

ذكاء صناعي 2

محاضرة 5

Self-Organizing Maps (SOM)

د. فادي متوج

- ❑ الخرائط ذاتية التنظيم Self-Organizing Maps تسمى أيضاً شبكات كوهونين Kohonen Maps
- ❑ اقترحها العالم الفنلندي Teuvo Kohonen في عام 1982
- ❑ هي شبكة عصبية اصطناعية يتم تدريبها بالاعتماد على **التعلم غير المراقب Unsupervised Learning**، بمعنى آخر إن هذا النوع من الشبكات العصبية تدرّب نفسها بنفسها (تتميز بعدم وجود هدف target)
- ❑ تستخدم SOM بكثرة في عملية **تصنيف البيانات** (data classification and clustering) وتحليلها (data analysis) وإظهارها (data visualization)
- ❑ تقدم SOM أيضاً وسيلة **لتمثيل البيانات متعددة الأبعاد** (multidimensional data) في فضاءات ذات أبعاد أقل من ذلك بكثير - عادةً بعد واحد أو بعدين (**تخفيض أبعاد البيانات**) مما يسهل عملية فهم هذه البيانات وتحليلها
- ❑ عملية تخفيض أبعاد البيانات، هي في الأساس تقنية ضغط بيانات معروفة باسم **vector quantisation**
- ❑ وبالإضافة إلى ذلك، فإن شبكة Kohonen تقوم بتخزين المعلومات بطريقة يتم بها الاحتفاظ بأية علاقات ضمن مجموعة التدريب. لذلك تدعى أيضاً بخرائط الحفاظ على الخواص

SOM Architecture

□ تتكون الشبكة ذاتية التنظيم SOM من مجموعة من **العقد (العصبونات)** المنتظمة على

خريطة ثنائية البعد **2D**

□ كل عصبون (عقدة) مرتبط ارتباطاً كاملاً بطبقة الدخل

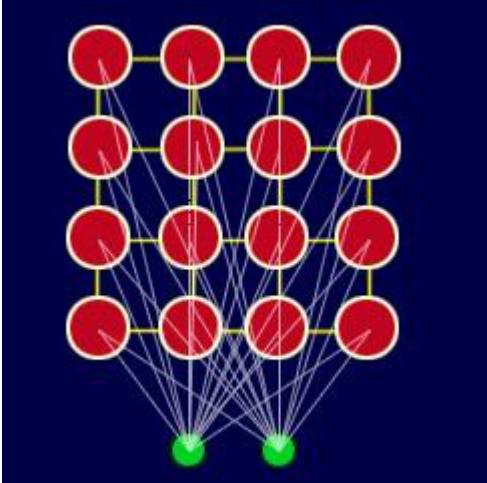
□ يمكن أن يتراوح عدد العصبونات من بضع عشرات إلى عدة آلاف

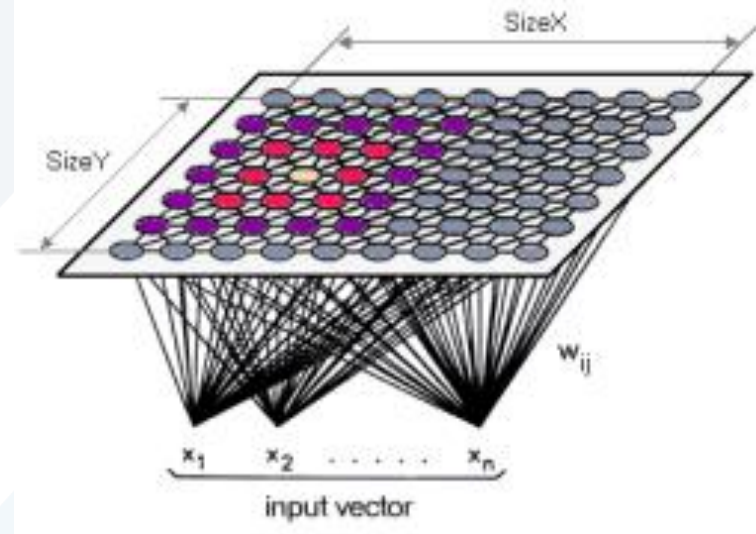
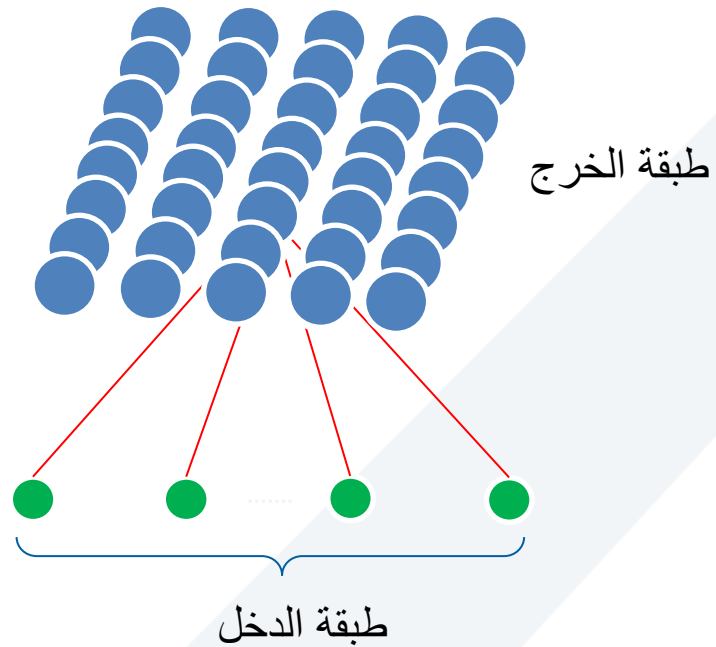
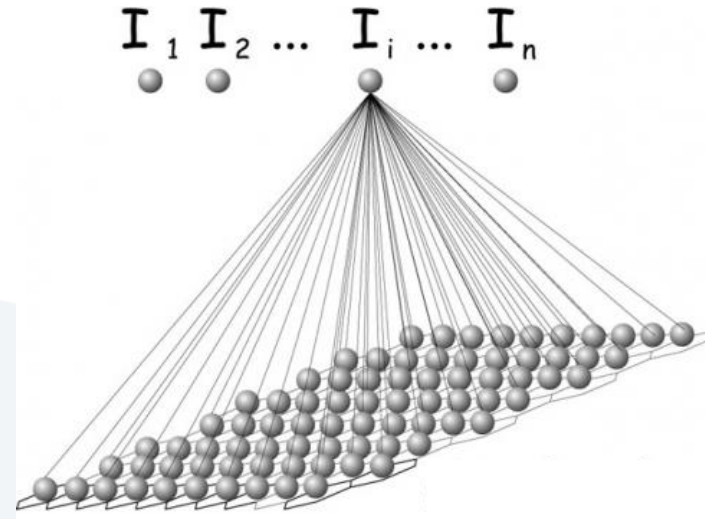
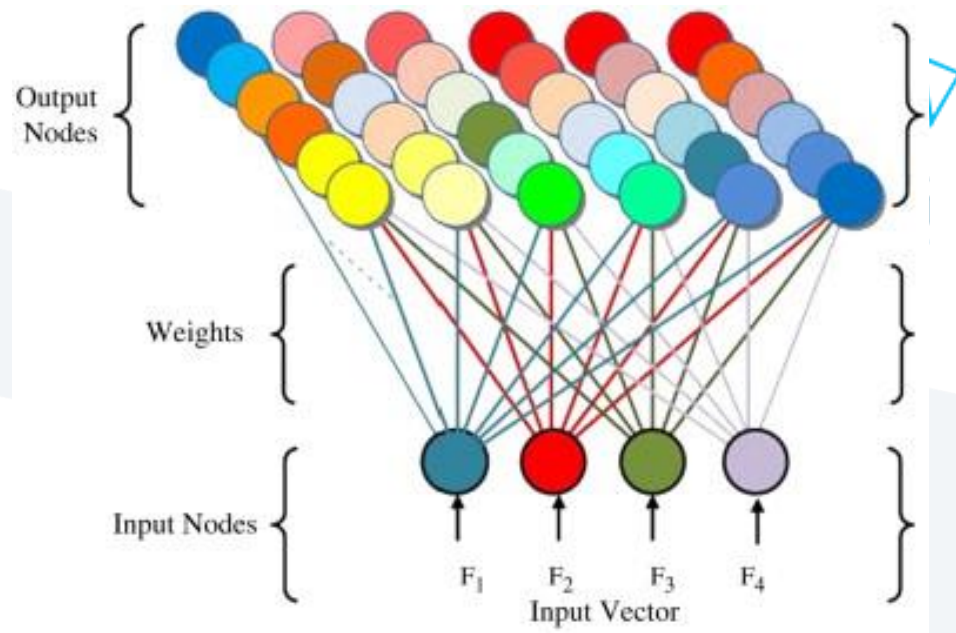
□ كل عصبون لديه موقع محدد (إحداثي x ، إحداثي y) في الشبكة و يمثل بواسطة شعاع وزن بعده يساوي بعد أشعة الدخل. أي إذا كانت بيانات التدريب تتألف من أشعة V ذات n بعد

$V_1, V_2, V_3, \dots, V_n$ فإن كل عصبون يمثل بشعاع W ذات بعد n $W_1, W_2, W_3, \dots, W_n$

□ لا توجد وصلات بين العقد في الشبكة.

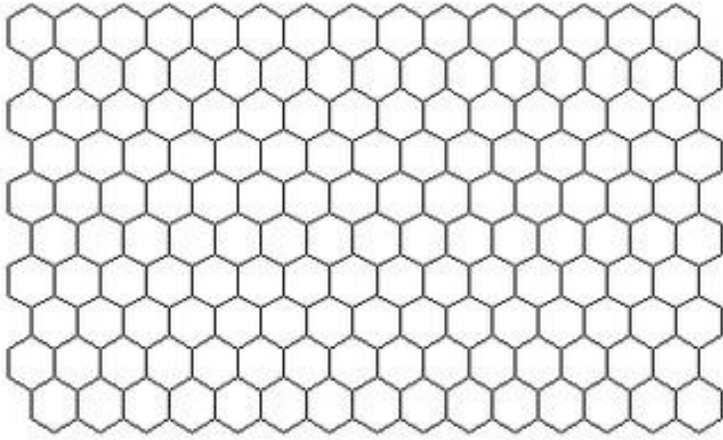
□ ويبين الشكل شبكة Kohonen صغيرة جداً مؤلفة من 4X4 عقدة متصلة بطبقة الدخل والتي تمثل شعاع ثنائي البعد.



