COMP502-HW07

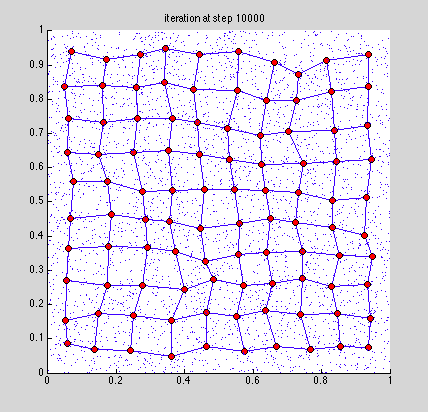
Xihao Zhu xz36

P1.

For my Kohonen SOM, I use 10000 steps and set learning rate as variable according to steps. Initial learning rate is as high as 0.6; while when steps approaches 10000 step, rate decreases uniformly to 0.01. Therefore the equation is like rate=init\_rate -(init\_rate –last\_rate)\*current#\_step/(total#\_step), where init\_rate=0.6, last\_rate=0.01, total#\_step=10000.

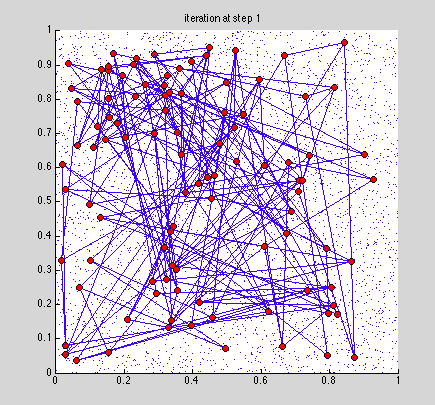
Initial weight is set to random floats between 0 to 1.

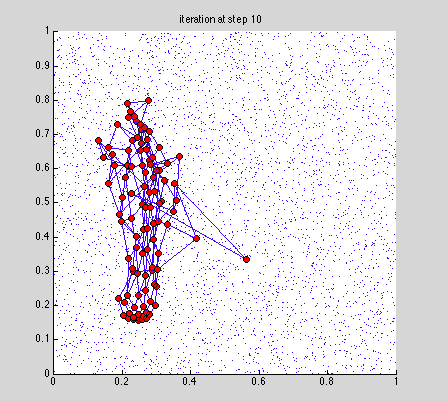
Final weight vectors is like this:

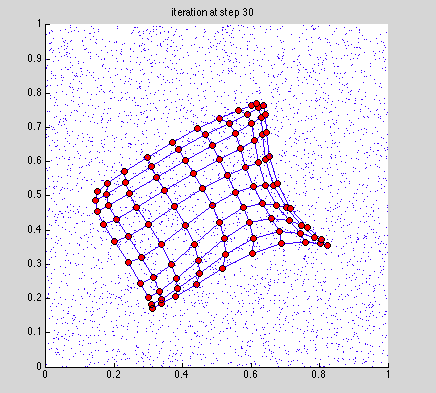


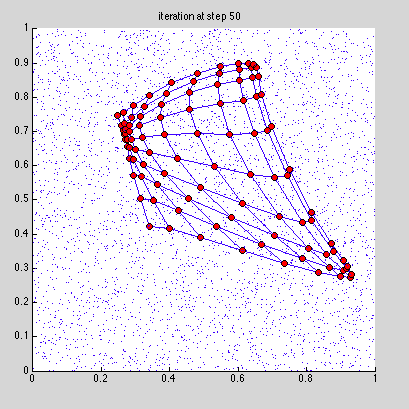
Since inputs are random floats spread in the whole plot above, the result weight is reasonable since it’s like a uniformly web that tries to cover all points.

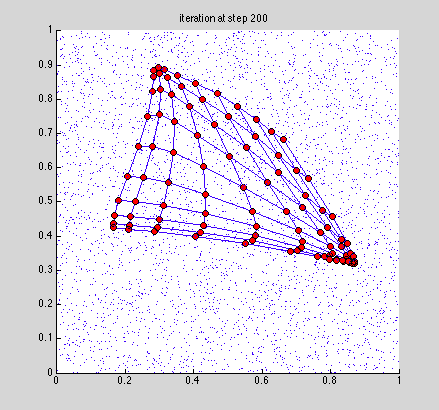
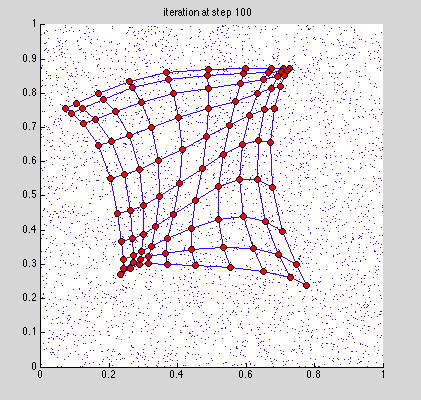
Historical weight vectors at step 1, 10, 30, 50, 100, 500, 2000, 10000:

