1. Solve the following problems:

   Chapter 3: 3.10 3.47 (EM problem)

   Chapter 5: 5.4

2. Neural network

For the above multi-layer neural network, derive the formulas of update rule to train weights $w_1$, $w_2$ and $w_3$, $w_4$ using the idea of back-propagation. The processing function of the neurons is sigmoid function $S(x)$

\[
S(x) = \frac{1}{1 + e^{-x}}
\]

\[
S'(x) = -\frac{1}{(1 + e^{-x})^2} \cdot (1 + e^{-x})'
\]

\[
= -\frac{1}{(1 + e^{-x})^2} \cdot (-e^{-x})
\]

\[
= \frac{e^{-x}}{1 + e^{-x}}
\]

\[
= S(x)(1 - S(x))
\]
3) Applying HMM to solve an application problem.

Download HMM code from here http://www.kanungo.com/software/software.html
Read its document and know how to define HMM models

We are going to create two miniature languages. We'll train one HMM with sentences from the first language, and train another HMM with sentences from the second language. We'll then classify some test sentences by evaluating which HMM is more likely to generate each test sentence.

Our languages make use of the following possible words: a, b, c, i, n, and t. The training set for the first language consists of the following five sentences: “c a t”, “c a b”, “c a n”, “c i t”, and “c i b”. The training set for the second language consists of the sentences: “t a c”, “b a c”, “n a c”, “t i c”, and “b i c”. The test set consists of the sentences: “c i n” and “n i c”.

Train an HMM with 4 states on each training set. The maximum number of iterations should be set to a large number (e.g., 50). Then test each HMM with the sentences from the test set. Compare the log likelihood to determine which language the test sentences may come from. Which HMM has larger log likelihood for each test sentence? Does this result make sense?