Neural Networks, CAP 6615 Spring 2011 Homework 7, Due 18 April 2011

1. (Haykin Problem 11.3) Consider the Markov chain depicted in Figure P11.3, which is reducible. Identify the classes of states contained in this state transition diagram.

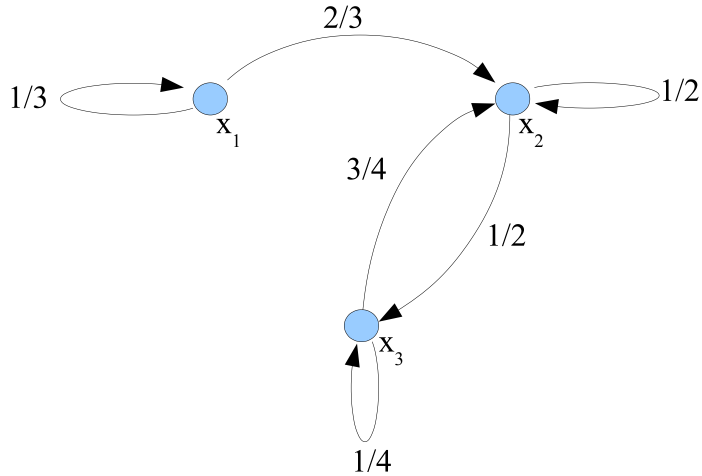

Figure P11.3

The classes of this Markov chain are $\{x_1\}$ and $\{x_2, x_3\}$

2. (Haykin Problem 11.4) Calculate the steady-state probabilities of the Markov chain shown in Fig. P11.4

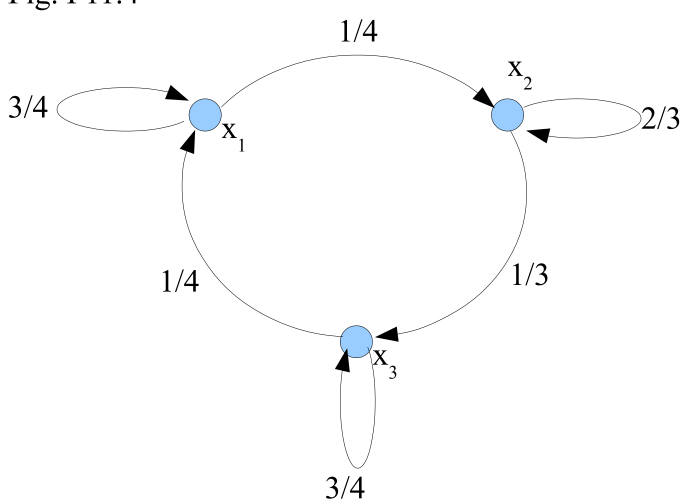

Figure P11.4

The stochastic matrix for this Markov chain is

$$P=\begin{bmatrix} 3/4 & 1/4 & 0 \\ 0 & 2/3 & 1/3 \\ 1/4 & 0 & 3/4 \end{bmatrix} \text{ yielding equations } \begin{array}{l} \pi_i=\pi_i(3/4)+\pi_2(0)+\pi_3(1/4) \\ \pi_2=\pi_1(1/4)+\pi_2(2/3)+\pi_3(0) \\ \pi_3=\pi_1(0)+\pi_2(1/3)+\pi_3(3/4) \end{array} \text{ thus, } \pi_1=\pi_3 \text{ ,}$$

$\pi_2=\pi_1(3/4)$. We also have $\pi_1+\pi_2+\pi_3=1$ by definition. Thus, $\pi_1+\pi_1(3/4)+\pi_1=1$ so $\pi_1=\pi_3=4/11$ and $\pi_2=3/11$.

3. (Haykin Problem 11.7) In this problem, we consider the use of simulated annealing for solving the *traveling-salesman problem* (TSP). You are given the following:
   - N cities
   - the distance between each pair of cities, d
   - a tour represented by a closed path visiting each city once, and only once.

   The objective is to find a tour (i.e., permutation of the order in which the cities are visited) that is of minimal total length L. In this problem, the different possible tours are the configurations, and the total length of a tour is the cost function to be minimized.

   (a) Devise an iterative method of generating valid configurations.

   One iterative scheme to generate valid configurations is to start with the simplest valid configuration, namely, [1, 2, … , N], then generate another configuration by randomly choosing a pair of cities i and j, then swapping the positions of i and j in the current configuration. e.g., performing this change from the initial configuration yields this state: [1, 2, … , i-1, j, i+1, … , j-1, i, j+1, … , N].

   (b) The total length of a tour is defined by

   $$L_p = \sum_{i=1}^{N} d_{P(i)P(i+1)}$$

   where P denotes a permutation with $P(N+1)=P(1)$. Correspondingly, the partition function is

   $$Z = \sum_{P} e^{-L_p/T}$$

   where T is a control parameter. Set up a simulated-annealing algorithm for the TSP.

