Developing a Neural Natural Language Generation Approach for a Medical Spoken Dialogue System with PyTorch

Sarah Days-Merrill
Mentor: Dr. Joohyun Kim
CCT REU 2018
Terminology

- Natural Language Generation
  - Convert data into sequence of human language
  - Focus on “how to say” given “what to say”
- Deep Learning
  - Scalable with large data
  - Representation learning
  - Examples: DNN, CNN, RNN
Recurrent Neural Network – based Seq2Seq Model

- Model: multi-layers (input, hidden state, output)
- Hidden states hold the memory from past inputs
Baseline Model

- Encoder-Decoder with Attention
  - (Encoder) Embeds input with word2vec
    - DA type, slot-value pairs, words
  - (Decoder) Takes embedding, runs through Attention and LSTM, and outputs text
(Soft) Attention

- Most relevant information for decoding the output
- Find weights to multiply

\[
\beta_{t,i} = q^T \tanh(W_h m h_{t-1} + W_m m z_i)
\]

\[
\omega_{t,i} = e^{\beta_{t,i}} / \sum_i e^{\beta_{t,i}}
\]

\[
d_t = a \oplus \sum_i \omega_{t,i} z_i
\]
Long Short-Term Memory

- Inner Architecture of the hidden states using LSTM
- 3 gates: forget gate, input gate, output gate
- Prevents vanishing and exploding gradient
- Allows for longer control of memory within hidden states
Training the Model

- Process input – Dialogue Act type and Slot-Value pairs
- Compute the loss function
- Calculate the gradient with Backpropagation
- Update the weights – using Stochastic Gradient Descent
  - Find the minimum of the loss function where the derivative is zero or convergence criterion

SGD formula: $\omega \leftarrow \omega - \eta \nabla L(\omega)$
RNNLG Reference Code

- https://github.com/shawnwun/RNNLG
  - Written using Python Theano library
- Goals:
  - Rewrite the code using PyTorch library
  - Organize file structure
Theano vs PyTorch

- Theano
  - Early programming libraries for Deep Learning
  - Uses NumPy arrays
  - Explicitly define parameters for optimization

- PyTorch
  - Newer – released January 2017
  - Uses Tensors
  - Built-in functions
    - Parameters defined behind the scenes
My Contribution

- GitHub repository at https://github.com/sdaysmerrill/Summer-REU-Project/tree/master/Devel
- Create beginning workings of the program
  - Convert reference code written in Theano to PyTorch
  - Construct separate files for data preparation, model, and training
  - Once training is complete, create visualization for the loss function
Future Work

1. Optimize the Encoder-Decoder model with standard techniques such as dropout, batch normalization, etc.
2. Apply techniques for better training such as scheduled sampling, copying mechanisms, and decoding with the Monte Carlo Tree Search approach
3. Implement Medical training dataset using MIMIC-III
4. Incorporate grammar-based RNN for syntactically correct sequence generation
Acknowledgements

- Dr. Joohyun Kim
- Louisiana State University
- CCT REU
- Work supported by the National Science Foundation (NSF) award #ACI-1560410.
References


Questions?

Thank you!