Solution 10.1: The McCulloch-Pitts Neuron

1. This question is formulated very openly and it is possible to come up with very many answers. Here are a few relatively relevant ones. Similarities:

   - Biological and McCulloch-Pitts neurons can be active or inactive.
   - They have a directionality (input/output).
   - The activation of a neuron is dependent on a weighted function of other neurons.

   Differences:

   - Real neurons exist in continuous time, whereas McCulloch-Pitts neurons operate in discrete time.
   - Real neurons have degrees of activation, not just on/off states.
   - The activation as a function of the inputs of a real neuron (typically) is not linear (or thresholded linear).

2. Realizable are NAND and AND, not realizable are XOR and NOT(XOR). To see this, imagine the Perceptron as drawing a straight line in the 2D input space (in n dimensions a McCulloch-Pitts Neuron draws a $n-1$ dimensional hyperplane). A successful 'classification' of the inputs means that all inputs of any given value lie on the same side of this line. Arrangements of points that cannot be separated by a line correspond to mappings that cannot be realized by the Perceptron; the XOR and NOT(XOR) are examples of such mappings.

3. (a) Does not change realizability.
   (b) The parameters of a McCulloch-Pitts neuron can be rescaled by a common factor; the relevant numbers are their ratios to the threshold. Using integers these ratios are always rational numbers, which can approximate any real number with arbitrary precision. The same functions can thus be implemented.
   (c) In this case the separation line will always intersect the origin. This does affect the possible classifications. If it is allowed to add an input that is always positive, one can get back to the case with threshold.

Solution 10.2: The Perceptron Learning Algorithm

x_1	x_2	desired output	current output	w_1	w_2	w_3
1	0	**1**	**0**	–0.4	0.5	–0.4
0	1	1	1	–0.1	0.5	–0.1
1	1	1	1			
0	0	0	0			
1	0	**1**	**0**			
0	1	1	1	0.2	0.5	0.2
1	1	1	1			
0	0	**0**	**1**			
1	0	1	1	0.2	0.5	–0.1
0	1	1	1			
1	1	1	1			
0	0	0	0			

The Perceptron learned the OR function.