EECS 16ML Introduction to RNNs
Fall 2020 Team Chairun Quiz

RNN Quiz

1. For which of these tasks would you use an RNN, as opposed to a feed forward network?

   (a) Sentiment Classification for Text
   (b) Image Classification, i.e. classify an image as cat or not cat
   (c) Audio transcription, i.e. take an audio file and convert it to text
   (d) Predicting changes in the business cycle (i.e. boom or bust phases)

   Answer: Sentiment Classification and Predicting changes in the business cycle. Both are sequence style problems that make use of other terms in the sequence to predict any given. Image classification and audio transcription (as defined in the options above) on the other hand do not have this term dependence.

2. You are given a many to many RNN, that has output $y^{\langle t\rangle}$ at time step t. At the t-th time step, what is the RNN estimating from a probabilistic point of view?

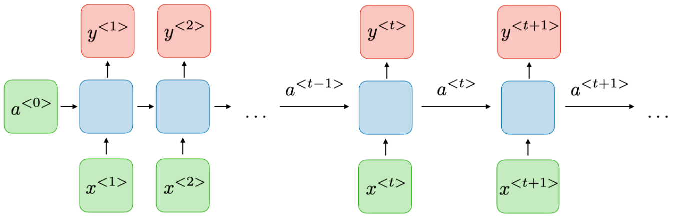

Figure 1: Traditional RNN architecture

   Answer: The RNN estimates $\mathbb{P}(y^{\langle t\rangle} \mid y^{\langle 1\rangle}, \ldots, y^{\langle t-1\rangle})$.

3. You have seen how the conventional RNN architecture is prone to the vanishing/exploding gradient problem due to backpropagation, and how GRUs prevent this from happening. Intuitively speaking, is there an activation function from the ones you know of (sigmoid, tanh, ReLU, softmax) that does the same, and why?

   Answer: the ReLU activation function – it is defined as $\max(0, x)$ for a given input x, which means that for a given input x, it does not try and "squash" it into a smaller domain, for example, sigmoid and softmax both change the range to [0, 1]. This repeated squashing in a neural network can also lead to excessive minimization, which causes a vanishing gradient.

4. Imagine you are given a sequence such that $a^{\langle t\rangle} = a^{\langle t+1\rangle} \cdot k + b$, that is, future terms determine prior terms. Explain how would use an RNN for predicting terms in the sequence without using the

bidirectional architecture.

Answer: One could reverse the input and pass into an RNN, essentially learning the backward connections in the conventional manner.

5. How can adding an $L2$ norm regularization term to layer weights affect an RNN? If it helps, consider extreme examples to make your case.

   Answer: Adding $L2$ regularization penalizes high values, and in extreme cases (oversmoothing), the weights become too small, which can exacerbate the vanishing gradient problem that RNNs are already prone to.

6. Can our Bi-GRU based stock price classifier be used for real time stock predictions?

   Answer: No, we constructed our model to be Bidirectional, that is, we need inputs from both the "left" and the "right" side, which means that we cannot use it to make live predictions. This is true for any Bidirectional architecture.

7. Explain how you would unroll an RNN through time. What neural network does the unrolled RNN look like? Provide a simple example to support your answer.

   Answer: An unrolled RNN will look like a very deep feedforward neural network. Unrolling an RNN is done by representing each state (at some given timestep) as a separate layer.

8. What is the primary difference between conventional backpropagation and backpropagation through time for RNNs?

   Answer: In an RNN, total prediction error/loss for a sequence is given by the sum of errors of each individual term in the sequence. For example, if the sequence was a sentence, then each timestep would be a word. Hence, in Backpropagation through time (BPTT), we calculate overall gradient of the loss function with respect to the weight matrices by summing gradients at each time step.

9. How would you deal with an exploding gradient problem in an RNN without changing the architecture of the network?

   Answer: As in the case of vanishing gradients, where we used ReLU to prevent values from getting too small, in the case of exploding gradients we can use something called gradient clipping – i.e.

setting a maximum threshold for gradient value at each iteration, preventing any massive descents from taking place.

10. Explain why the LSTM performed better than the simple RNN in the first section of the project notebook.

    Answer: Since the problem involved cumulative sum at each term in the sequence, the classification was dependent on every term prior to any given term. Essentially, the Long term aspect of LSTMs allows the model to better remember earlier terms, which especially impacts sequences where high baggage weights show up in the beginning of the sequence.