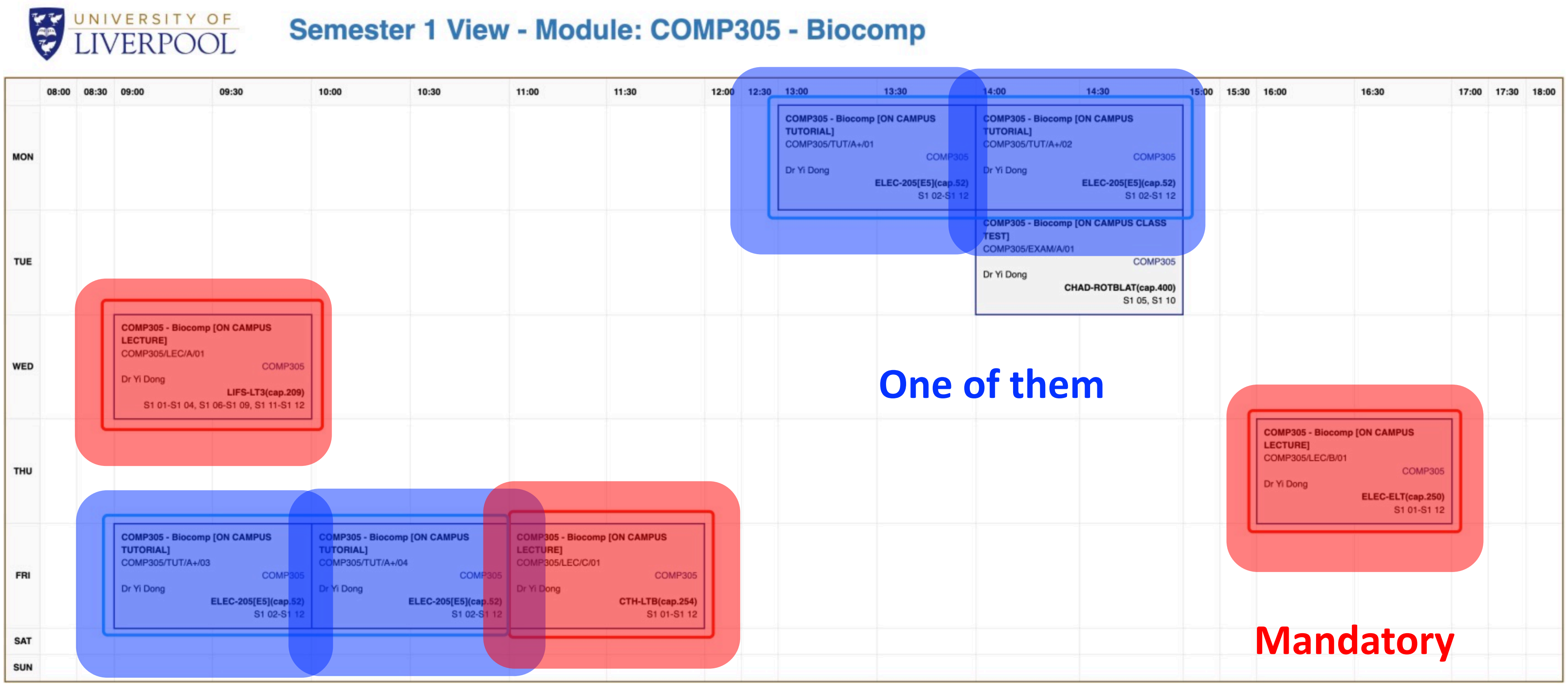
Comp305

Biocomputation

Lecturer: Yi Dong

Comp305 Module Timetable



There will be **26-30** lectures, thee per week. The lecture slides will appear on Canvas. Please use Canvas to access the lecture information. There will be **9** tutorials, one per week.

Lecture/Tutorial Rules

Questions are welcome as soon as they arise, because

1. Questions give feedback to the lecturer;

2. Questions help your understanding;

3. Your questions help your classmates, who might experience difficulties with formulating the same problems/doubts in the form of a question.

Comp305 Part I.

Artificial Neural Networks

Topic 5.

Kohonen’s Rule (Competitive Learning)

Topic of Today's Lecture

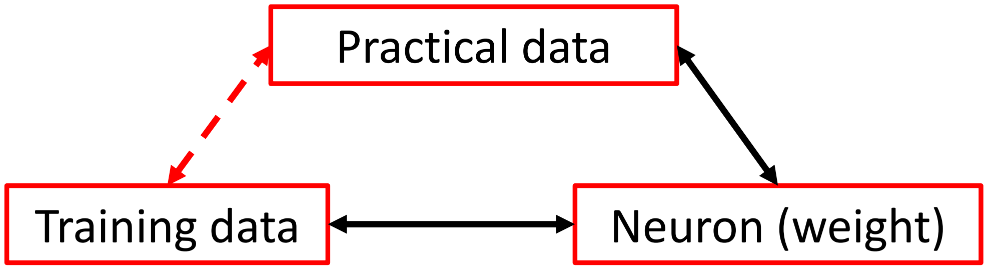
Kohonen's rule and self-organization map.

Family of Hebb's Rules – Associative Learning

This weighted sum S will be bigger if the vectors

a and W are similar,

i.e., close to each other.

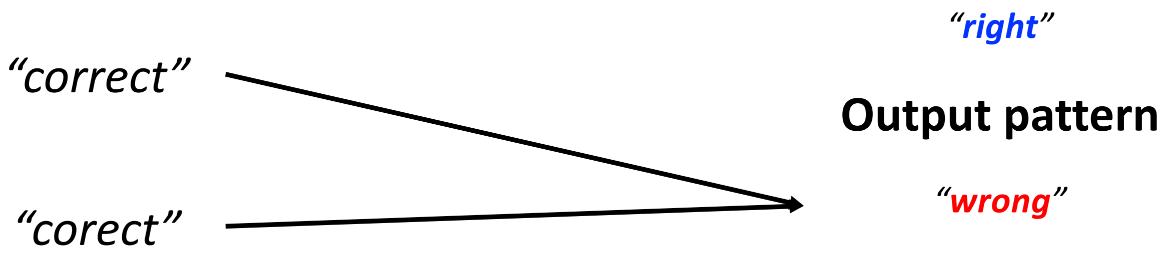


Definition: Association is the task of mapping patterns to patterns.

- An ***associative memory*** is to learn and remember the mapping between two unrelated pattens.
- For instance, a person may learn a word and its meaning before (learn the mapping between ***word*** and ***meaning***). When the person reads the word that is spelled incorrectly, she or he may know it has the same meaning with the correct word by association (remember the mapping).

Key Points

- We do not label any input in the data set during learning. The weight updating in each iteration only involves **the input, the output and the current weight**.
- We do not care what the output pattern means. We **care which inputs are considered similar** by the network. That is, unsupervised learning.



They point to the same pattern?

Unsupervised Learning

- Unsupervised learning is a type of machine learning in which the algorithm **is not provided with any pre-assigned labels or scores for the training data**. As a result, unsupervised learning algorithms must first **self-discover any naturally occurring patterns** in that training data set.

- A common example is ***clustering***, that is, an unsupervised network can group similar sets of input patterns into ***clusters*** predicated upon a predetermined set of criteria relating the components of the data.

- ***Clustering*** can be achieved when we extend the single neuron to the network with multiple outputs.

Kohonen Rule: Competitive Learning

- We consider a one-layer neural network with multiple outputs.

- In competitive learning, as the name implies, an output neuron of a network compete among all the outputs to be updated (regarding the weights).

- Whereas in a neural network based on Hebbian learning several output neurons may be updated simultaneously, **in competitive learning only a single output neuron is updated at any instant**.

- This makes competitive learning highly suited to discover statistically important features that may be used to classify a set of input patterns.

Kohonen Rule: Competitive Learning

- **Teuvo Kohonen** suggested a network in which the output neurons compete for the right to respond to an input.

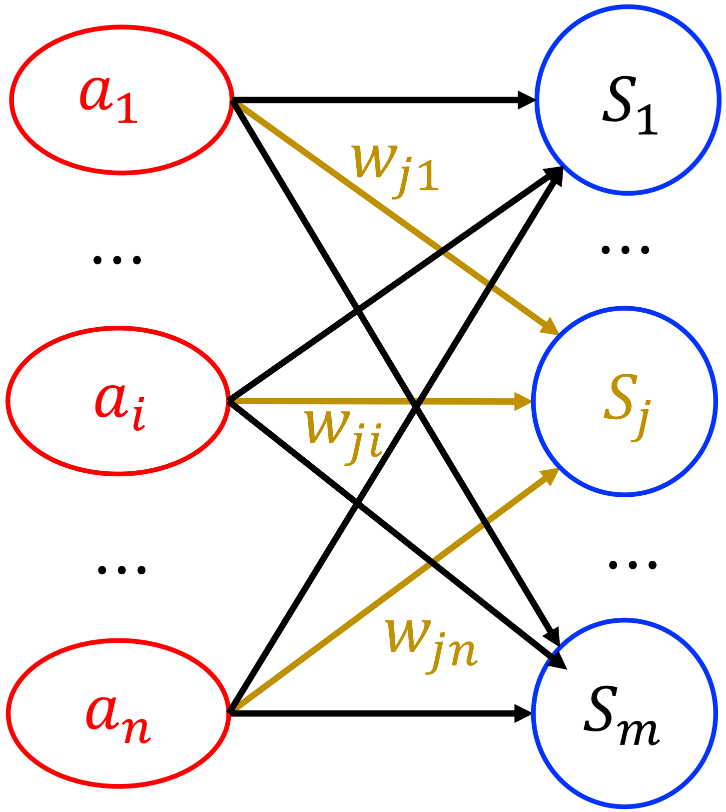
- The learning rule first assumes that one of the output neurons, say the j-th, has maximum value of the instant state S_j at that instant.

- That neuron is declared to be the ***winner,*** and only the weights of the winner's connections

$$w_j = \left(w_{j1}, w_{j2}, \cdots, w_{jn}\right)$$

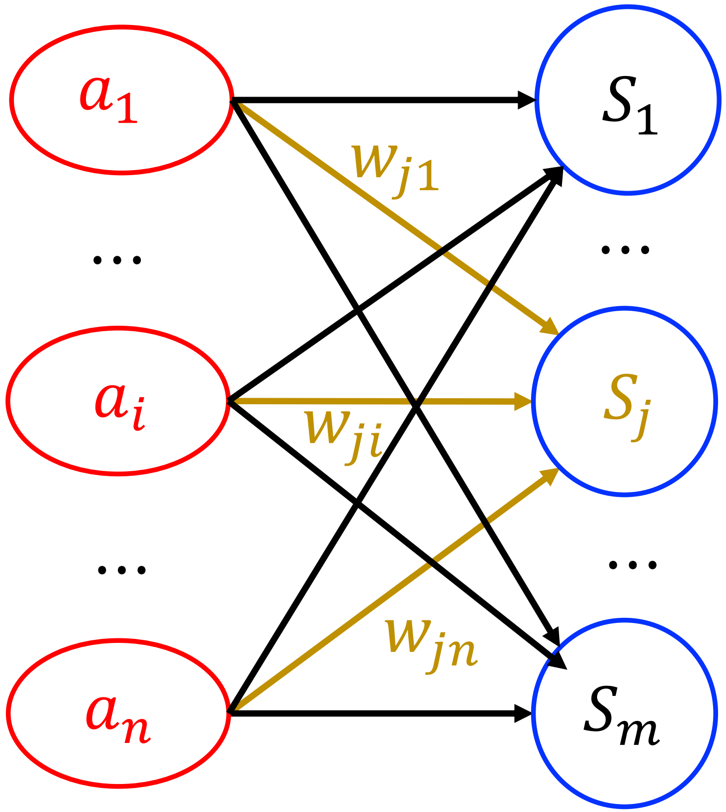
  is updated.