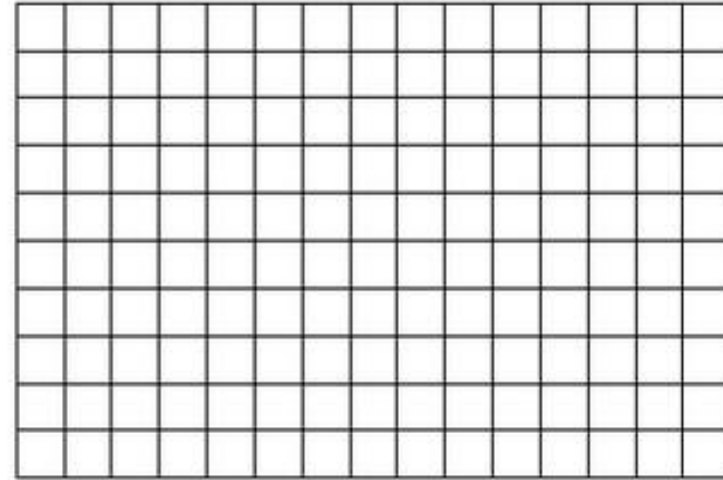


- البنية المستخدمة الأكثر شيوعاً للخريطة هي **المستطيلة Rectangular** والسداسية **Hexagonal**.

Hexagonal SOM grid

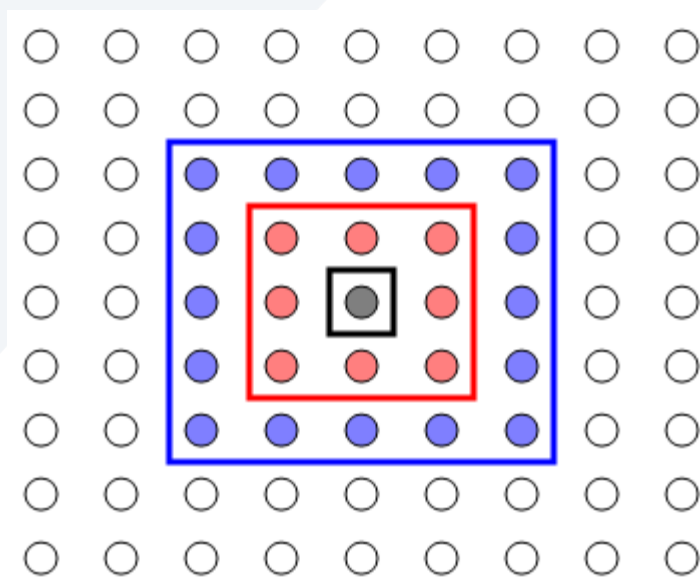
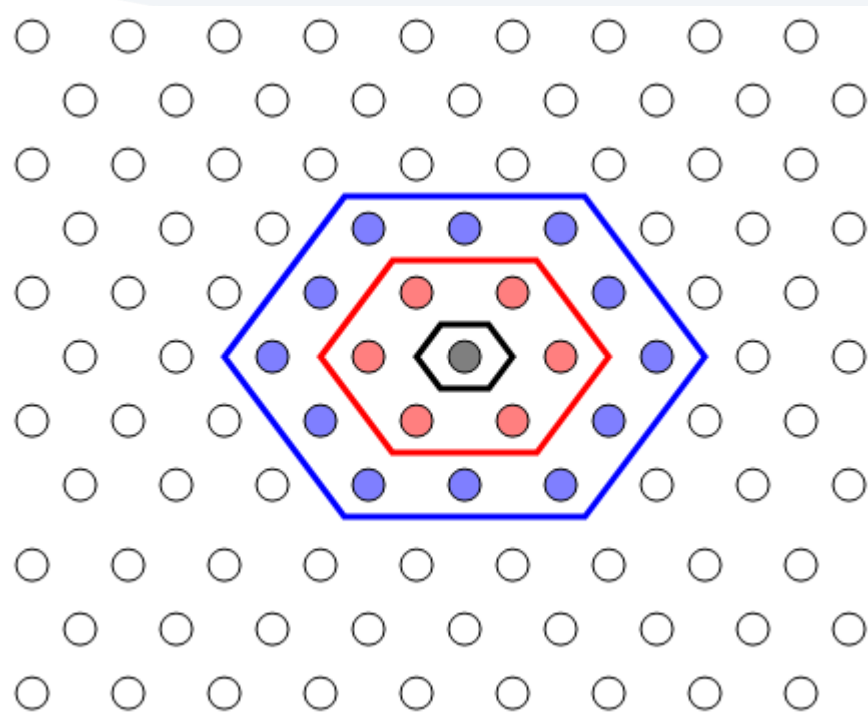


Rectangular SOM grid





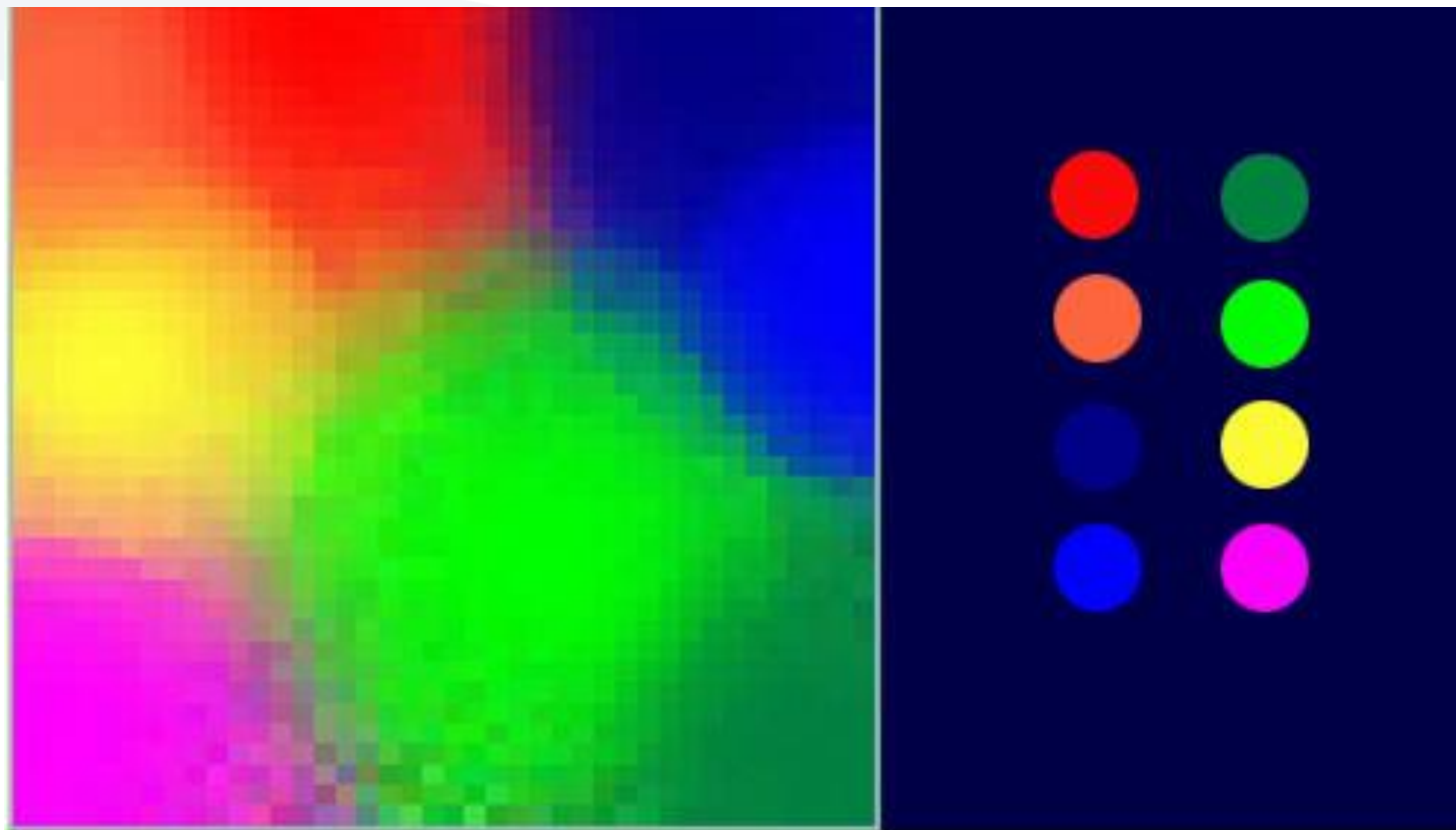
جامعة
المنارة
MANARA UNIVERSITY



- من الأمثلة الشائعة المستخدمة للمساعدة في فهم SOM هو **تمثيل الألوان** من مكوناتها **ثلاثية الأبعاد** (الأحمر والأخضر والأزرق) في **بعدين**.
- يبين الشكل مثالا على SOM تم تدريبها على التعرف على ثمانية ألوان مختلفة تظهر على اليمين.
- تم عرض الألوان على الشبكة على شكل أشعة ثلاثية الأبعاد (بعد واحد لكل من مكونات اللون) وتعلمت الشبكة على تمثيلهم في فضاء 2D.
- يمكن ملاحظة أنه بالإضافة إلى تجميع الألوان في مناطق مختلفة، فإن المناطق ذات الخصائص المتشابهة هي مجاورة لبعضها البعض.



