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.

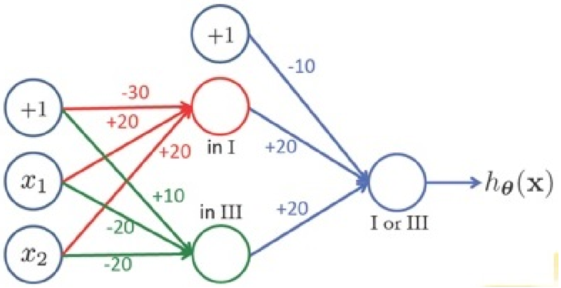
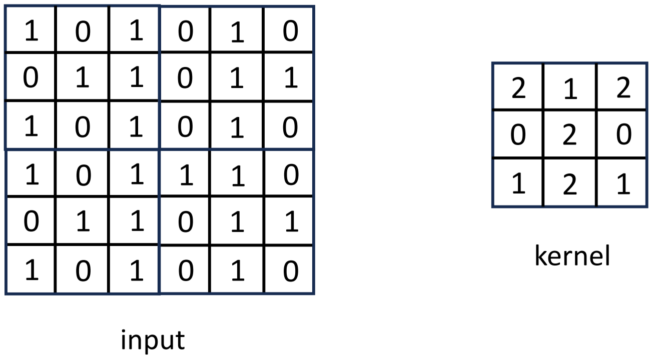


Figure 1: Neural network that approximate `not (XOR)` function.

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



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.

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

(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).