9.641 Neural Networks
Problem Set 6: Deconvolution with Neural networks

(Due on Friday, Apr. 8)

You are expected to turn in any code you might have used to answer a question.

1. In this probem, you will help Helmut, your Eastern-German apprentice-astrophysicist-friend, to recover good data from the noisy data he collected with the cheap russian telescope he got for his birthday. Because this is a cheap telescope, the optics are very low quality and tend to blur the image. This can be modeled by convolving the true signal x_i with a Gaussian filter $g_i = \exp\left(-\frac{(i-i_0)^2}{\sigma^2}\right)$, where the index i runs from $-\infty$ to ∞ and adding noise:

$$n = g * x + \epsilon$$

The convolution of signal x with filter g is defined as

$$(g * x)_i = \sum_{j=-\infty}^{\infty} g_{i-j} x_j \tag{1}$$

This definition is applicable to signals. If g and x are finite, they can be extended to infinite length by adding zeros at both ends. After this trick, called zero padding, the definition in Eq. (1) becomes applicable. For example, the sum in Eq. (1) becomes

$$(g * x)_i = \sum_{j=0}^{n-1} g_{i-j} x_j \tag{2}$$

for the finite time series $x_0, \ldots, x_{n-1}$.

(a) Matrix form of convolution. Show that the convolution of g_0, g_1, g_2 and x_0, x_1, x_2 can be written as

$$g * x = Gx$$

where the matrix G is defined by

$$G = \begin{pmatrix} g_0 & 0 & 0 \\ g_1 & g_0 & 0 \\ g_2 & g_1 & g_0 \\ 0 & g_2 & g_1 \\ 0 & 0 & g_2 \end{pmatrix} \tag{3}$$

and $g * x$ and x are treated as column vectors.

(b) Helmut called his local Eastern-German dealer and they gave him the width of the Gaussian that best accounts for the defect in the optics:

$$\sigma = 2.5$$

Compute the convolution matrix G so that your true signal $x \approx G^{-1} * n$, where n is the noisy signal collected by the telescope. Download the file 'telescope.mat'. It contains both the noisy signal from the telescope as well as the true signal (for you to evaluate the quality of your algorithm). Compare the x signal obtained with the true signal we gave you . How good is the fit? Explain.

Hint: Use the matlab command *toeplitz* to form the matrix G.

(c) As we saw in class you can also use neural networks to perform deconvolution. Because they constrained the solution, they might help cope with noise. Deconvolving the signal is equivalent to minimizing the objective function $L = ||n - Gx||^2$. Write down a neural network equation so that it minimises L.

Hint: Write down the general form of the Lyapunov function L for a network with global inhibition and consider $\frac{\partial L}{\partial x}$.

(d) Simulate the neural network and compare the result with the true signal. How does it compare to direct deconvolution? Explain.

2. More about Lyapunov functions

(a) Construct a conserved quantity $H(x, y)$ for the dynamics

$$\dot{x} = f(y) \qquad \dot{y} = g(-x)$$

of two scalar variables x and y. Prove $dH/dt = \ 0$

(b) Construct a Lyapunov function $L(x, y)$ for the dynamics

$$\dot{x} + x = f(y) \qquad \dot{y} + y = g(-x)$$

of two scalar variables x and y. Prove that $dL/dt \leq 0$, with equality only at steady states. Use the assumption that f and g are increasing functions.

(c) Let **1** be the column vector of all ones. Recall that the all-to-all inhibitory network

$$\dot{x} + x = [b + \alpha x - \beta \mathbf{1}(\mathbf{1}^T x)]^+$$

performs the computation $\min_x V(x)$ where

$$V = \frac{1-\alpha}{2} x^T x - b^T x + \frac{\beta}{2}(\mathbf{1}^T x)^2$$

Show that

$$V(x) \ = \ \max_y S(x, y)$$

where y is a scalar variable and S is the saddle function

$$S \equiv \frac{1-\alpha}{2} x^T x - b^T x + \sqrt{\beta} y (\mathbf{1}^T x) - \frac{1}{2} y^2$$

This means that the all-to-all inhibitory network effectively computes a solution to the minimax problem:

$$\min_x \max_y S(x, y)$$

(d) A variant of the all-to-all inhibitory network is one in which there is a global inhibitory neuron mediating inhibitory interactions between all the excitatory neurons,

$$\dot{x} + x \ = \ [b + \alpha x - \sqrt{\beta} \mathbf{1} y]^+ \tag{4}$$

$$\tau_y \dot{y} + y \ = \ \sqrt{\beta} \mathbf{1}^T x \tag{5}$$

In the special case $\tau_y = 0$, this is equivalent to all-to-all inhibition. Show that this dynamics can be written as

$$\dot{x} + x \ = \ \left[x - \frac{\partial S}{\partial x} \right]^+ \tag{6}$$

$$\tau_y \dot{y} \ = \ \frac{\partial S}{\partial y} \tag{7}$$

This is gradient ascent in the y variable, and "pseudo" gradient descent in x.