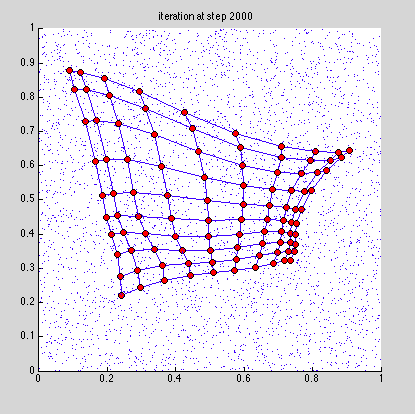
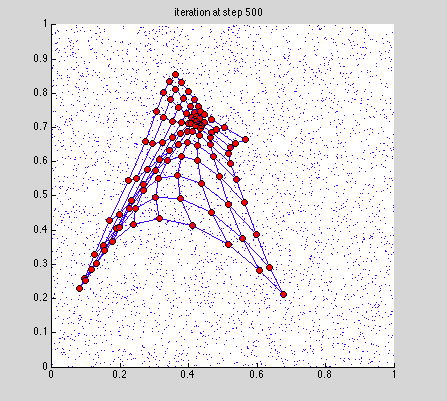


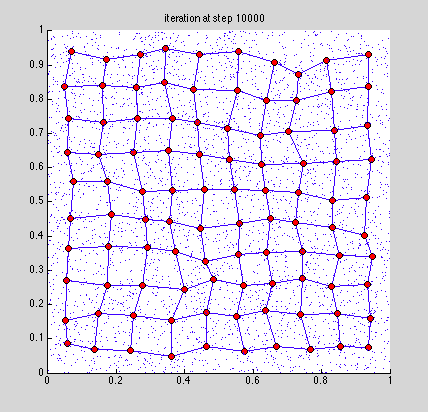
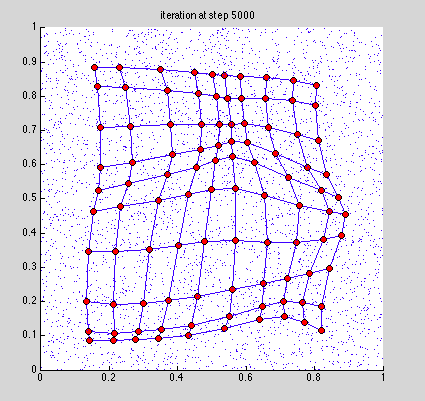












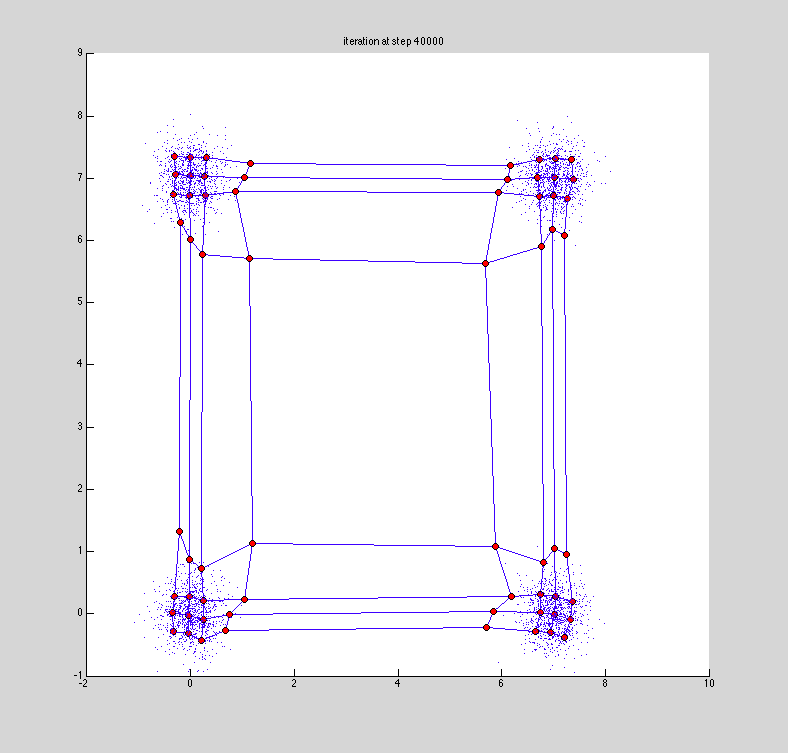
P2.

(a)

For this Kohonen SOM, I use 40000 steps and set learning rate as variable according to steps. Initial learning rate is as high as 0.05; while when steps approaches 40000 step, rate decreases uniformly to 0.005. Therefore the equation is like rate=init\_rate -(init\_rate –last\_rate)\*current#\_step/(total#\_step), where init\_rate=0.05, last\_rate=0.005, total#\_step=40000.

Initial weight is set to random floats between 0 to 1.

Final weight vectors is like this:



The resulting weights show the structure of 4 clusters, as we expected.

To get a first feel of PE’s input mapping number, I use following NN toolbox methods:

net=selforgmap([8 8])

net=train(net, x\_rand)

plotsomhits(net,x\_rand)

and get following figure:

