Andrew Elysee (aelys2)

CS446

Homework 6

1. 1. Lets define to be the expected number of steps taken to get from to . Assume we have made it to state then there is a 50% chance that we either need 1 more step giving us if we take or a 50% chance we need 1 more step plus another steps if we take giving us thus we can define recursively to be:  
      To get a non-recursive definition, we can unroll this:  
      The base case can easily be reasoned using the geometric distribution (the expected value of the geometric distribution is 1/p which is 0.5 in this case). Also, .  
      Thus, the expected number of steps to go from to is steps.  
        
      **SOURCE: https://www.youtube.com/watch?v=2PtrzCEjBTs**
   2. The general formula given in class for discounted reward setting is:  
      For our base case, let’s consider what happens when ,   
      Since there is a 50% chance that we take or , the expected reward for the remained steps would be . And we can also use the closed form for the sum of an infinite geometric series   
      Now for the more general case:  
      We first need to consider how many more steps it will take to reach because for this many steps we are guaranteed to have 0 reward based on the problem definition luckily we can do this using our function in 1a . The probability that we are successfully able to reach n from our current location is mean there is a probability we need to take steps. So, our expected number of steps to reach is , this can probably be further simplified but I can’t of how. Using this information we get:  
      Once again, we can use the sum of infinite geometric series:  
      It is important to note that this rule would not work for because there are only self-loops on   
      For the general case for this is a lot simpler because we are being reset back to the beginning so the expected number of steps is . Thus, we get   
      Once again, It is important to note that this rule would not work for because there are only self-loops on .  
      We can combine this into a piece-wise function giving us that:
   3. For we have 2 different situations to consider, when and .

First when , it is important to note that because ,, thus, . Therefore, when , because . For our other case is , since this means that raising it to a greater power will make it smaller, (this is a basic property of numbers in this range so I don’t think I need to prove it) so because is less than for all (this is because ) then when .

* 1. Given this new greedy policy of , the expected new number of steps to get from to would be . This is because thus, the argmax would resolve with taking every time progressing through the states from never resetting back to .

1. 1. In python file
   2. In python file
   3. For kernel of size 10 the model is able to provide significantly better accuracies than a kernel size of 3. I believe this is because there is probably some relation between the beginning of the sequence and the end of the sequence that makes it so that a kernel size of 3 is incapable of seeing that the 2 are correlated because the window is too small.  
        
      GRAPH ON NEXT PAGE  
        
      Chart, line chart

      Description automatically generated
   4. It appears that the attention without positional encoding was unable to properly capture the data. It appears that without the position data it was incapable of identifying where the key features of the data were at.   
      Chart, line chart

      Description automatically generated
   5. It appears that almost all the sequences are mapped to the the very last value of the sequences with little influence coming from the other 9 positions of the sequence. Attentions are different from convolutions because they can relate non-continuous/local parts of the data while convolutions can only relate parts of the data that are local to each other within the kernel size. This explains why the convolutional network with kernel size 3 was able to see that the last element had so much influence.   
         
      A picture containing chart

      Description automatically generated