جامعة
المنارة
MANARA UNIVERSITY



- لتجنب الخلط بين هذا النوع من الشبكات العصبية و كل ما تعلمناه سابقاً عن الشبكات العصبية، يجب نسيان كل ما تعلمته حول الشبكات العصبية حيث لا يجب النظر إلى شبكات SOM على أنها مكونة من عصبونات لها توابع تفعيل ووصلات تغذية أمامية/عكسية بل من الأفضل النظر إليها على أنها نوع جديد مختلف

Learning Algorithm



1. تهيئة أوزان كل عصبون بشكل عشوائي بحيث $0 < w < 1$
2. اختيار شعاع دخل بشكل عشوائي من مجموعة بيانات الدخل و عرضه على الشبكة
3. فحص كل عصبون لإيجاد العصبون الذي تكون أوزانه مشابهة أكثر لشعاع الدخل. العصبون الفائز يعرف عادةً **بأفضل وحدة مطابقة (BMU) Best Matching Unit**
4. حساب **نصف قطر المنطقة المجاورة لـ BMU**. حيث تكون هذه القيمة في البداية كبيرة، عادة تعطى قيمة مساوية إلى نصف قطر الخريطة، ولكن تقل مع كل خطوة زمنية
5. **تعديل أوزان** العصبون الفائز و أوزان العصبونات المجاورة له (العصبونات التي وجدناها في الخطوة 4) لجعلها أكثر قرباً من شعاع الدخل. كلما كان العصبون أقرب لـ BMU كلما كان تغير أوزانه أكبر
6. تكرار الخطوة الثانية N مرة

- يمكن وصف عملية تعلم SOM كمنافسة **competition** بين العصبونات لتمثيل أشعة الدخل.
- العصبون الذي شعاع وزنه أقرب إلى شعاع الدخل المقدم يفوز في المنافسة.
- يتم تعديل شعاع الوزن للعصبون الفائز وكذلك للعصبونات في المنطقة المجاورة للفائز لكي تشابه أكثر شعاع الدخل

Finding the Best Matching Unit

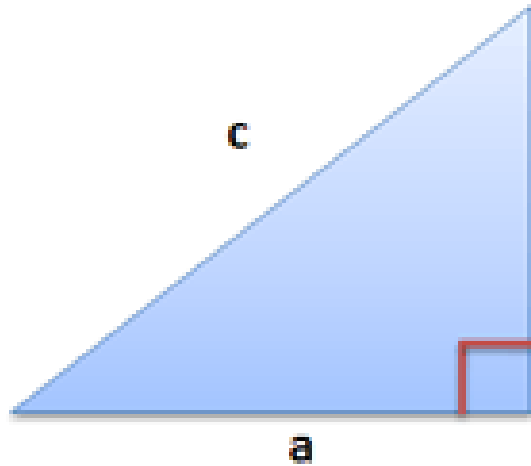
- إحدى الطرق لتحديد أفضل وحدة مطابقة، هي بحساب **المسافة الإقليدية Euclidean distance** بين شعاع وزن كل عقدة و شعاع الدخل الحالي. العصبون الذي يكون شعاع وزنه أقرب إلى شعاع الدخل يكون هو ال BMU

المسافة الإقليدية

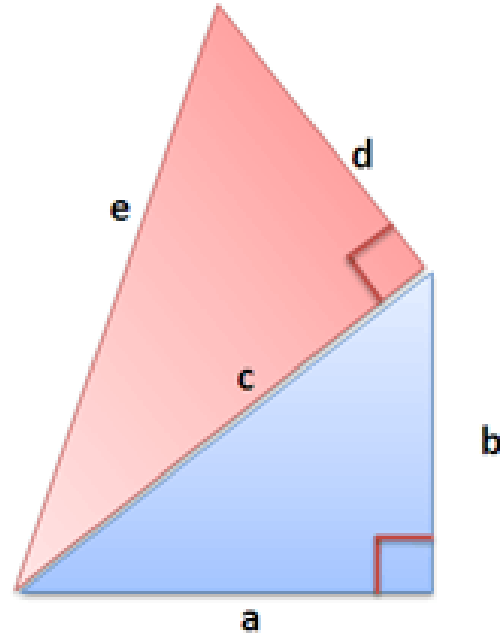
$$D = \sqrt{\sum_{i=0}^{i=n} (V_i - W_i)^2}$$

