Optimizing the Topology of Complex Neural Networks

Fei Jiang, Hugues Berry and Marc Schoenauer
Project-team ALCHEMY & TAO
INRIA & LRI, Orsay France

UNIVERSITÉ PARIS-SUD 11
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA FUTURS

Outline

- Introduction
  - Related works
  - Handwritten digit recognition using Kohonen maps (SOM)
- Experiments
  - Direct problem : Given a SOM network , observe its performances with different topologies.
  - Inverse problem: Given a task to perform, find the optimal topology of the SOM network.
- Conclusion

Objective

- Long term goal: Investigate computing architectures involving a large number of computer units.
- First step: Given a large size ANN and a task, study the relationship between:
  - its topology (Small-world, Scale-free)
  - its performance (Fitness, Robustness)
- This talk:
  - ANN: Self organizing map (SOM / Kohonen maps)
  - Task: Recognition of hand written digits

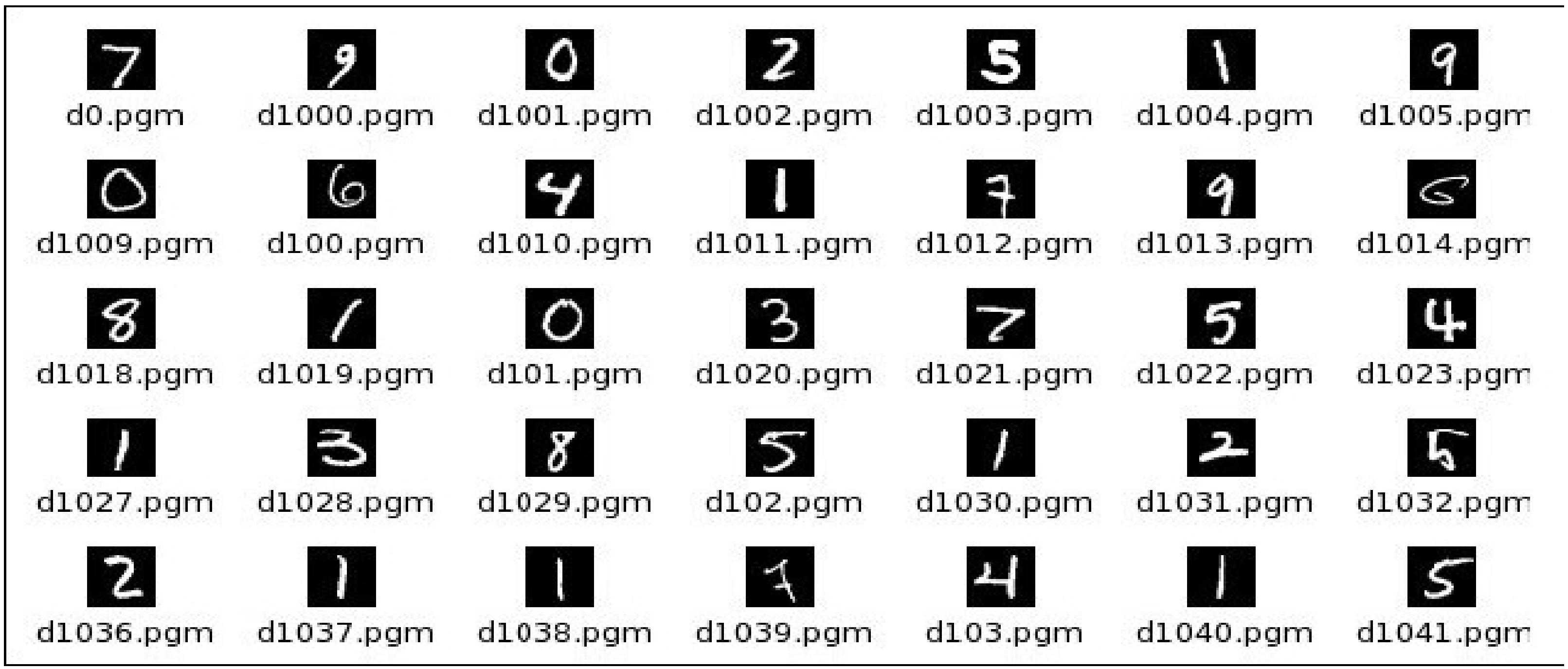
Related works

- Introducing a “small-world” topology in a ‘monolayer perceptron’ increases the learning rate of the network [1].
- Using evolutionary algorithms to optimize the topology of 1D-cellular automata networks (majority problem), the optimal topologies were systematically close to “small-world” ones. [2]
- Optimizing the topology of boolean networks, the evolution of networks with random topology was very different from networks with “scale-free” topology[3].

- [1] Simar et al. (2005) Phys. Lett. A 336:8-15.
- [2] Tomassini et al. (2005) Complex Systems 15:261-284.
- [3] Oikonomou & Cluzel (2006) Nature Physics 2:532-536.

Recognition of hand written digits

- MNIST : (28 * 28 pixels, 255 grey levels)
- Training set : 60.000, Test set : 10.000



Yann Le-Cun (Courant Institute, NYU) and Corinna Cortes (Google Labs, New York).

Classical unsupervised SOM Learning

1 For each iteration, chose uniformly one example (I) as input.

2 For each neuron i, compute its distance to (I)

$$d_i = \sum_{j=1}^{M} (I_j - W_{ij})^2$$

3 Find the BMU and update its weights, k = Arg min (d_i)

4 Update the weights of BMU's neighbours

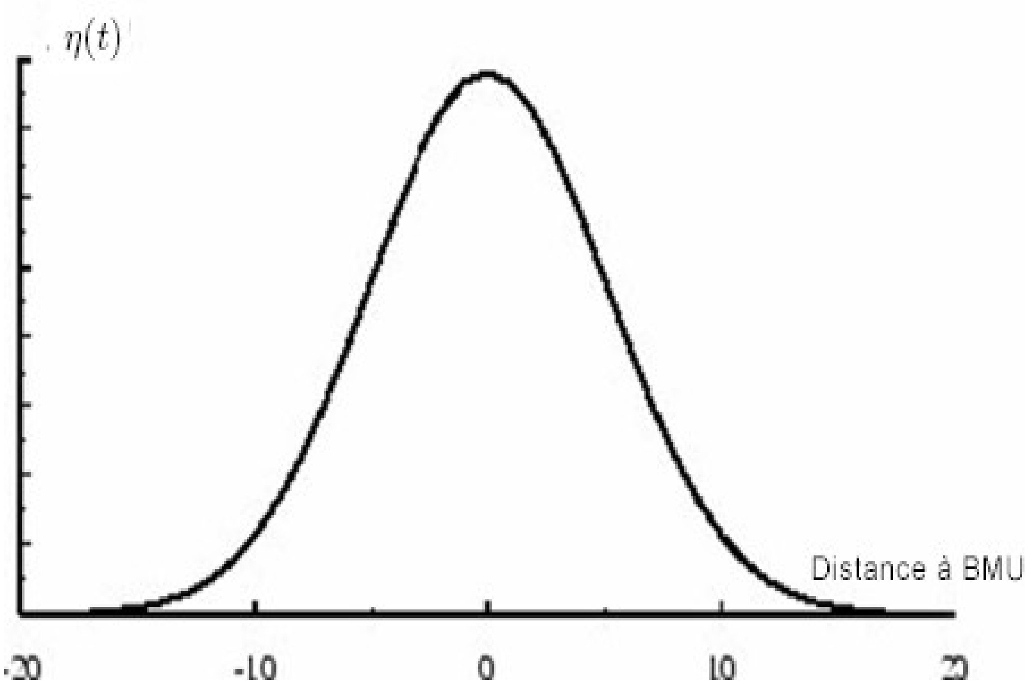
$$W_{kj}(t+1) = W_{kj}(t) + \eta(t) \times (I_j(t) - W_{kj}(t))$$

