

CPSC 636-600 Homework 5 (Total 100 points)

See the course web page for the **due date** and **submission info**.

Instructor: Yoonsuck Choe

April 18, 2008

Problem 1 (Program: 40 pts): Fill in the two lines in the `hopfield.m` skeleton code from the course web page, run the three examples (noisy input and partial input), and report your results.

`http://courses.cs.tamu.edu/choe/08spring/src/hopfield.m`

There are several other files that you may need. **Important:** For matlab, use a dogfilter of size 7 instead of 17. Or, better yet,

1. Fix the last line in `dogfilter.m` to this `M = real(conv2(X,g,'valid'));`
2. When calling `dogfilter`, do `dogfilter(rand(36,36),17)`, instead of `dogfilter(rand(20,20),17)`. This is when you construct the input matrix.

Problem 2 (Written: 20 pts): Test your implementation and see how much noise the Hopfield network can tolerate. Change the noise level and see beyond which point the network does not settle to the correct attractor.

Adjusting the inequality part (0.6) will allow you to control the noise level: `(rand(1,20*20)<0.6)`.

Problem 3 (Written: 20 pts): Test how much omitted information the Hopfield network can tolerate. This part `inp(1:10,:)=zeros(10,20);` controls how much input to eliminate.

Problem 4 (Written: 20 pts): Try increasing the number of memory patterns and check if you get the α_c value as determined in page 694 of the textbook.