حيث: V هو شعاع الدخل
 W هو شعاع الوزن

حساب المسافة الإقليدية باستخدام نظرية فيثاغورث



$$a^2 + b^2 = c^2$$

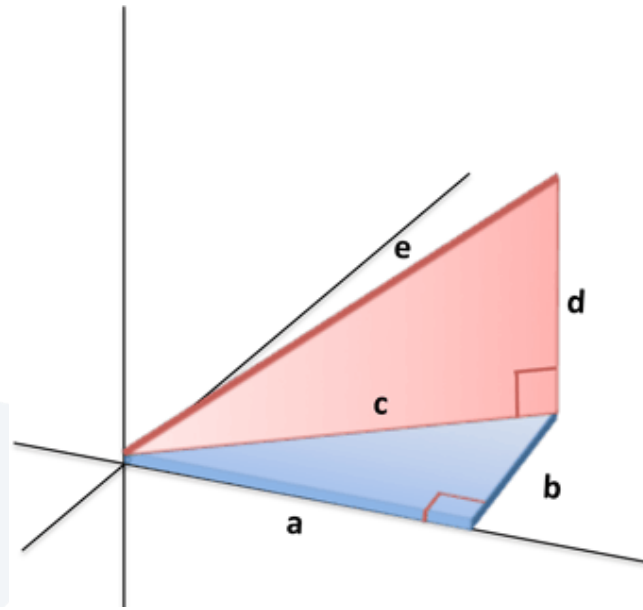


$$\begin{aligned} a^2 + b^2 &= c^2 \\ c^2 + d^2 &= e^2 \end{aligned}$$

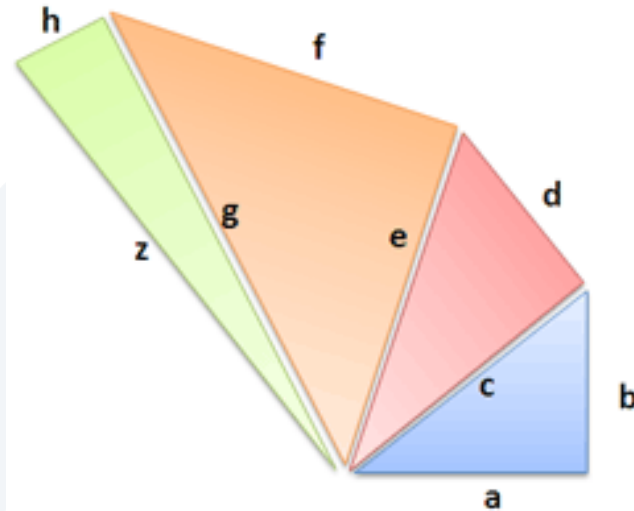


$$a^2 + b^2 + d^2 = e^2$$

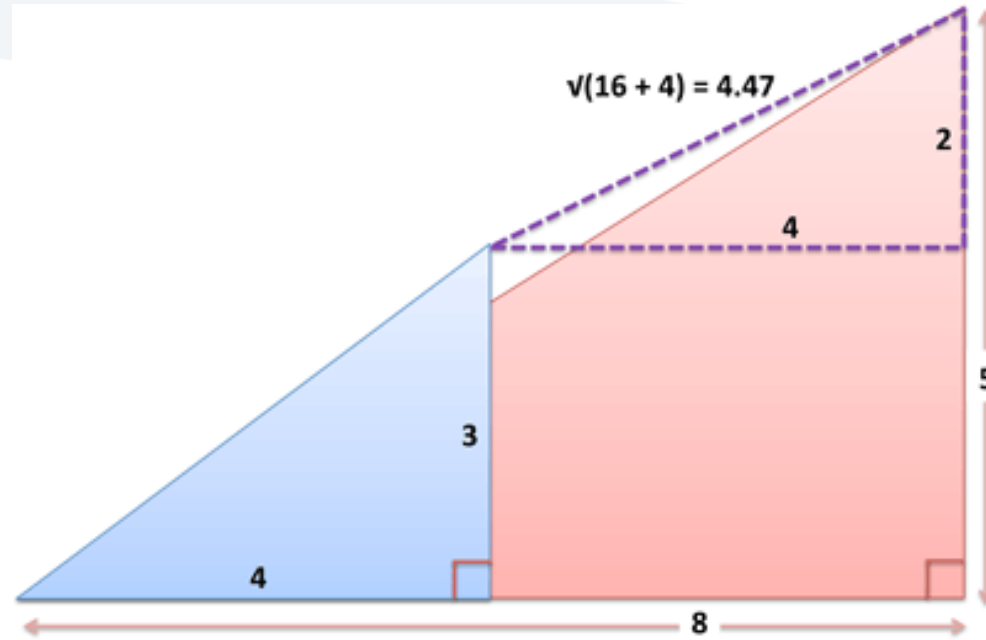
$$x^2 + y^2 + z^2 = \text{distance}^2$$



Use Any Number of Dimensions

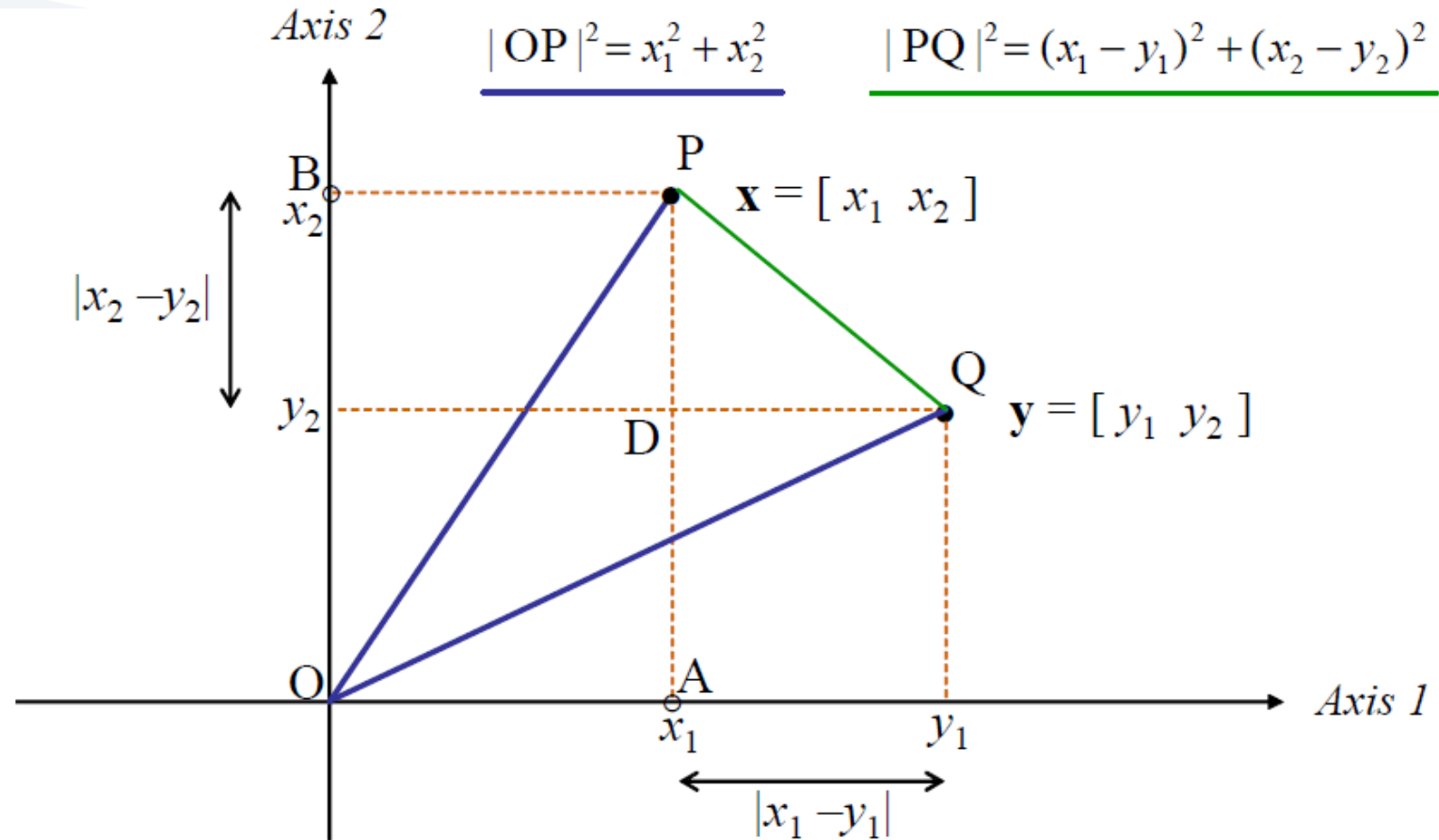


$$a^2 + b^2 + d^2 + f^2 + h^2 = z^2$$



$$distance^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$$

distance between two vectors $\mathbf{x} = [x_1 \ x_2]$ and $\mathbf{y} = [y_1 \ y_2]$



$$d_{x,y}^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2$$

$$d_{x,y} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Example



وكمثال على ذلك، لحساب المسافة بين شعاع دخل (1، 0، 0) و شعاع وزن عشوائي (0.1، 0.4، 0.5)

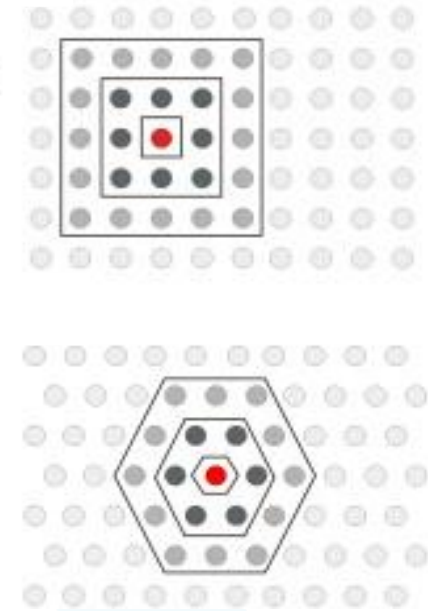
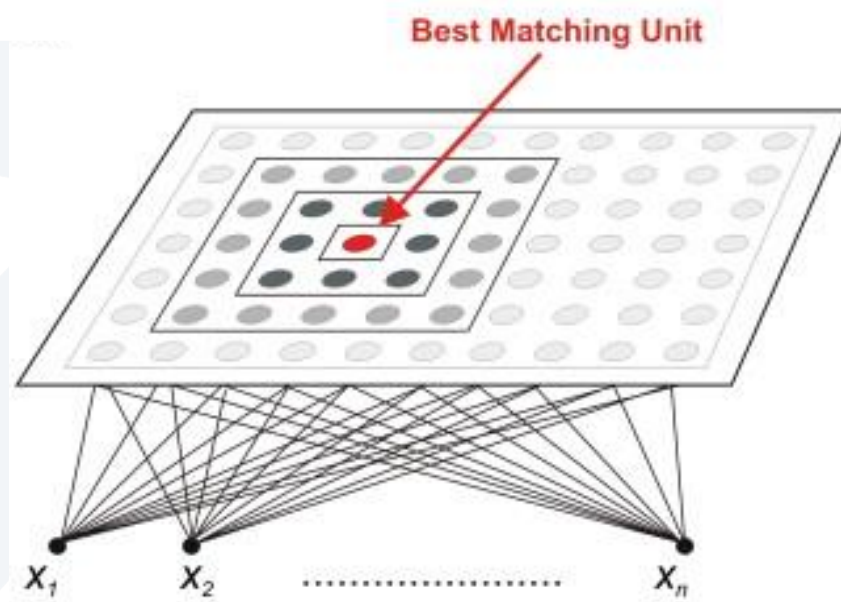
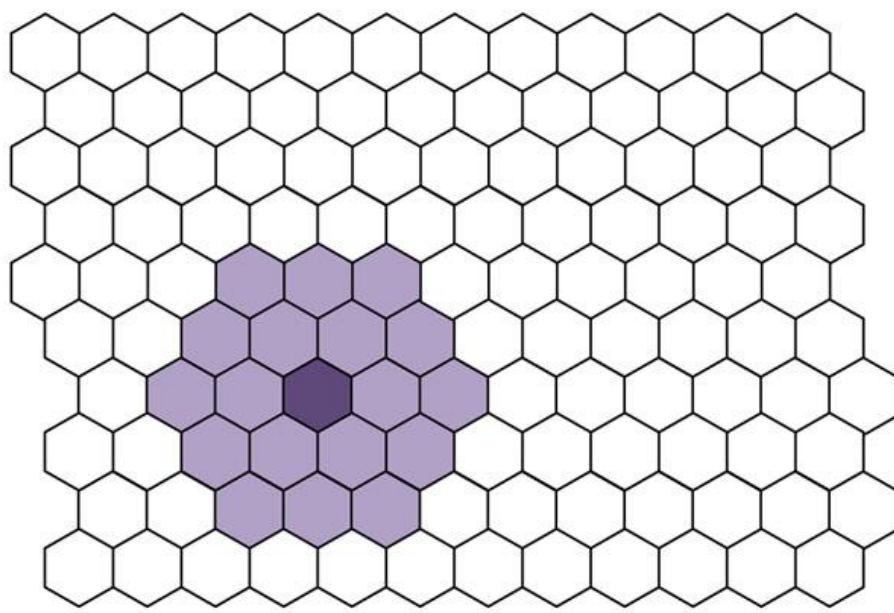
$$\text{distance} = \text{sqrt}((1 - 0.1)^2 + (0 - 0.4)^2 + (0 - 0.5)^2)$$

$$= \text{sqrt}((0.9)^2 + (-0.4)^2 + (-0.5)^2)$$

$$= \text{sqrt}(0.81 + 0.16 + 0.25)$$

$$= \text{sqrt}(1.22)$$

$$\text{distance} = 1.106$$



العلاقة الرياضية لتعديل الأوزان

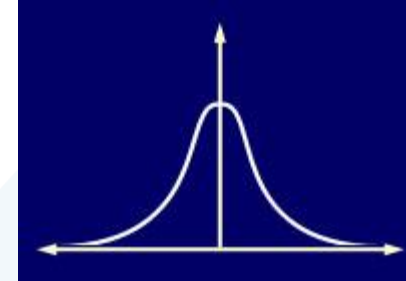
- يتم تعديل أوزان العصبون الفائز و العصبونات المجاورة له وفق العلاقة

معدل التعلم

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 1, 2, 3, \dots$$

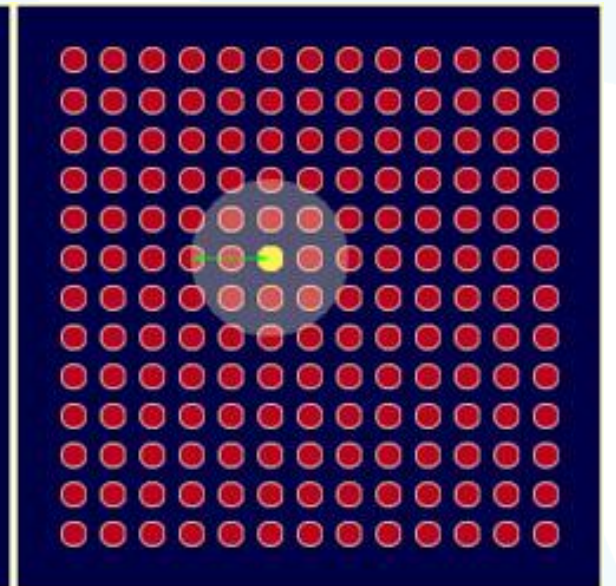
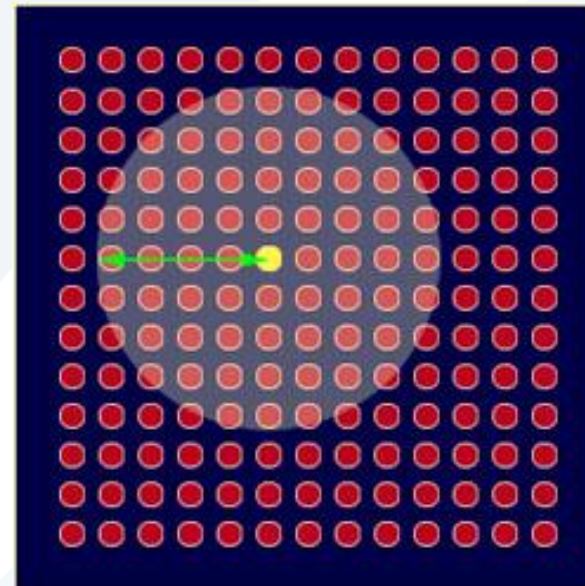
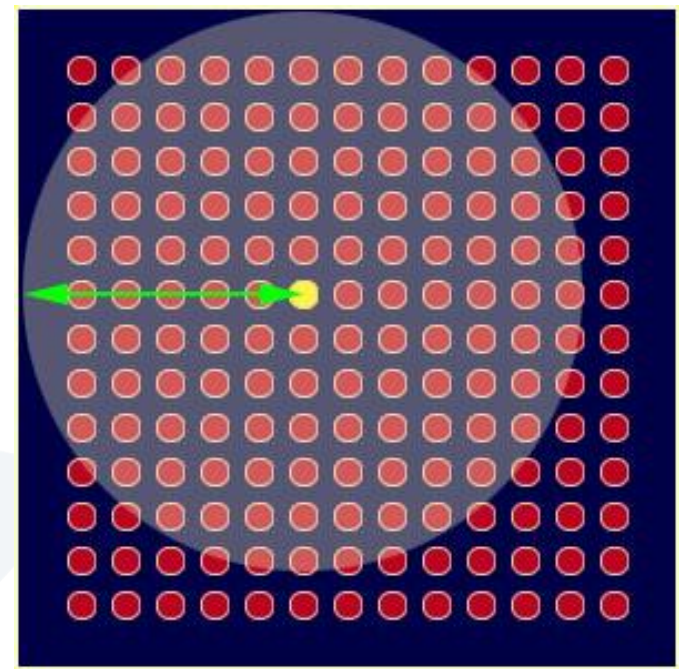
تابع الجوار

