Self-organizing Maps (SOM)

- Overview
- Self-organizing Maps (SOM)
- Demonstration: Recognition of situations
- Learning Vector Quantization (LVQ)
- Demonstration: TSP solver

Overview of SOM

- The process of formation of a topologically ordered mapping from the signal space onto the neural network is defined by the Self-organizing map (SOM) algorithm.
- The “feature maps” thereby realized can often effectively be used for the preprocessing of patterns for their recognition, or, if the neural network is a regular two-dimensional array, to project and visualize high-dimensional signal space on such a two-dimensional display.
- As a theoretical scheme, on the other hand, the adaptive SOM processes, in a general way, may explain the organizations found in various brain structures.

Demonstrations

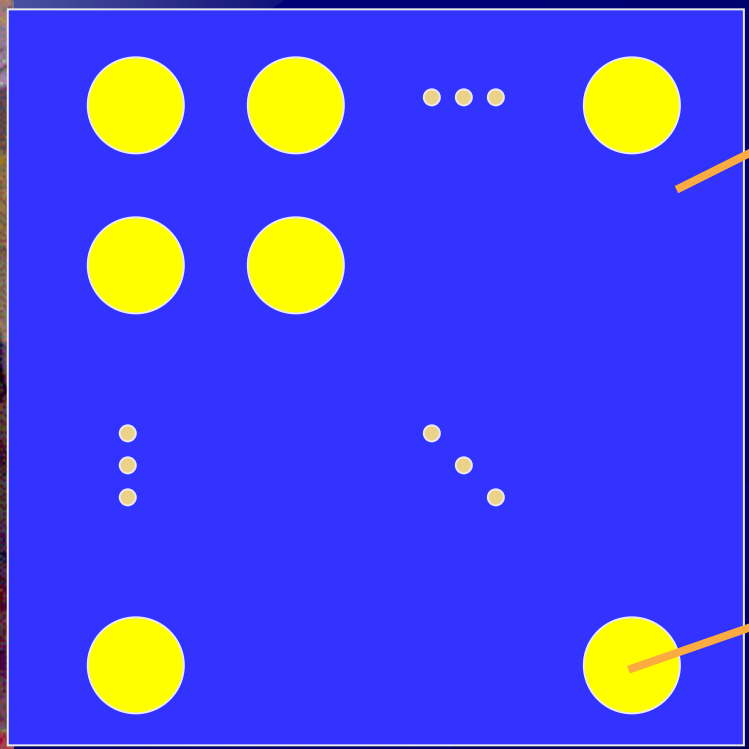
- Mapping “colors” onto 2D plane.
- A color is a vector (R, G, B).
- Map size: 10x10 units
- Initial state: randomized

SOM can categorize the colors (order 3 vectors) onto a 2D map without any teacher signal.

Each trial generates different map.

SOM array: a “map”

Input vector $x(t)$ (order n)



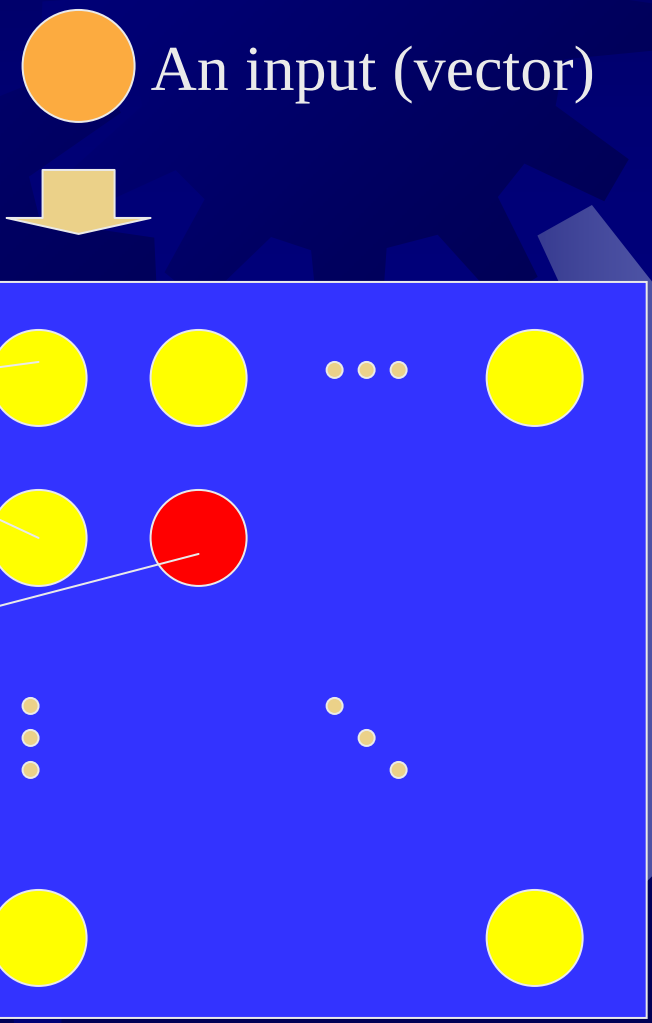
SOM array

$m \times m$ units are allocated

Each unit $m_i(t)$ is $n \times 1$ vector.

Training SOM = Changing $\mathbf{m_i(t)}$

Schematic view of SOM



By iterative training, near units has near vectors.

Winner node: an unit that has the smallest norm among the input.

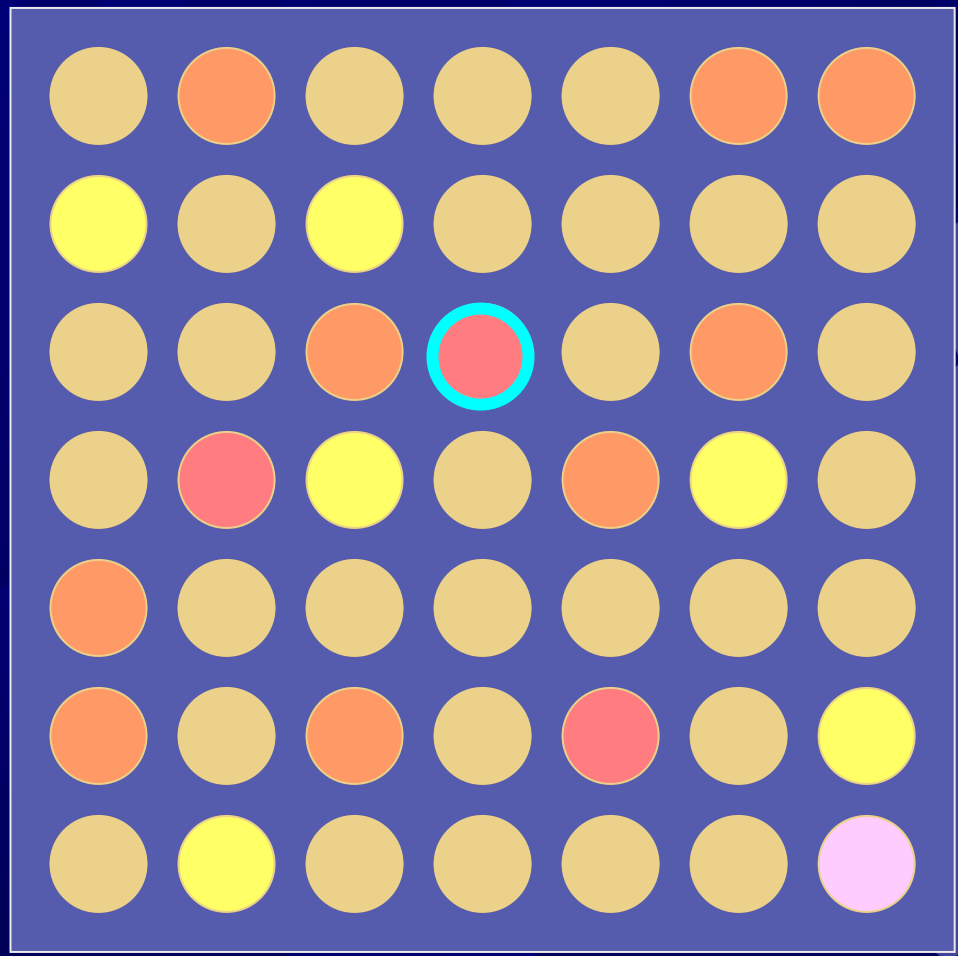
According to the winner node, we can categorize input signals.

To train a SOM

- We have to feed many data.
- We have to fix the learning time (step) T.
- We do not need any teacher signal.