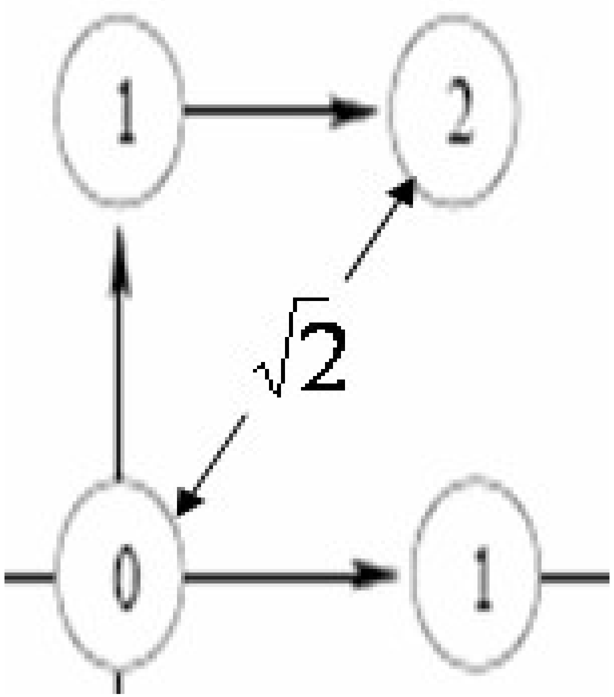
BMU : B*est Matching Unit*

Change euclidean distance to graph distance

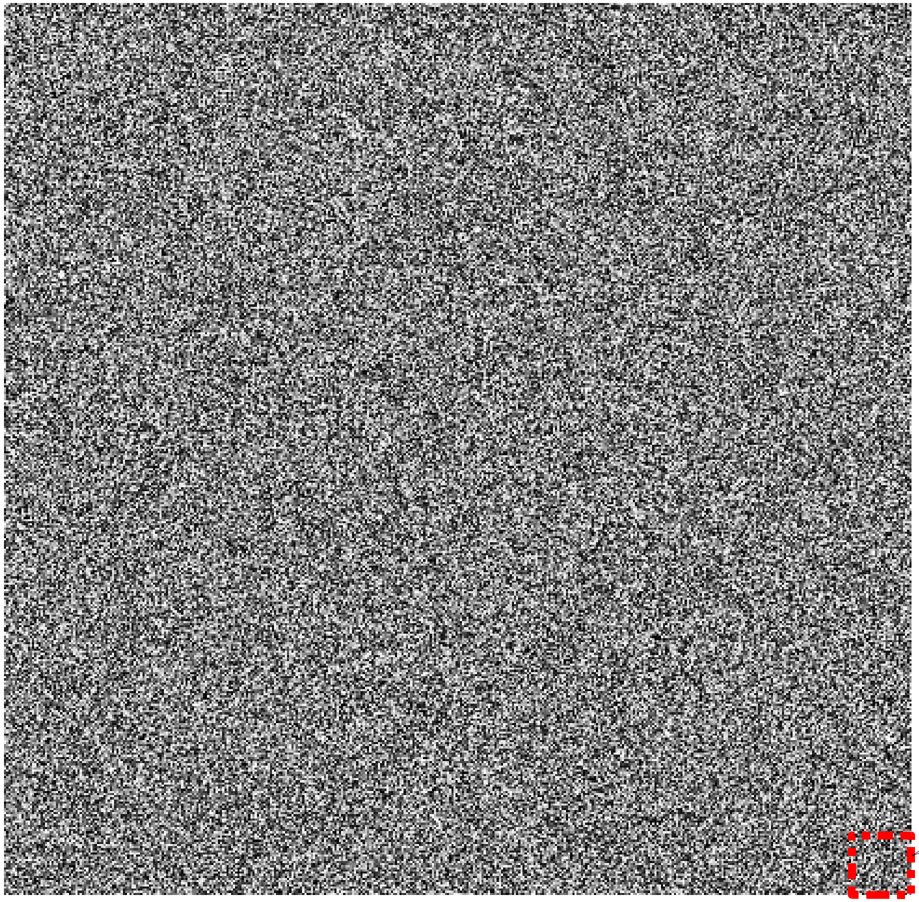
$$W_{kj}(t+1) = W_{kj}(t) + \eta(t) \times (I_j(t) - W_{kj}(t))$$

4 *h* is a Gaussian function of the **graph distance** from the BMU. Its variance decays during learning