   1. Use some reasonable schedule for adjusting the temperature T.
   2. Initialize the tour as above.
   3. Calculate the energy in the current state.
   4. Generate a new configuration and calculate the energy due to swapping two cities.
   5. If the energy difference is negative or zero, accept the new tour, otherwise accept the change with the probability defined by the Metropolis algorithm.
   6. Go to step 3 unless the required number of iterations is complete.
   7. Reduce T according to the annealing schedule and go to step 3 unless the final temperature has been reached.

4. (Haykin Problem 11.8) Consider a stochastic, two-state neuron k operating at temperature T. This neuron *flips* from state x_k to state $-x_k$ with probability

   $$P(x_k \rightarrow -x_k) = \frac{1}{1+\exp(-\Delta E_k/T)}$$

   where ΔE_k is the energy change resulting from such a flip. The total energy of the Boltzmann machine is defined by

   $$E = -\frac{1}{2}\sum_{\substack{i \\ i\neq j}}\sum_{j} w_{ji} x_i x_j$$

   where w_{ji} is the synaptic weight from neuron i to neuron j, with $w_{ji}=w_{ij}$ and $w_{jj}=0$.

   (a) Show that

$$\Delta E_k = -2x_k v_k$$

where v_k is the induced local field of neuron k.

Neuron k flips from state x_k to state $-x_k$ at temperature T with probability

$P(x_k \to -x_k) = \dfrac{1}{1+e^{-\Delta E_k/T}}$ where ΔE_k is the energy difference resulting from this flip.

The Boltzmann machine's energy function is

$E = \dfrac{-1}{2}\sum_{\substack{i \\ i \neq j}}\sum_j w_{ji}x_ix_j$. Note that this is really identical to $E = -\dfrac{1}{2}\sum_i\sum_j w_{ji}x_ix_j$ since each weight w_{ii} is 0. Both of these expressions count each unique pair of x's twice (reversing the roles of i and j) then divide by two. We could rewrite this as $E = -\sum_{i \leq j} w_{ji}x_ix_j$, removing this double counting, and finally write this as $E = -\sum_{i<j} w_{ji}x_ix_j$, once more eliminating the terms with identical subscripts.

Returning to calculation of the energy resulting from a flip of neuron x_k to $-x_k$, using our third formulation, we have

ΔE_k = (energy with neuron k in state $-x_k$) – (energy with neuron k in state x_k)

$$= \left(-\left[-x_k\sum_{i<k} w_{ki}x_i + \sum_{\substack{i<j \\ i\neq k}} w_{ji}x_ix_j - x_k\sum_{k<j} w_{jk}x_j\right]\right)$$
$$-\left(-\left[x_k\sum_{i<k} w_{ki}x_i + \sum_{\substack{i<j \\ i\neq k}} w_{ji}x_ix_j + x_k\sum_{k<j} w_{jk}x_j\right]\right) .$$

In each of the two factors of this expression, the first subfactor contains those elements where $i < k$, the second subfactor contains all elements not involving k, and the third subfactor collects those elements where $k > i$. In the first of the two factors, the terms involving k are negative (as required) and in the second they are positive. Continuing, we get

$$\Delta E_k = \left(-\left[-x_k\sum_{i<k} w_{ki}x_i + \sum_{\substack{i<j \\ i\neq k}} w_{ji}x_ix_j - x_k\sum_{k<j} w_{jk}x_j\right]\right)$$
$$-\left(-\left[x_k\sum_{i<k} w_{ki}x_i + \sum_{\substack{i<j \\ i\neq k}} w_{ji}x_ix_j + x_k\sum_{k<j} w_{jk}x_j\right]\right)$$
$$= \left(x_k\sum_i w_{ki}x_i\right) + \left(x_k\sum_i w_{ki}x_i\right)$$
$$= 2x_k\sum_i w_{ki}x_i$$
$$= 2x_k v_k.$$

(b) Hence, show that for an initial state $x_k = -1$, the probability that neuron k is flipped into state +1 is $1/(1+\exp(-2v_k/T))$.

Note first that the result stated above has the wrong sign for the exponential. The result, however, does not depend on the sign of the initial value of x_k, so substituting -1 for x_k in the result above yields $\frac{1}{1+e^{-\Delta E^k/T}}=\frac{1}{1+e^{2v_k/T}}$.

(c) Show that the same formula in part (b) holds for neuron *k* flipping into state -1 when it is initially in state +1.

Recall that $\frac{1}{1+e^x}=\frac{1}{1+e^x}+\frac{e^x}{1+e^x}-\frac{e^x}{1+e^x}=1-\frac{e^x}{1+e^x}=1-\frac{1}{1/e^x+1}=\frac{1}{1-e^{-x}}$ so $\frac{1}{1+e^{-2v_k/T}}=1-\frac{1}{1+e^{2v_k/T}}$. The left side of this equation is the probability of flipping from +1 to -1. The equation tells us that the probability of the complement of flipping x_k from +1 to -1, that is, the probability of flipping from +1 to -1, is given by the right side. In contrast to what part (c) of this problem asserts, these probabilities are **not** identical, and the formula does not hold for flipping from state +1 to state -1. The formula developed in part (a) does, of course hold because it does not depend on the initial value of x_k.