Andrew Elysee (aelys2)
CS446

Homework 6

1.

a. Lets define $f(n)$ to be the expected number of steps taken to get from s_1 to s_n. Assume we have made it to state s_{n-1} then there is a 50% chance that we either need 1 more step giving us $f(n-1)+1$ if we take a_1 or a 50% chance we need 1 more step plus another $f(n)$ steps if we take a_2 giving us $f(n-1)+f(n)+1$ thus we can define $f(n)$ recursively to be:

$$f(n) = \frac{1}{2}(f(n-1)+1) + \frac{1}{2}(f(n-1)+f(n)+1\ )$$
$$2 \times f(n) = f(n-1)+1+f(n-1)+f(n)+1$$
$$f(n) = 2 \times f(n-1)+2$$

To get a non-recursive definition, we can unroll this:

$$f(n) = 2(2(2(2f(n-4)+2)+2)+2)+2$$
$$f(n) = 2^{n-1} * f(1) + (2^{n-1}+2^{n-2}+\cdots+2^1)$$

The base case $f(1)=2$ 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, $(2^{n-1}+2^{n-2}+\cdots+2^1) = 2^n - 2$.

$$f(n) = 2^{n-1} \times 2 + 2^n - 2 = 2^n + 2^n - 2 = 2 \times 2^n - 2 = 2^{n+1} - 2$$

Thus, the expected number of steps to go from s_1 to s_n is $2^{n+1}-2$ steps.

SOURCE: https://www.youtube.com/watch?v=2PtrzCEjBTs

b. The general formula given in class for discounted reward setting is:

$$Q(s,a) = (1-\gamma)\mathbb{E}_{s_0=s,a_0=a\sim\pi} \sum_{i\geq 0} r_i \gamma^i$$

For our base case, let's consider what happens when $s = s_n$,

$$Q(s_n,a_1) = (1-\lambda)\mathbb{E}\left(1+\sum_{i\geq 1} r_i\gamma^i\right)$$
$$Q(s_n,a_2) = (1-\gamma)\mathbb{E}\left(\sum_{i\geq 1} r_i\gamma^i\right)$$

Since there is a 50% chance that we take a_1 or a_2, the expected reward for the remained steps would be $\frac{1}{2}$. And we can also use the closed form for the sum of an infinite geometric series

$$Q(s_n,a_1) = (1-\lambda)\left(1+\frac{\frac{1}{2}\gamma}{1-\gamma}\right) = 1-\gamma+\frac{\frac{\gamma}{2}(1-\gamma)}{1-\gamma} = 1-\frac{\gamma}{2}$$
$$Q(s_n,a_2) = (1-\gamma)\left(\frac{\frac{1}{2}\gamma}{1-\gamma}\right) = \frac{\frac{\gamma}{2}(1-\gamma)}{1-\gamma} = \frac{\gamma}{2}$$

Now for the more general case:

$$Q(s_i, a_1) = (1-\gamma)\mathbb{E}\left(\sum_{i\geq 0} r_i \gamma^i\right)$$

We first need to consider how many more steps it will take to reach s_n 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 $f(n) = 2^{n+1} - 2$. The probability that we are successfully able to reach n from our current location is $\frac{1}{2}^{n-i-1}$ mean there is a $1 - \frac{1}{2}^{n-i-1}$ probability we need to take $2^{n+1} - 2$ steps. So, our expected number of steps to reach s^n is $\left(1 - \frac{1}{2}^{n-i-1}\right)(2^{n+1} - 2\,)$, this can probably be further simplified but I can't of how. Using this information we get:

$$Q(s_i, a_1) = (1-\gamma)\left(0\left(1 - \frac{1}{2}^{n-i-1}\right)(2^{n+1} - 2\,) + \sum_{i \geq \left(1 - \frac{1}{2}^{n-i}\right)(2^{n+1}-2\,)} \frac{1}{2}\gamma^i\right)$$

$$= (1-\gamma)\left(\sum_{i \geq \left(1 - \frac{1}{2}^{n-i}\right)(2^{n+1}-2\,)} \frac{1}{2}\gamma^i\right)$$