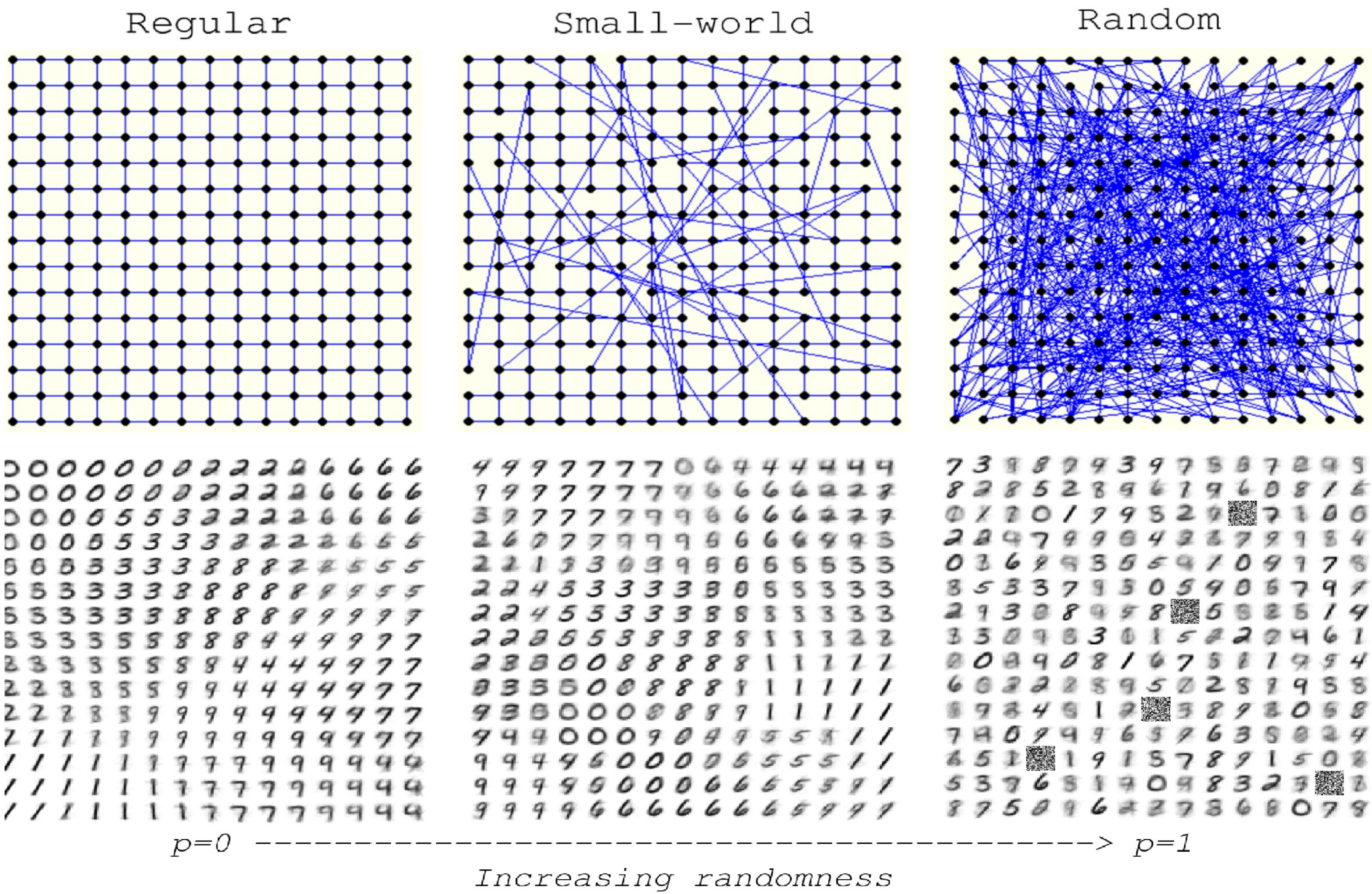


Visualisation of the neurons' weights



Example: 15*15 neurons

Initialisation


Regular
Small-world
Random
p=0 --> p=1
Increasing randomness

Labelling & Classifying (supervised part)

