Recurrent Language Models

CMSC 470
Marine Carpuat
Toward a Neural Language Model

Figures by Philipp Koehn (JHU)
Count-based n-gram models vs. feedforward neural networks

- Pros of feedforward neural LM
  - Word embeddings capture generalizations across word types

- Cons of feedforward neural LM
  - Closed vocabulary
  - Training/testing is more computationally expensive

- Weaknesses of both types of model
  - Only work well for word prediction if the test corpus looks like the training corpus
  - Only capture short distance context
Language Modeling with Recurrent Neural Networks

Figure by Philipp Koehn
Recurrent Neural Networks (RNN)

The hidden layer includes a recurrent connection as part of its input

Unrolling the RNN over the time sequence as a feed-forward network

The hidden layer from the previous time step plays the role of memory, remembering earlier context

Figures from Jurafsky & Martin
Unrolled RNN illustrated

weights U, V, W are shared across all timesteps
Prediction/Inference with RNNs

**function** `FORWARDRNN(x, network)` **returns** output sequence $y$

$h_0 \leftarrow 0$

for $i \leftarrow 1$ to `LENGTH(x)` do

$h_i \leftarrow g(U h_{i-1} + W x_i)$

$y_i \leftarrow f(V h_i)$

**return** $y$

For language modeling, $f = \text{softmax}$ function to provide normalized probability distribution over possible output classes.
Training RNNs with backpropagation

- Training goal: estimate parameter values for U, V, W
- Use same loss as for feedforward language models
- Given unrolled network, run forward and backpropagation algorithms as usual
Training RNNs with backpropagation
Practical Training Issues: minibatch

- Compute parameter updates based on a "minibatch" of examples
  - instead of using one example at a time
- More efficient
  - matrix-matrix operations as opposed to multiple matrix-vector operations
- Can lead to better model parameters
  - middle ground between online and batch training

Figure by Graham Neubig
Practical Training Issues: vanishing/exploding gradients

Multiple ways to work around this problem:

- ReLU activations help
- Dedicated RNN architecture (Long Short Term Memory Networks)

Figure 16: An example of the vanishing gradient problem.

Figure by Graham Neubig
Aside: Long Short Term Memory Networks
What do Recurrent Language Models Learn?

Figure from Karpathy 2015
What do Recurrent Language Models Learn?

Figure from Karpathy 2015
What do Recurrent Language Models Learn?

• Parameters are hard to interpret, so we can gain insights by analyzing their output behavior instead

• Can capture (some) long-distance dependencies

After much economic progress over the years, the country has...

The country, which has made much economic progress over the years, still has...
Recurrent neural network language models

• Have all the strengths of feedforward language model

• And do a better job at modeling long distance context

• However
  • Training is trickier due to vanishing/exploding gradients
  • Performance on test sets is still sensitive to distance from training data