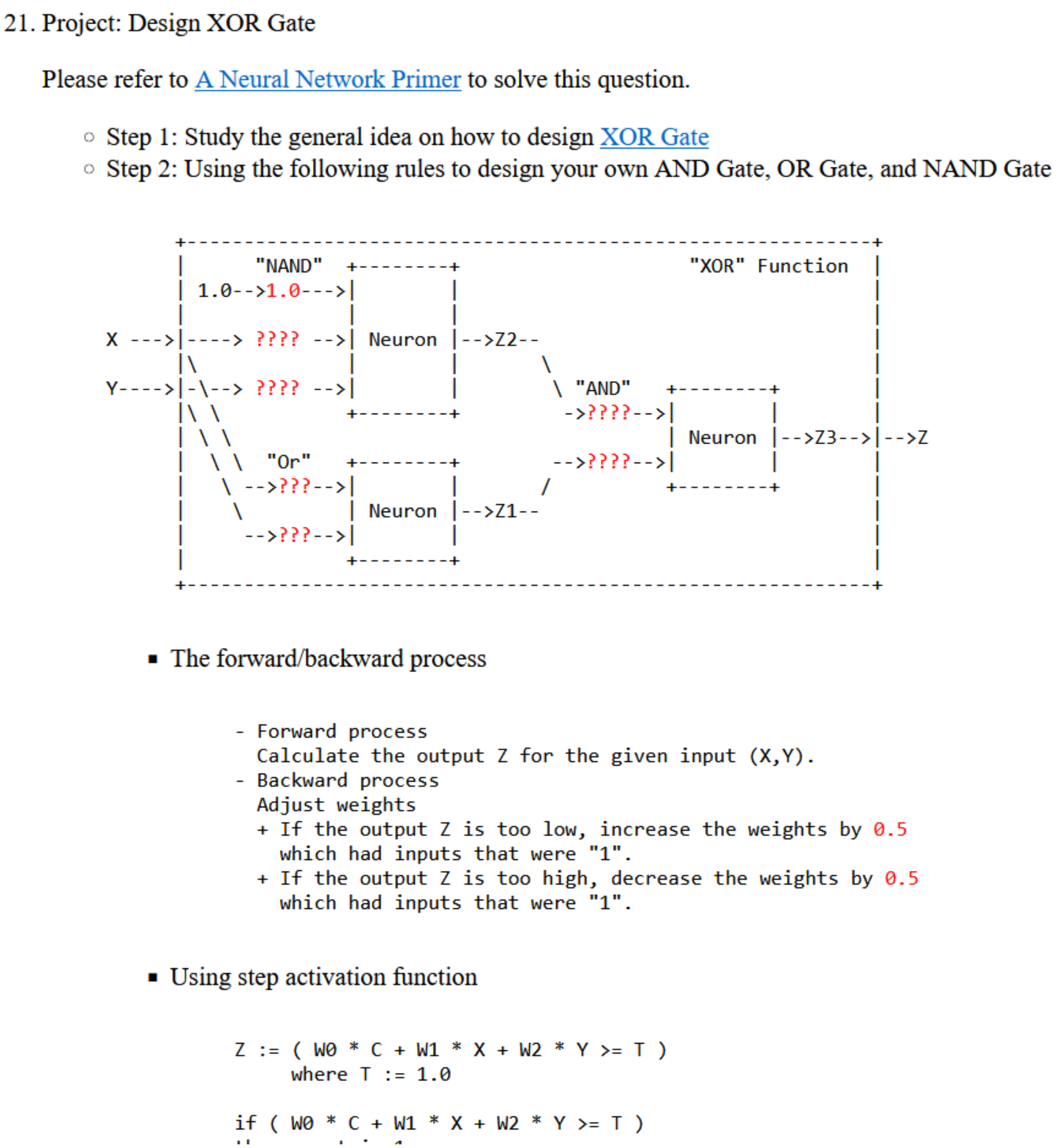


21. Project: Design XOR Gate

Please refer to A Neural Network Primer to solve this question.