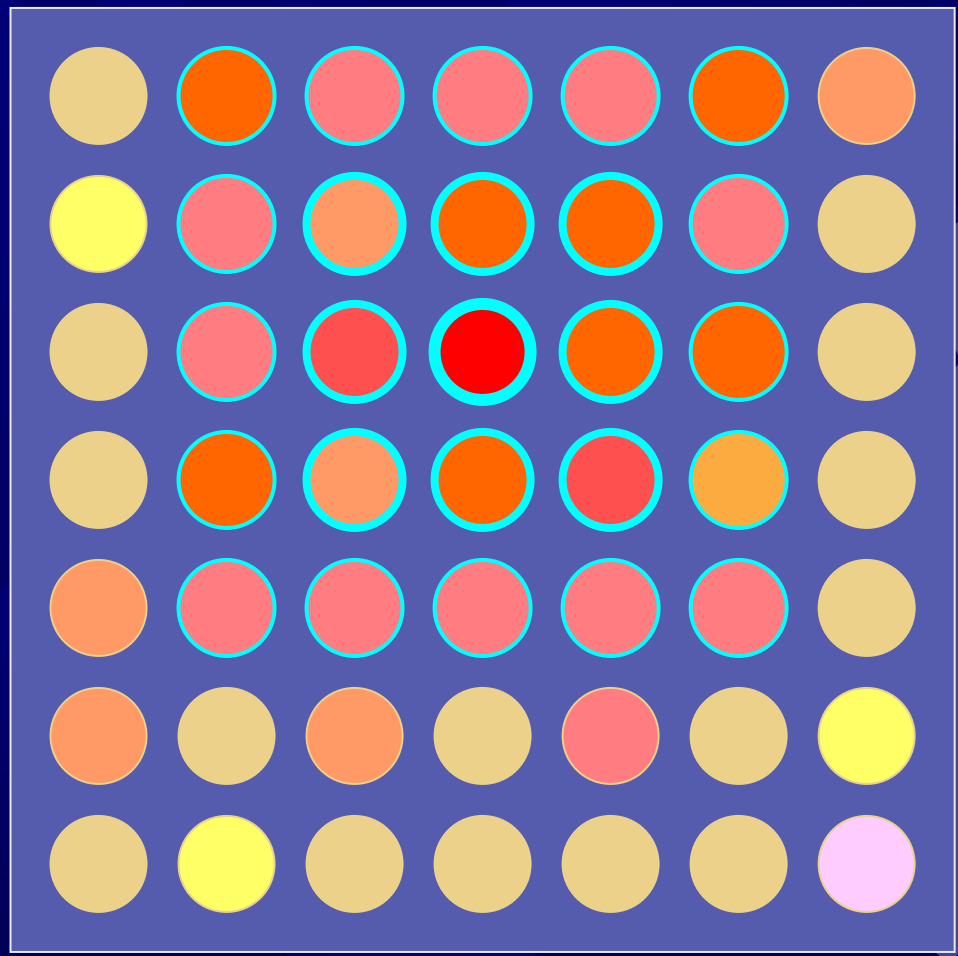
Procedure of training

New input
(ex. a color)



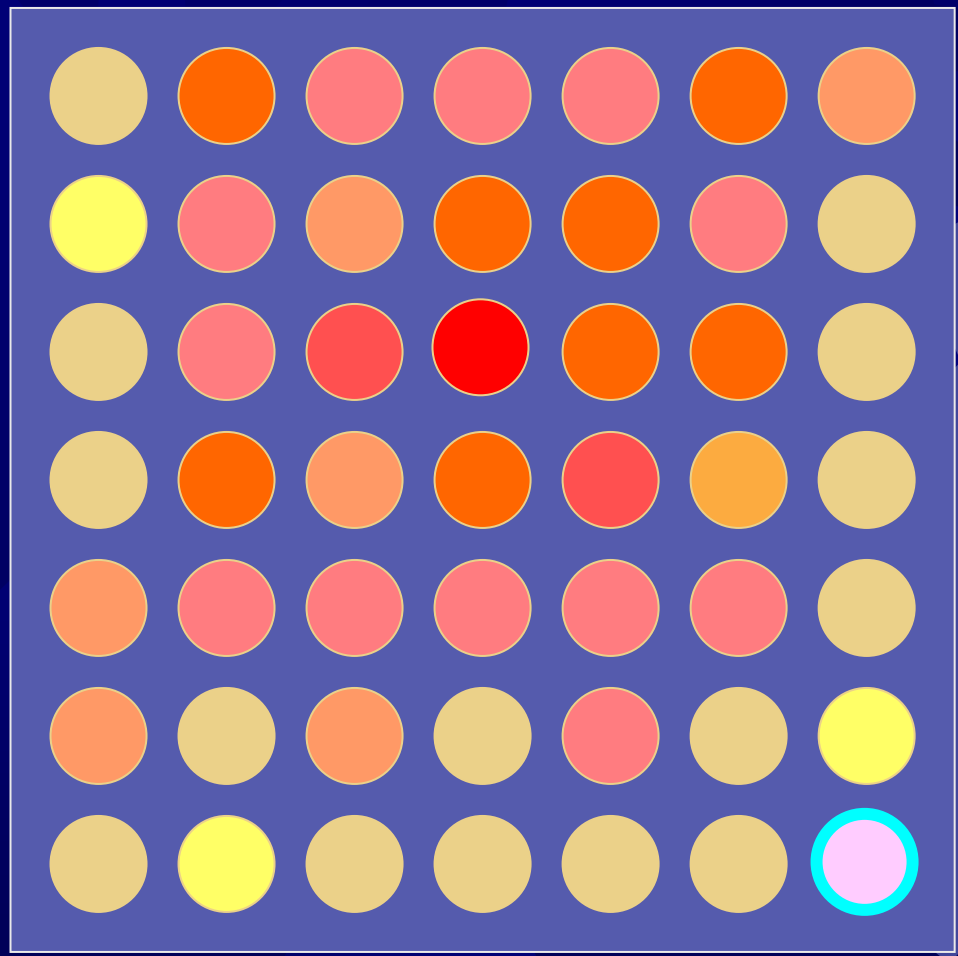
Find a “winner node” that has smallest difference between the new input.

Procedure of training

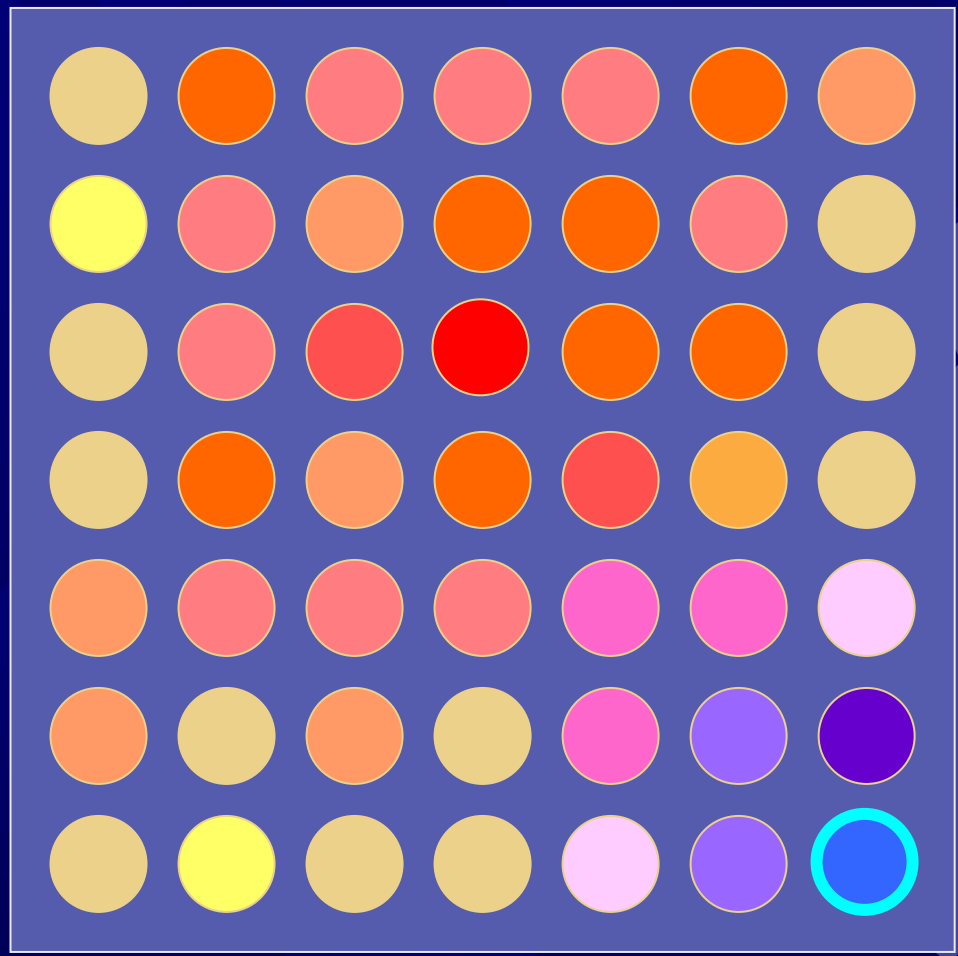


The winner node affects values of nodes located around it.

Procedure of training



Procedure of training



Training procedure of SOM

(1) Finding a winner node $m_c(t)$

$$\{m_c(t) \mid |x(t)-m_c(t)| = \min |x(t)-m_i(t)| \}$$

(2) Renew unit $m_i(t+1)$

$$m_i(t+1) = m_i(t) + h_{ci}(t)\,(x(t)-m_i(t))$$

Nodes that is located within N_c from the winner node $m_c(t)$

Nc becomes smaller

$$\begin{cases} h_{ci}(t) = \alpha(t) & (i \in N_c) \\ h_{ci}(t) = 0 & (\text{other cases}) \end{cases}$$

