Journal Report 6
10/7/19 – 10/13/19
Valerie Nayak
Computer Systems Research Lab
Period 2, White

Daily Log

Monday October 7

Today, I continued the lecture on sentence probability and word prediction using RNNs. I learned that RNNs carry the same weight vector throughout and you accumulate the loss over the model rather than having separate weights and losses. This makes an RNN good for language models, because you're applying the same weights to each word. Using a trained language model, you can generate a sentence (or multiple sentences) in the style of the training corpus. For example, the lecture showed us some generated text in the style of the Harry Potter books. Next class, I plan on finishing up this lecture and working more on using PyTorch for training my dependency parsing model.

Tuesday October 8

I finished the lecture on RNNs, sentence probability, and sentence classification. I got PyTorch properly installed (thanks, Mr. White!) and have been working on implementing the neural network for my dependency parsing model. I've found that using PyTorch is almost exclusively API calls which I'm not used to yet, so I'm still in the process of figuring out the implementation process and getting used to the research involved in going through the documentation.

Thursday October 10

Today, I watched the 7th lecture video of my online class, which covers the vanishing gradient problem and more complex variants of RNNs. (The simpler RNN I learned about on Tuesday is known as a "vanilla RNN".)

The vanishing or gradient problem is when the gradients for a neural network are small (less than 1), so the gradients get exponentially smaller as you backpropagate. This problem means that RNNs aren't good at forming associations with words that were further back in a sentence because the gradients from those effectively "vanish" in a large network. The opposite is the exploding gradient problem, in which the gradients are larger than 1 and grow exponentially. This is also problematic because if the gradients are too large, your model will take large steps and could significantly increase your loss.

The lecture also talked about established solutions to the vanishing and exploding gradient problems. Exploding gradients can be fixed with gradient clipping, where you scale down a gradient if its magnitude exceeds a certain value. The vanishing gradient problem can be improved with different types of RNNs – LSTMs and GRUs, which are better at retaining information long-term.

Timeline

Date	Goal	Met
Sunday Oct 6	Complete sentence classification methods with neural networks and word windows. Get started on week 4 materials with RNNs.	I'm still finishing up the dependency parsing because of the aforementioned issues I had with PyTorch. I've also in the middle of the sixth lecture video.
Sunday Oct 13	Finish my dependency parsing neural network model. Work on week 4 materials with RNNs for sentence classification.	Finished dependency parsing and almost finished watching week 4 lectures.
Sunday Oct 20	Train my first RNN model working for translation tasks. Finish the 4th week of lecture videos if I don't finish that in the previous week.	
Sunday Oct 27	Finish training the RNN model for translation and work on week 5 lecture videos. Start reading literature for my question answering project.	

Reflection

In the next one to two weeks, I'll need to start using a GPU accelerator for my model training, as the models will now become too large to train on my own machine.

In working on the dependency parsing model with PyTorch, I've found that most of the coding is based on API calls. It's great and convenient that all these methods have been implemented for me, but I'm also not used to using this API yet. While working on the model, I spent a lot of time trying to understand which methods do what – which data structures they use and how/when I'm supposed to call them – but I think I'll get more used to it with time and practice. I'm hoping the next task, the translation model, will help me get more used to it.