Kohonen Rule: Competitive Learning



- We further relax the restriction on the input type, i.e., we allow real inputs in the competitive learning.
- Only the winner outputs a 1, while others output 0.

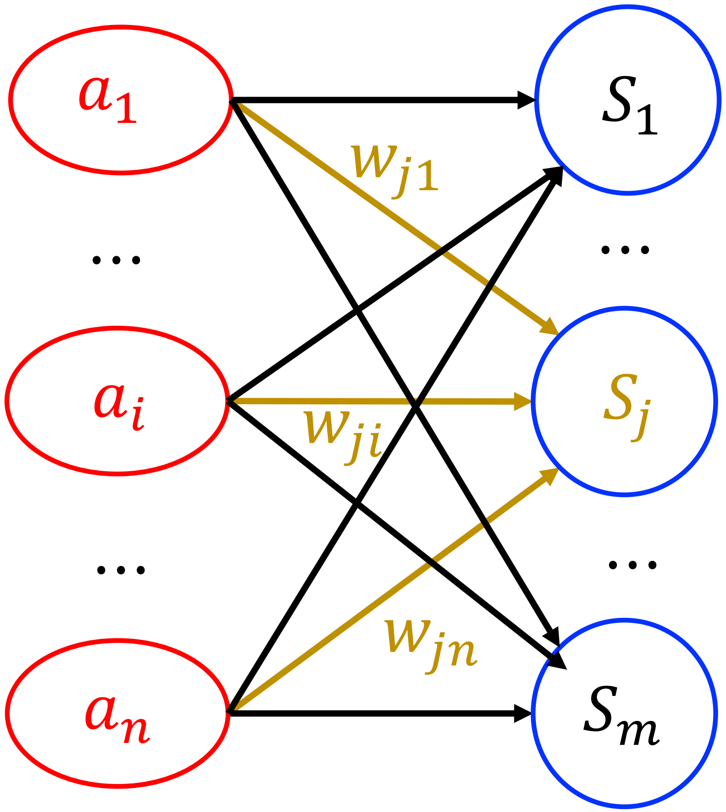
Kohonen Rule: Competitive Learning



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

- The learning rule first assumes that one of the output neurons, say the j-th, has maximum value state S_j at that instant.
$$S_j = \max(S_1, \cdots, S_m)$$
- That neuron is declared the ***winner,*** and only its vector of connections weights.
$$w_j = (w_{j1}, w_{j2}, \cdots, w_{jn})$$
is updated.

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

- Assume that the j-th output neuron has maximum weighted input at that instant t.

$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$

- Then its weight is updated as

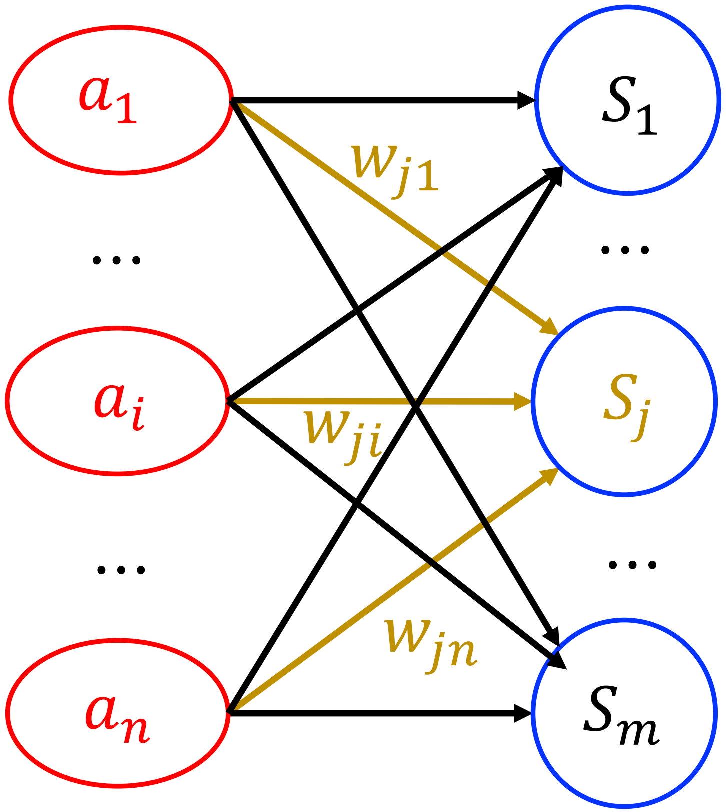
$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$.

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

- Assume that the j-th output neuron has maximum weighted input at that instant t.

$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$

Winner takes it all!

- Then its weight is updated as

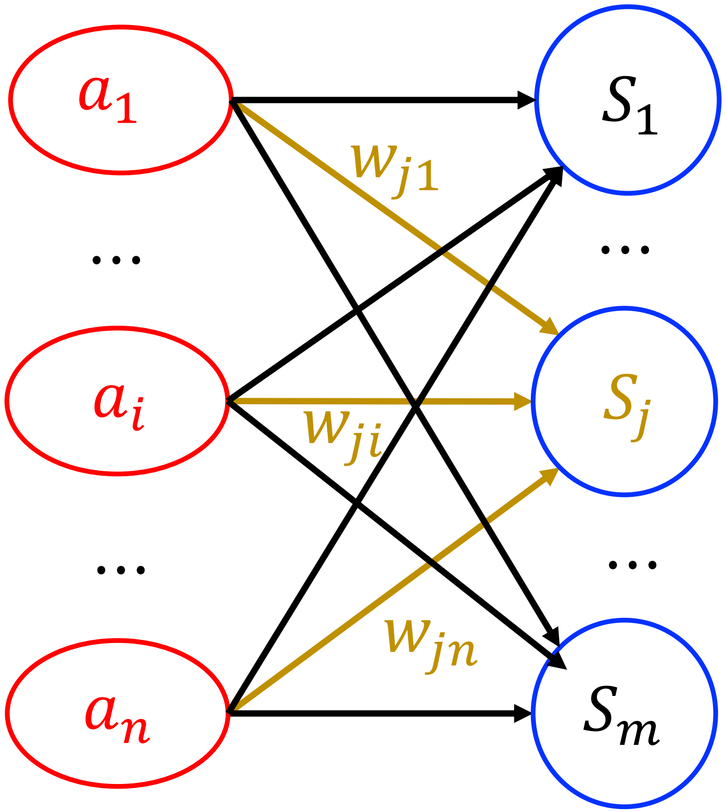
$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$.

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

- Assume that the j-th output neuron has maximum weighted input at that instant t.

$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$

Winner takes it all!

- Then its weight is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

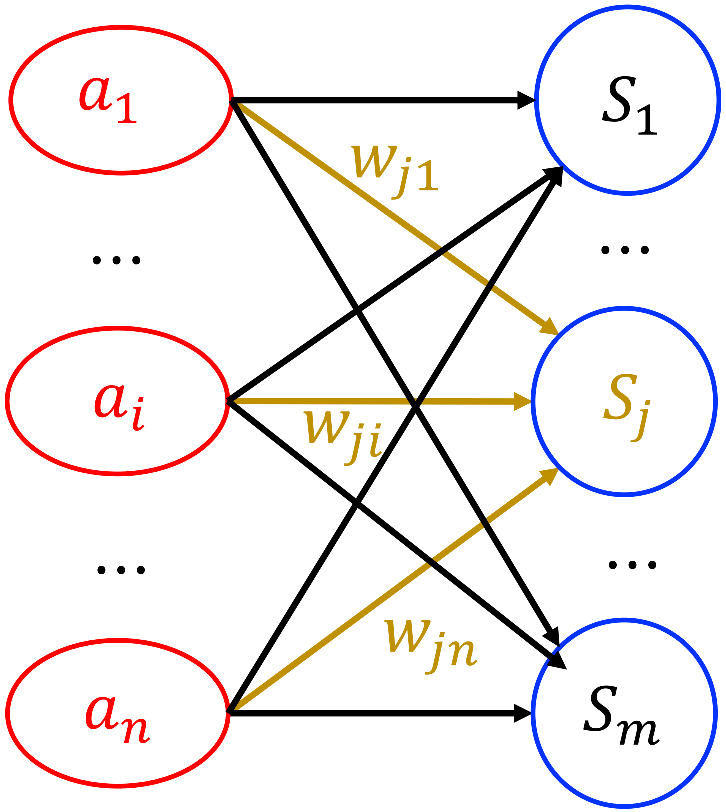
where $i = 1, \cdots, n$.

Difference between a and w

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

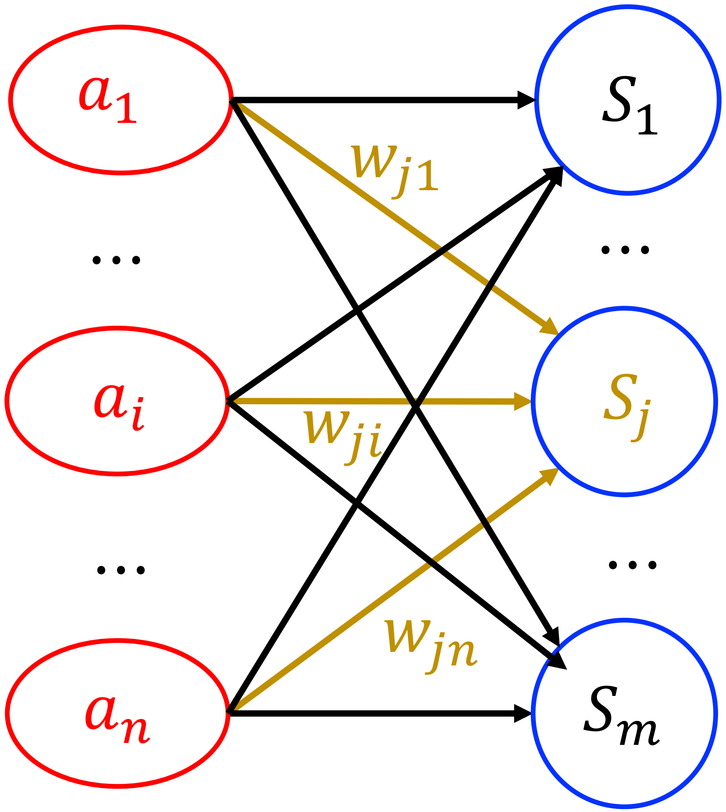
- The j-th output neuron has maximum weighted input at that instant t.
$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$
- Then its weight is updated as
$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$.

- The incremental term Δw_{ji}^t:
$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

Having the maximum weighted input, means that the winning output neuron has the vector of connection weights most similar to the input vector.

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

- The j-th output neuron has maximum weighted input at that instant t.
$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$
- Then its weight is updated as
$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$.

