Sequence to Sequence Models for Machine Translation

CMSC 723 / LING 723 / INST 725

Marine Carpuat
Machine Translation

• Translation system
  • Input: source sentence F
  • Output: target sentence E
  • Can be viewed as a function

\[ \hat{E} = \text{mt}(F) \]

• Statistical machine translation systems

\[ \hat{E} = \underset{E}{\text{argmax}} \ P(E \mid F; \theta) \]

• 3 problems

• Modeling
  • how to define \( P(.) \)?

• Training/Learning
  • how to estimate parameters from parallel corpora?

• Search
  • How to solve argmax efficiently?
Introduction to Neural Machine Translation

- Neural language models review
- Sequence to sequence models for MT
  - Encoder-Decoder
  - Sampling and search (greedy vs beam search)
  - Practical tricks
- Sequence to sequence models for other NLP tasks
A feedforward neural 3-gram model

\[ m = \text{concat}(M_{.,e_{t-2}}, M_{.,e_{t-1}}) \]
\[ h = \tanh(W_{mh}m + b_h) \]
\[ s = W_{hs}h + b_s \]
\[ p = \text{softmax}(s) \]
A recurrent language model

\[ m_t = M_{eq} e_{t-1} \]

\[ h_t = \begin{cases} 
\tanh(W_{mh} m_t + W_{hh} h_{t-1} + b_h) & t \geq 1, \\
0 & \text{otherwise.} 
\end{cases} \]

\[ p_t = \text{softmax}(W_{hs} h_t + b_s). \]
A recurrent language model

\[ m_t = M_{t:e_t-1} \]
\[ h_t = \text{RNN}(m_t, h_{t-1}) \]
\[ p_t = \text{softmax}(W_{hs}h_t + b_s). \]
Examples of RNN variants

• **LSTMs**
  • Aim to address vanishing/exploding gradient issue

• **Stacked RNNs**

• ...
Training in practice: online

Algorithm 1 A fully online training algorithm

1: procedure ONLINE
2:   for several epochs of training do
3:     for each training example in the data do
4:       Calculate gradients of the loss
5:       Update the parameters according to this gradient
6:   end for
7: end for
8: end procedure
Training in practice: batch

Algorithm 2 A batch learning algorithm

1: procedure BATCH
2:   for several epochs of training do
3:     for each training example in the data do
4:       Calculate and accumulate gradients of the loss
5:     end for
6:   Update the parameters according to the accumulated gradient
7: end for
8: end procedure
Training in practice: minibatch

- Compromise between online and batch

- Computational advantages
  - Can leverage vector processing instructions in modern hardware
  - By processing multiple examples simultaneously

Operations w/o Minibatching
\[
\begin{align*}
\text{tanh}(W x_1 + b) & \quad \text{tanh}(W x_2 + b) & \quad \text{tanh}(W x_3 + b)
\end{align*}
\]

Operations with Minibatching
\[
\begin{align*}
x_1 x_2 x_3 & \quad \text{concat} & \quad \text{broadcast} & \quad b
\end{align*}
\]
\[
\text{tanh}(W X + B)
\]
Problem with minibatches: in language modeling, examples don’t have the same length

- 3 tricks
  - Padding
    - Add </s> symbol to make all sentences same length
  - Masking
    - Multiply loss function calculated over padded symbols by zero
  - + sort sentences by length
Introduction to Neural Machine Translation

• Neural language models review

• Sequence to sequence models for MT
  • Encoder-Decoder
  • Sampling and search (greedy vs beam search)
  • Training tricks

• Sequence to sequence models for other NLP tasks
Encoder-decoder model
Encoder-decoder model

\[ m_t^{(f)} = M_{:,f_t} \]

\[ h_t^{(f)} = \begin{cases} 
\text{RNN}^{(f)}(m_t^{(f)}, h_{t-1}^{(f)}) & t \geq 1, \\
0 & \text{otherwise}.
\end{cases} \]

\[ m_t^{(e)} = M_{:,c_{t-1}} \]

\[ h_t^{(e)} = \begin{cases} 
\text{RNN}^{(e)}(m_t^{(e)}, h_{t-1}^{(e)}) & t \geq 1, \\
 h_t^{(f)} & \text{otherwise}.
\end{cases} \]

\[ p_t^{(e)} = \text{softmax}(W_{hs} h_t^{(e)} + b_s) \]
Generating Output

• We have a model $P(E|F)$, how can we generate translations?

• 2 methods

  • **Sampling**: generate a random sentence according to probability distribution

  • **Argmax**: generate sentence with highest probability
Ancestral Sampling

- Randomly generate words one by one
- Until end of sentence symbol
- Done!

\[
\text{while } y_{j-1} \neq "<\text{s}>": \\
y_j \sim P(y_j \mid X, y_1, \ldots, y_{j-1})
\]
Greedy search

• One by one, pick single highest probability word

• Problems
  • Often generates easy words first
  • Often prefers multiple common words to rare words

```
while y_{j-1} != "</s>":
y_j = \arg\max P(y_j | X, y_1, \ldots, y_{j-1})
```
Greedy Search

Example
Beam Search

Example with beam size b = 2

We consider b top hypotheses at each time step.
Introduction to Neural Machine Translation

• Neural language models review

• Sequence to sequence models for MT
  • Encoder-Decoder
  • Sampling and search (greedy vs beam search)
  • Practical tricks

• Sequence to sequence models for other NLP tasks