$$\alpha(t) = \alpha_0(1-t/T)$$

$$N_c(t) = N_c(0)(1-t/T)$$

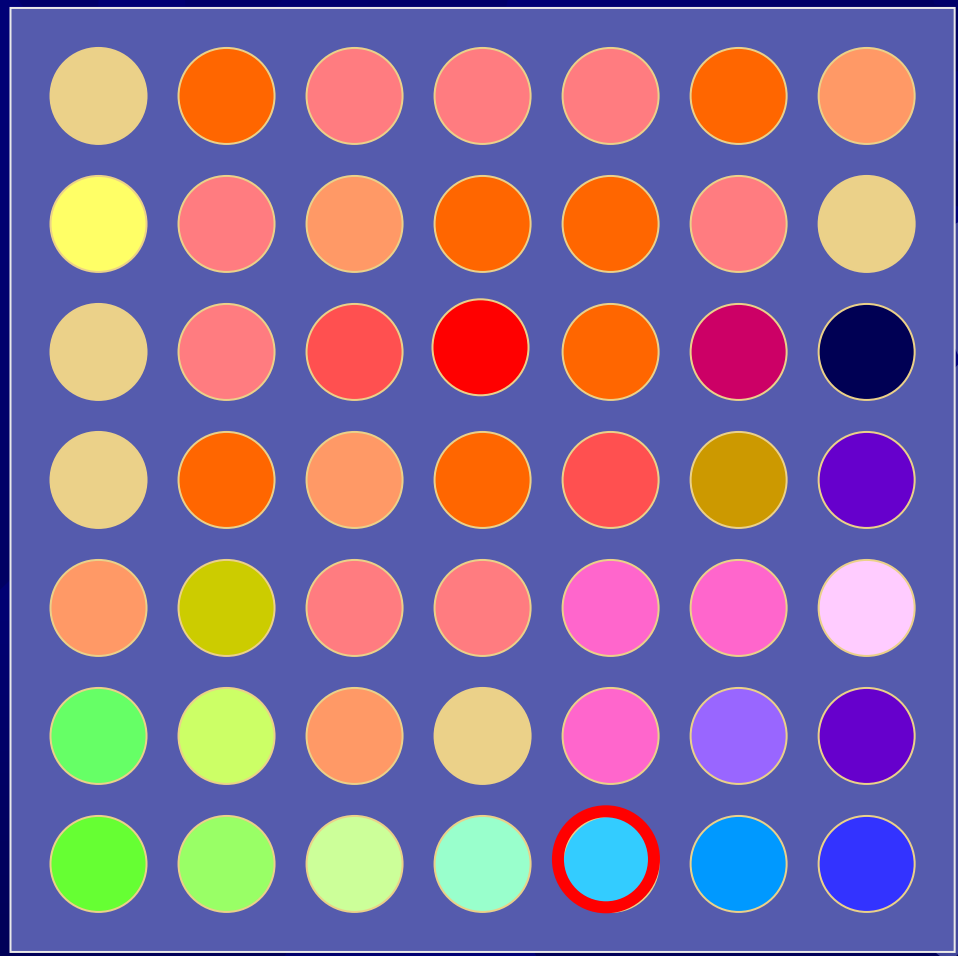
※Blue parameters are given in advance.

Using trained SOM

Unknown data

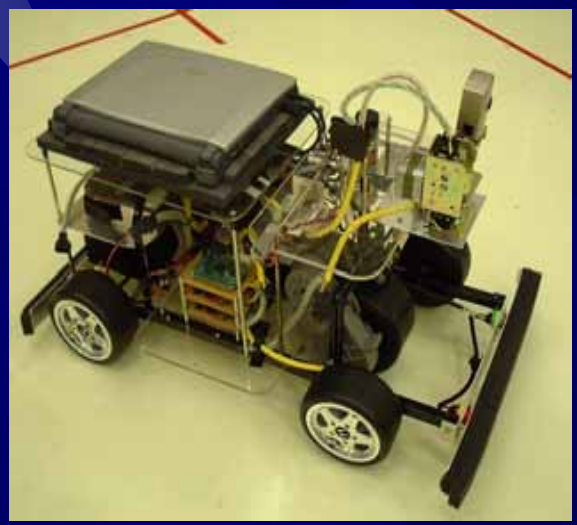
Finding a “winner node”, we can understand the nearest approximation for the input.

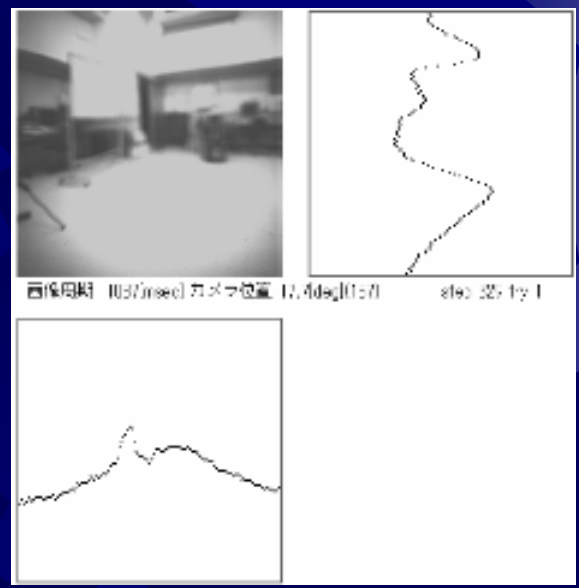
Locations of nodes indicate categories of known data.



Demonstration

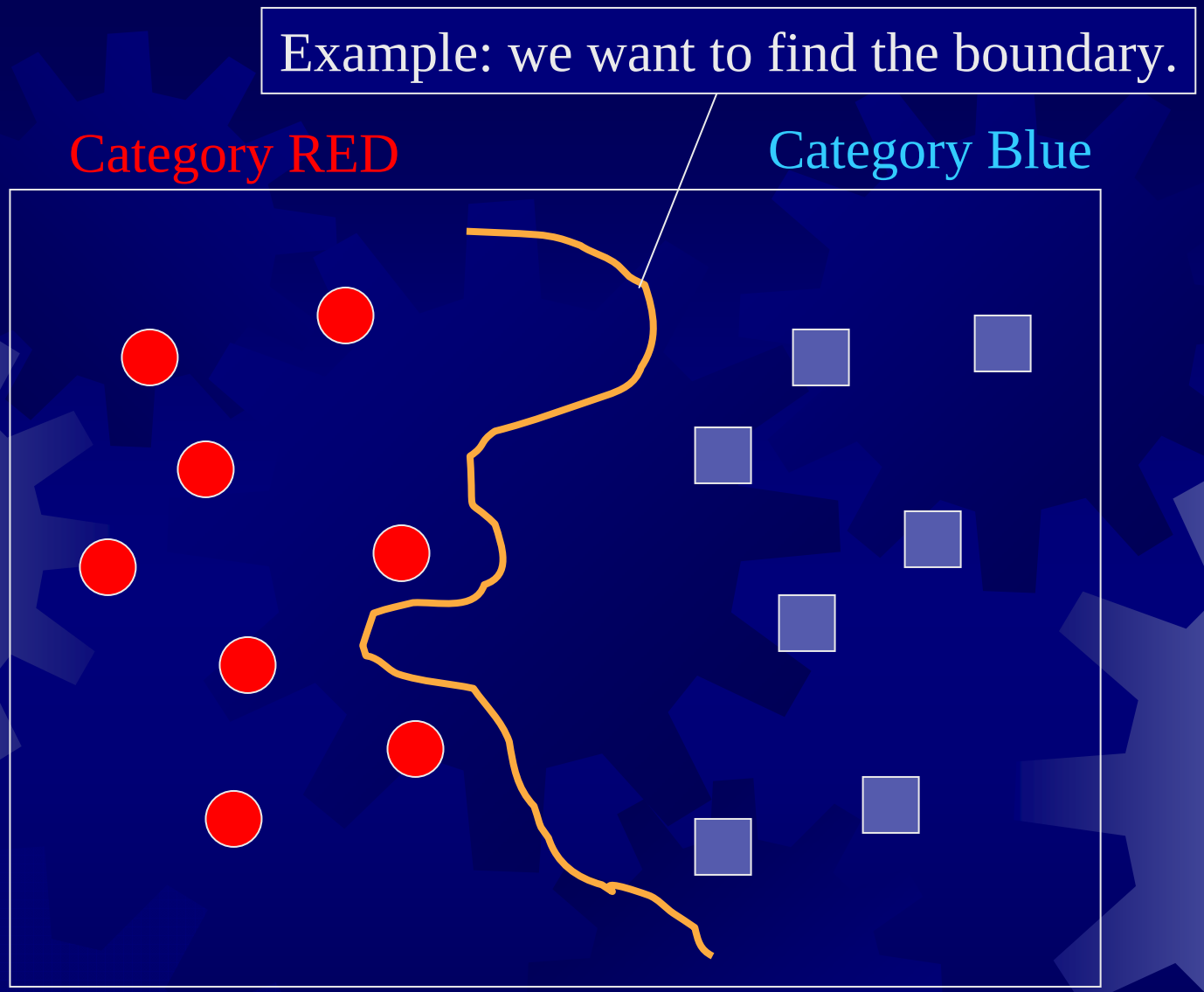
- SOM categorize vision data from a camera on the top of a mobile robot.
- The robot estimate its situation according to the trained result of SOM.