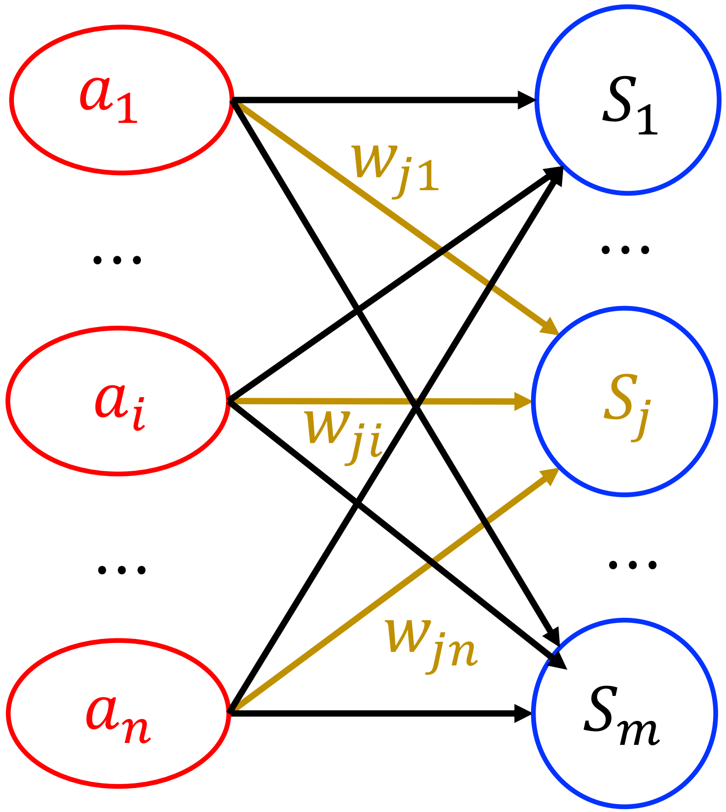
- The incremental term Δw_{ji}^t:
$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

a_1^4	a_2^4	a_3^4	a_4^4
1	0	1	0

w_1^4	w_2^4	w_3^4	w_4^4
0.71	0.02	0.71	0.02

Share the same idea of Hebb's rules

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

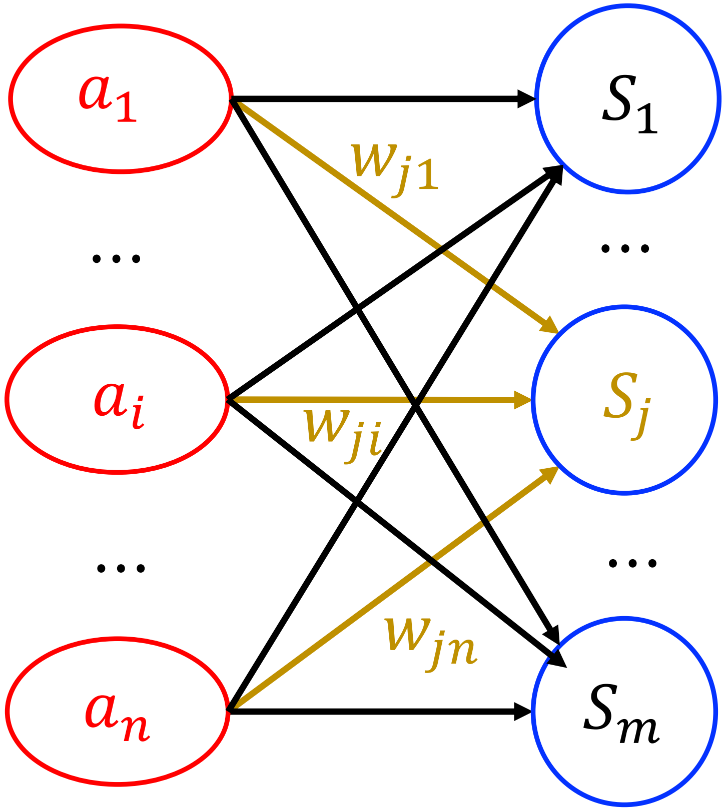
- The j-th output neuron has maximum weighted input at that instant t.
$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$
- Then its weight is updated as
$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$.

- The incremental term Δw_{ji}^t:
$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

The ***winner-takes-it-all*** learning rule modifies the winner's (only) connection weights by a fraction (learning rate) of the difference between the current input vector and the current weight vector of the winner neuron.

Kohonen Learning Rule



Updated are the ***winner's*** output unit weights of connections only: $S_j = \max(S_1, \cdots, S_m)$.

- The j-th output neuron has maximum weighted input at that instant t.

$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$

- Then its weight is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$.

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

After the adjustment, the winner connection weights tend to even ***better*** correlate with the input pattern.

Q & A About Kohonen Learning Rule

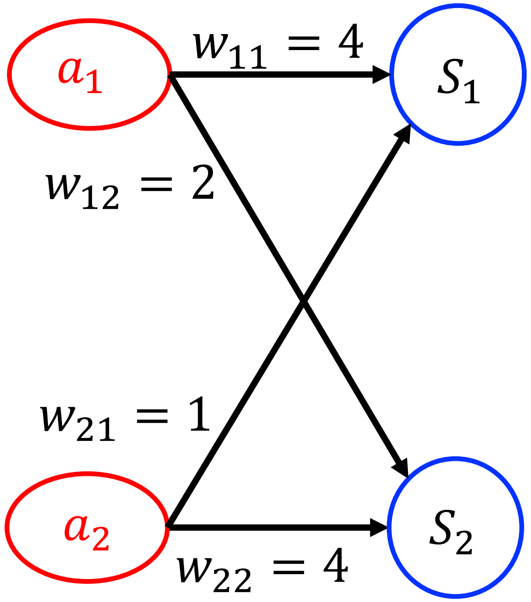
- Question: Does an input update the same output neuron?

Q & A About Kohonen Learning Rule

- Question: Does an input update the same output neuron?
- Answer: No. Consider an input a updates an output neuron s, 1) Even though s wins, the weight of s may decrease. 2) If another input b updates another output neuron r, the weight of r may increase. Both may lead to the change of output neuron for a.

Q & A About Kohonen Learning Rule

- Question: Does an input update the same output neuron?
- Answer: No. Consider an input a updates an output neuron s, 1) Even though s wins, the weight of s may decrease. 2) If another input b updates another output neuron r, the weight of r may increase. Both may lead to the change of output neuron for a.



Consider the input $(2,1)$. Learning rate is 0.5.

- $t = 0$, use the input, i.e.,

$$a_1 = 2, a_1 = 1.$$

Compute the instant states for two outputs, respectively.

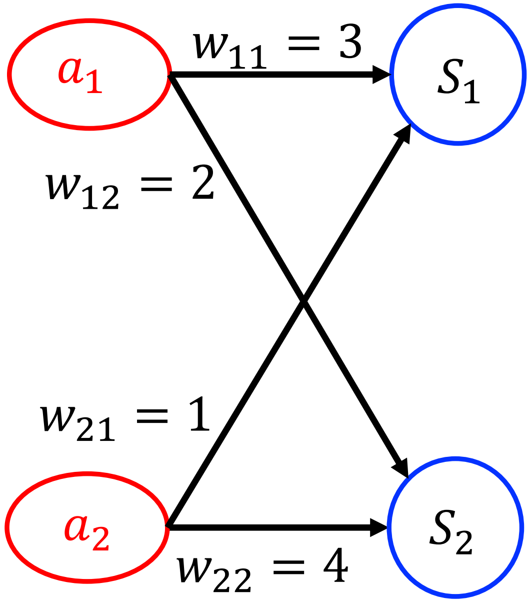
$$S_1 = 4\times 2 + 1\times 1 = 9$$
$$S_2 = 2\times 2 + 4\times 1 = 8$$

$S_1 > S_2$, thus we update the weights of the 1st output.

$$w_{11} = w_{11} + \Delta w_{11} = 4 + 0.5\times(2 - 4) = 3$$
$$w_{21} = w_{21} + \Delta w_{21} = 1 + 0.5\times(1 - 1) = 1$$

Q & A About Kohonen Learning Rule

- Question: Does an input update the same output neuron?
- Answer: No. Consider an input a updates an output neuron s, 1) Even though s wins, the weight of s may decrease. 2) If another input b updates another output neuron r, the weight of r may increase. Both may lead to the change of output neuron for a.



Consider the input $(2,1)$. Learning rate is 0.5.

- $t = 1$, use the input, i.e.,

$$a_1 = 2, a_1 = 1.$$

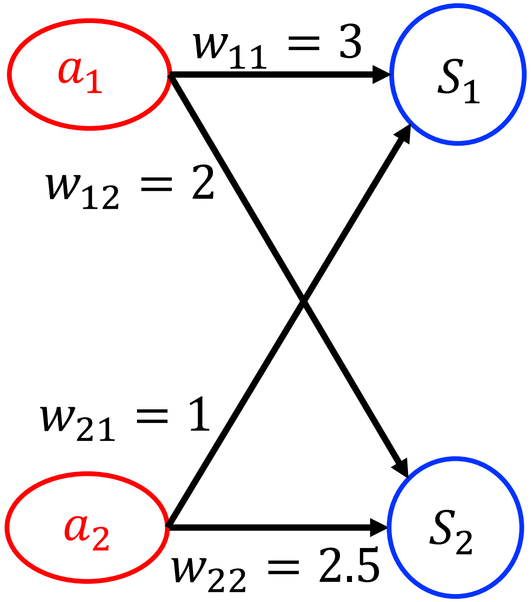
Compute the instant states for two outputs, respectively.

$$S_1 = 3\times 2 + 1\times 1 = 7$$
$$S_2 = 2\times 2 + 4\times 1 = 8$$

$S_1 < S_2$, thus we update the weights of the 2nd output.

Q & A About Kohonen Learning Rule

- Question: Does an input update the same output neuron?
- Answer: No. Consider an input a updates an output neuron s, 1) Even though s wins, the weight of s may decrease. 2) If another input b updates another output neuron r, the weight of r may increase. Both may lead to the change of output neuron for a.



Consider the input $(2{,}1)$. Learning rate is 0.5.

- $t = 1$, use the input, i.e.,

$$a_1 = 2, a_1 = 1.$$

Compute the instant states for two outputs, respectively.

$$S_1 = 3\times 2 + 1\times 1 = 7$$
$$S_2 = 2\times 2 + 4\times 1 = 8$$

$S_1 < S_2$, thus we update the weights of the 2nd output.

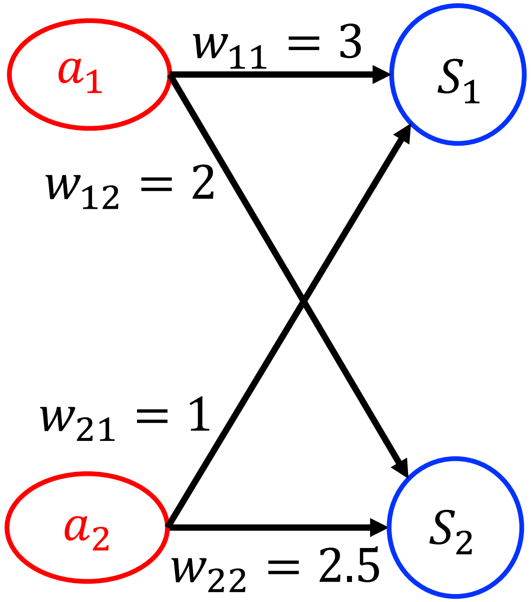
$$w_{12} = w_{12} + \Delta w_{12} = 2 + 0.5\times(2-2) = 2$$
$$w_{22} = w_{22} + \Delta w_{22} = 4 + 0.5\times(1-4) = 2.5$$

Q & A About Kohonen Learning Rule

- Question: Does an input update the same output neuron?