- Step 1: Study the general idea on how to design XOR Gate
- Step 2: Using the following rules to design your own AND Gate, OR Gate, and NAND Gate

```
      +------------------------------------------------------------+
      |      "NAND"  +--------+                    "XOR" Function  |
      | 1.0-->1.0--->|        |                                    |
      |              |        |                                    |
X --->|----> ???? -->| Neuron |-->Z2--                             |
      |\             |        |       \                            |
Y---->|-\--> ???? -->|        |        \ "AND"   +--------+        |
      |\ \           +--------+         ->????-->|        |        |
      | \ \                                      | Neuron |-->Z3-->|-->Z
      |  \ \  "Or"   +--------+        -->????-->|        |        |
      |   \ -->???-->|        |       /          +--------+        |
      |    \         | Neuron |-->Z1--                             |
      |     -->???-->|        |                                    |
      |              +--------+                                    |
      +------------------------------------------------------------+
```

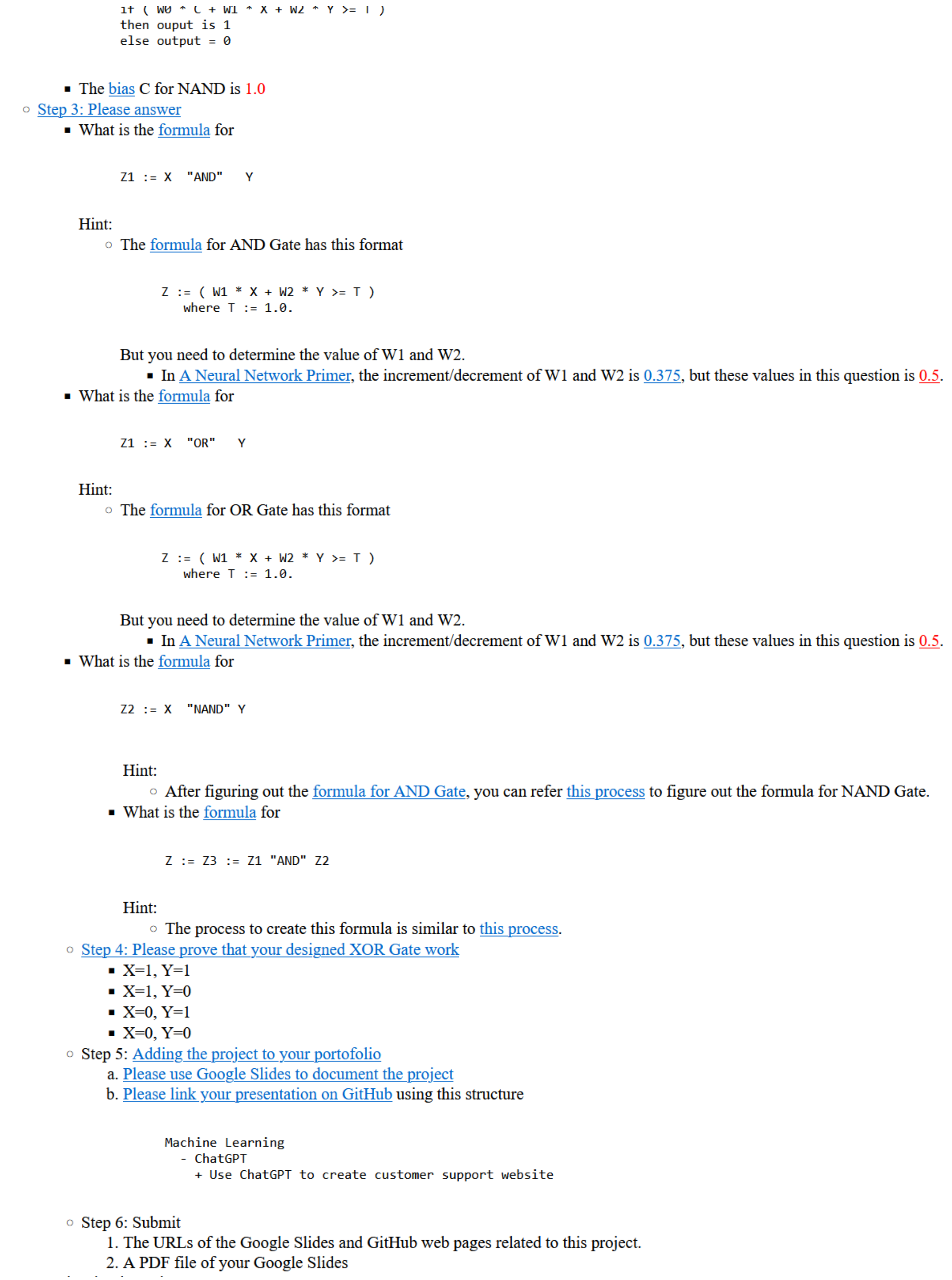
- The forward/backward process

```
- Forward process
  Calculate the output Z for the given input (X,Y).
- Backward process
  Adjust weights
  + If the output Z is too low, increase the weights by 0.5
    which had inputs that were "1".
  + If the output Z is too high, decrease the weights by 0.5
    which had inputs that were "1".
```

