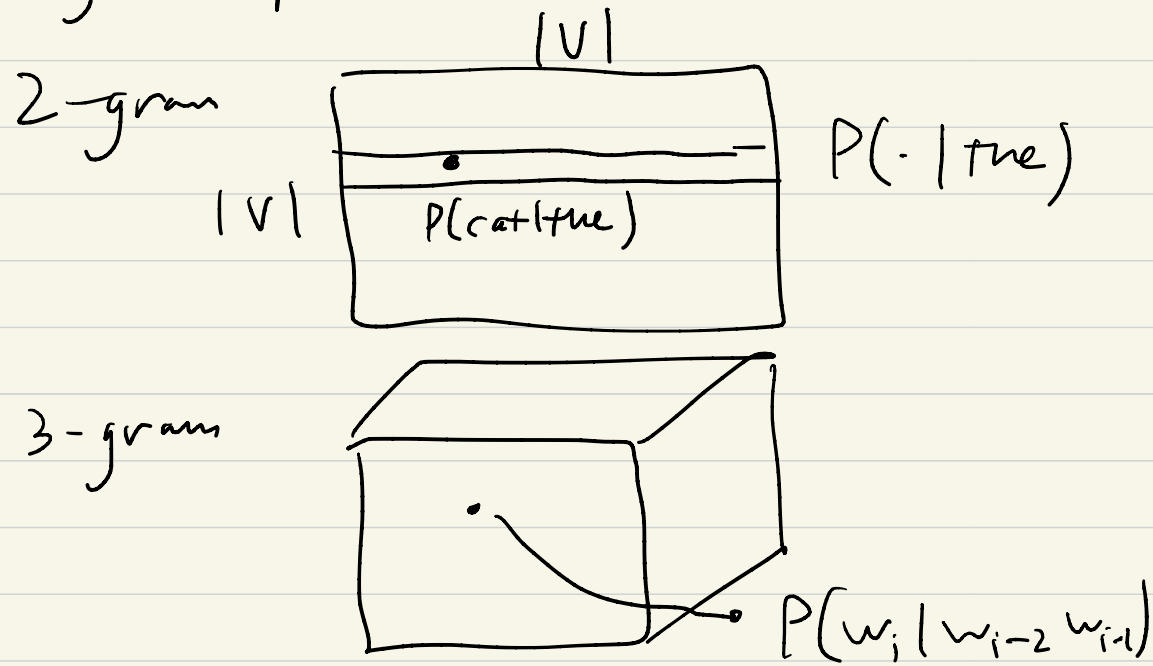
$$P_2(\bar{w}) = P(\text{the} \mid \langle S \rangle)\,P(\text{cat} \mid \text{the}) \cdot P(\text{ran} \mid \text{cat}) \cdot P(\text{STOP} \mid \text{ran})$$

($\langle S \rangle$: start of sent)

3-gram: $P(\text{the} \mid \langle S \rangle \langle S \rangle) \cdot P(\text{cat} \mid \langle S \rangle \text{the})\, P(\text{ran} \mid \text{the cat})\, P(\text{STOP} \mid \ldots)$

Parameters: transitions in HMM

Big lookup table



Get these params: count + normalize

5-gram: P(Maui | hate to go to) = 0

Nobody has said this before, but it's still reasonable!

Smoothing

Very important in n-gram LMs

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

$$\Rightarrow \lambda_1 P^{raw}(w_i \mid w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1})$$
$$+ \lambda_2 P^{raw}(w_i \mid w_{i-3}, w_{i-2}, w_{i-1}) \quad \text{4-gm!}$$
$$+ \lambda_3 P^{raw}(w_i \mid w_{i-2}\, w_{i-1})$$
$$+ \lambda_4 P^{raw}_{raw}(w_i \mid w_{i-1})$$
$$+ \lambda_5 P(w_i)$$

↳ > 0! $P(\text{Maui})$

$$\lambda_1 \cdot 0 + \lambda_2 \cdot 0 + \lambda_3 \cdot 10^{-4} + \lambda_2 \cdot 10^{-2} + \ldots$$

$\Rightarrow$ reasonable prob.

Set λ carefully so this is a valid prob. dist.