We can avoid that by normalization!

- Answer: No. Consider an input a updates an output neuron s, 1) Even though s wins, the weight of s may decrease. 2) If another input b updates another output neuron r, the weight of r may increase. Both may lead to the change of output neuron for a.



Consider the input $(2,1)$. Learning rate is 0.5.

- $t = 1$, use the input, i.e.,

$$a_1 = 2, a_1 = 1.$$

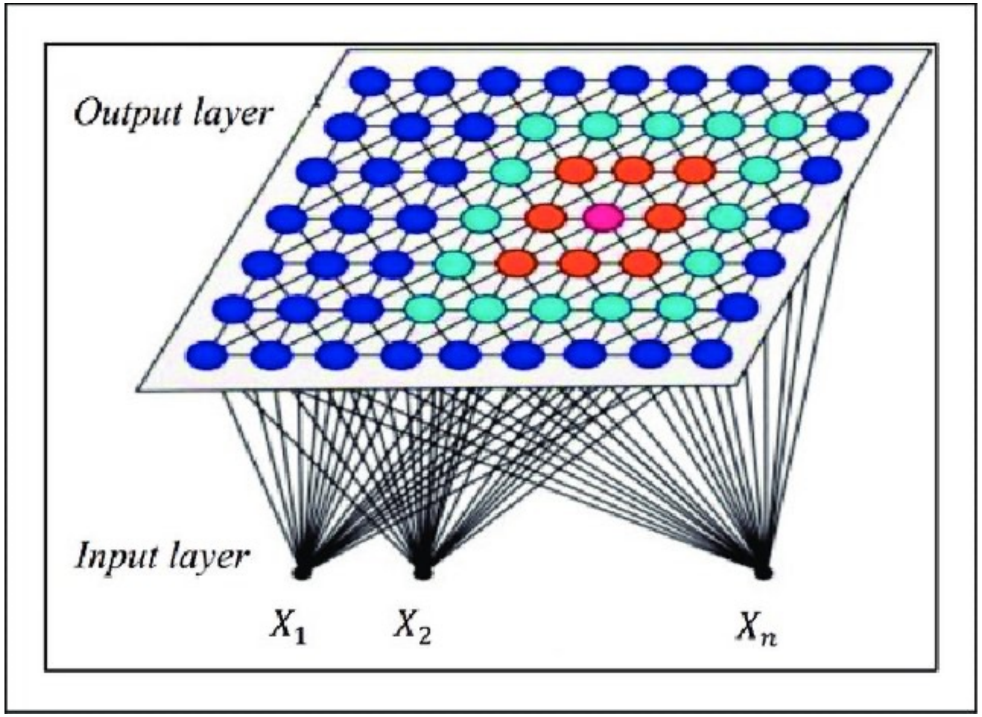
Compute the instant states for two outputs, respectively.

$$S_1 = 3 \times 2 + 1 \times 1 = 7$$
$$S_2 = 2 \times 2 + 4 \times 1 = 8$$

$S_1 < S_2$, thus we update the weights of the 2nd output.

$$w_{12} = w_{12} + \Delta w_{12} = 2 + 0.5 \times (2 - 2) = 2$$
$$w_{22} = w_{22} + \Delta w_{22} = 4 + 0.5 \times (1 - 4) = 2.5$$

Self-Organizing Map (SOM)

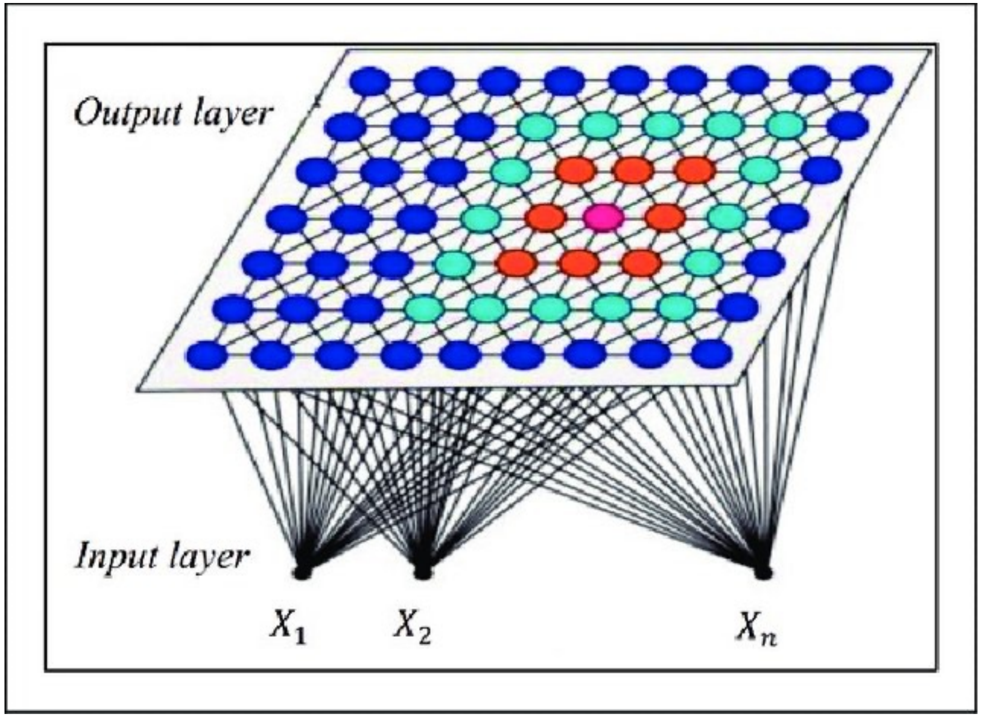


Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- We are interested in a specific application of Kohonen learning rule (competitive learning), self-organizing map (SOM).

- SOM is used to produce a low-dimensional (typically **two-dimensional**) representation of a higher dimensional data set while preserving the topological structure of the data. For instance, in left figure, a SOM maps a n-dimensional input to a 2-dimensional space. It can be used for **clustering or visualization**.

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j-th output neuron has maximum weighted input at that instant t.

$$S_j^t = \max(S_1^t, \cdots, S_m^t)$$

- Then its weight is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

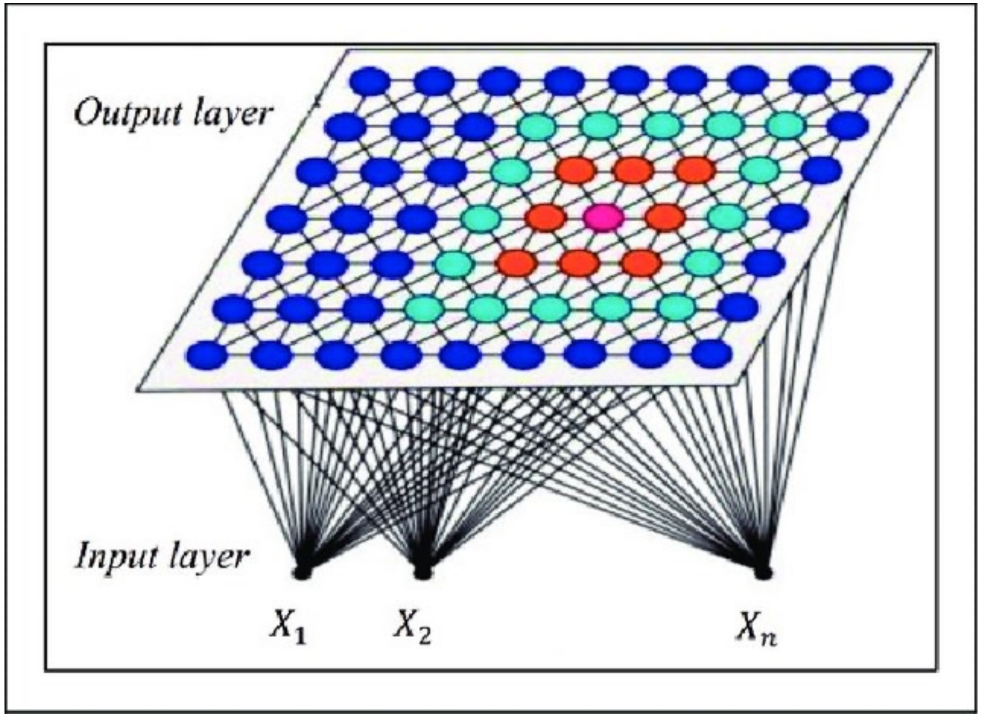
where $i = 1, \cdots, n$.

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)$$

The training of such specific networks is based on Kohonen learning rule (competitive learning).

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j^*-th output neuron has maximum weighted input at that instant t.

$$S_{j^*}^t = \max(S_1^t, \cdots, S_m^t)$$

- Then the weight of the j-th neuron is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$, for $j = 1, \cdots, m$.

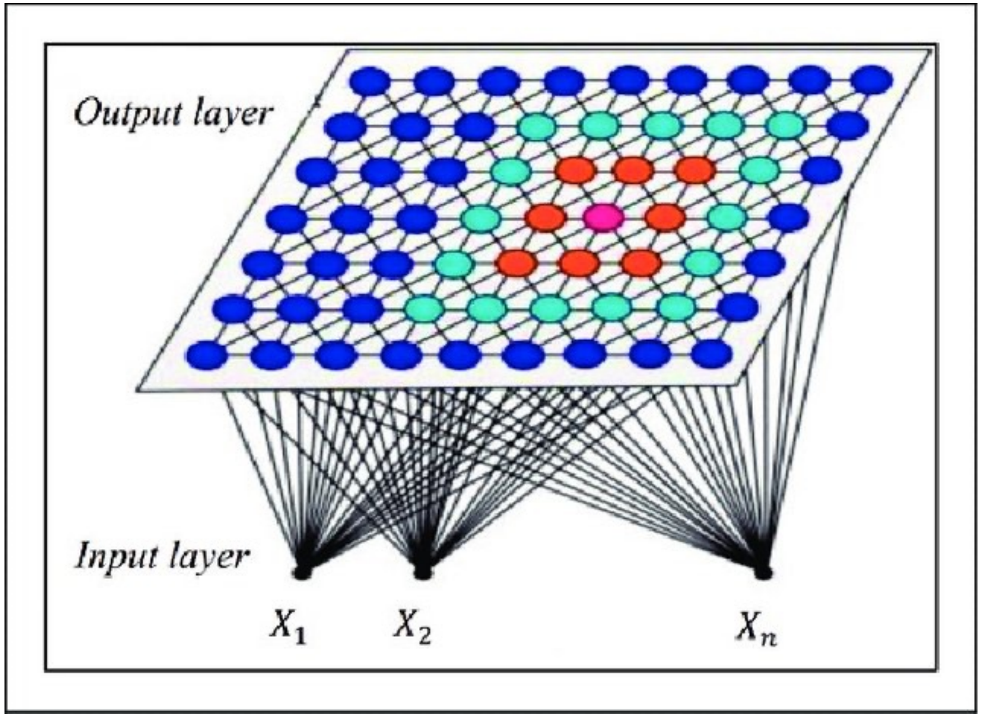
- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Where $\theta(j, j^*)$ is a restraint function due to the distance between neuron j and j^*.

