1. Verify that the neural network (with sigmoid activation functions) in Figure 1 approximates the `not(XOR)` function by computing the values of the hidden units (i.e., the outputs of the hidden nodes) and the final output $h_\theta(\mathbf{x})$ for the given inputs below.

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

Table 1: Input and output of `not(XOR)` function.

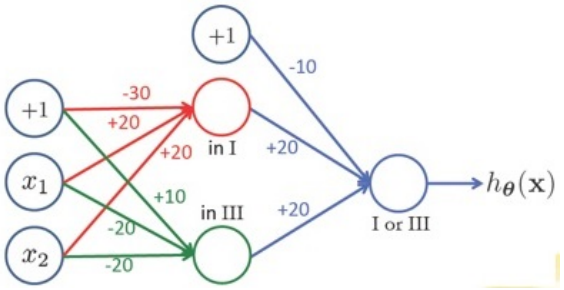

Figure 1: Neural network that approximate `not(XOR)` function.

Ans: The values of hidden units and the output are shown in the table below.

x_1	x_2	HU 1	HU 2	$\hat{y}$
0	0	$g(-30) \approx 0$	$g(10) \approx 1$	$g(10) \approx 1$
0	1	$g(-10) \approx 0$	$g(-10) \approx 0$	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$	$g(-10) \approx 0$	$g(-10) \approx 0$
1	1	$g(10) \approx 1$	$g(-30) \approx 0$	$g(10) \approx 1$

Table 2: Input and output of `not(XOR)` function.

2. Consider a convolutional layer with the kernel shown in Figure 2.

   (a) Compute the output for the given input using the kernel with zero-padding.

   Ans: The output from the convolutional layer is shown in Figure 3.

   (b) Suppose that a 2×2 max-pooling layer with stride 2 is used following the convolutional layer. Compute the output from the max-pooling layer with the input from part (a).

   Ans: The output from the maxpool layer is shown in Figure 4.

1	0	1	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	0	1	1	1	0
0	1	1	0	1	1
1	0	1	0	1	0

input

2	1	2
0	2	0
1	2	1

kernel

Figure 2: The kernel for a convolutional layer and the input for problem 2.

```
output =

     3     3     5     2     5     3
     3     8     5     6     5     5
     6     5     8     8     8     4
     4     7     6     8     6     5
     3     8     7     7     7     5
     4     3     5     4     5     3
```

Figure 3: Answer to problem 2(a).

```
maxoutput =

     8     6     5
     7     8     8
     8     7     7
```

Figure 4: Output from maxpool layer.