$$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right) \quad t = 1, 2, 3, \dots$$

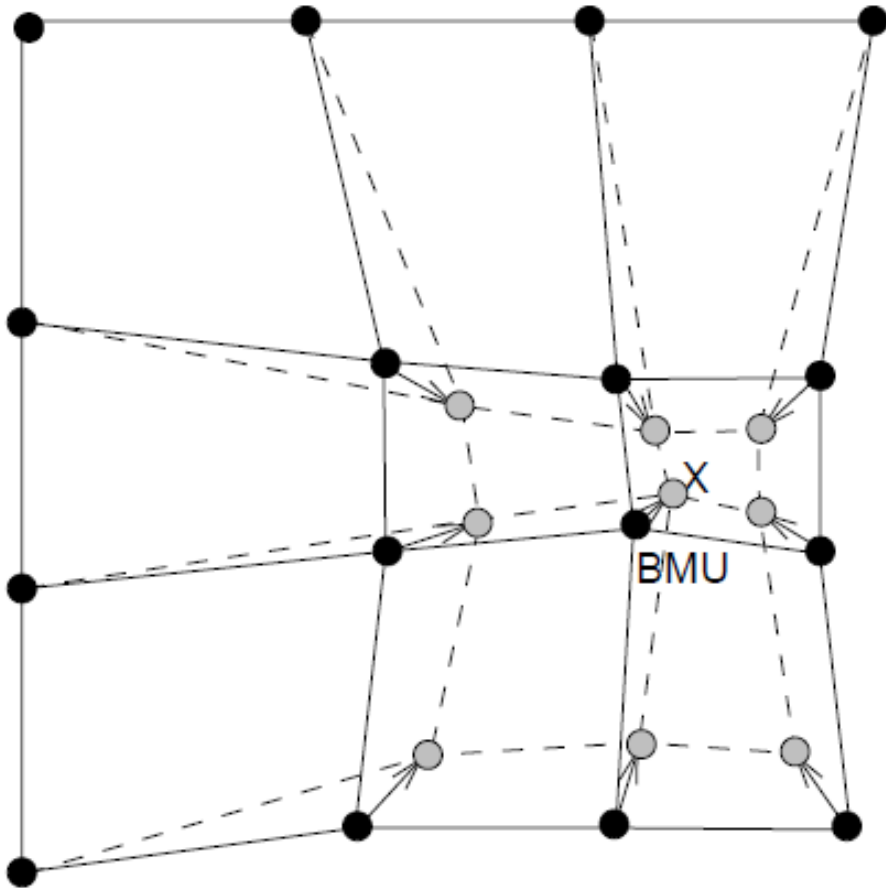


- تتناقص منطقة الجوار مع الزمن مع تقدم عملية التدريب.
- كذلك يتناقص معدل التعلم مع تقدم عملية التدريب

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 1, 2, 3, \dots$$



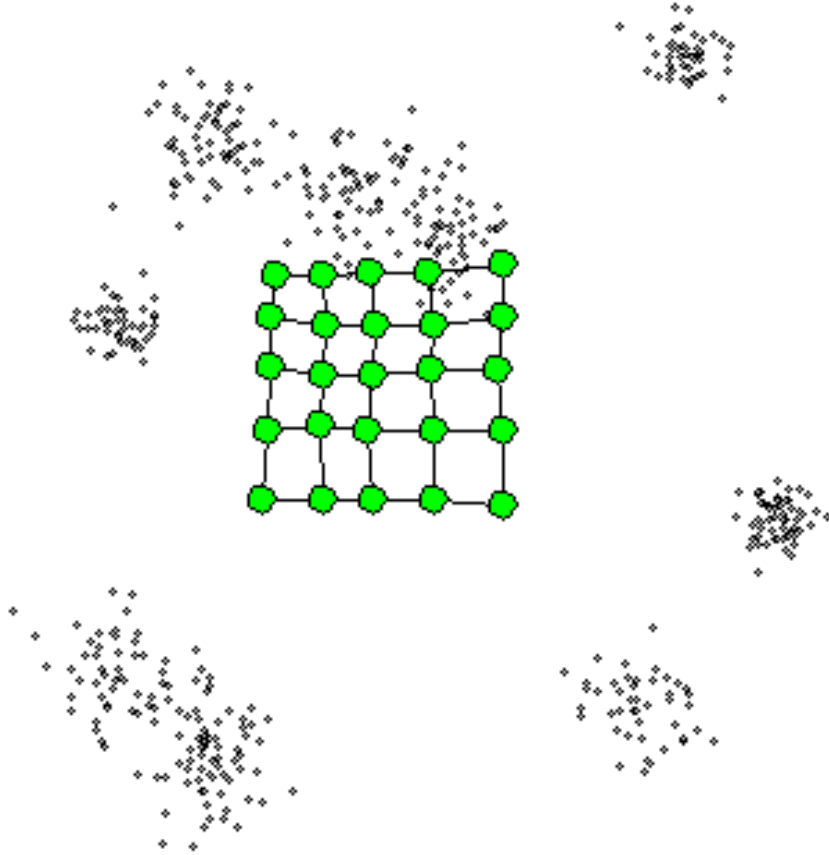
تحديث أفضل وحدة مطابقة (BMU) وجيرانها باتجاه عينة الدخل التي يرمز لها x



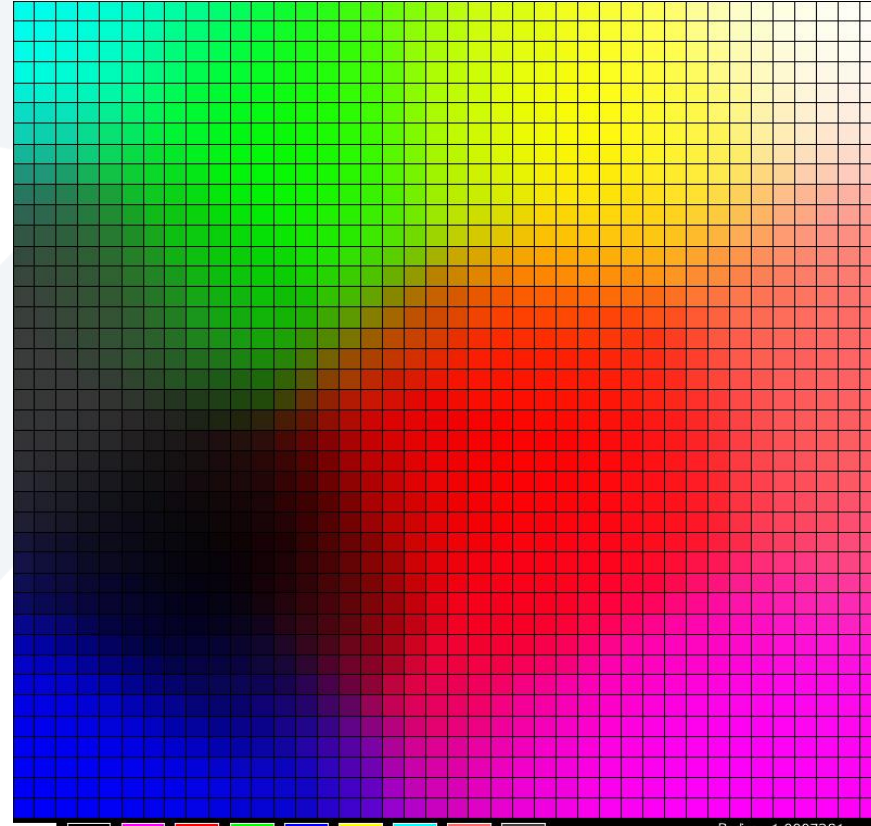
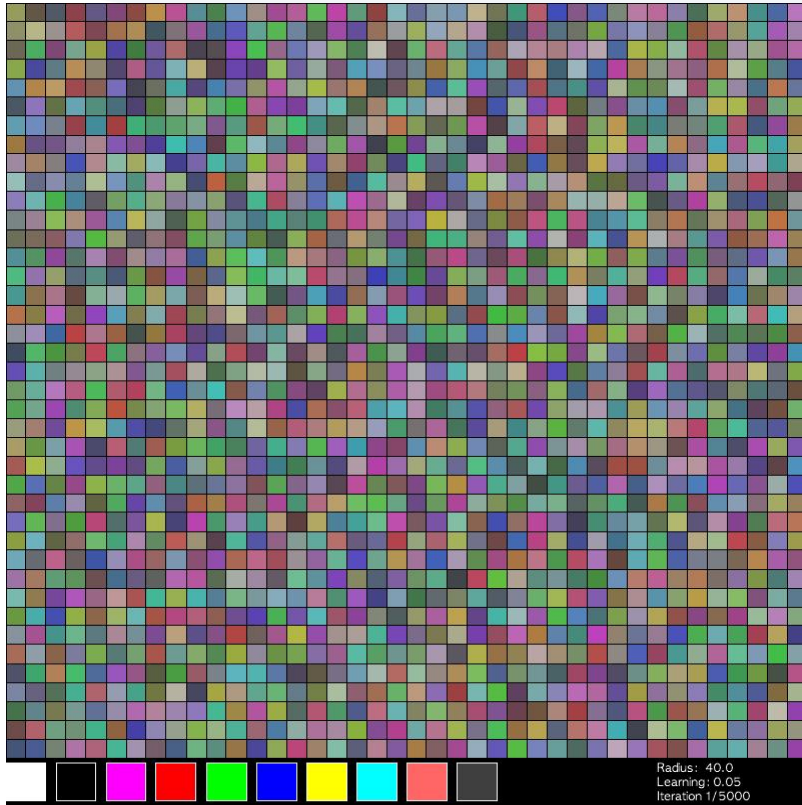
الخطوط المستمرة والمتقطعة تقابل الحالة قبل وبعد تعديل الأوزان، على التوالي.