- Sometimes, the winning neighbourhood is extended beyond the single neuron winner, so that it includes the neighbouring neurons, for which some corrections to the connection weights may also be done.

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j^*-th output neuron has maximum weighted input at that instant t.

$$S_{j^*}^t = \max(S_1^t, \cdots, S_m^t)$$

- Then the weight of the j-th neuron is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$, for $j = 1, \cdots, m$.

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

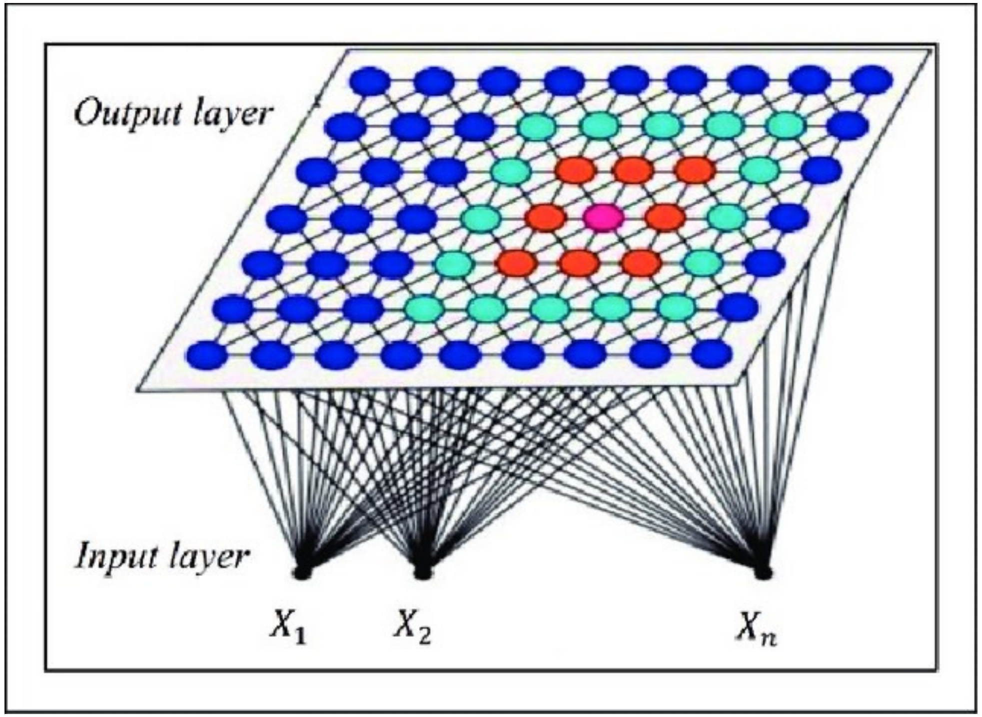
Where $\theta(j, j^*)$ is a restraint function due to the distance between neuron j and j^*.

- In addition, C may sometimes decrease along with the time, for convergence purpose, i.e.,

$$C(t) \longrightarrow 0, \qquad t \longrightarrow 0$$

Feng, J. F., and B. Tirozzi. "Convergence theorems for the Kohonen feature mapping algorithms with vlrps." *Computers & Mathematics with Applications* 33.3 (1997): 45-63.

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j^*-th output neuron has maximum weighted input at that instant t.

$$S_{j^*}^t = \max(S_1^t, \cdots, S_m^t)$$

- Then the weight of the j-th neuron is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$, for $j = 1, \cdots, m.$

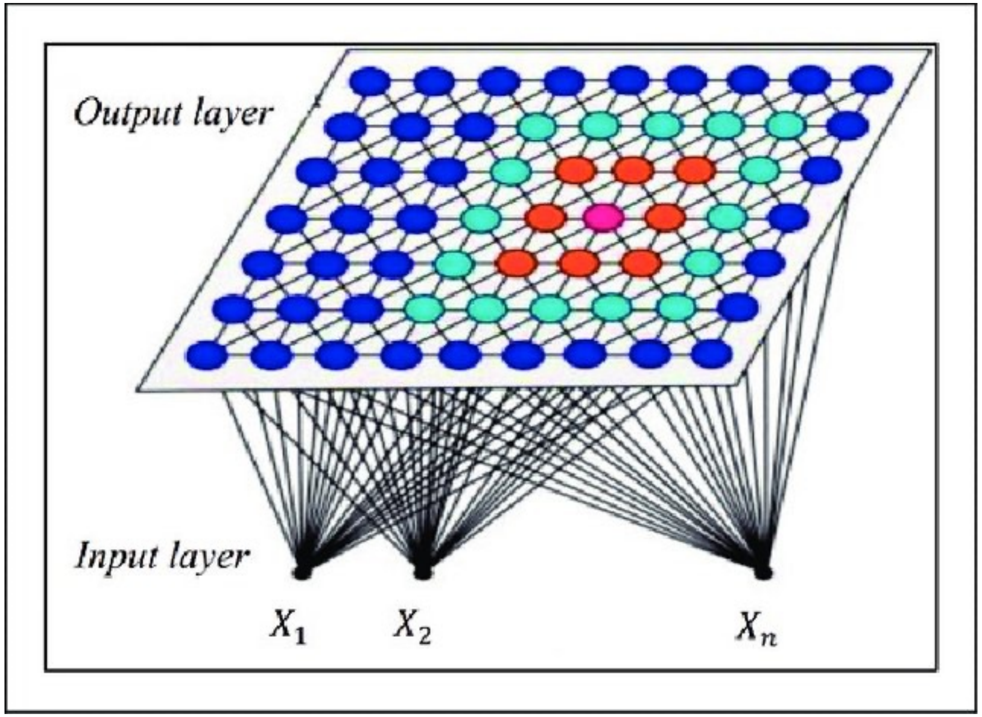
- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Where $\theta(j, j^*)$ is a restraint function due to the distance between neuron j and j^*.

- In the map, location of the most strongly excited neurons (winner) is correlated with the certain input signals.
- Neighbouring excited neurons correspond to inputs with similar features.

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j^*-th output neuron has maximum weighted input at that instant t.

$$S_{j^*}^t = \max(S_1^t, \cdots, S_m^t)$$

- Then the weight of the j-th neuron is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$, for $j = 1, \cdots, m.$

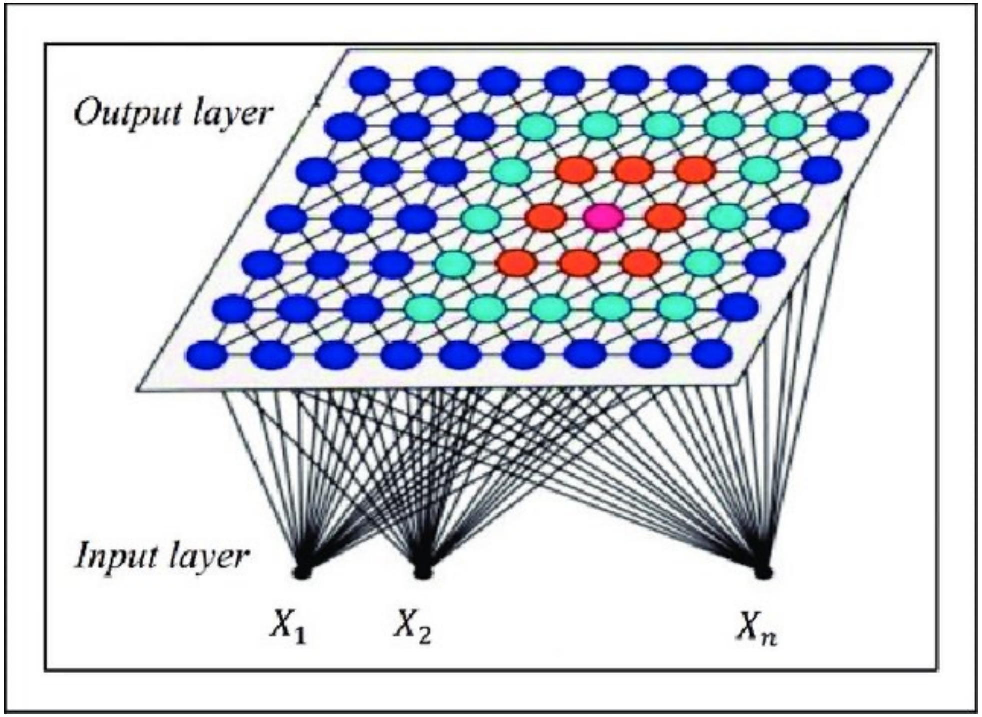
- The incremental term Δw_{ji}^t:

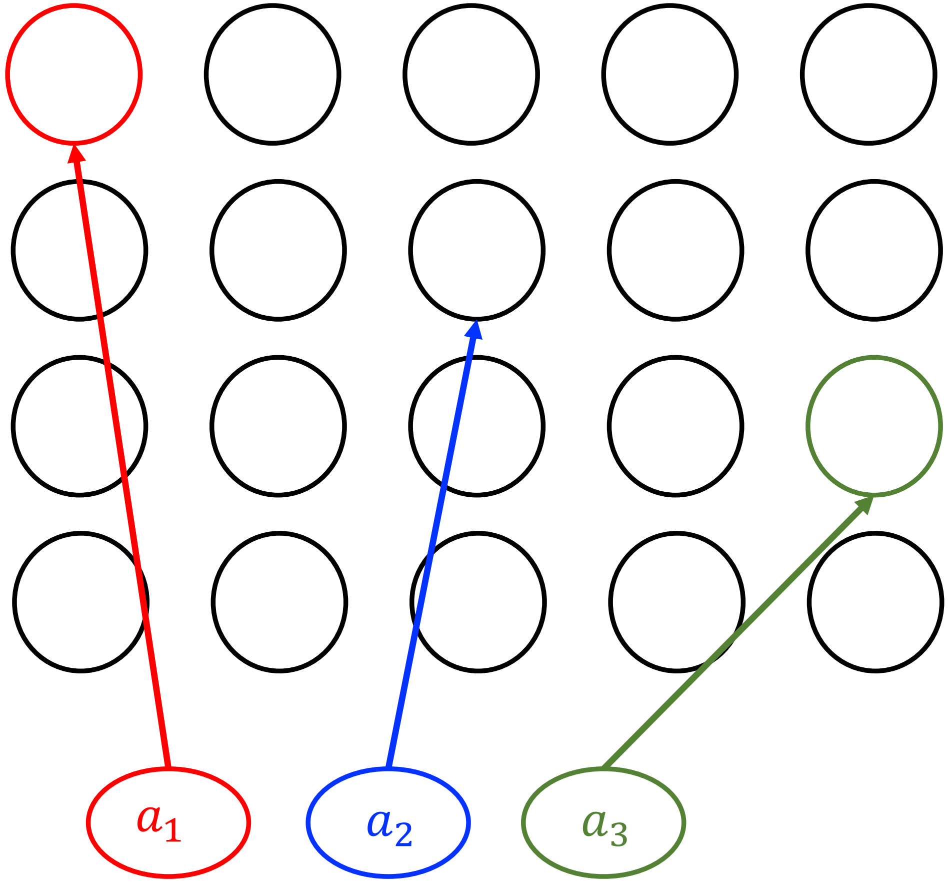
$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Where $\theta(j, j^*)$ is a restraint function due to the distance between neuron j and j^*.