1 After learning, give to each neuron the label of the image class for which it was most often BMU.

2 Classification of an unknown example:

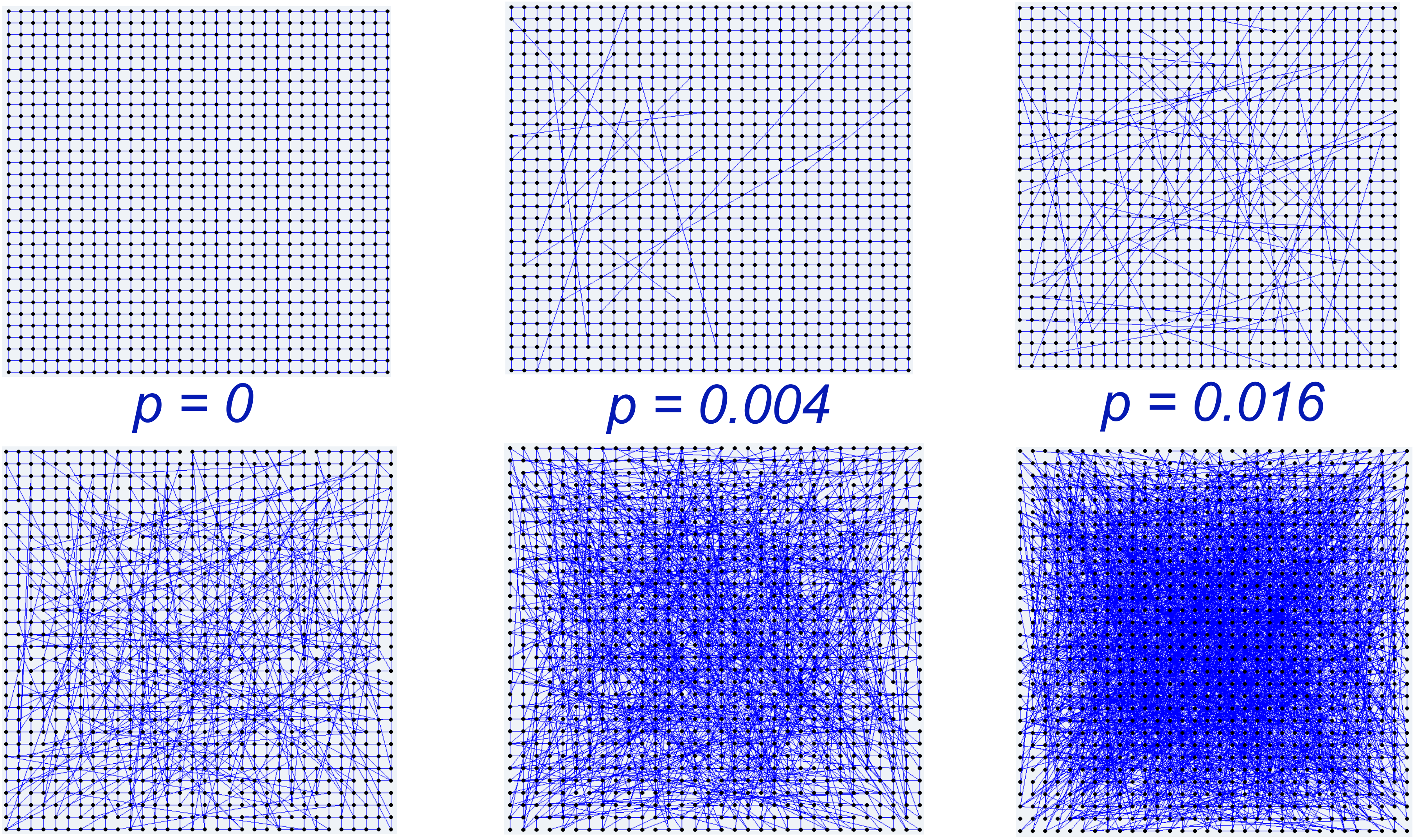
1. Find the neuron which has a minimum distance to this example.
2. Assign the neuron label to the example (guessed label)
3. Compare guessed label of the example to its real one.