Once again, we can use the sum of infinite geometric series:

$$Q(s_i, a_1) = (1-\gamma)\left(\frac{\frac{1}{2}\gamma^{\left(1-\frac{1}{2}^{n-i}\right)(2^{n+1}-2\,)}}{1-\gamma}\right) = \frac{1}{2}\gamma^{\left(1-\frac{1}{2}^{n-i-1}\right)(2^{n+1}-2\,)}$$

$$Q(s_i, a_1) = \frac{1}{2}\gamma^{\left(1-\frac{1}{2}^{n-i}\right)(2^{n+1}-2\,)}$$

It is important to note that this rule would not work for $i = n$ because there are only self-loops on s_n

For the general case for a_2 this is a lot simpler because we are being reset back to the beginning so the expected number of steps is $2^{n+1} - 2$. Thus, we get

$$Q(s_i, a_2) = (1-\gamma)\mathbb{E}\left(\sum_{i\geq 0} r_i \gamma^i\right)$$

$$Q(s_i, a_2) = (1-y)\left(0(2^{n+1}-2) + \sum_{i \geq 2^{n+1}-2} \frac{1}{2}\gamma^i\right) = (1-y)(\sum_{i \geq 2^{n+1}-2} \frac{1}{2}\gamma^i)$$

$$Q(s_i, a_2) = (1-y)\left(\frac{\frac{1}{2}\gamma^{(2^{n+1}-2\,)}}{1-\gamma}\right) = \frac{1}{2}\gamma^{(2^{n+1}-2\,)}$$

$$Q(s_i, a_2) = \frac{1}{2}\gamma^{(2^{n+1}-2\,)}$$

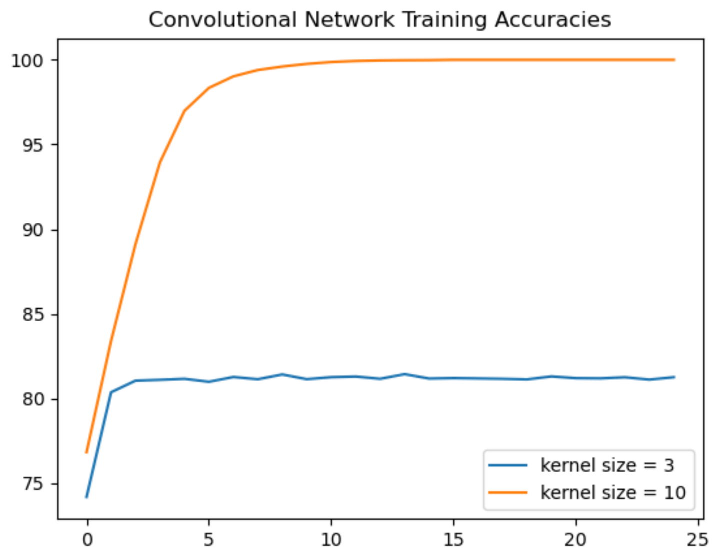
Once again, It is important to note that this rule would not work for $i = n$ because there are only self-loops on s_n.
We can combine this into a piece-wise function giving us that:

$$Q(s_i, a_j) = \begin{cases} 1 - \frac{\gamma}{2} \quad If\ i = n\ and\ j = 1 \\ \frac{\gamma}{2} \quad If\ i = n\ and\ j = 2 \\ \frac{1}{2}\gamma^{\left(1-\frac{1}{2}^{n-i}\right)(2^{n+1}-2)} \quad If\ i \neq n\ and\ j = 1 \\ \frac{1}{2}\gamma^{(2^{n+1}-2)} \quad If\ i \neq n\ and\ j = 2 \end{cases}$$