- Because in the training phase weights of the whole neighbourhood are moved in the same direction, similar items tend to excite adjacent neurons. Therefore, SOM forms a semantic map where similar samples are mapped close together and dissimilar ones apart.

Example: If a_1, a_2 are similar, a_3 is different ...

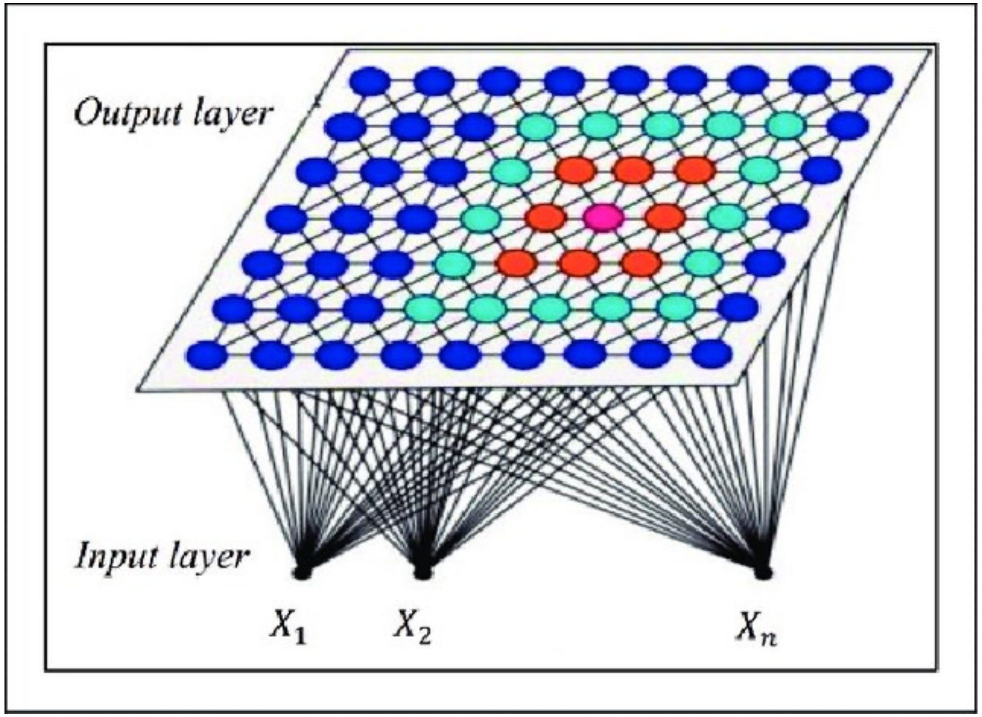


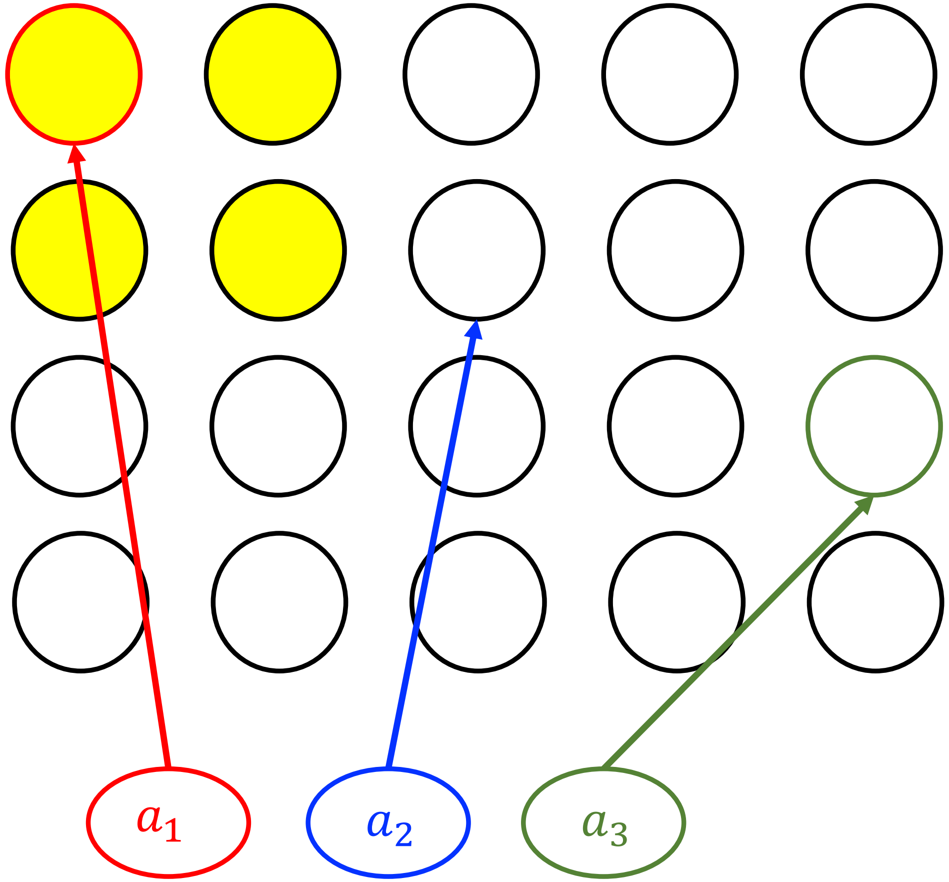


- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Example: If a_1, a_2 are similar, a_3 is different …

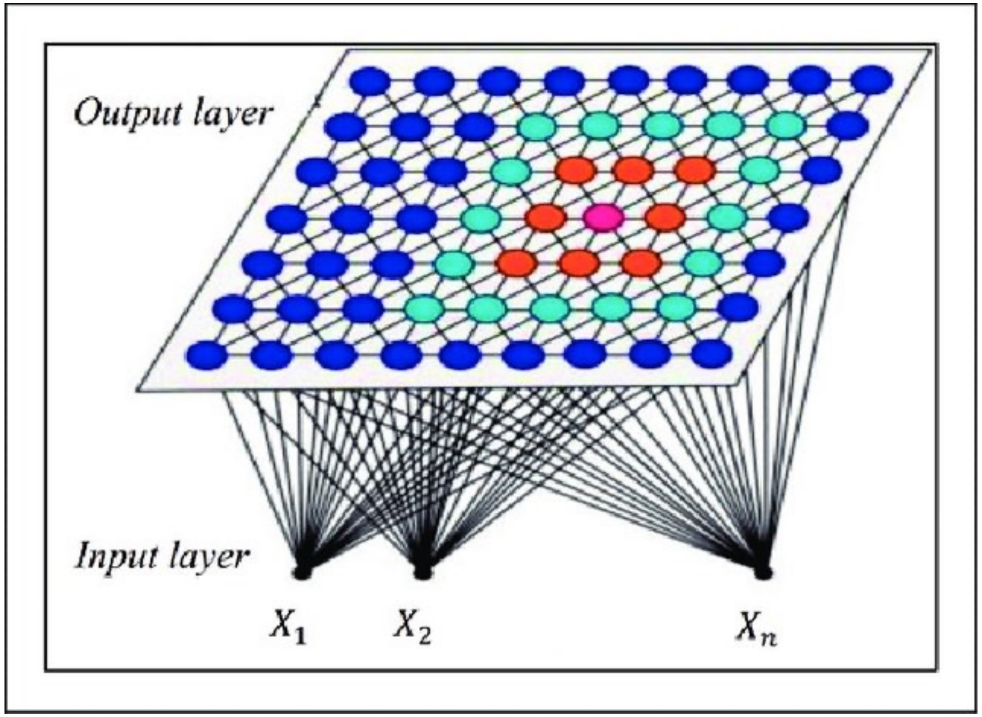




- The incremental term Δw_{ji}^t:

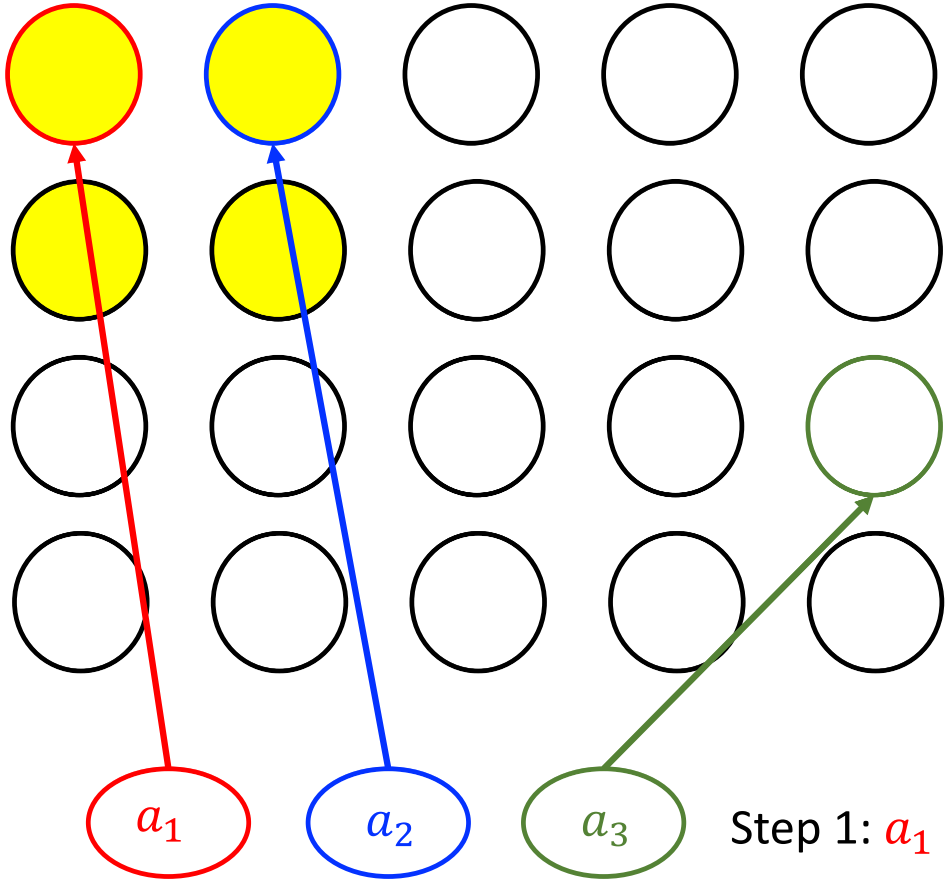
$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Example: If a_1, a_2 are similar, a_3 is different …

