Recitation 6: RNN Basics
Joseph Konan and Kinori Rosnow
Sequential Data

Sequential Dependence - outcome depends on previous/next steps of sequence

Finish the lyric:
“Never gonna to give you up. Never gonna to let you ______.”

Fill in the blank:
“Hey, I just met you and this is ______, but here’s my number. So call me, maybe”

What is this word:
“heh-low-wer-l-d”

Other Tasks:
- speech transcription
- text generation
- ...
Data Types

one to one

one to many

many to one

many to many

https://i.stack.imgur.com/b4sus.jpg
Recurrent Neural Networks

- Cyclic connection
  - Previous step informs next step
- Differs from CNN
  - Considers sequential dependencies
- Architectural variants to address certain issues

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
RNN Unrolled - through time/steps
RNNs Unrolled

\[ h_t \quad \text{=} \quad A \quad \begin{array}{l} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_t \end{array} \quad \begin{array}{l} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_t \end{array} \]

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Text vector representation

- **Input: One hot encoding**
  - “Never going to give you up” \{N=6\}
    - One Hot Encoding: \(\text{Never} = [1, 0, 0, 0, 0, 0]\)

- **Input/Post-processing: Word embedding**
  - Efficient use of space (denser)
  - Can represent relationships (information rich)

- **Output: Probability Distribution**
  - “Never going to give you \text{up}” \{N=6\}
    - \([\text{Never, going, to, give, you, up}]\)
    - \(P(w) = [0.01, 0.01, 0.03, 0.04, 0.05, 0.86]\)
  - “Never going to give you up. Never going to let you \text{down}.” \{N=8\}
    - \([\text{Never, going, to, give, you, up, let, down}]\)
    - \(P(w) = [0.01, 0.01, 0.02, 0.03, 0.03, 0.43, 0.03, 0.44]\)

https://nlp.stanford.edu/projects/glove/
Prediction Example

H0 = \text{Never}
H1 = p(w1 \mid H0, \text{Never})
H2 = p(w2 \mid H1, \text{going})
H3 = p(w3 \mid H2, \text{to})
H4 = p(w4 \mid H3, \text{give})
H5 = p(w5 \mid H4, \text{you})

Never going to give you ___
Never going to give you up
Generation Example

\[ H_0 = p(w_1 \mid H_0, \text{Never}) \]
\[ H_1 = p(w_2 \mid H_1, \text{going}) \]
\[ H_2 = p(w_3 \mid H_2, \text{to}) \]
\[ H_3 = p(w_4 \mid H_3, \text{give}) \]
\[ H_4 = p(w_5 \mid H_4, \text{you}) \]

\text{Never going __ _____ __ __}

\text{Never going to give you up}
Backpropagation
Problems Training

● After many iterations
  ○ Short term memory
  ○ Vanishing Gradients
  ○ LSTMs and GRUs combat these issues

● Early training for tasks like generation
  ○ Cold start - can use teacher forcing
  ○ Lack of exploration - can use noise

● Long term/non-local dependencies may be reduced or lost
  ○ Attention

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Resources

1. http://colah.github.io/posts/2015-08-Understanding-LSTMs/
2. Vanishing/Exploding Gradients (C2W1L10) - YouTube - https://www.youtube.com/watch?v=qhXZsFVxGKo