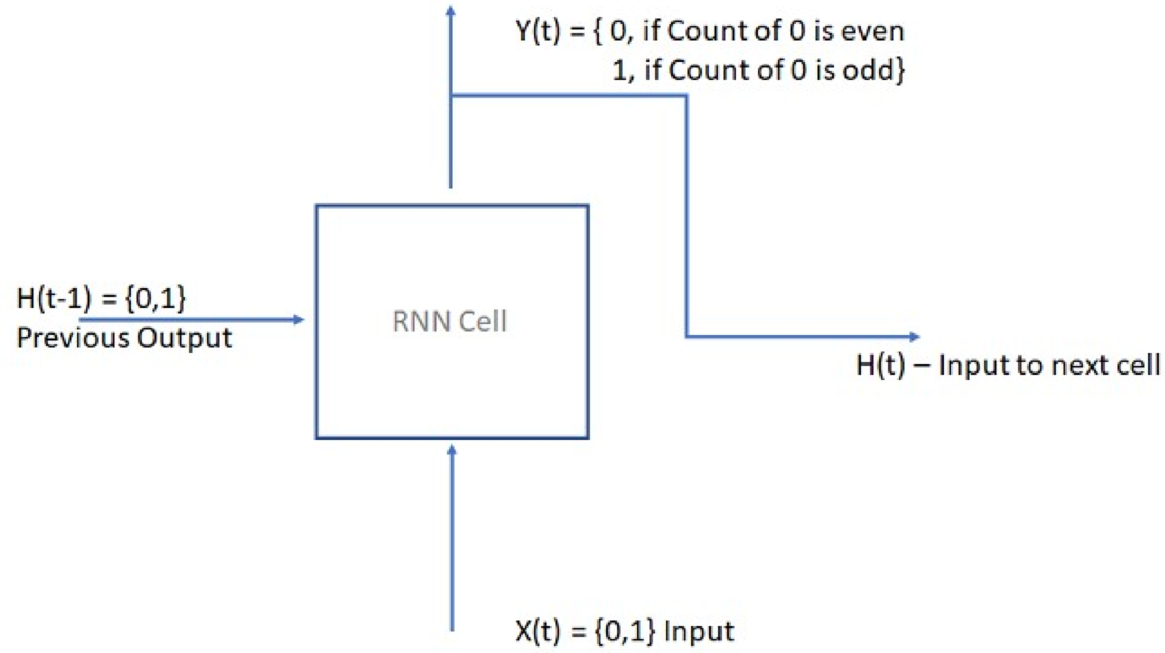
DSECLZG524 - Assignment 2 - G299

Assignment Q2: Let us define a sequence parity function as a function that takes in a sequence of binary inputs and returns a sequence indicating the number of 0’s in the input so far; specifically, if at time t the 0’s in the input so far is even it returns 1, and 0 if it is odd. For example, given input sequence [0, 1, 0, 1, 1, 0], the parity sequence is [0, 0, 1, 1, 1, 0].

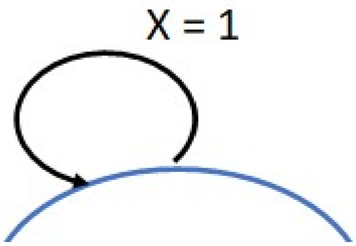
Implement the minimal vanilla recurrent neural network to learn the parity function. Explain your rationale using a state transition diagram and parameters of the network.

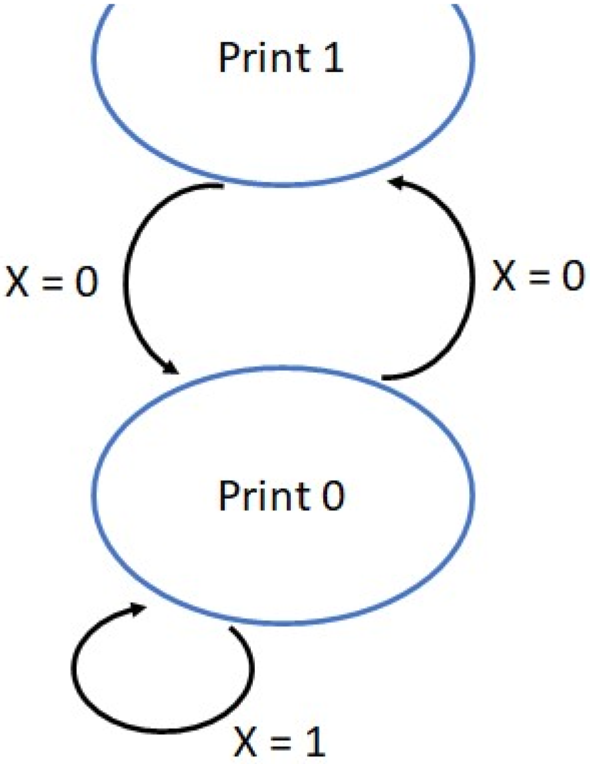
Answer:

Following is the basic RNN cell for generating the parity sequence. The output of the previous cell gives the state of 0's and thus along with the current input will give the output. If the Previous out put was 0 implying the count of 0's till T-1 was even. Thus if current input is 1 then output is 0 else output is 1.



State Trasition Diagram: The state changes if the input 'X' is 0 as the count becomes either Even or Odd. State remains the same if 'X' is 1. Same has been shown the the below diagram.





RNN Implemenation:

In [3]:
```
import numpy as np
```

In [4]:
```
def func_RNN(h_prev, x_curr):

    #Case where input so far has odd 0's
    if h_prev == 0:
        if x_curr == 0:
            h_curr = 1
        else:
            h_curr = 0
    else:
    #Case where input so far has even 0's
        if x_curr == 0:
            h_curr = 0
        else:
            h_curr = 1


    return h_curr
```

In [5]:

```
#input array
x = np.array([0,1,0,1,1,0])
len = x.shape
#Starting intialization for H is 0
h_prev = 0

Y_list = np.array([])


for cnt in range(0,len[0]):
    print('Input value to RNN cell at Time T-{} is {}'.format(cnt,x[cnt]))

    #Output function takes previous output from the hidden cell and current
    y = func_RNN(h_prev, x[cnt])

    #Initilizes current output of the Hiddent cell as input to next cell
    h_prev = y
    print('Output value of RNN cell at Time T-{} is {}'.format(cnt,y))
    print('------------------------------------------------')

    #Creating output array

    Y_list = np.append(Y_list, int(y))


print('Input to RNN is {}'.format(x))
print('Output of RNN is {}'.format(Y_list))
```

```
Input value to RNN cell at Time T-0 is 0
Output value of RNN cell at Time T-0 is 1
------------------------------------------------
Input value to RNN cell at Time T-1 is 1
Output value of RNN cell at Time T-1 is 1
------------------------------------------------
Input value to RNN cell at Time T-2 is 0
Output value of RNN cell at Time T-2 is 0
------------------------------------------------
Input value to RNN cell at Time T-3 is 1
Output value of RNN cell at Time T-3 is 0
------------------------------------------------
Input value to RNN cell at Time T-4 is 1
Output value of RNN cell at Time T-4 is 0
------------------------------------------------
Input value to RNN cell at Time T-5 is 0
Output value of RNN cell at Time T-5 is 1
------------------------------------------------
Input to RNN is [0 1 0 1 1 0]
Output of RNN is [1. 1. 0. 0. 0. 1.]
```

In [ ]: