First, note that $\partial L/\partial\theta_{H2}$ will have the same value except that θ and $(1-\theta)$ are reversed, so the parameters for bags 1 and 2 will move in lock step if $\theta = 0.5$. Second, note that $\partial L/\partial\theta_{H1} = 0$ when $\theta_{H1} = 550/(450 + 550)$, i.e., exactly the observed proportion of candies with holes. Finally, we can calculate second derivatives and evlauate them at the fixed point. For example, we obtain

$$\frac{\partial^2 L}{\partial\theta_{H1}^2} = N\theta^2\theta_{H1}(1-\theta_{H1})(2\theta_{H1}-1)$$

which is negative (indicating the fixed point is a maximum) only when $\theta_{H1} < 0.5$. Thus, in general the fixed point is a saddle point as some of the second derivatives may be positive and some negative. Nonetheless, EM can reach it by moving along the ridge leading to it, as long as the symmetry is unbroken.

20.11 XOR (in fact any Boolean function) is easiest to construct using step-function units. Because XOR is not linearly separable, we will need a hidden layer. It turns out that just one hidden node suffices. To design the network, we can think of the XOR function as OR with the AND case (both inputs on) ruled out. Thus the hidden layer computes AND, while the output layer computes OR but weights the output of the hidden node negatively. The network shown in Figure S20.3 does the trick.

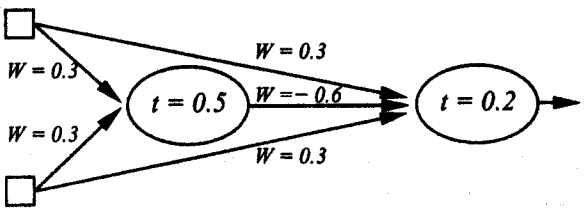

Figure S20.3 A network of step-function neurons that computes the XOR function.

20.12 The examples map from $[x_1, x_2]$ to $[x_1, x_1, x_2]$ coordinates as follows:
$[-1,-1]$ (negative) maps to $[-1,+1]$
$[-1,+1]$ (positive) maps to $[-1,-1]$
$[+1,-1]$ (positive) maps to $[+1,-1]$
$[+1,+1]$ (negative) maps to $[+1,+1]$
Thus, the positive examples have $x_1x_2 = -1$ and the negative examples have $x_1x_2 = +1$. The maximum margin separator is the line $x_1x_2 = 0$, with a margin of 1. The separator corresponds to the $x_1 = 0$ *and* $x_2 = 0$ axes in the original space—this can be thought of as the limit of a hyperbolic separator with two branches.

20.13 The perceptron adjusts the separating hyperplane defined by the weights so as to minimize the total error. The question assumes that the perceptron is trained to convergence (if possible) on the accumulated data set after each new example arrives.

There are two phases. With few examples, the data may remain linearly separable and the hyperplane will separate the positive and negative examples, although it will not represent