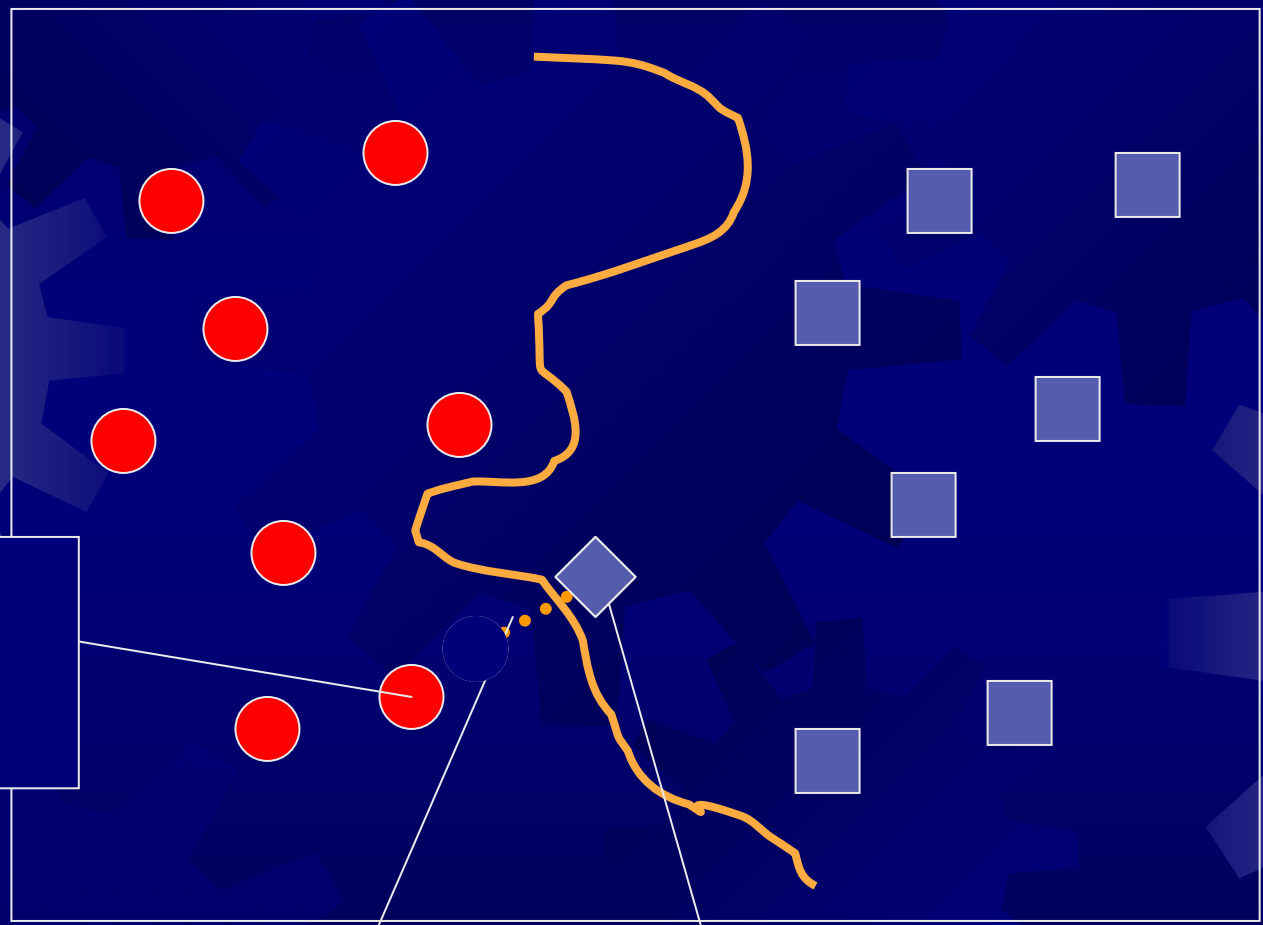
Automatic boundary generation (Learning Vector Quantization: LVQ)

- LVQ generates boundaries of classified categories on a SOM. In this method, the location of a unit will be changed.
- On LVQ procedure, a map itself represents a space of input vectors. So, each point in a map represents an input signal, and position of each node represents its status.
- Training procedures of LVQ need teacher signal. (SOM does not need any teacher)
- Not contents of a cell but its position will be changed during training procedure.



- Input: a point on the map. In this case, a map has large dimension.
- We can distinguish which category a point belongs to by a winner node (nearest node).

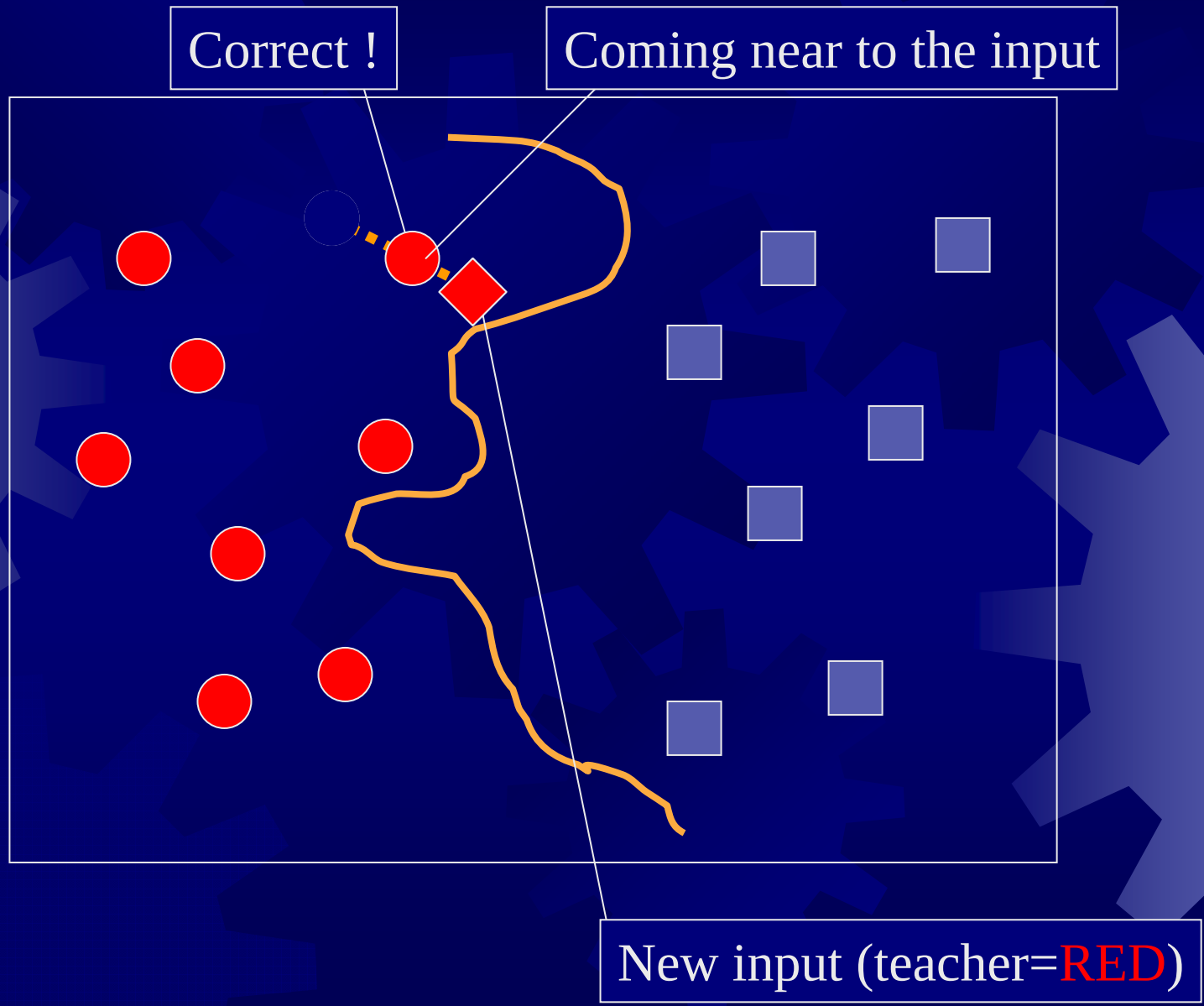
On LVQ procedure, a map itself represents a space of input vectors. So, each point in a map represents an input signal, and position of each node represents its status.

