Problem Set 3 (due Oct 23)
Lyapunov functions and Contrastive Hebbian Learning

October 16, 2006

The stability of

$$\dot{x_i} + x_i = \left[b_i + \sum_j W_{ij} x_j \right]^+$$

can be analyzed in the nonnegative orthant using the Lyapunov function

$$L(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{x} - \mathbf{b}^T\mathbf{x}.$$

In this problem you will prove that L is a Lyapunov function for the specific case of the "winner-take-all" network

$$\dot{x}_i + x_i = \left[b_i + \alpha x_i - \beta \sum_j x_j \right]^+ .$$

1. Lyapunov function for the WTA network.
   (a) Specialize the above general expression for L to the WTA dynamics.
   (b) Prove that L is nonincreasing ($dL/dt \leq 0$) on trajectories of the dynamics, with equality only at steady states of the dynamics.
   *H*int: $\frac{dL}{dt} = \sum_i \frac{\partial L}{\partial x_i} \dot{x}_i$
   (c) Prove that L is lower bounded if $\alpha < 1 + \beta$ in the nonnegative orthant, and is not lower bounded if $\alpha > 1 + \beta$.
   (d) Prove that L is *radially unbounded* ($L(\mathbf{cx}) \to \infty$ as $c \to \infty$) for $\alpha < 1 + \beta$ and x in the nonnegative orthant. This completes the proof that L is a Lyapunov function of the network dynamics.
   *N*ote: In class we talked about co-positivity of the $I - W$ matrix which is a sufficient condition for L to be radially unbouded.

2. Contrastive Hebbian Learning.
   (a) Define a network with an input layer $\mathbf{x}^0$, hidden layer $\mathbf{x}^1$, and output layer $\mathbf{x}^2$. Let $\mathbf{W}^1$ contain the connections from $\mathbf{x}^0$ to $\mathbf{x}^1$, and $\mathbf{W}^2$ the connections from $\mathbf{x}^1$ to $\mathbf{x}^2$. Since the network is symmetric, the transposed matrix $\mathbf{W}^{2T}$ contains the connections from $\mathbf{x}^2$ to $\mathbf{x}^1$. (We don't have to worry about connections from $\mathbf{x}^1$ to $\mathbf{x}^0$, because the input layer is always clamped in what follows.) We consider two types of dynamics, one in which the output layer is free and the other in which it is clamped at desired values. The nonlinear activation function is denoted by f, which is assumed to be monotone increasing. When applied to a vector argument, f acts on each component of the vector, MATLAB style.

i. Free dynamics. The update for the hidden layer is

$$\mathbf{x}^1 = f\left(\mathbf{W}^1\mathbf{x}^0 + \mathbf{W}^{2T}\mathbf{x}^2 + \mathbf{b}^1\right) \tag{1}$$

Note that the hidden layer receives both bottom-up and top-down drive from $\mathbf{x}^0$ and $\mathbf{x}^2$ respectively. The top-down drive $\mathbf{W}^{2T}\mathbf{x}^2$ is what distinguishes the network from a multilayer perceptron.
The update for the output layer is

$$\mathbf{x}^2 = f\left(\mathbf{W}^2\mathbf{x}^1 + \mathbf{b}^2\right) \tag{2}$$

The free dynamics consists of alternating between these updates.
Find a Lyapunov function for the dynamics and prove that it is nonincreasing under either of the two updates above. This implies that the free dynamics finds a local minimum of the Lyapunov function with respect to $\mathbf{x}^1$ and $\mathbf{x}^2$.

ii. Clamped dynamics. Suppose that the output layer is clamped at some desired values. Then just one evaluation of Eq. (1) is required to reach a steady state. Show that this minimizes the Lyapunov function with respect to $\mathbf{x}^1$.

(b) Contrastive Hebbian learning is defined as follows. Iterate the free dynamics to a steady state and make the anti-Hebbian update

$$\begin{aligned}
\Delta\mathbf{W}^1 &= -\eta\mathbf{x}^1\mathbf{x}^{0T} \\
\Delta\mathbf{W}^2 &= -\eta\mathbf{x}^2\mathbf{x}^{1T} \\
\Delta\mathbf{b}^1 &= -\eta\mathbf{x}^1 \\
\Delta\mathbf{b}^2 &= -\eta\mathbf{x}^2
\end{aligned} \tag{3}$$

Now clamp the output layer to the desired values, and update the hidden layer according to Eq. (1). Now make a Hebbian update, which is the same as Eq. (3) but with opposite sign.

Here we are making an anti-Hebbian update after the free dynamics, and a Hebbian update after the clamped dynamics. In class, we discussed a slightly different form of this, which was to update the parameters after both the free and clamped dynamics are over, combining the Hebbian and anti-Hebbian terms in a single change. For this form of contrastive Hebbian learning, one can write down a cost function for which the learning update is gradient descent. Write down this cost function and show that your answer is correct.

(c) In a previous assignment, you trained a multilayer perceptron to recognize handwritten digits from the MNIST database. The perceptron had one hidden layer, and was trained using backpropagation.

Train a network of the above form on the MNIST database using contrastive Hebbian learning. Your network should be of size 784-25-10. Run the algorithm until it converges and submit the following results:

- The training error as a function of epoch number, using the cost function you found in part (b).
- The weight matrix $\mathbf{W}^1$, visualized as a set of images. How do these weights compare to those found by the backprop network?