- Using step activation function

```
Z := ( W0 * C + W1 * X + W2 * Y >= T )
     where T := 1.0

if ( W0 * C + W1 * X + W2 * Y >= T )
```



```
if ( W0 * C + W1 * X + W2 * Y >= T )
then ouput is 1
else output = 0
```

- The bias C for NAND is 1.0

- Step 3: Please answer
  - What is the formula for

    ```
    Z1 := X  "AND"   Y
    ```

    Hint:
    - The formula for AND Gate has this format

      ```
      Z := ( W1 * X + W2 * Y >= T )
         where T := 1.0.
      ```

      But you need to determine the value of W1 and W2.
      - In A Neural Network Primer, the increment/decrement of W1 and W2 is 0.375, but these values in this question is 0.5.
  - What is the formula for

    ```
    Z1 := X  "OR"   Y
    ```

    Hint:
    - The formula for OR Gate has this format

      ```
      Z := ( W1 * X + W2 * Y >= T )
         where T := 1.0.
      ```

      But you need to determine the value of W1 and W2.
      - In A Neural Network Primer, the increment/decrement of W1 and W2 is 0.375, but these values in this question is 0.5.
  - What is the formula for

    ```
    Z2 := X  "NAND" Y
    ```

    Hint:
    - After figuring out the formula for AND Gate, you can refer this process to figure out the formula for NAND Gate.
    - What is the formula for

      ```
      Z := Z3 := Z1 "AND" Z2
      ```

      Hint:
      - The process to create this formula is similar to this process.
  - Step 4: Please prove that your designed XOR Gate work
    - X=1, Y=1
    - X=1, Y=0
    - X=0, Y=1
    - X=0, Y=0
  - Step 5: Adding the project to your portofolio

    a. Please use Google Slides to document the project

    b. Please link your presentation on GitHub using this structure

    ```
    Machine Learning
      - ChatGPT
        + Use ChatGPT to create customer support website
    ```
  - Step 6: Submit
    1. The URLs of the Google Slides and GitHub web pages related to this project.
    2. A PDF file of your Google Slides

Answer:

The forward/backward process

```
- Forward process
  Calculate the output Z for the given input (X,Y).
- Backward process
  Adjust weights
  + If the output Z is too low, increase the weights by 0.5
    which had inputs that were "1".
  + If the output Z is too high, decrease the weights by 0.5
    which had inputs that were "1".
```

Using step activation function

```
Z := ( W0 * C + W1 * X + W2 * Y >= T )
     where T := 1.0

if ( W0 * C + W1 * X + W2 * Y >= T )
then ouput is 1
else output = 0
```

Desired Function for “XOR”:

```
  OR                  NAND               XOR

 X  Y | Z1           X  Y | Z2          X  Y | Z3
----------          ----------         ----------
 0  0 | 0            0  0 | 1           0  0 | 0
 0  1 | 1     AND    0  1 | 1     =     0  1 | 1
 1  0 | 1            1  0 | 1           1  0 | 1
 1  1 | 1            1  1 | 0           1  1 | 0
```

Step 3:

“OR” Gate Formula:

Z := (W1 * X + W2 * Y >= T)

where T := 1.0

Desired Function for “OR”:

```
X  Y  |  Z
-----------
0  0  |  0
0  1  |  1
1  0  |  0
1  1  |  1
```

Loop 1:

W1 = W2 = 1.0

```
X   Y   |   Z1
-------------------
0   0   |   0      Z ok
0   1   |   1      Z ok
1   0   |   1      Z ok
1   1   |   1      Z ok
```

W1 = 1.0, W2 = 1.0 for “OR” Gate

“Nand” Gate Formula:

Z := (W0 * C + W1 * X + W2 * Y >= T)

where T := 1.0

Desired Function for “NAND”:

```
C   X   Y   |   Z2
---------------
1   0   0   |   1
1   0   1   |   1
1   1   0   |   1
1   1   1   |   0
```

Loop 1:

W0 = 0.0

W1 = W2 = 1.0

```
C   X   Y   |   Z2
---------------
1   0   0   |   0   Z too low, increase W0 by 0.5, W0=0.5
1   0   1   |   1   Z ok
1   1   0   |   1   Z ok
1   1   1   |   1   Z too high, decrease all by 0.5, W0=0.0, W1=0.5, W2=0.5
```

Loop 2:

W0 = 0.0

W1 = W2 = 0.5

```
C   X   Y   |   Z2
---------------
1   0   0   |   0   Z too low, increase W0 by 0.5, W0=0.5
1   0   1   |   0   Z too low, increase W0, W2 by 0.5, W0=1.0, W2=1.0
```

```
1  1  0  |  0  Z too low, increase W0, W1 by 0.5, W0=1.5, W1=1.0
1  1  1  |  1  Z too high, decrease all by 0.5, W0=1.0, W1=0.5, W2=0.5
```

Loop 3:

W0 = 1.0

W1 = W2 = 0.5

```
C  X  Y  |  Z2
---------------
1  0  0  |  1  Z ok
1  0  1  |  1  Z ok
1  1  0  |  1  Z ok
1  1  1  |  1  Z too high, decrease all by 0.5, W0=0.5, W1=0.0, W2=0.0
```

Loop 4:

W0 = 0.5

W1 = W2 = 0.0

```
C  X  Y  |  Z2
---------------
1  0  0  |  0  Z too low, increase W0 by 0.5, W0=1.0
1  0  1  |  0  Z too low, increase W0, W2 by 0.5, W0=1.5, W2=0.5
1  1  0  |  0  Z too low, increase W0, W1 by 0.5, W0=2.0, W1=0.5
1  1  1  |  0  Z ok
```

Loop 5:

W0 = 0.5

W1 = W2 = 0.0

```
C  X  Y  |  Z2
---------------
1  0  0  |  1  Z ok
1  0  1  |  1  Z ok
1  1  0  |  1  Z ok
1  1  1  |  1  Z too high, decrease all by 0.5, W0=1.5, W1=0.0, W2=0.0
```

Loop 6:

W0 = 1.5

W1 = W2 = 0.0

```
C  X  Y  |  Z2
---------------
1  0  0  |  1  Z ok
1  0  1  |  1  Z ok
```

```
1   1   0   |   1   Z ok
1   1   1   |   1   Z too high, decrease all by 0.5, W1=1.0, W2=-0.5, W3=-0.5
```

Loop 7:

W0 = 1.0

W1 = W2 = -0.5

```
C   X   Y   |   Z2
---------------
1   0   0   |   1   Z ok
1   0   1   |   1   Z too low, increase W0, W2 by 0.5, W0=1.5, W2=0.0
1   1   0   |   1   Z too low, increase W0, W2 by 0.5, W0=1.5, W2=0.0
1   1   1   |   1   Z ok
```

Loop 8:

W0 = 2.0

W1 = W2 = 0.0

```
C   X   Y   |   Z2
---------------
1   0   0   |   1   Z ok
1   0   1   |   1   Z ok
1   1   0   |   1   Z ok
1   1   1   |   1   Z too high, decrease all by 0.5, W1=1.5, W2=-0.5, W3=-0.5
```

Loop 9:

W0 = 1.5

W1 = W2 = -0.5

```
C   X   Y   |   Z2
---------------
1   0   0   |   1   Z ok
1   0   1   |   1   z ok
1   1   0   |   1   Z ok
1   1   1   |   0   Z ok
```