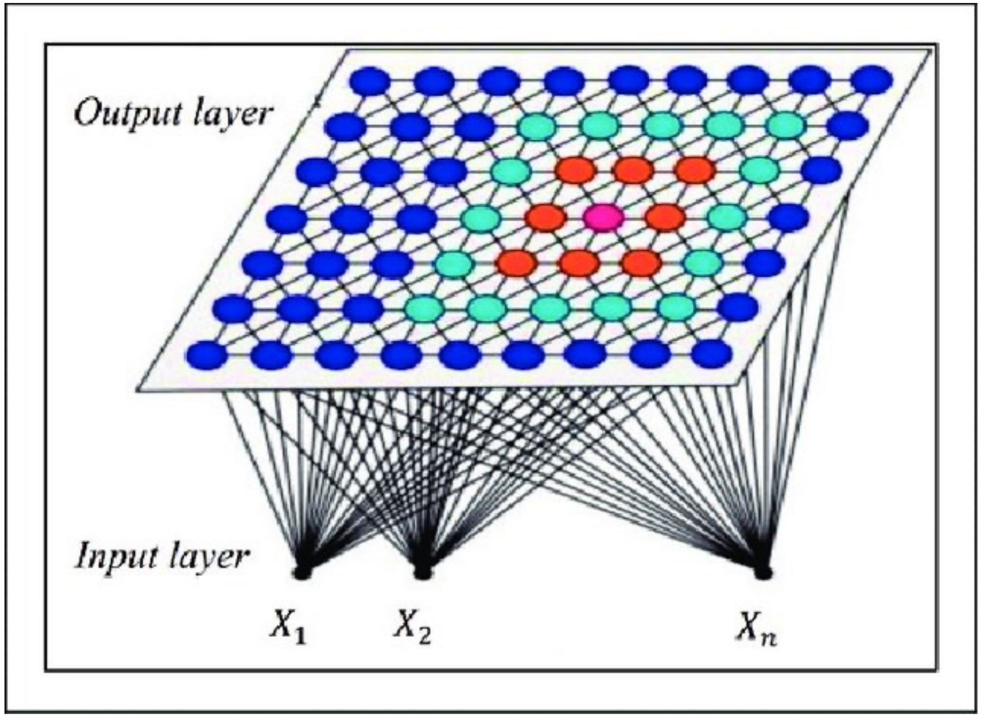


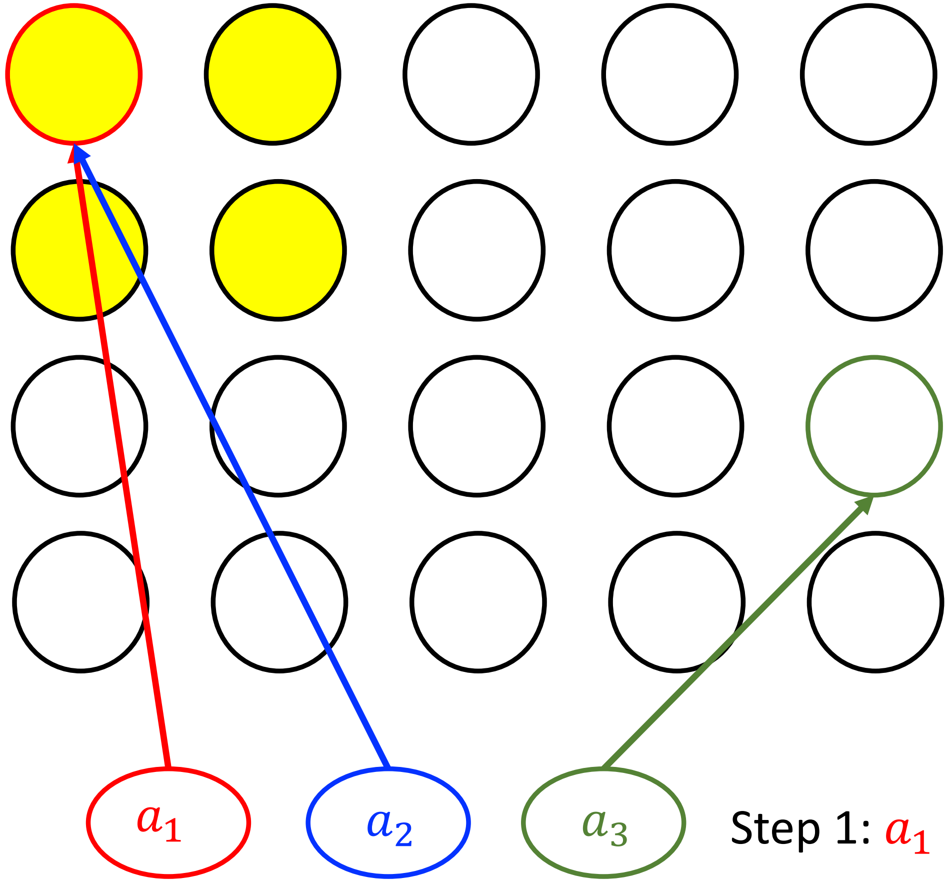
- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$



Example: If a_1, a_2 are similar, a_3 is different …

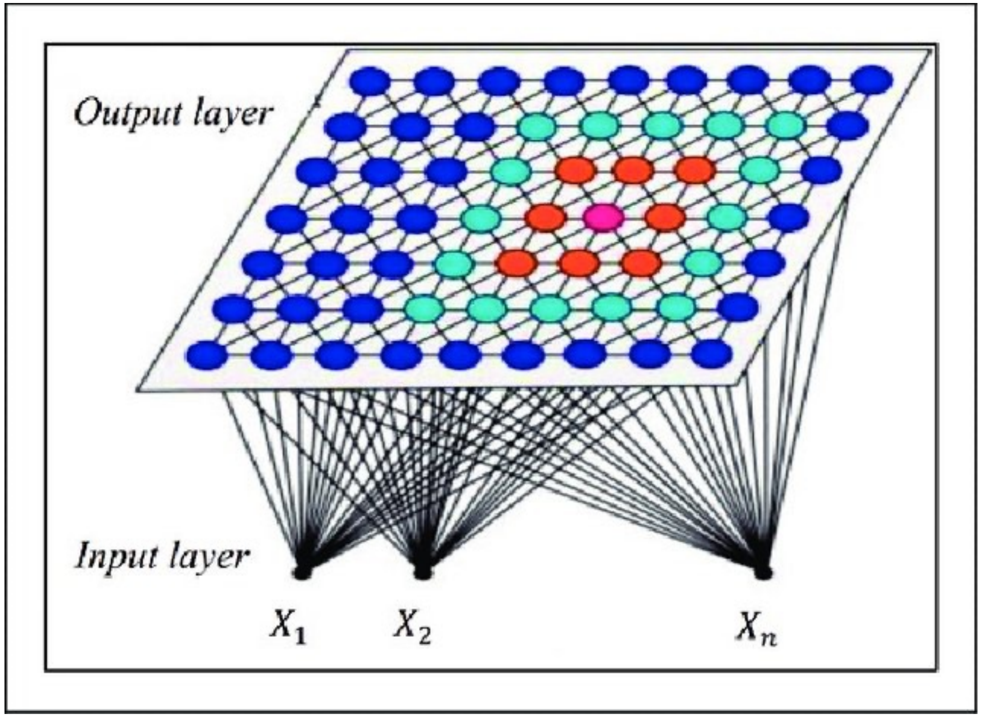




- The incremental term Δw_{ji}^t:

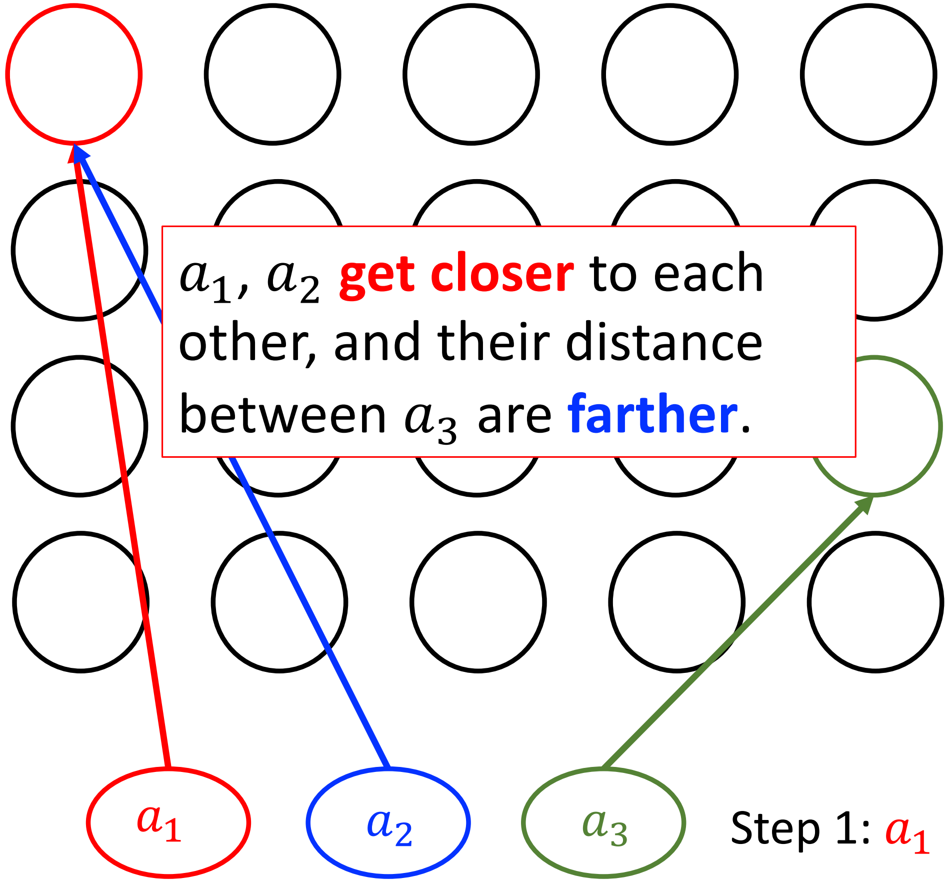
$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Example: If a_1, a_2 are similar, a_3 is different …



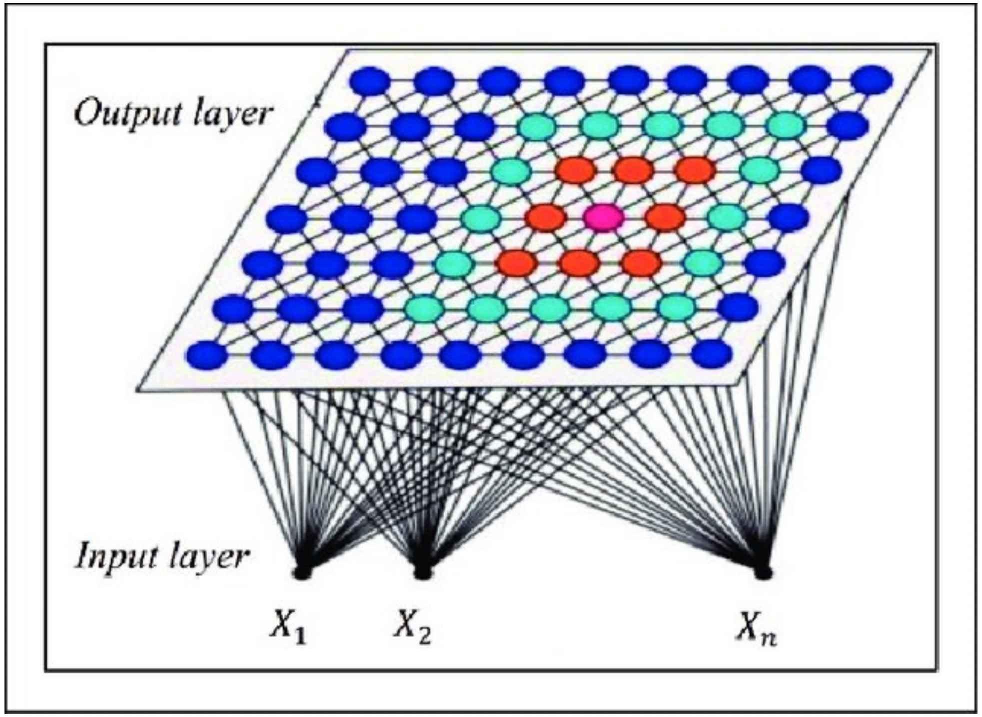
- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$