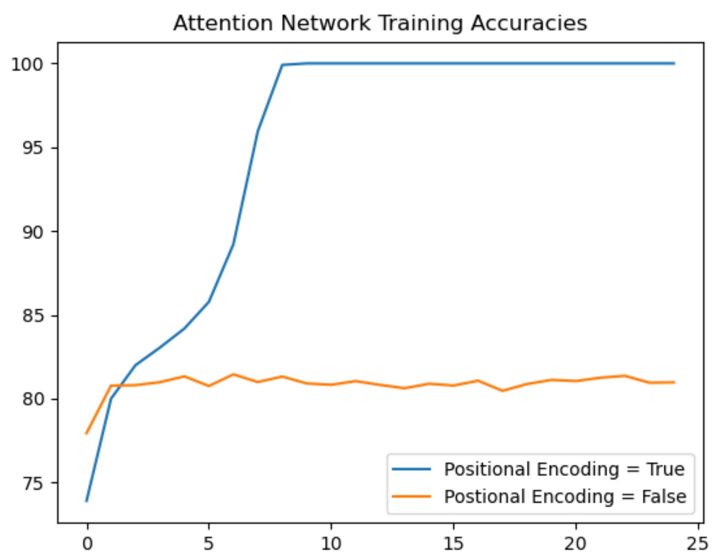
c. For we have 2 different situations to consider, when $i = n$ and $i \neq n$.
   First when $i = n$, it is important to note that because $\gamma \in [0,1]$,$\frac{\gamma}{2} \leq \frac{1}{2}$, thus, $1 - \frac{\gamma}{2} > \frac{1}{2}$. Therefore, when $i = n$, $Q(s_n, a_1) > Q(s_n, a_2)$ because $1 - \frac{\gamma}{2} > \frac{\gamma}{2}$. For our other case is $i \neq n$, since $\gamma \in [0,1]$ 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 $\left(1 - \frac{1}{2}^{n-i-1}\right)(2^{n+1} - 2\ )$ is less than $(2^{n+1} - 2\ )$ for all $i < n$ (this is because $1 - \frac{1}{2}^{n-i-1} \leq\ 1 - \frac{1}{2}^{n-(n-1)} = 1 - \frac{1}{2}^{1} = \frac{1}{2}$) then $Q(s_i, a_1) > Q(s_i, a_2)$ when $i \neq n$.
d. Given this new greedy policy of $\pi(s_i) = argmax_a Q(s_i, a)$, the expected new number of steps to get from s_1 to s_k would be $k - 1$. This is because $\forall i < n :$ $Q(s_i, a_1) > Q(s_i, a_2)$ thus, the argmax would resolve with taking a_1 every time progressing through the states from $s_1 \to s_2 \to \cdots \to s_k$ never resetting back to s_1.

2.
   a. In python file
   b. In python file
   c. 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.

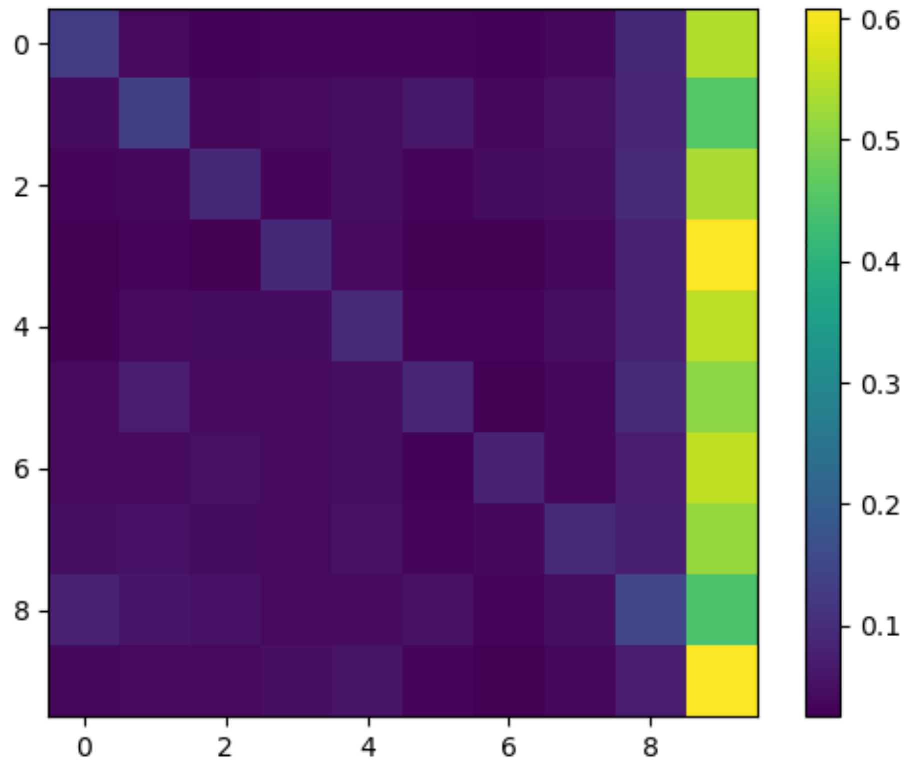
   GRAPH ON NEXT PAGE



d. 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.



e. 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.


0
2
4
6
8
0
2
4
6
8
0.6
0.5
0.4
0.3
0.2
0.1