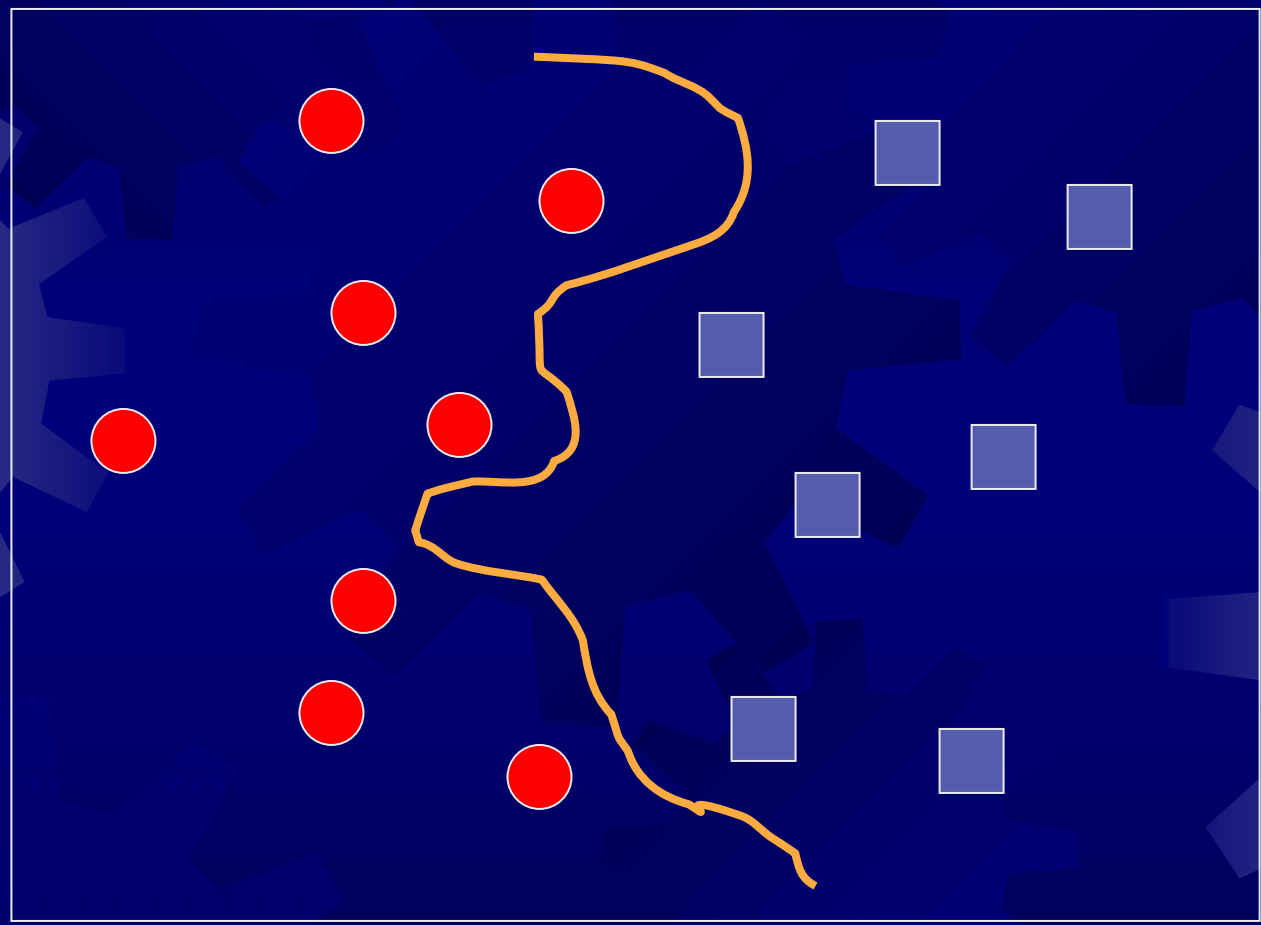


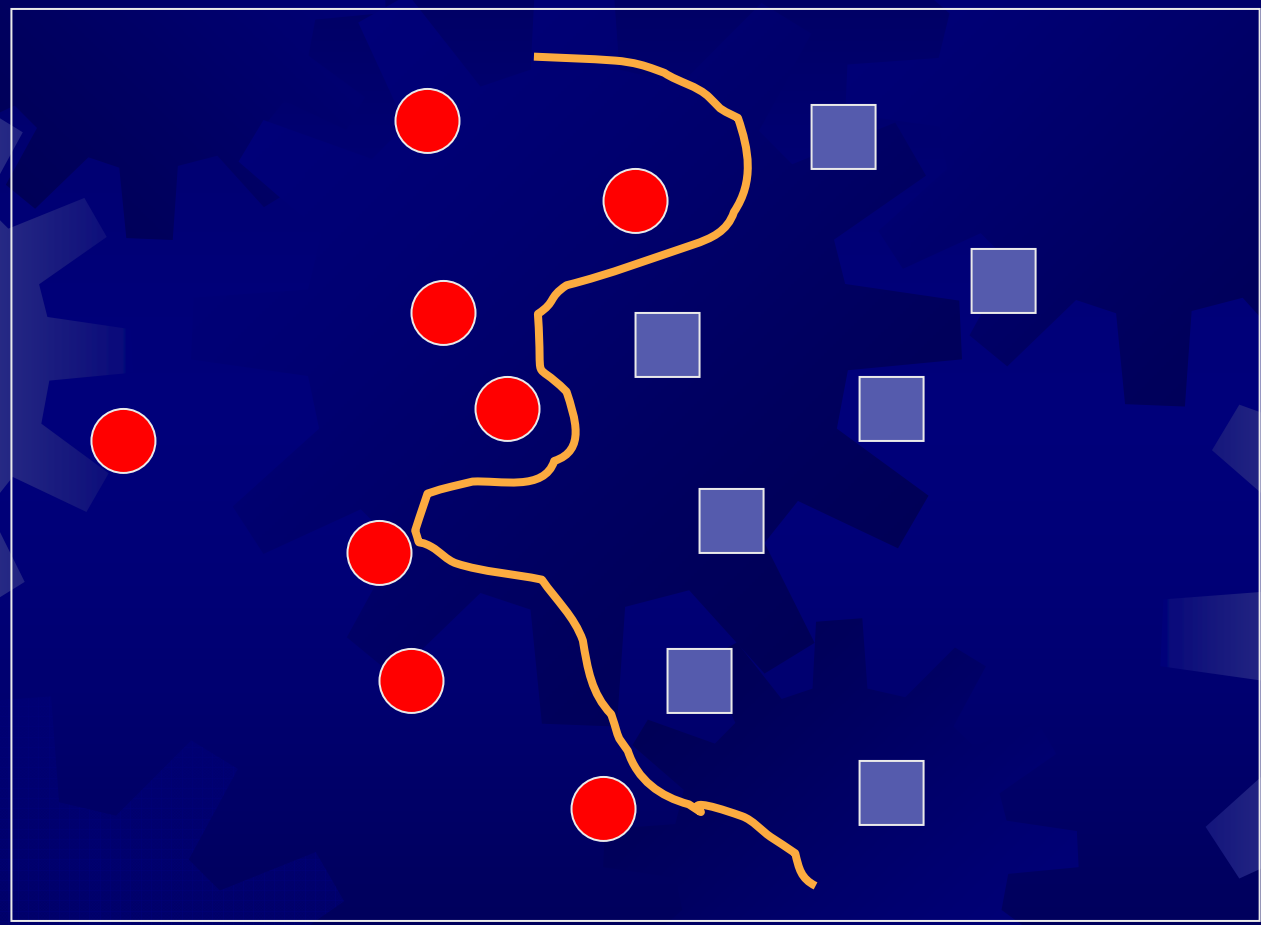
Get away from the input

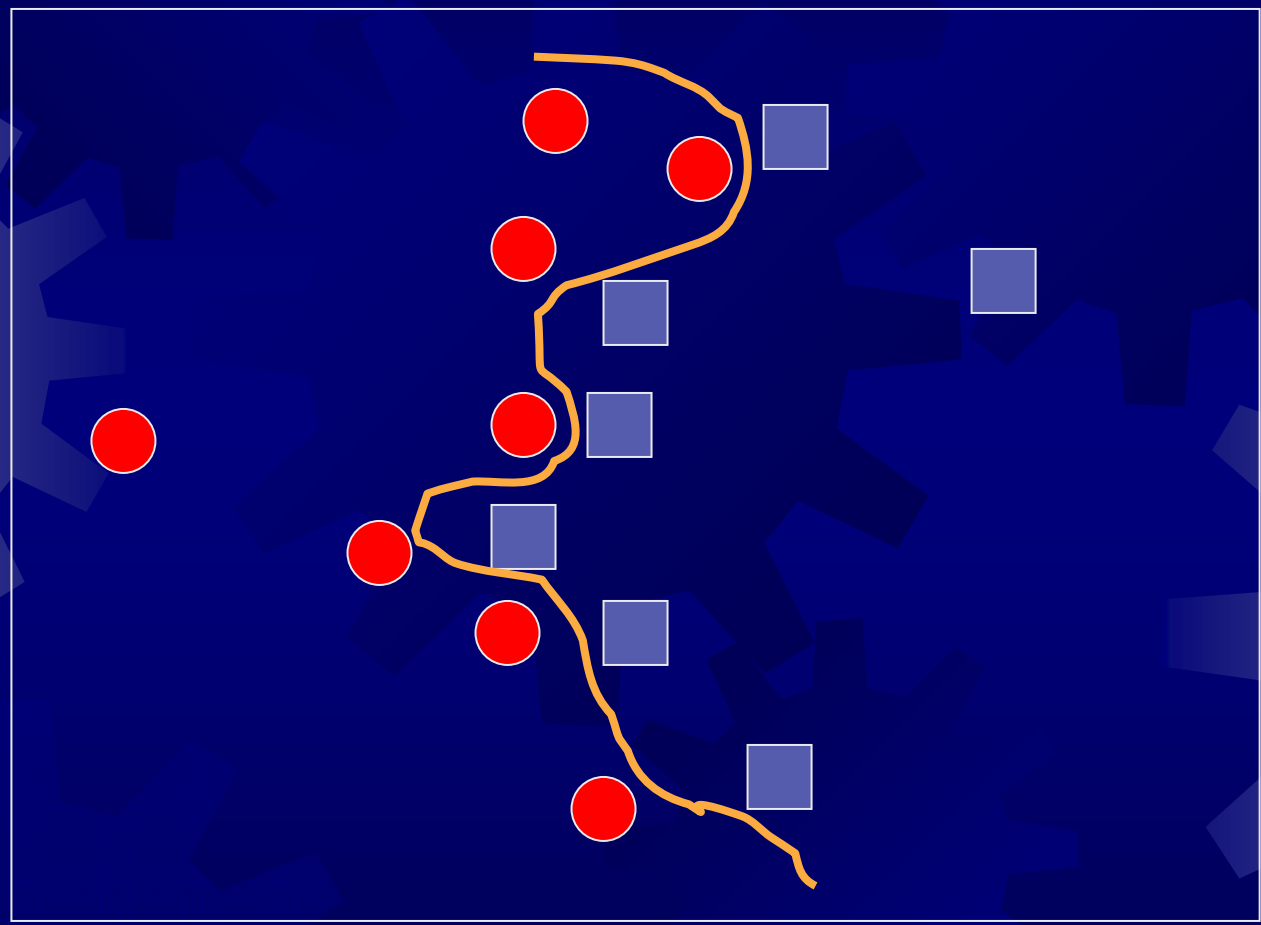
Winner (nearest) = RED: categorizing failed e (teacher)

If a winner node (nearest to a input) belongs to same category as a teacher, current categorizing is correct.









