CS 378 Lecture 15:
Language Modeling, RNNs

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

Announcements
- Midterm, A3 grading
- AU
- FP Custom proposals (optional!)
  due Monday (changed)

Recap (so far)

Classify (sentiment)

Text \rightarrow POS
\rightarrow NER
\rightarrow Syntactic parse trees

Text \rightarrow Label or structure
Next few weeks: text -> text
(ex: machine translation) seq2seq models
dialogue summarization
... every thing?

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

\( \overline{w}_{t,1} \) two candidates. Check if \( P(\overline{w}_{t,1}) > P(\overline{w}_{t,2}) \) return higher
**N-gram language modeling**

By the chain rule of probability:

\[
P(\overline{w}) = P(w_1) P(w_2 | w_1) P(w_3 | w_1 w_2) \\
\cdots \\
P(w_n | w_1 w_2 \cdots w_{n-1})
\]

(not Markov assumption)

If we make an assumption:

only depend on past n-1 words

\[
P(\overline{w}) = \prod_{i=1}^{n} P(w_n | w_{i-n+1} \cdots w_{n-1})
\]

2-gram model: the cat ran

\[
P_2(\overline{w}) = P(\text{the} | \langle s \rangle) P(\text{cat} | \text{the}) P(\text{ran} | \text{cat}) P(\text{STOP} | \text{ran})
\]
3-gram: \( P(\text{the } <s> <s>) \ P(\text{cat} <s> \text{ the}) \)
\( P(\text{ran} (\text{the cat})) \)

Poll

I saw the dog —
wagging, barking, in, on, jump

I saw the dog. —

It, any word that starts a sentence

less restrictive context
N-gram parameterization \( V = \text{vocabulary} \)

Big lookup table (like HMM transitions)

2-gram

\[
P(cat|\text{the})
\]

3-gram

\[
P(w_i|w_{i-2} w_{i-1})
\]

Count + normalize over a big corpus
To do well, we need $n \geq 5$ (5-gram)

I hate to go to Maui

5 words

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

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

$\Rightarrow$ incorrect, should be $> 0$

**Smoothing** (in n-gram LMs)

$P_{5}(w_i \mid w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1})$ = raw counts

$\approx \lambda P_{5}^{\text{raw}}(w_i \mid w_{i-4}, \ldots, w_{i-1})$

$+ (1-\lambda) P_{n}(w_i \mid w_{i-3}, w_{i-2}, w_{i-1})$
\[ P_u = \lambda P_u^{\text{raw}} + (1-\lambda) P_3 \]

\[ P_5 = \lambda_1 P_5^{\text{raw}} + \lambda_2 P_u^{\text{raw}} + \lambda_3 P_3^{\text{raw}} + \lambda_4 P_2^{\text{raw}} + \lambda_5 P_1^{\text{raw}} \]

\[ P_1^{\text{raw}} = \frac{\text{count (Maui)}}{\text{size of corpus}} > 0 \]

\[ P_5 \text{ (Maui ...)} > 0 \]

Lots of very complex tricks here!
Neural Language Models

\[ P(w_i | w_1, \ldots, w_{i-1}) \Rightarrow \text{model w/a NN} \]

all prev words, not just \( n-1 \)

\( w_i \): next word

\[ T \rightarrow \text{FFNN} \]

\[ \text{reduce} \]

\[ w_1 \quad w_2 \quad w_3 \quad \ldots \quad w_{i-1} \]

GPT-3 reduce: Transformer

For us reduce: RNN

Also possible reduce: DAN

FFNN over \( n-1 \) words
\[ \vec{h} = \text{neural net}(w_1, \ldots, w_{i-1}) \]

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

\[ W: |V| \times d \text{ matrix} \]

weight matrix

multiclass w/ |V| classes

A simple ex: DAN no ordering

\[ \vec{h} = \text{avg}(\vec{w}_1, \ldots, \vec{w}_{i-1}) \]

\[ \text{d-dim vector} \]

\[ WH: |V| \text{-dim vector} \]

\[ \text{Softmax: } P(w_i | \cdot) \text{ prob dist over } |V| \]

FFNN doesn't scale to large n

\[ \vec{h} = \text{concat} (\vec{w}_{i-n+1}, \ldots, \vec{w}_{i-1}) \]

concat

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

n-1 word embs
RNN encodes a sequence of vectors into a single vector "in order" to predict the next word. RNN outputs tend of \( \hat{h} \), \( \hat{y} \), \( \hat{h} \) at each step.

Elman network:

\[
\bar{h}_i = \tanh (W_i x_i + V \bar{h}_{i-1}) \\
\bar{y}_i = \tanh (U \bar{h}_i)
\]
Properties

Only params are \( W, V, U \)
Params don't depend on seq. len!
Copy-paste single RNN cell

\[
\square \rightarrow \square \rightarrow \square \rightarrow \square \rightarrow h \rightarrow \prod P(w_{i+1})
\]

differentiable!

Compute gradients for \( W, V, U \)

w/ backprop

Form loss \((-\log P(w_{i+1}))\)

Compute gradients