COGSCI 108f: Homework Assignment 3,  
due: Wednesday 2005-05-19 at the beginning of class

You can work in teams of up to three. Clearly label each part of your solution and don’t forget to put your name and the names of co-workers on the solution. Attach all matlab listings. Have fun!

(A) (1 point) Show that entropy is non-negative. Only treat the discrete case.

(B) (4 point, after Ballard, 1997.) At the Willy Wonka Chocolate Factory they are constantly making chocolate bars. The computer in charge of the process codes good bars as a zero. Once in a while the machine makes an error, which the computer records as a one. Studies show that the probability of a mistake $P(1) = 1/16$, so that $P(0) = 15/16$. A sample record is given by 00001000010100000010.

- (1 point) Based on the probabilities given, what is the entropy or information rate?
- (2 points) These data are taking up a lot of space. As a newly employed systems analyst, you suggest that the company encode the data in blocks of two bits using the code shown in the table. What is the average number of bits per chocolate bar for the new code? How many bits do you minimally need for the example record above?

<table>
<thead>
<tr>
<th>Data block</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>0</td>
<td>10</td>
<td>110</td>
<td>111</td>
</tr>
</tbody>
</table>

- (1 point) How could you make an even better code?

(C) (2 points) Let $\mathcal{X} = \{0, 1\}$ and consider two distributions $p$ and $q$ on $\mathcal{X}$. Let $p(0) = 1 - r$, $p(1) = r$, and let $q(0) = 1 - s$, $q(1) = s$.

- Give expressions for $D(p||q)$ and $D(q||p)$.
- Evaluate these expressions for $r = s$, and for $r = 1/2$, $s = 1/4$. Is $D(.||.)$ symmetric with respect to switching its arguments?

(D) (3 points) Consider an exponential density given by:

$$p(x) = \frac{1}{\mu} e^{-x/\mu} \text{ for } x \geq 0 \text{ and } 0 \text{ otherwise}.$$  

- (1 point) Show that $p(x|\theta)$ is a valid pdf.
- (2 points) What is the differential entropy of the distribution?

(E) (3 points) Show that for independent random variables, the joint entropy is the sum of the marginal entropies, i.e., show that

$$H(X, Y) = H(X) + H(Y).$$

Treat both the discrete and the continuous case.

The subsequent questions deal with the Hopfield network.

(F) (3 points) Write a function `computeWeights.m` to store a number of patterns into the network. Each pattern can be represented as a binary vector and the set of patterns to be stored should be passed to the function as a binary matrix. The function is to return the weight matrix for the network.

(G) (3 points) The state evolves over time by application of the McCulloch Pitts update rule. Write a function `updateState.m` to perform a synchronous update of the network’s state, such that all units are updated at the same time. The function should take the old state and the weight matrix as inputs and return the new state.
(H) (3 points) Write a function `energy.m` to compute the energy of a state of the network. The function should take the state and the weight matrix as inputs and return the energy.

(I) (3 points) Write a function `settle.m` that takes the network’s weight matrix and its initial state as arguments and that lets the network settle. To this end, the function should repeatedly call `updateState.m` and `energy.m` until the energy does not change anymore. The function should return the final state of the network.

(J) (10 points) Write a program `hopfield.m` that first computes the weights for a particular set of patterns to be stored. It then initializes the state of the network to a random initial state and lets it settle. It finally, compares the resulting state to all stored patterns and prints out if the resulting state matches any of the stored patterns and if so which one it matches. Now do the following experiment with the program: Use the network to store 5 random patterns. Initialize the network with a pattern that is a version of one of the stored patterns where 20% of the bits have been flipped. Does the network retrieve the original pattern? Run the program 20 times and note how often it retrieved the stored pattern.

(K*) (5 points extra credit) Analyze the storage capacity experimentally. Use the network to store between 1 and 20 random patterns. In each case, initialize the network with all of the patterns stored and see what fraction of the stored states are stable. Plot this percentage as a function of the number of patterns you stored. Explain the result.