Training procedure of LVQ

Finding winner m_c to input x

Here, we denote that x and m_c belongs to S_r and S_s respectively.

Then, we renew the value (position) of the winner as following.

$\alpha(t)$ $(0<\alpha(t)<1)$ is a coefficient of learning.

Demonstration: Solving TSP by LVQ

TSP: Traveling Salesman Problem

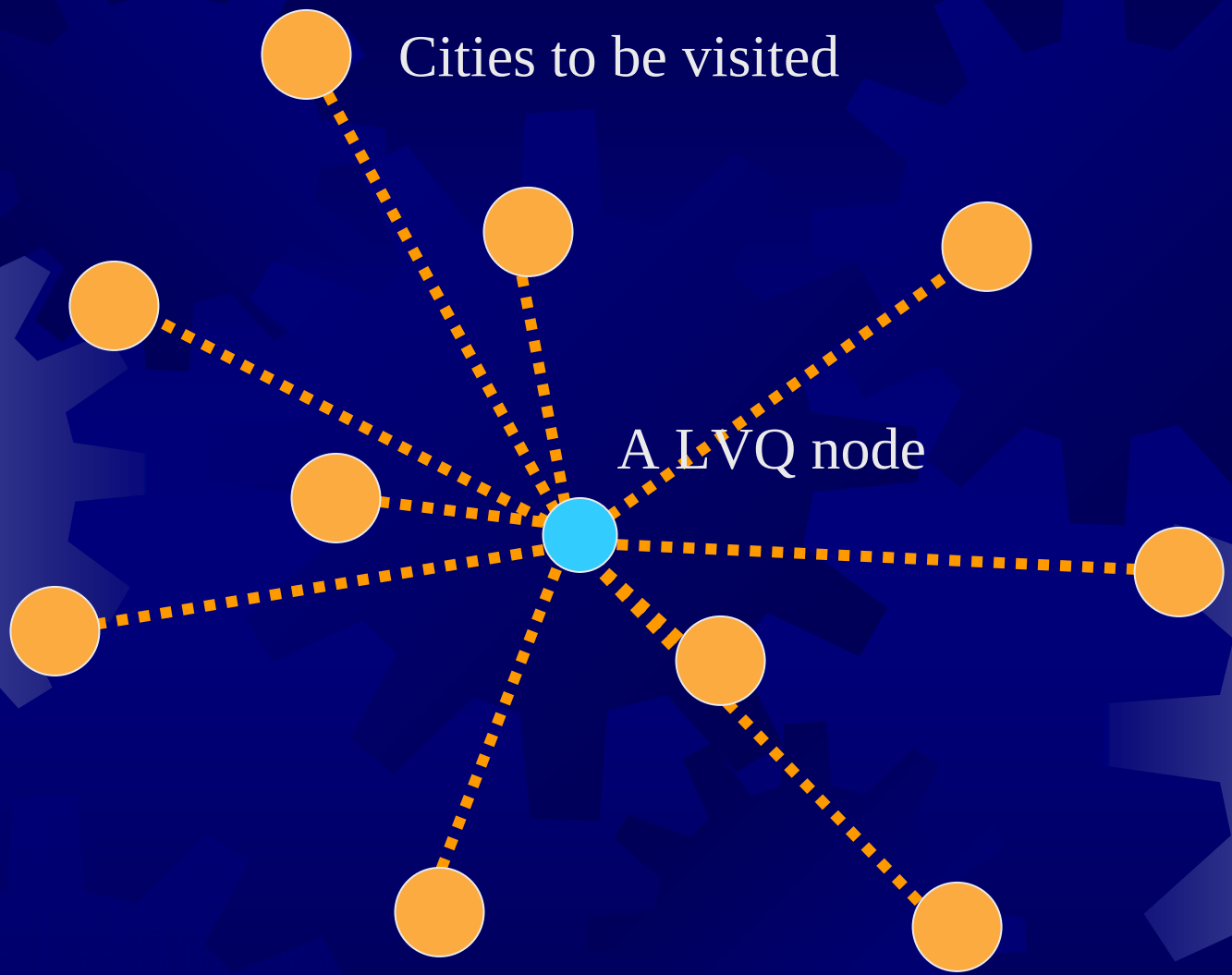
- Let us assume a salesman who starting from his home city, is to visit exactly once each city on a given list and then return home.
- A TSP problem is a problem such that he selects the order in which he visits the cities so that the total of the distances traveled in his tour is minimum.
- Assume that he knows, for each pair of cities, the distance from one to the other. Then he has all the data necessary to find the minimum, but it is by no means obvious how to use these data in order to get the answer.
- So, TPS is difficult problem.

Overview of TSP solver

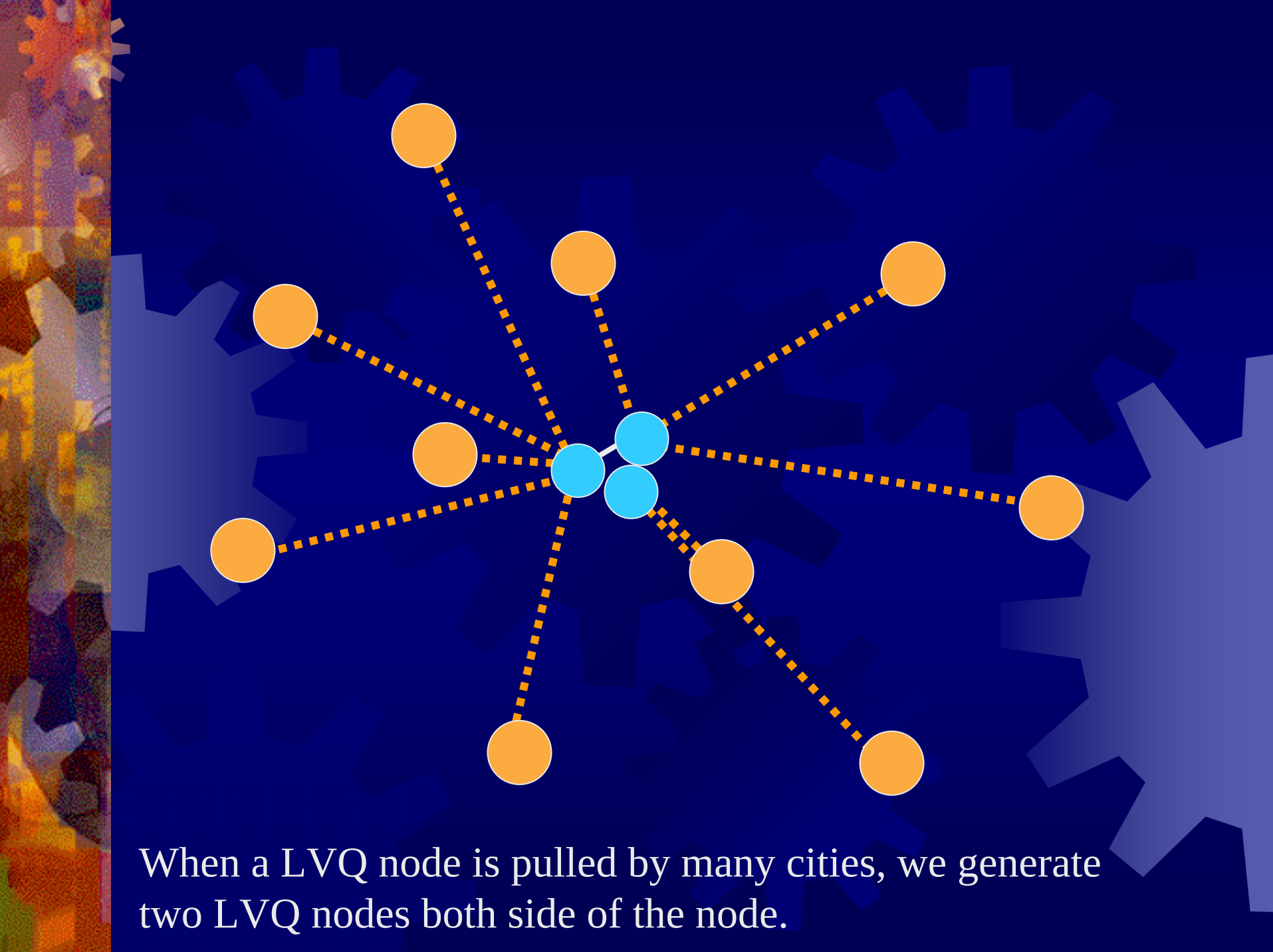
Input: locations (x,y) of cities to be visited.
We construct a “ring” of LVQ nodes.

By following rules, LVQ nodes covers cities successively.