3 Fitness of topology:

$$F = n_{err} / N_{test}$$

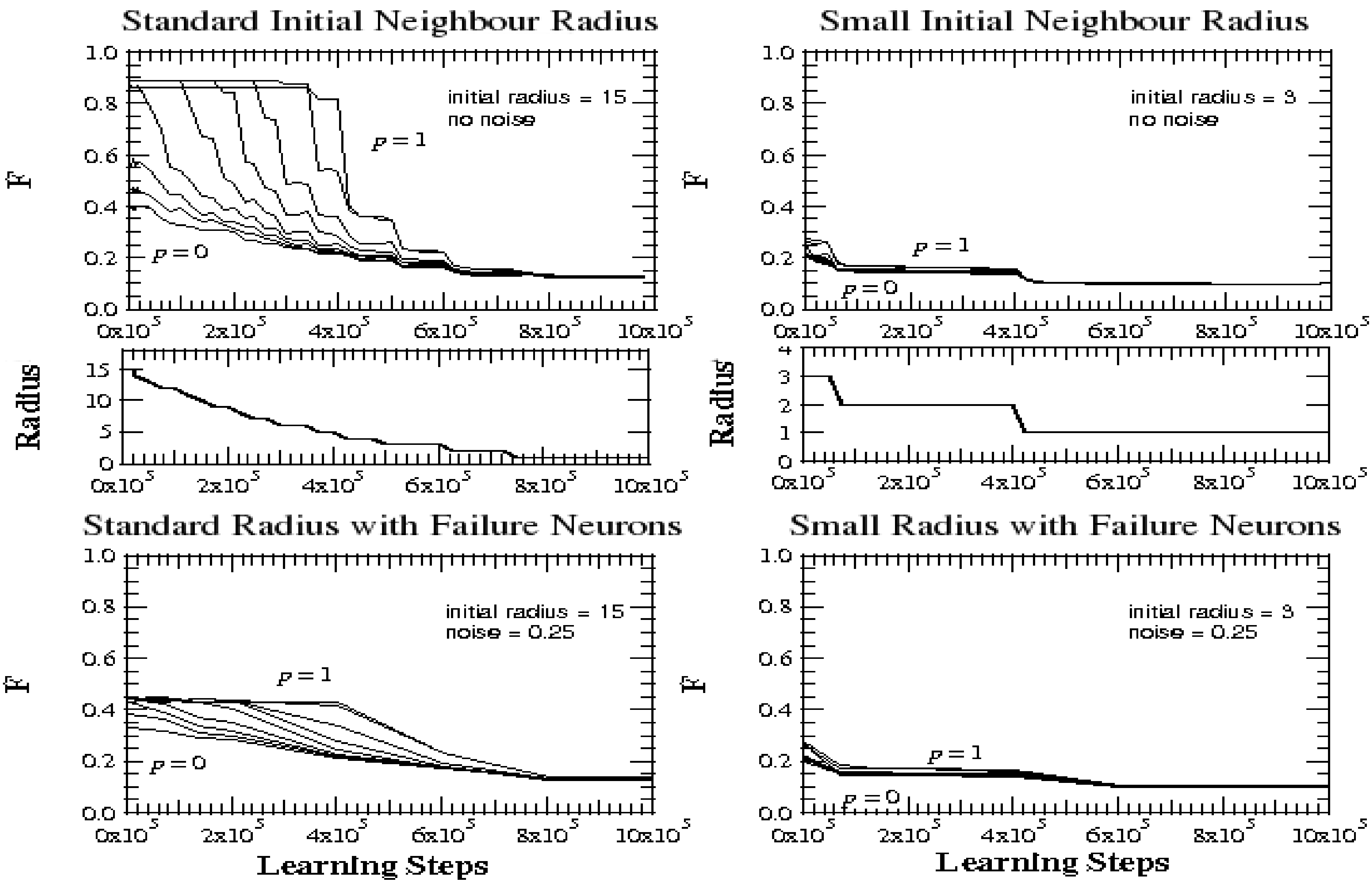
Direct Problem: experimental parameters

- SOM :
  - Size : 32*32, Learning steps : 1 million
  - Full MNIST database
- Rewiring probability:
  - p∈[0, 0.001, 0.002, 0.004, 0.008, 0.016, 0.032, 0.064, 0.128, 0.256, 0.512, 1.000]
- Noise: (randomly chosen) 25 % of the neurons **deactivated** at each generation