W0 = 1.5, W1 = -0.5, W2 = -0.5 for "NAND" Gate

“AND” Gate Formula:

```
Z := ( W1 * X + W2 * Y >= T )
where T := 1.0
```

Desired Function for “AND”:

X	Y	Z3
0	0	0
0	1	0
1	0	0
1	1	1

Loop 1:

W1 = W2 = 1.0

X	Y	Z3	
0	0	0	Z ok
0	1	1	Z too high, decrease W2 by 0.5, W2=0.5
1	0	1	Z too high, decrease W1 by 0.5, W1=0.5
1	1	1	Z ok

Loop 2:

W1 = W2 = 0.5

X	Y	Z3	
0	0	0	Z ok
0	1	1	Z ok
1	0	1	Z ok
1	1	1	Z ok

W1 = 0.5, W2 = 0.5 for “AND” Gate

“XOR” Gate formula (plugging W0, W1, and W2s from previous gates):

```
Z1 := X "Or" Y
Z2 := X "NAND" Y
Z := Z3 := Z1 "AND" Z2
Z := ( X "Or" Y ) "AND" ( X "NAND" Y )
Z := ( 1.0 * X + 1.0 * Y >= 1.0 ) "AND" ( 1.5 * 1.0 + -0.5 * X + -0.5 * Y >=
1.0 )
Z := ( 0.5 * ( 1.0 * X + 1.0 * Y >= 1.0 ) + 0.5 * ( 1.5 + -0.5 * X + -0.5 * Y
>= 1.0 ) >= 1.0 )
```

Desired Function for “XOR”:

X	Y	Z3
0	0	0
0	1	1

```
1   0   |   1
1   1   |   0
```

Step 4:

When X=0, Y=0:

```
Z3 = (0.5 * ( 1.0 * 0 + 1.0 * 0 >= 1.0 ) + 0.5 * ( 1.5 + -0.5 * 0 + -0.5 * 0
>= 1.0 ) >= 1.0 )
= (0.5 * ( 0.0 >= 1.0 ) + 0.5 * ( 1.5 >= 1.0 ) >= 1.0 )
= (0.5 * ( false ) + 0.5 * ( true ) >= 1.0 )
= 0.5 * 0 + 0.5 * 1 >= 1.0
= 0.5 >= 1.0
= false
= 0 ➔ same as desired
```

When X=0, Y=1:

```
Z3 = (0.5 * ( 1.0 * 0 + 1.0 * 1 >= 1.0 ) + 0.5 * ( 1.5 + -0.5 * 0 + -0.5 * 1
>= 1.0 ) >= 1.0 )
= (0.5 * ( 1.0 >= 1.0 ) + 0.5 * ( 1.0 >= 1.0 ) >= 1.0 )
= (0.5 * ( true ) + 0.5 * ( true ) >= 1.0 )
= 0.5 * 1 + 0.5 * 1 >= 1.0 = 1.0 >= 1.0
= true
= 1 ➔ same as desired
```

When X=1, Y=0:

```
Z3 = (0.5 * ( 1.0 * 1 + 1.0 * 0 >= 1.0 ) + 0.5 * ( 1.5 + -0.5 * 1 + -0.5 * 0
>= 1.0 ) >= 1.0 )
= (0.5 * ( 1.0 >= 1.0 ) + 0.5 * ( 1.0 >= 1.0 ) >= 1.0 )
= (0.5 * ( true ) + 0.5 * ( true ) >= 1.0 )
= 0.5 * 1 + 0.5 * 1 >= 1.0 = 1.0 >= 1.0
= true
= 1 ➔ same as desired
```

When X=1, Y=1:

```
Z3 = (0.5 * ( 1.0 * 1 + 1.0 * 1 >= 1.0 ) + 0.5 * ( 1.5 + -0.5 * 1 + -0.5 * 1
>= 1.0 ) >= 1.0 )
= (0.5 * ( 2.0 >= 1.0 ) + 0.5 * ( 0.5 >= 1.0 ) >= 1.0 )
= (0.5 * ( true ) + 0.5 * ( false ) >= 1.0 )
= 0.5 * 1 + 0.5 * 0 >= 1.0 = 0.5 >= 1.0
= false
= 0 ➔ same as desired
```

All Z3 same as desired in "XOR" Gate!