يوضح المثال التالي مبدأ عمل SOM

(النقاط السوداء = البيانات، النقاط الخضراء = العصبونات):

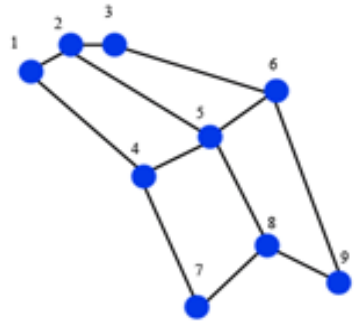


مثال : تصنيف الألوان



إظهار البيانات عن طريق SOM

• مثال 1: SOM 3×3

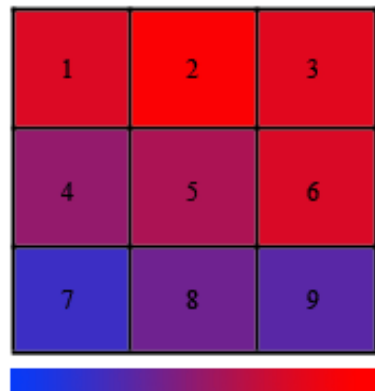


1	2	3
4	5	6
7	8	9

X



Y



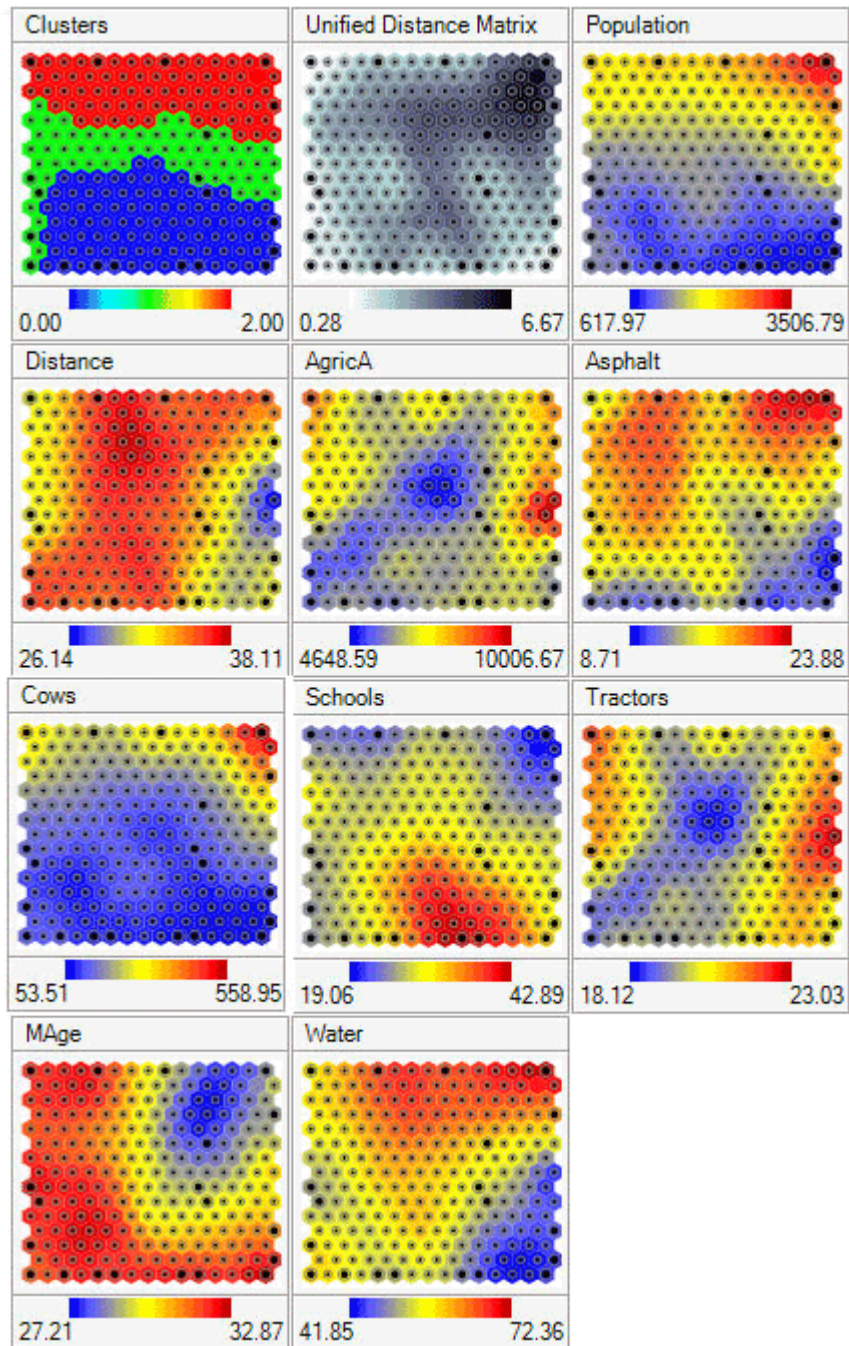
Unit 1: Low X value, High Y value
Unit 9: High X value, Low Y value

مثال على تطبيق SOM:



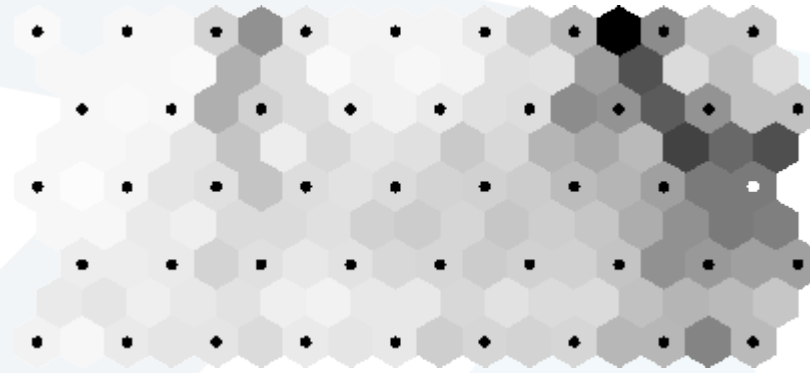
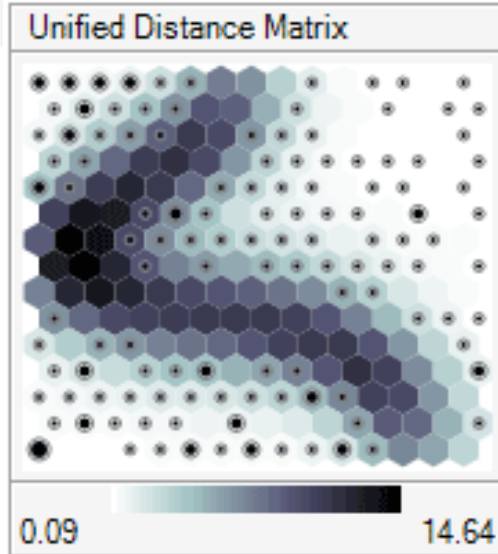
- لدينا بيانات تحتوي عددا من المتغيرات المقاسة في قرى مختلفة. حيث تمثل كل عينة قرية

Population - عدد السكان الذين يعيشون في القرية
Distance - المسافة إلى أقرب مدينة (كم)
AgricA - منطقة زراعية (هكتار)
Asphalt - الطرق الإسفلتية (كم)
Cows - عدد الأبقار في القرية
Schools - عدد المدارس
Tractors - عدد الجرارات
MAge - متوسط العمر (بالسنوات)
Water - نوعية المياه (0-100%)