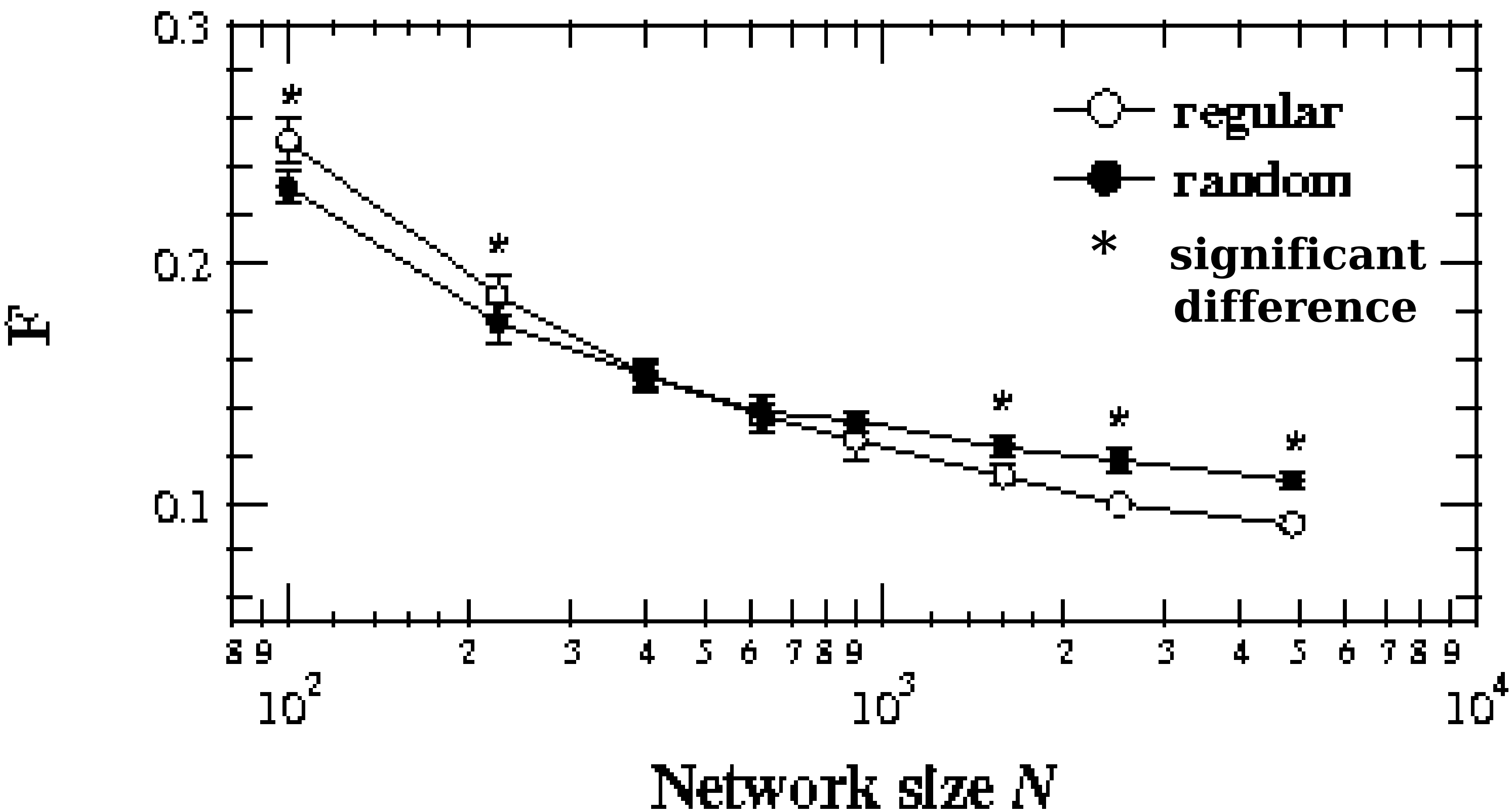


$p = 0$ $p = 0.004$ $p = 0.016$

$p = 0.064$ $p = 0.256$ $p = 1$


Standard Initial Neighbour Radius
initial radius = 15
no noise
$p=1$
$p=0$
F
Radius
Small Initial Neighbour Radius
initial radius = 3
no noise
$p=1$
$p=0$
F
Radius
Standard Radius with Failure Neurons
initial radius = 15
noise = 0.25
$p=1$
$p=0$
F
Learning Steps
Small Radius with Failure Neurons
initial radius = 3
noise = 0.25
$p=1$
$p=0$
F
Learning Steps
0x10^5 2x10^5 4x10^5 6x10^5 8x10^5 10x10^5