Step 1: a_1

Check the Result after Training



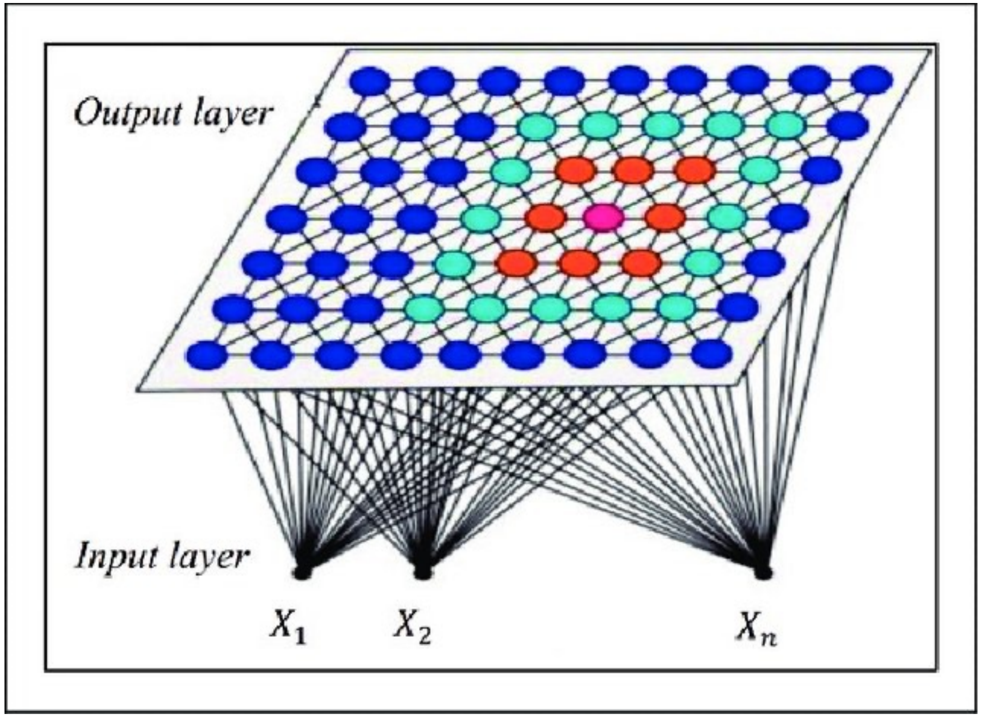
2	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0

- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

a_1 a_2 a_3

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j^*-th output neuron has maximum weighted input at that instant t.

$$S_{j^*}^t = \max(S_1^t, \cdots, S_m^t)$$

- Then the weight of the j-th neuron is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$, for $j = 1, \cdots, m.$

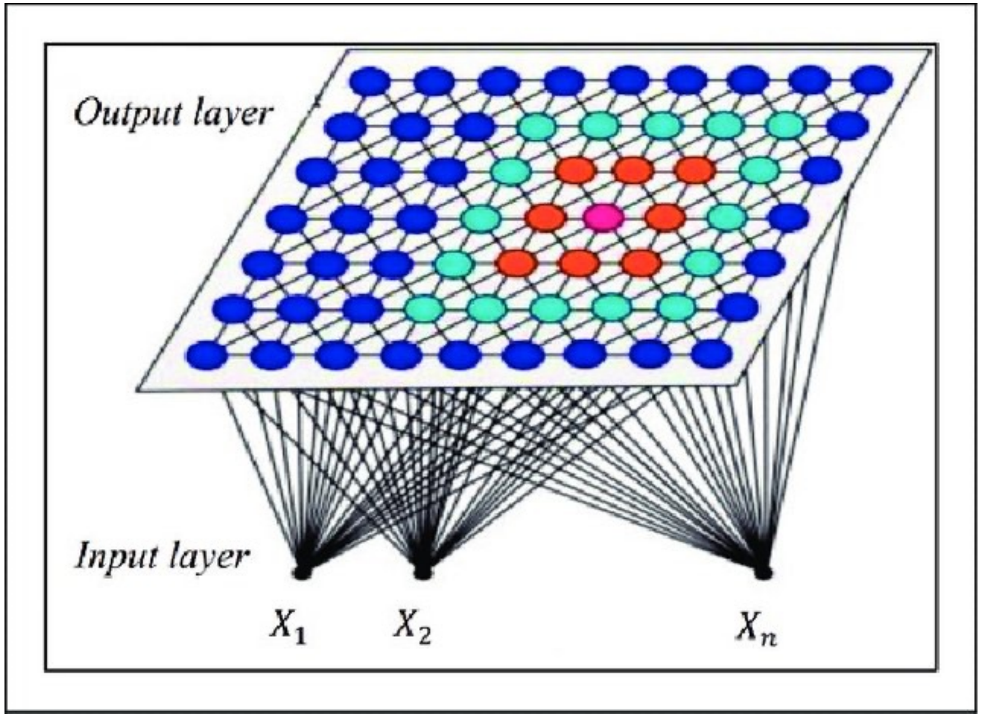
- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

Where $\theta(j, j^*)$ is a restraint function due to the distance between neuron j and j^*.

- After successful training, individual neurons of the network learn to specialise on ensembles of similar patterns;
- in so doing they become ***feature detectors*** for different classes of input patterns.

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.

- The j^*-th output neuron has maximum weighted input at that instant t.

$$S_{j^*}^t = \max(S_1^t, \cdots, S_m^t)$$

- Then the weight of the j-th neuron is updated as

$$w_{ji}^{t+1} = w_{ji}^t + \Delta w_{ji}^t$$

where $i = 1, \cdots, n$, for $j = 1, \cdots, m.$

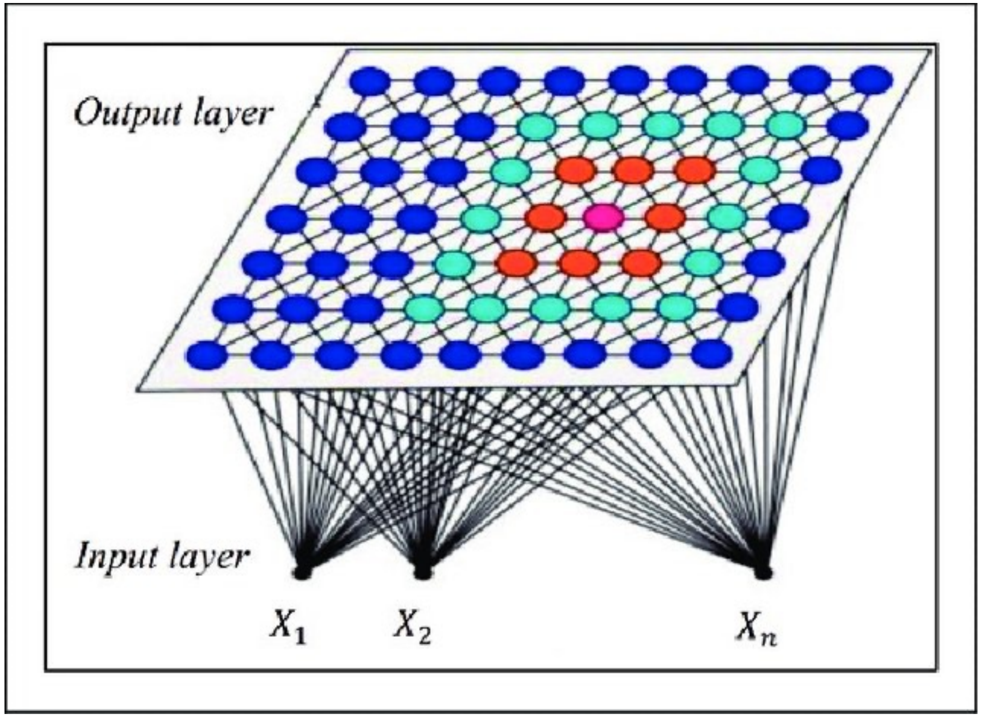
- The incremental term Δw_{ji}^t:

$$\Delta w_{ji}^t = C(t)\left(a_i^t - w_{ji}^t\right)\theta(j, j^*)$$

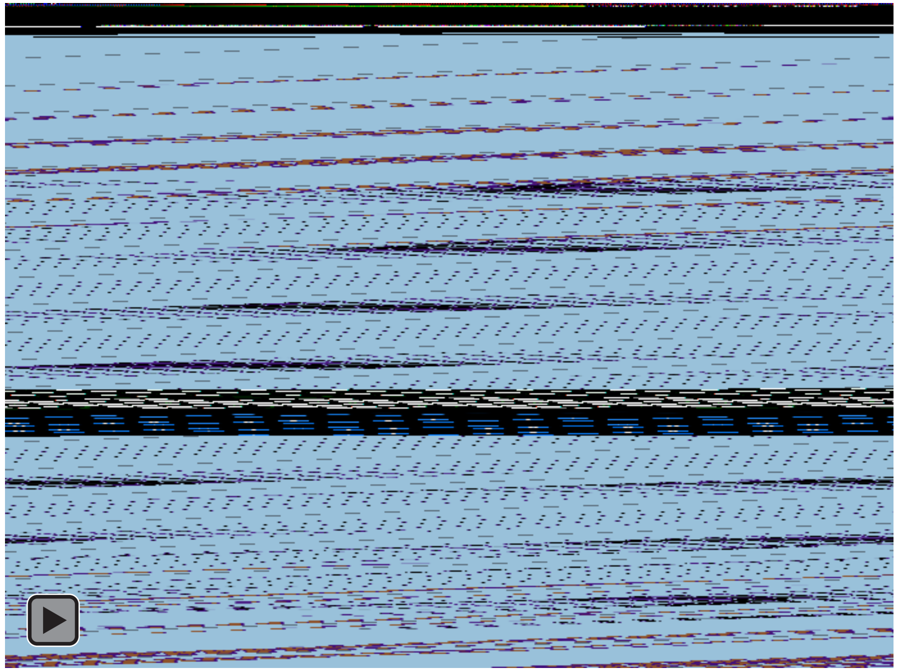
Where $\theta(j, j^*)$ is a restraint function due to the distance between neuron j and j^*.

- The same as in Hebb's rule, to avoid individual neurons being driven into saturation, sometimes the network is provided with some form of normalisation of weights of connections.

Self-Organizing Map (SOM)



Source: Zair M, Rahmoune C, Benazzouz D. Multi-fault diagnosis of rolling bearing using fuzzy entropy of empirical mode decomposition, principal component analysis, and SOM neural network. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2019, 233(9): 3317-3328.



The above figure visualizes the training process of SOM: **similar samples are mapped close together** and **dissimilar ones apart**.