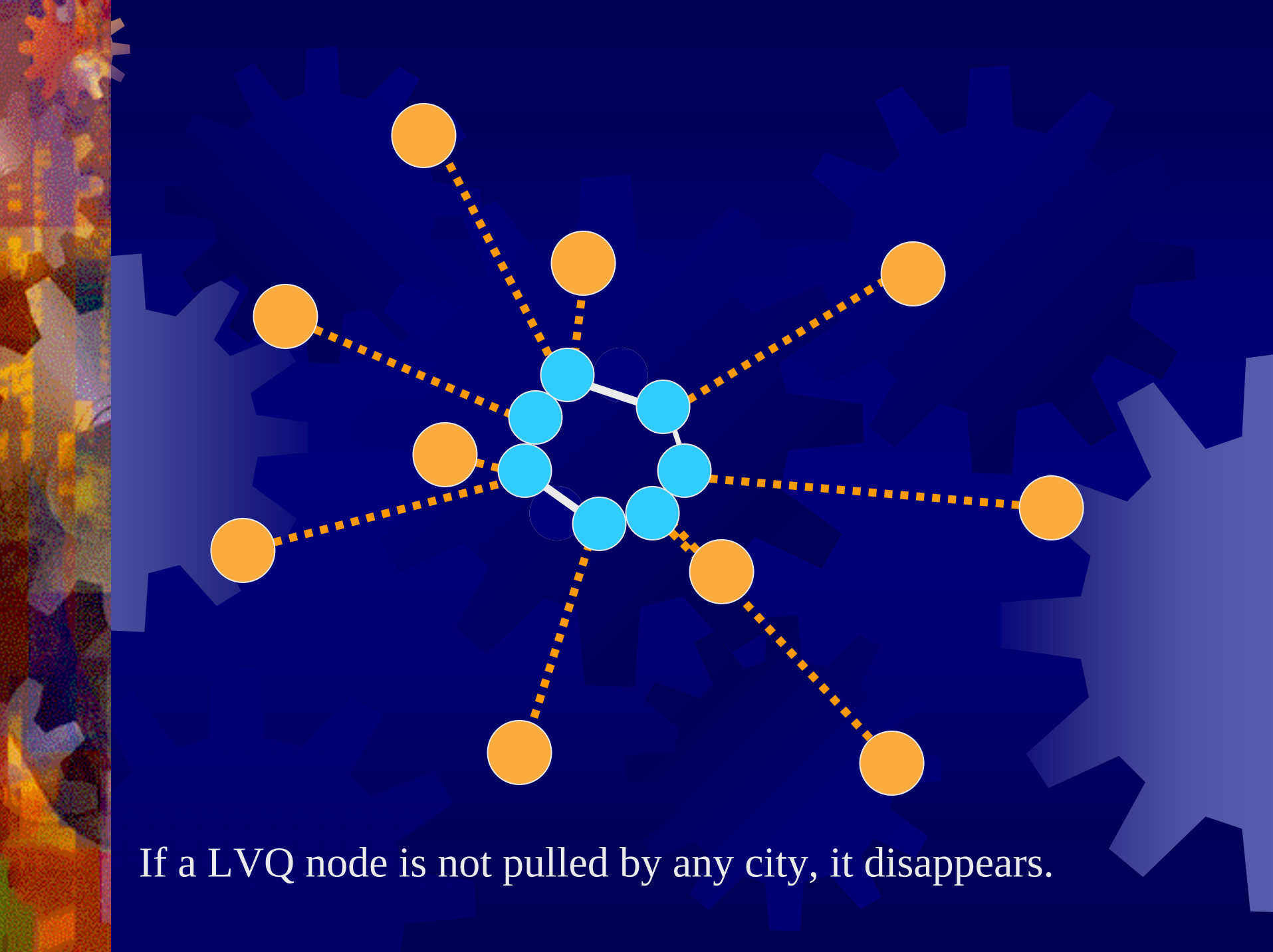
At the beginning, we put only one LVQ node.



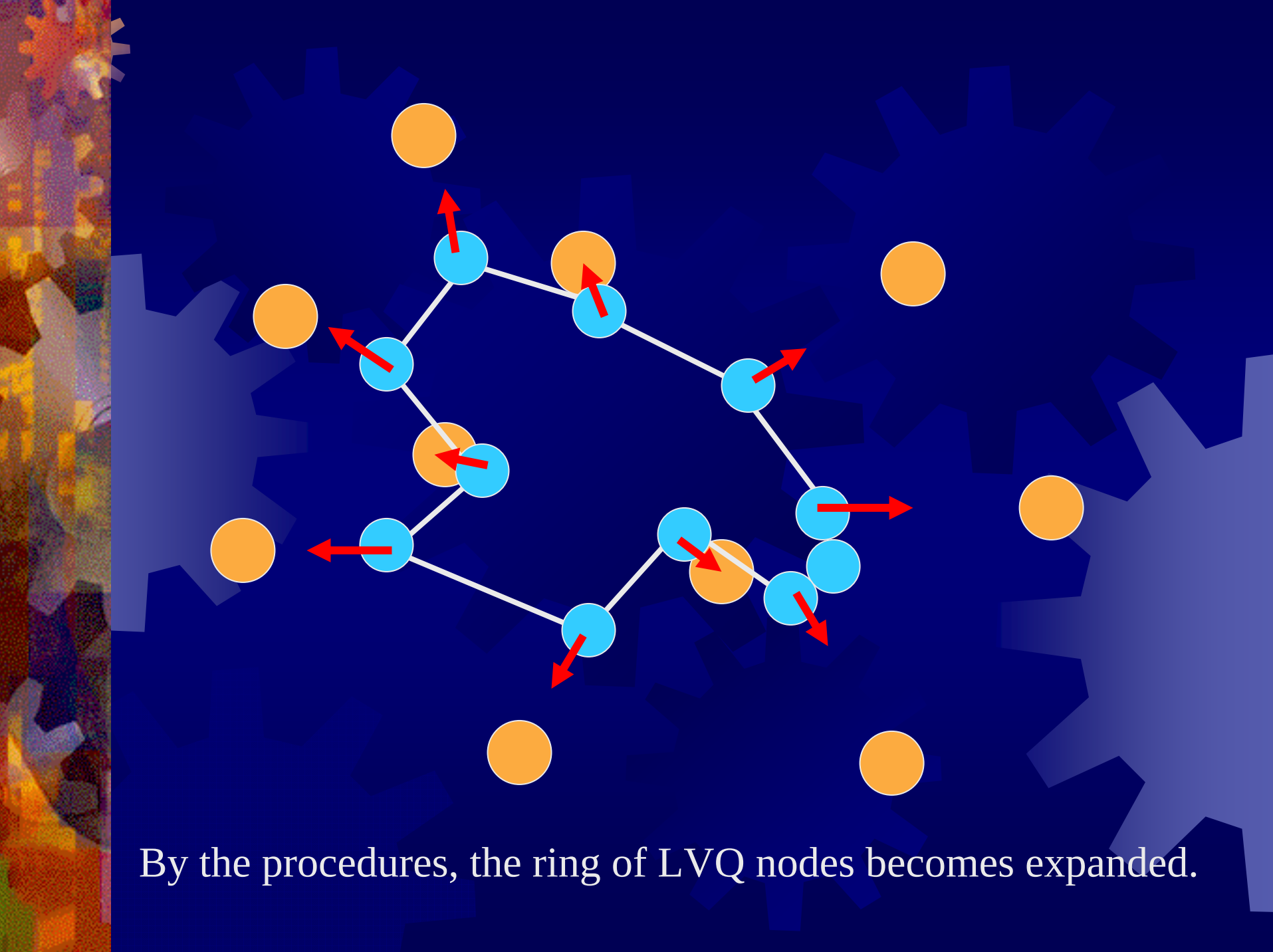
A city pulls the nearest LVQ nodes toward its position.



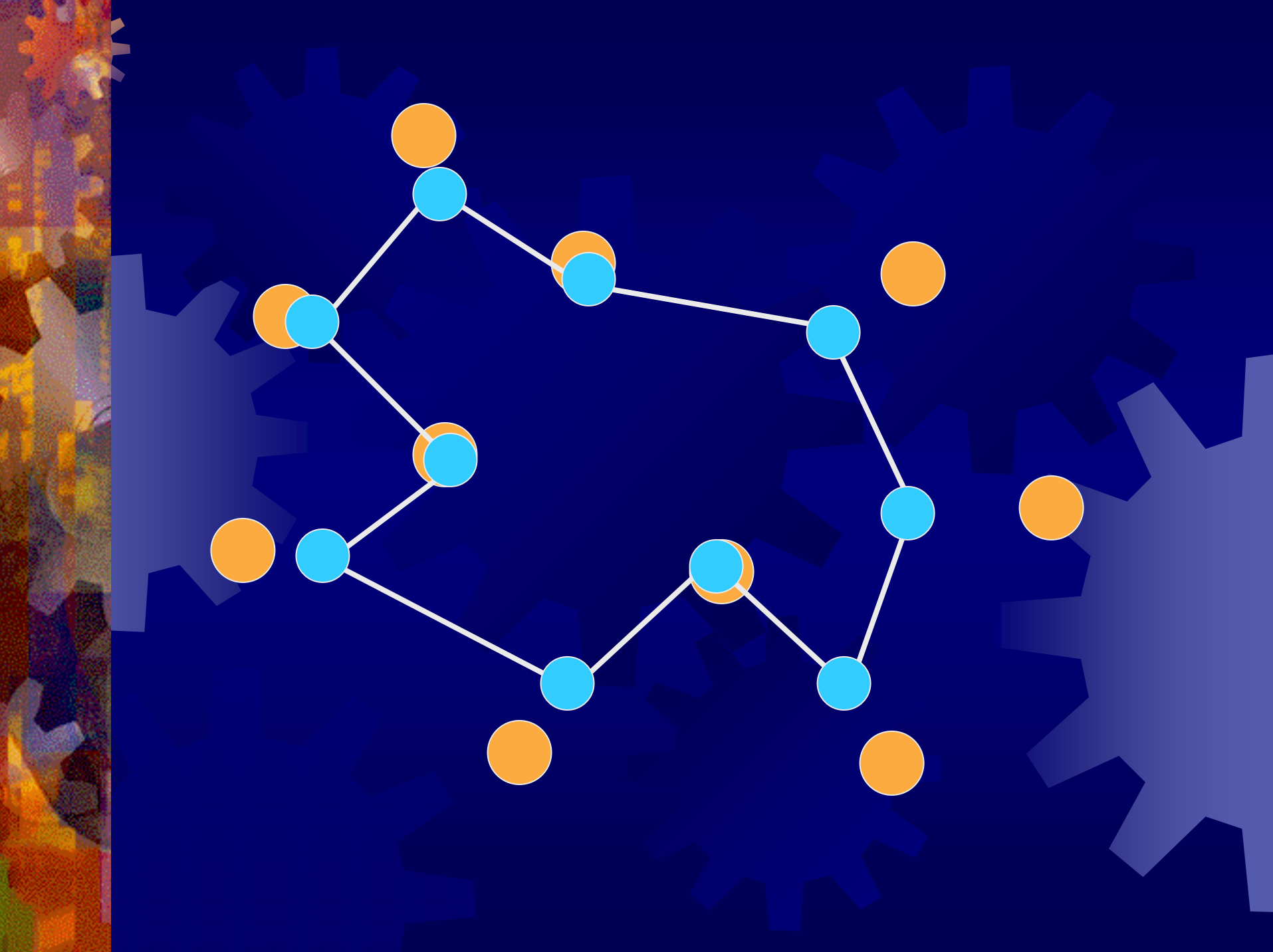
When a LVQ node is pulled by many cities, we generate two LVQ nodes both side of the node.

