# PROJECT 1

### Due October 2, 2018 (before start of class)

Use the feedforward neural network with a single output layer (as was done in lecture 2) to classify input patterns either as a ”C” or not a ”C”. We have here the following two training patterns

```
.##
#..
#..
#..
.##
```

with a target $t^{(1)} = 1$, and

```
#.#
#.#
.#.
#.#
#.#
```

with a target $t^{(2)} = -1$ (for not a ”C”).

Following what was done in class, discretize the patterns to form two training bipolar vectors, each having $5 \times 3 = 15$ elements. Use Hebb's rule to compute the weight vector and bias assuming initial zero values. Compute them analytically.

Check to find out if the resulting NN correctly classify the two training vectors. Perform the calculation analytically using the method done in class.

Next, write a computer program using a modern language to investigate the performance of this NN. Using brute force (exhaustive) method to find out what happens if the original pattern for ”C” has one mistake in one of its pixels. Try all different pixels. Repeat the process for two pixels, and then for three, etc. Produce a vector where the $k$-th element gives the total number of ”C” that is misclassified when it is corrupted by having $k$ different pixels flipped from black to white and from white to black.

Next instead of corrupted ”C” patterns, use $k$ undetermined (unmeasured) pixels by assigning a zeros to those pixels. Again try $k = 1$, $k = 2$, etc., exhaustively. For the lowest $k$ that first gives a pattern that the NN classifies incorrectly, what are the components of that vector? In case that there are multiple vectors for the same $k$ that are misclassified, just specify the components of the first such vector that you find. Compare that vector with ”C” and ”X”. Is the result what you expect?

Submit a hard copy of the complete program that you used to obtain your results.