Neural language models

Use a neural net to model

$P(w_i | w_1, \ldots, w_{i-1})$

whole context, not just $n-1$ words

For now: simplify, consider

$P(w_i | w_{i-1})$

vectors

Skip-gram: $P(\text{context} | \text{word})$

One idea: take context $\xrightarrow{h}$ NN $\rightarrow P(w_i | w_{i-1})$

w_1 w_2 ... w_{i-1}

embed with pre-trained skip-gram vectors

$P(w_i | w_1, \ldots, w_{i-1})$ dist over $|V|$ words

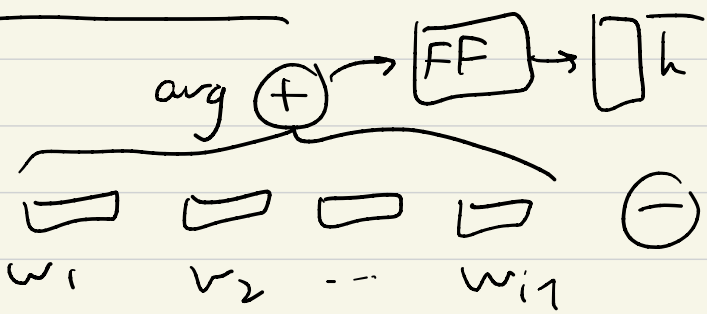
$\bar{h} = \text{neural net}(w_1, \ldots, w_{i-1})$ d-dim

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

W: param matrix $|V| \times d$

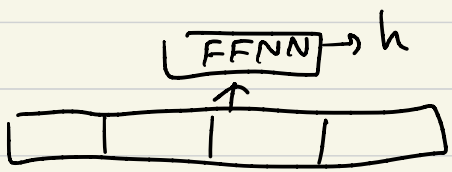
Ways to do this

DAN:



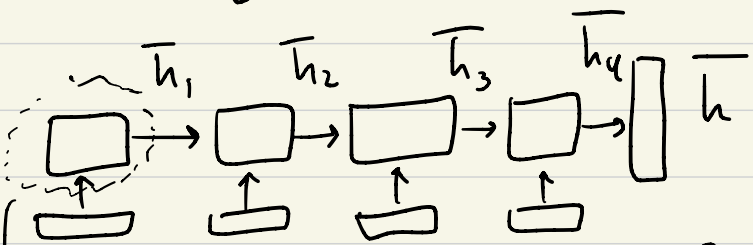
⊖ no ordering

FFNN: n-1 words 5-gram



⊖ doesn't scale to large n

RNN: encodes a sequence of arbitrary length into a single vector

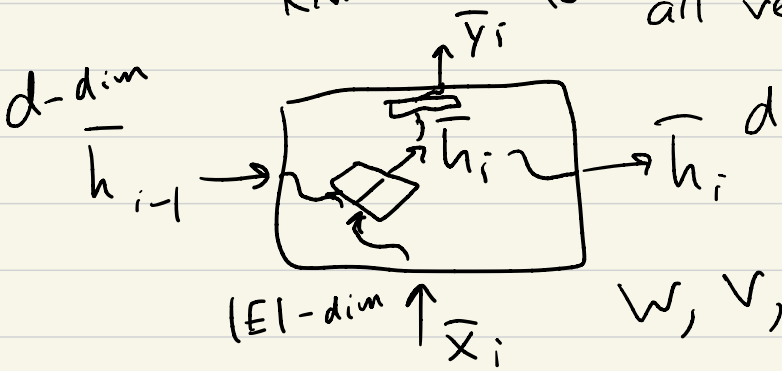


RNN outputs: $\bar{y}_i, \bar{h}_i$ at each step

Zoom in

RNN cell

101-dim all vectors



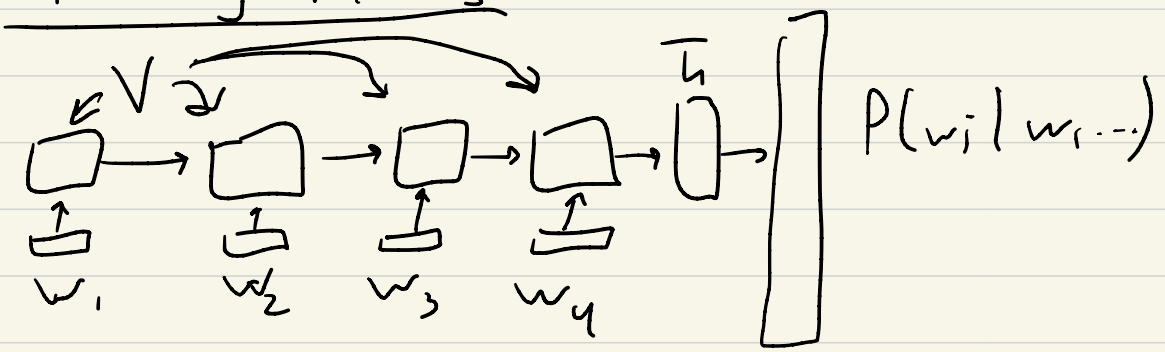
W, V, U params.

Elman network

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

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

Training RNNs



This is a differentiable computation!

Some layers share params