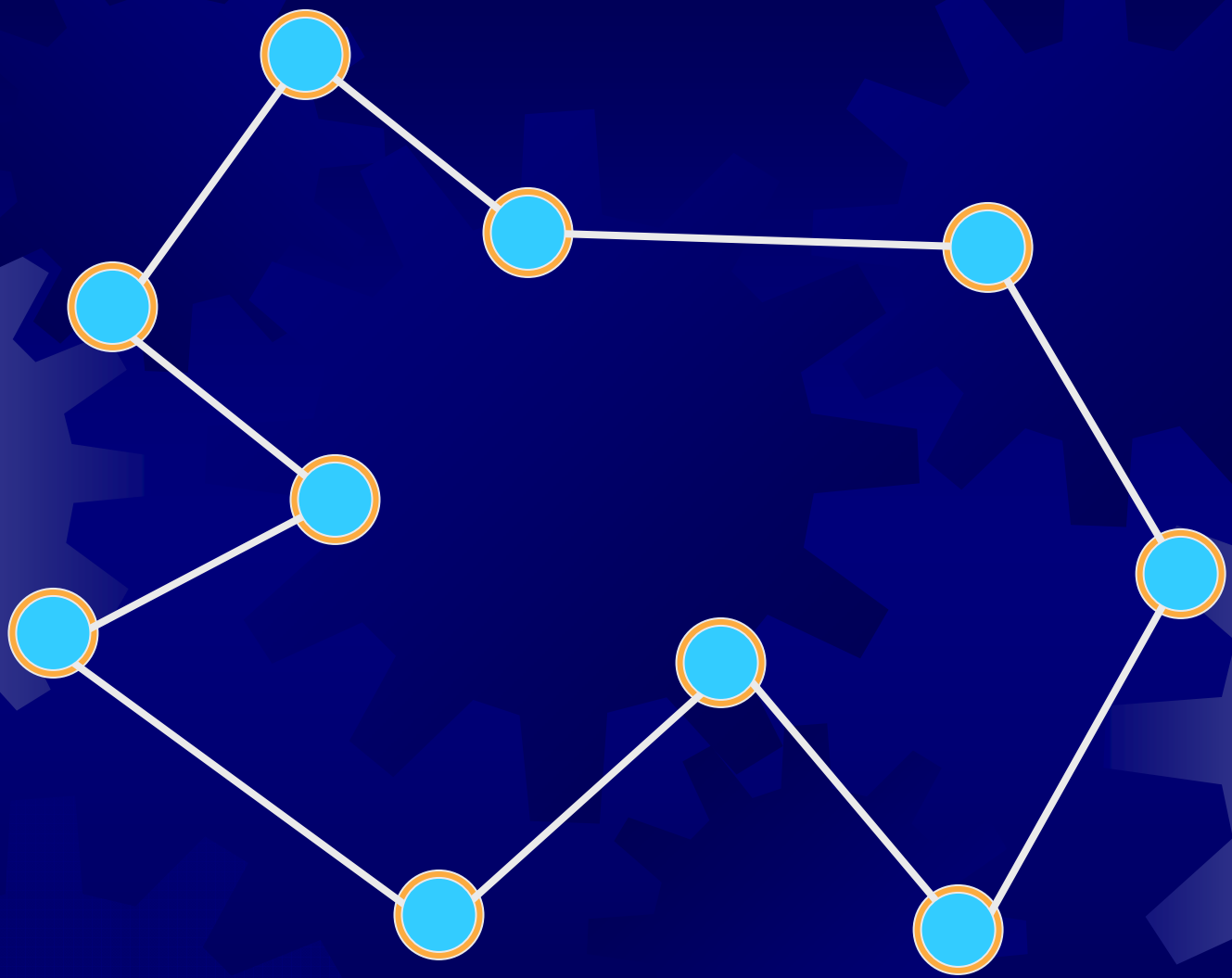


If a LVQ node is not pulled by any city, it disappears.

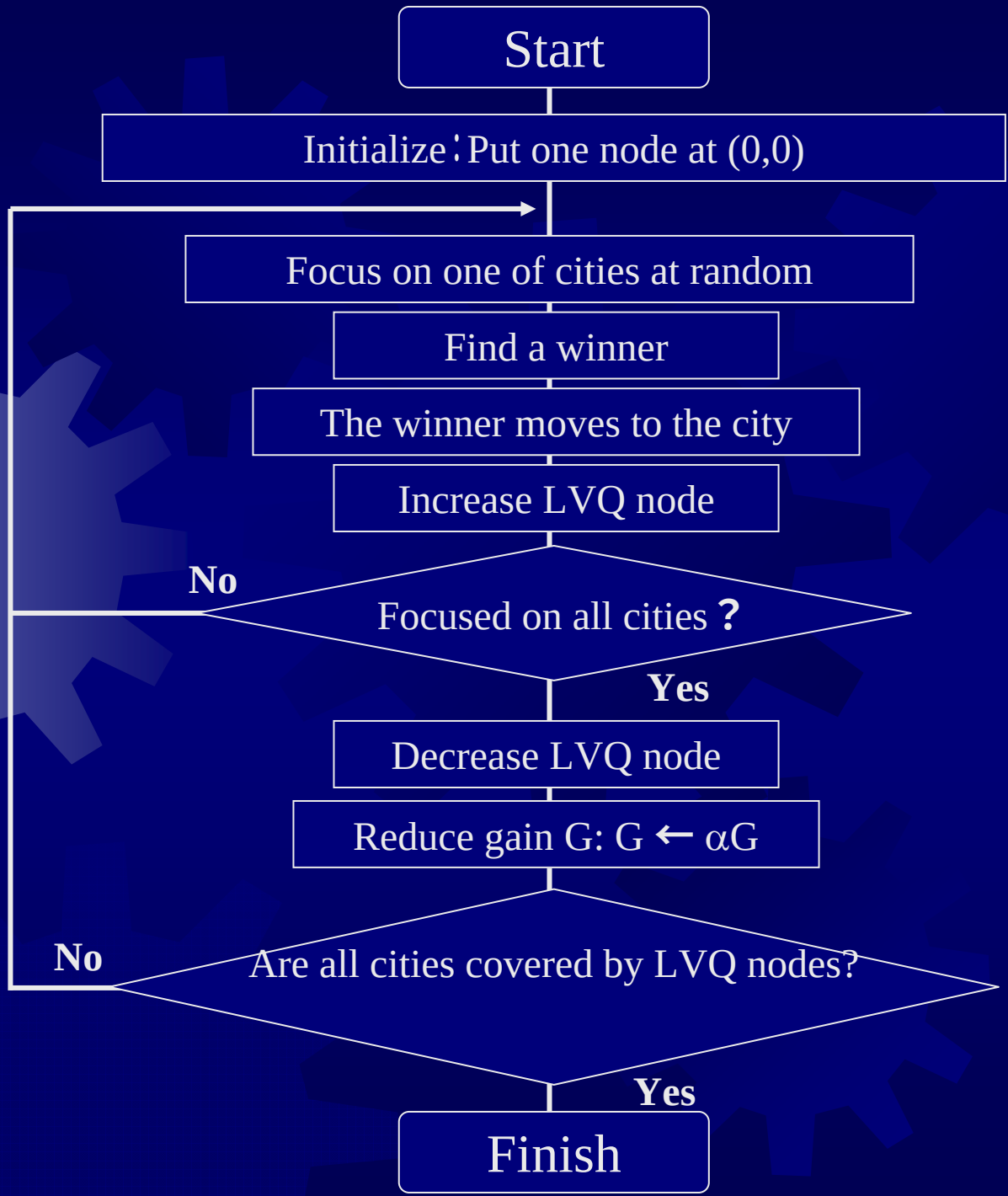


By the procedures, the ring of LVQ nodes becomes expanded.





Finally, the ring of LVQ nodes covers all cities.


Start
Initialize：Put one node at (0,0)
Focus on one of cities at random
Find a winner
The winner moves to the city
Increase LVQ node
No
Focused on all cities ?
Yes
Decrease LVQ node
Reduce gain G: G ← αG
No
Are all cities covered by LVQ nodes?
Yes
Finish

Conclusion

- Self-organizing Map (SOM)
  - Categorize unknown data without teacher signal.
  - A result is represented as a map on a 2D plane.
  - Example: state recognition of a robot.
- Learning Vector Quantizaton (LVQ)
  - Generating boundaries according to teacher signals.
  - Example: TSP solver.