Performance of network size

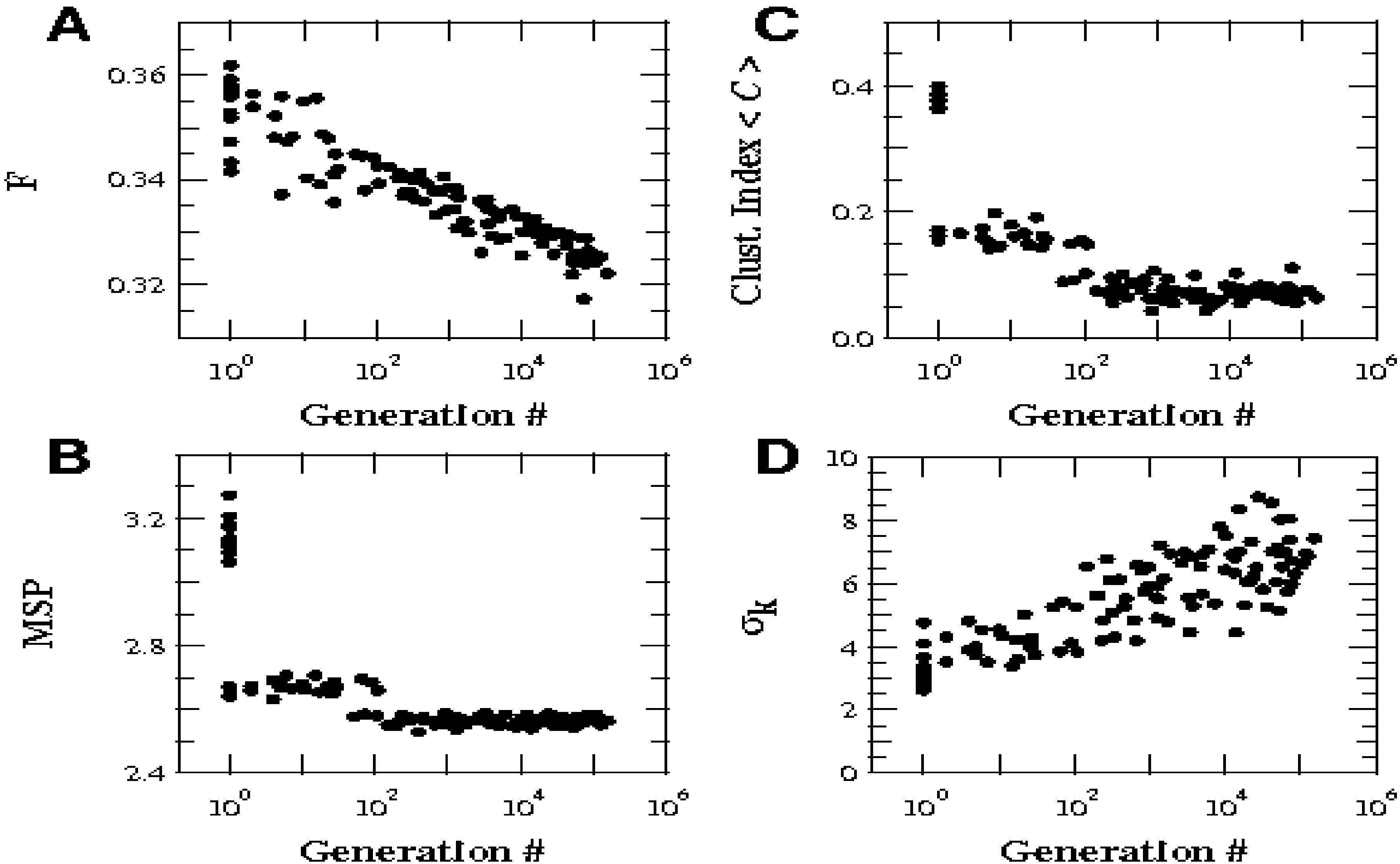


Inverse problem

- SOMs :
  - Size : 10*10, Learning Iterations : 10 000
  - Training set : 2 000, Test set : 5 000
- Objective: optimize the topology for the classification task
  - Fitness (F)
- During evolution, measure the variation of:
  - Clustering (<C>)
  - Mean shortest path (MSP)
  - Connectivity distribution (P(k))

Steady-State Genetic Algorithm

1. Initialize 100 networks: start from 100 regular networks (4 links per neuron) then rewire them (p=5%)
2. For 200 000 “generations”:
   - Draw uniformly 2 networks, select the best one (N)
   - N' = mutation(N) (rewire C% of its links)
   - Rewiring Rate : C%, decays with generation
     - Start: C=30 %;
     - End: C=0.3 % (rewire 1 link only);
   - Compute fitness(N') using average of 5 different network initialisations
   - Draw uniformly 6 networks from the population, replace worse by N'


A
F
0.36
0.34
0.32
Generation #
B
MSP
3.2
2.8
2.4
Generation #
C
Clust. Index < C >
0.4
0.2
0.0
Generation #
D
σ_k
10
8
6
4
2
0
Generation #
10^0 10^2 10^4 10^6

Conclusions

Direct problem:

- Regular topology learns faster when compared with the small-world or random topologies.
- In the long run, no significant difference between the regular and the others.

Inverse problem :

- The final networks given by artificial evolution are 'more random' than the initial networks (MSP and Clustering are smaller)
- The variance of the connectivity distribution increases (more heterogeneous).

Thank you for you attention.

Any questions?