U-Matrix

Unified Distance Matrix



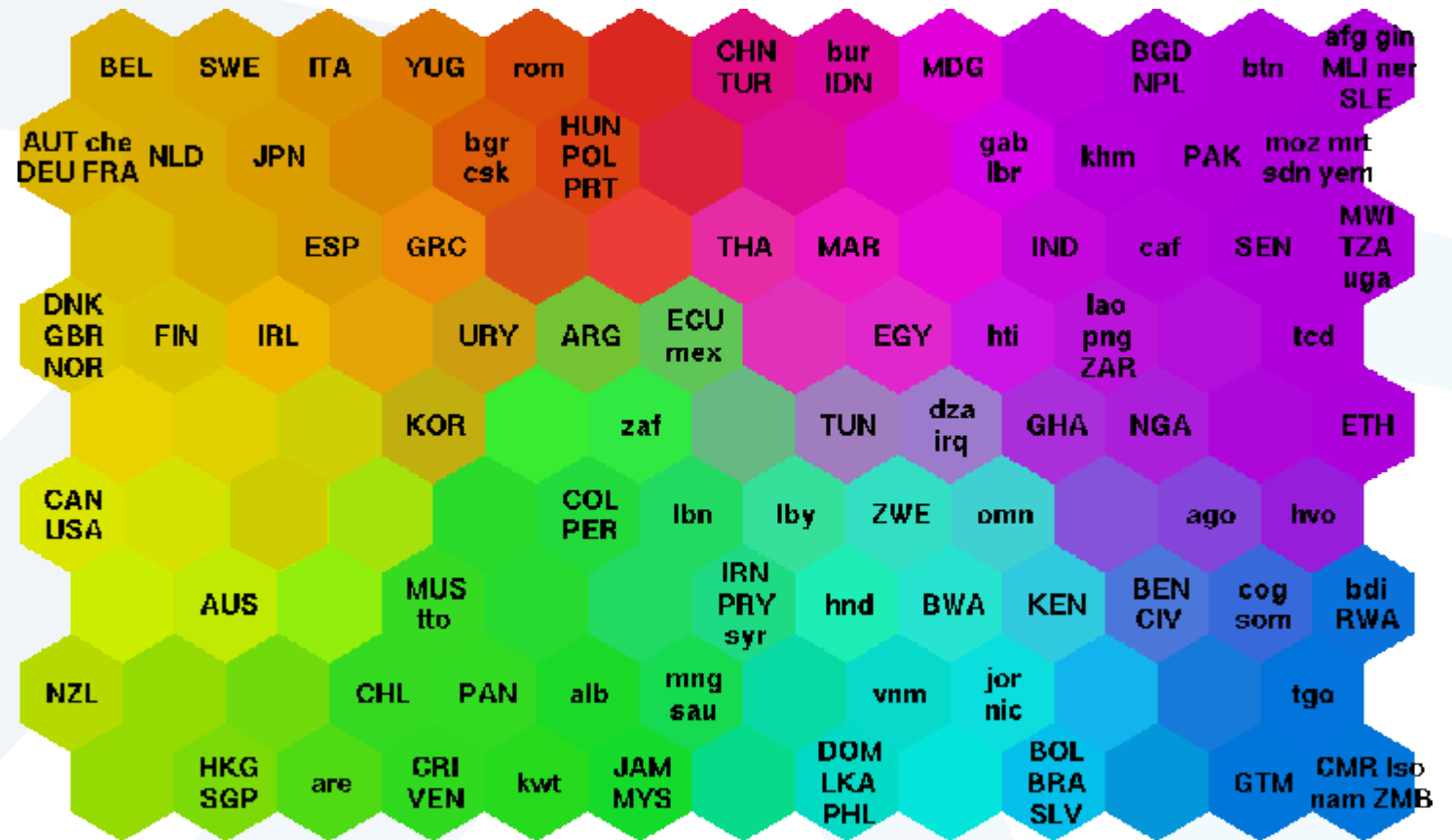
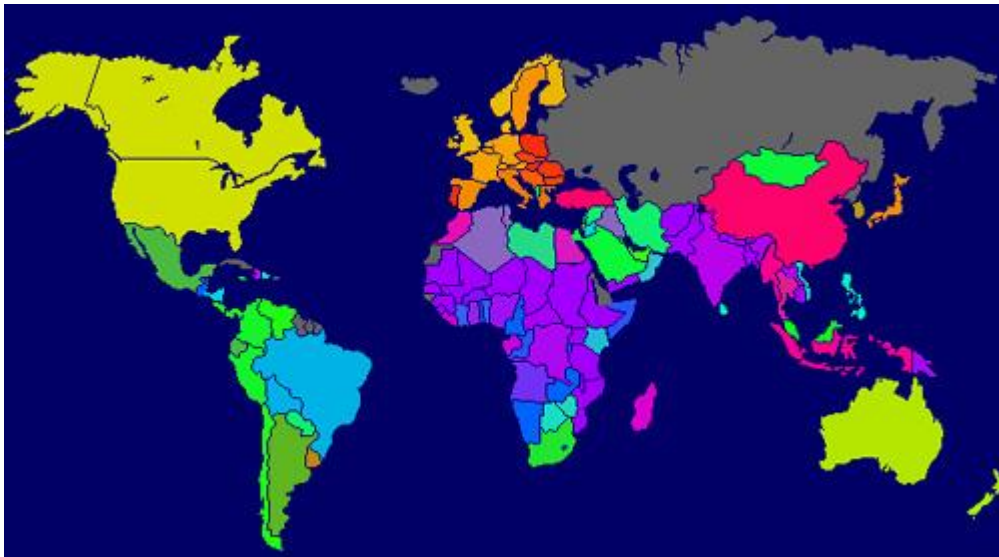
المناطق المظلمة تعني أن المسافة بين العقد كبيرة
المناطق الفاتحة تعني أن العقد قريبة من بعضها البعض.

مثال :خريطة الفقر العالمية

تكونت هذه البيانات من إحصاءات البنك الدولي حول البلدان في عام 1992. واعتمدت على 39 مؤشراً تصف مختلف العوامل المتعلقة بنوعية الحياة، مثل الحالة الصحية، والتغذية، والخدمات التعليمية، وغيرها.



جامعة
المنارة
MANARA UNIVERSITY



مثال تصنيف صور مناظر طبيعية



مثال تصنيف موديلات سيارات



Example



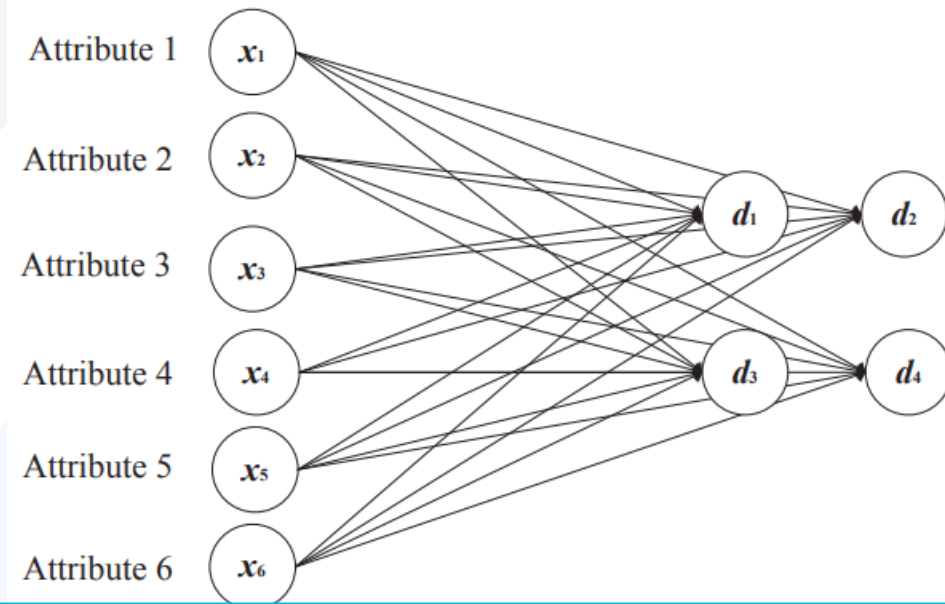
- An illustrative SOM example about psychographic segmentation analysis will be presented. The aim of using SOM is to divide a heterogeneous market into smaller homogeneous subsets consisting of customers who have relatively similar characteristics. The data collected through a survey consists of customer responses ($k = 20$) to items of personality traits ($j = 6$) including being social (x_1), innovative (x_2), confident (x_3), logical (x_4), cheerful (x_5), and practical (x_6). Table below presents the customer responses as input data. Input values range from 0 to 1 where 1 denotes “strongly agree” and 0 denotes “strongly disagree”

Input space

Customer	Personality Traits (Attributes)					
	x_1	x_2	x_3	x_4	x_5	x_6
1	1	0.25	0.75	0	0.50	0.50
2	0.25	0.75	0.75	0.50	0.25	0.50
3	0.75	0.25	0.50	0.50	0.50	0.25
4	0.50	1	1	0.75	0.75	0
...
20	0	0.25	0.75	0.25	0.50	0.75

Example (continued)

- Since there are 20 input samples, at least $20 * (10\%) = 2$ clusters can be used to represent the output layer. However, in order to construct a square-type two dimensional map, a topologic structure with 2×2 network size has been chosen.
- As shown in Figure below, the SOM in this case consists of 6 input nodes and 4 output nodes arranged in form of a 2×2 map. Because all input nodes are measured on the same scale, input vectors do not need to be normalized.



Example (continued)



Initial Weights:

- The formation of the SOM starts first by initializing the weight vectors $W_i = (w_{i1} \ w_{i2} \ \dots \ w_{im})'$ where $i = 1, 2, \dots, n$ denotes the number of output nodes in the network.
- Weights are links that connect the input nodes to the output nodes and are updated through the learning process.
- The most common initialization method assigns preferably small random values to the weight vectors of nodes in the network.

Example (continued)

Similarity Matching:



- Learning in SOM does not rely on predefined target outcomes that would guide the process. The output nodes compete among themselves to become activated. Only the node whose weight vector is most similar to the input vector will be activated and declared as the winner.
- To find this **best matching node**, the distances between an input data (x) and all the weight vectors (w_i) of the SOM are computed using different measurement methods, such as Euclidean distance.
- The **Euclidean distance** between a sample x , chosen randomly from the input dataset, and all the weight vectors at iteration t is calculated by using the following formula:

$$d_i(t) = \|x(t) - w_i(t)\| = \sqrt{\sum_{j=1}^m (x_{tj} - w_{tji})^2} \quad i = 1, 2, \dots, n$$

- The weight vector w_i of each output node i has the same dimension as the input vector x .
- At the end of the similarity matching process, the best-matching (winning) node c at iteration t is determined by using the minimum distance Euclidean criterion:

$$c(t) = \arg \min_i \{\|x(t) - w_i(t)\|\}$$

Example (continued)



- In our market segmentation example, for all four output nodes the initial weight vectors are assigned random values as shown in Table below.
- The node whose weight vector is most similar to the customer response data, randomly chosen from the input space, will be activated and adjusted together with its neighboring units.
- The Euclidean distance is applied to calculate the distances between the input data and all the weight vectors. Suppose that the first sample record $x(0) = (0.25 \ 0.75 \ 0.75 \ 0.50 \ 0.25 \ 0.50)'$ randomly selected is the response of customer number two.

