Today
- Finish LSTMs and language modeling
- Visualizing LSTMs
- Machine translation
- Word alignment

Recap LSTMs

- \( \tilde{h}_1 \)
- \( c_1 \)
- \( \tilde{h}_2 \)
- \( c_2 \)
- \( \ldots \)
- \( \overline{X}_1 \)
- \( \overline{X}_2 \)

Main difference from Elman networks: gates

\[
\tilde{h}_i = \tilde{h}_{i-1} \Omega F + \ldots \text{ element-wise product vector } \in [0,1]^d
\]
Announcements
- Midterm
- Custom FP proposals
- Ay

Machine Translation
- Learn to translate from lang A into lang B

Input: source sentence S
Output: target sent T

Data: bitext. Set of sentences & their translations in the other lang.
Given a bitext, how do we learn to translate?

\[\text{Je fais un bureau} \mid \text{I make a desk}\]

\[\text{Qu'est-ce que tu fais?} \mid \text{What are you doing?}\]

\[\text{faire (Fr)} = \text{to make} \quad \text{to do}\]

1. Word mapping is not one-to-one
2. Some phrases need to translate as a unit

Step 1: Word alignment: discover word-level mapping between sentences in the bitext

Unsupervised learning problem
How to do MT?

French sentence → meaning → En Japanese → we don’t know what this is!

Bernard Vauquois (1968) interlingua

interlingua

semantic

syntactic

phrase-level

word-level

sweet spot
Phrase-based MT (this lecture + part of next) Then: neural MT

Bitext

\[ \text{word alignment} \]

Aligned bitext

\[ \text{extract phrases} \]

Phrase table

\[ \text{Decoder} \]

\{ Phrase T., this is a phrase translation pair \}

produce translation w/ high LM prob using high-scoring phrases
Word Alignment

Input: bitext, source sent $s$, target sent $t$

Output: Je fais un bureau

I am making a desk

Je vais le faire

I am going to do it

Alignments: one-to-many

Each word in $t$ aligns to 1 word in $s$
Define $\overline{a}$ as follows:

$a_i = \text{index in the source that the } i\text{th target word aligns to}$

$\overline{a} = 1/2 < 2 
\overline{s} = \text{Je vais le faire NULL}$

$\overline{t} = \text{I am going to do it}$

Alignment models: distribution $p(\overline{t}, \overline{a} | \overline{s})$, generative model of $\overline{t}$, $\overline{a}$

$\overline{t}$ like words in an HMM

$\overline{a}$ is like the tags
IBM Model 1 (1993) \( n \) target words

\[ \overline{a} = (a_1, \ldots, a_n) \quad \overline{t} = (t_1, \ldots, t_n) \]

\[ \overline{s} = (s_1, \ldots, s_m, \text{NULL}) \quad \text{in source words} \]

\[ P(\overline{t}, \overline{a} | \overline{s}) = \prod_{i=1}^{n} P(a_i) P(t_i | s_{a_i}) \]

Generative process: for each target word, pick a source index to align to \( P(a_i) = \text{uniform dist over} \ \frac{1}{m+1} \\text{options} \)

Generate \( t_i \) conditioned on \( s_{a_i} \)

the \( a_i \)th source word \( a \)
Model params: dictionary  
  target word

I do ---

\[ P(t | Je) \]
\[ P(t | \text{fais}) \]

Je  
\[ \text{fais} \]

Source:  
words

\( a_i \) is like a switch. Tells you which source word to condition on to generate the given target

\( a_1 = 1 \) vs. \( a_i = 2 \)

\[ P(I | Je) \quad P(I | \text{fais}) \] very low prob

very high prob