Initial weights

j	w_1	w_2	w_3	w_4
1	0.15	0.06	0.11	0.38
2	0.97	0.84	1.00	0.78
3	0.65	0.50	0.99	0.79
4	0.27	0.35	0.64	0.68
5	0.09	0.39	0.19	0.44
6	0.46	0.85	0.47	0.22

Example (continued)



- The Euclidean distances for the four output nodes are calculated:

$$d_1(\mathbf{0}) = \sqrt{(0.25-0.15)^2 + (0.75-0.97)^2 + (0.75-0.65)^2 + (0.5-0.27)^2 + (0.25-0.09)^2 + (0.5-0.46)^2} = 0.385$$

$$d_2(\mathbf{0}) = \sqrt{(0.25-0.06)^2 + (0.75-0.84)^2 + (0.75-0.50)^2 + (0.5-0.35)^2 + (0.25-0.39)^2 + (0.5-0.85)^2} = 0.521$$

$$d_3(\mathbf{0}) = \sqrt{(0.25-0.11)^2 + (0.75-1.00)^2 + (0.75-0.99)^2 + (0.5-0.64)^2 + (0.25-0.19)^2 + (0.5-0.47)^2} = 0.405$$

$$d_4(\mathbf{0}) = \sqrt{(0.25-0.38)^2 + (0.75-0.78)^2 + (0.75-0.79)^2 + (0.5-0.68)^2 + (0.25-0.44)^2 + (0.5-0.22)^2} = 0.408$$

- For the first input record, the winning node is the first output node ($c(\mathbf{0}) = 1$) that has the minimum Euclidean distance, $d_1(\mathbf{0}) = 0.385$. In other words, the first output node is most similar to the given input node.

Example (continued)

Weight Updating

- The weight vectors of the winner and its neighboring units in the output space are adjusted to become more representative of the features that characterize the input space.
- This updating towards the input sample requires the consideration of the following two parameters: **learning rate and neighborhood size**.
- The learning rate, $\alpha(t)$, controls the rate of change of the weight vectors and, as in all neural networks, it takes values between 0 and 1.
- In SOMs, the learning rate gradually decreases as a function of the iteration step index t . That means, while the step index increases, the learning rate might decrease linearly, exponentially or geometrically, or it can be inversely proportional to t .
- The learning rate should start with a reasonably high value that is close to unity and progressively reduce to small values. This procedure corresponds to larger corrections at the beginning of the training process (i.e. the ordering phase) than at the end where fine-tuning of the map takes place (i.e. the convergence phase).
- The ordering phase may take as many as 1000 iterations of the SOM algorithm, and possibly more, while the convergence phase of the adaptive process must be at least 500 times the number of neurons in the network

Example (continued)

Weight Updating

- The updated weight vector $w_i(t+1)$ at iteration $t+1$ is defined by

$$w_i(t+1) = w_i(t) + \alpha(t)[x(t) - w_i(t)]$$

- The rate of adaptation of the weights decreases away from the winning node, according to a neighborhood function $h_{ci}(t)$ where i denotes the index of the neighboring unit.
- The most preferred function of this type is the Gaussian function which decreases in both the spatial domain and time domain:

$$h_{ci}(t) = \exp\left(-\frac{d_{ci}^2}{2\sigma^2(t)}\right)$$

where d_{ci}^2 denotes the lateral distance between the winning neuron c and the excited neuron i , and $\sigma(t)$ represents the effective width or radius of neighborhood at iteration t .

- The Gaussian function is symmetric about the maximum point defined by $d_{ci} = 0$, and decrease monotonically to zero with increasing lateral distance (d_{ci})
- Here, $\sigma(t)$ is some monotonically decreasing function of time leading to a decrease in the value of the neighborhood function.

Example (continued)



- Finally, after defining the neighborhood set around the winning node c , a neighborhood function is chosen to update the weight vectors of the respective nodes. The following formula presents the update rule for the weight vector of unit i :

$$w_i(t + 1) = w_i(t) + \alpha(t)h_{ci}(t)[x(t) - w_i(t)]$$

- All three processes – competition, cooperation and adaptation – described here are repeated for the remaining training data until the weights converge and no noticeable changes in the low-dimensional output layer are observed.

Example (continued)

- For the first sample record selected, the winning node was the first output node ($c = 1$) that had the minimum Euclidean distance. Now the weight vectors of this best matching node and its neighborhood will be adjusted so that they become more similar to the current input sample $x(0)$. Setting the initial learning rate close to one ($\alpha(0) = 0.9$) and applying the adaptation formula, the updated weights for the winning node ($c = 1$) at $t = 0$ are calculated as shown below:

$$w_1(1) = w_1(0) + \alpha(0)[x(0) - w_1(0)]$$

$$\begin{bmatrix} w_{11}(1) \\ w_{12}(1) \\ w_{13}(1) \\ w_{14}(1) \\ w_{15}(1) \\ w_{16}(1) \end{bmatrix} = \begin{bmatrix} 0.15 \\ 0.97 \\ 0.65 \\ 0.27 \\ 0.09 \\ 0.46 \end{bmatrix} + 0.9 \left(\begin{bmatrix} 0.25 \\ 0.75 \\ 0.75 \\ 0.50 \\ 0.25 \\ 0.50 \end{bmatrix} - \begin{bmatrix} 0.15 \\ 0.97 \\ 0.65 \\ 0.27 \\ 0.09 \\ 0.46 \end{bmatrix} \right) = \begin{bmatrix} 0.24 \\ 0.77 \\ 0.74 \\ 0.48 \\ 0.23 \\ 0.50 \end{bmatrix}$$

Example (continued)



- If the learning rate was chosen equal to zero, the weights would not change, i.e. $w_1(1) = w_1(0)$.
- Next, applying the Gaussian neighborhood function with $\sigma(0) = 1$ and the adaptation formula given early the updated weights for the neighboring units at $t = 0$ are calculated as shown below:

$$h_{12}(0) = \exp\left(-\frac{d_{12}^2}{2\sigma^2(0)}\right) = \exp\left(-\frac{0.544^2}{2 \cdot 1^2}\right) = 0.862$$

$$h_{13}(0) = \exp\left(-\frac{d_{13}^2}{2\sigma^2(0)}\right) = \exp\left(-\frac{0.515^2}{2 \cdot 1^2}\right) = 0.876$$

$$h_{14}(0) = \exp\left(-\frac{d_{14}^2}{2\sigma^2(0)}\right) = \exp\left(-\frac{0.676^2}{2 \cdot 1^2}\right) = 0.796$$

where

$$d_{12} = \sqrt{(0.15-0.06)^2+(0.97-0.84)^2+(0.65-0.50)^2+(0.27-0.35)^2+(0.09-0.39)^2+(0.46-0.85)^2}=0.544$$

$$d_{13} = \sqrt{(0.15-0.11)^2+(0.97-1.00)^2+(0.65-0.99)^2+(0.27-0.64)^2+(0.09-0.19)^2+(0.46-0.47)^2}=0.515$$

$$d_{14} = \sqrt{(0.15-0.38)^2+(0.97-0.78)^2+(0.65-0.79)^2+(0.27-0.68)^2+(0.09-0.44)^2+(0.46-0.22)^2}=0.676$$

hence, we obtain for output node 2:

$$w_2(1) = w_2(0) + \alpha(0)h_{12}(0)[x(0) - w_2(0)]$$

Example (continued)

$$\begin{bmatrix} w_{21}(1) \\ w_{22}(1) \\ w_{23}(1) \\ w_{24}(1) \\ w_{25}(1) \\ w_{26}(1) \end{bmatrix} = \begin{bmatrix} 0.06 \\ 0.84 \\ 0.50 \\ 0.35 \\ 0.39 \\ 0.85 \end{bmatrix} + (0.9)(0.862) \left(\begin{bmatrix} 0.25 \\ 0.75 \\ 0.75 \\ 0.50 \\ 0.25 \\ 0.50 \end{bmatrix} - \begin{bmatrix} 0.06 \\ 0.84 \\ 0.50 \\ 0.35 \\ 0.39 \\ 0.85 \end{bmatrix} \right) = \begin{bmatrix} 0.21 \\ 0.77 \\ 0.69 \\ 0.47 \\ 0.28 \\ 0.58 \end{bmatrix}$$

The weights of the output nodes 3 and 4 are calculated similarly. The adjusted weight vectors of the winning node and its neighborhood are given below:

$$w_1(1) = (0.24 \ 0.77 \ 0.74 \ 0.48 \ 0.23 \ 0.50)'$$

$$w_2(1) = (0.21 \ 0.77 \ 0.69 \ 0.47 \ 0.28 \ 0.58)'$$

$$w_3(1) = (0.22 \ 0.80 \ 0.80 \ 0.53 \ 0.24 \ 0.49)'$$

$$w_4(1) = (0.29 \ 0.76 \ 0.76 \ 0.55 \ 0.30 \ 0.42)'$$

Summary of the SOM Algorithm



The basic steps of Kohonen's SOM algorithm can be summarized by the following iterative procedure:

- (i) *Initialization*. Choose the dimension and size of the output space. Assign random values or alternatively sample vectors drawn randomly from the training set to the initial weight vectors $w_i(0)$. Specify the neighborhood function as well as the functional form of the learning rate $\alpha(0)$ and radius of the neighborhood $\sigma(0)$. Assign starting values for $\alpha(0)$ and $\sigma(0)$. Normalize training data. Define a critical threshold value (T) for the maximum number of iterations. Set iteration index $t = 1$.
- (ii) *Sampling*. Randomly select an input vector $x(t)$ from the training data set.
- (iii) *Similarity Matching*. Compute the Euclidean distances between the input vector and each output node's weight vector and find the best matching node $c(t)$ at iteration t by applying the minimum distance criterion:

$$c(t) = \underset{i}{\operatorname{arg\,min}} \{ \|x(t) - w_i(t)\| \} \quad i = 1, 2, \dots, n$$

Summary of the SOM Algorithm (continued)



(iv) *Weight Updating.* Adjust the weights of the winning node and its neighborhood according to their distances to the winning node by using the update formula:

$$w_i(t+1) = w_i(t) + \alpha(t)h_{ci}(t)[x(t) - w_i(t)]$$

For the winning node the neighborhood function $h_{ci}(t)$ will be equal to 1.

- (v) *Parameter Adjustment.* Set $t = t + 1$. Adjust the neighborhood size and the learning rate.
- (vi) *Continuation.* Keep returning to Step 2 until the change of the weights is less than a prespecified threshold value or the maximum number T of iterations is reached. Otherwise stop.