

## جلسه اول: شبکه‌های عصبی خودسازمان‌ده (Self-Organizing Maps - SOM)

در روش شبکه عصبی SOM، فرآیند اولیه تا حدی مشابه با روش خوشه‌بندی K-Means است. در ابتدای کار، برای شروع آموزش شبکه، نیاز به مراکز داریم که معمولاً از طریق روش‌های خوشه‌بندی اولیه به دست می‌آیند. این مراکز خوشه‌ها در لایه‌ای از نورون‌ها قرار می‌گیرند که به آن **لایه رقابتی** گفته می‌شود.

در ادامه، داده‌های ورودی به شبکه داده می‌شوند. هر نورون (که نماینده یک خوشه است) فاصله خود را با داده ورودی محاسبه می‌کند (معمولاً با استفاده از فاصله اقلیدسی). نورونی که کمترین فاصله را با ورودی دارد، برنده رقابت می‌شود و به عنوان **نورون برنده - Best Matching Unit (BMU)** شناخته می‌شود.

در این مرحله، تنها نورون برنده فعال نمی‌شود، بلکه نورون‌های همسایه آن نیز تحت تأثیر قرار می‌گیرند. این تأثیر از طریق به‌روزرسانی وزن‌ها (وزن نورون‌ها به ورودی‌ها) صورت می‌گیرد، به گونه‌ای که نورون‌های برنده و همسایه‌های نزدیک آن، وزن‌های خود را به سمت داده ورودی حرکت می‌دهند. این فرآیند به تدریج باعث شکل‌گیری نگاشت منظم از داده‌ها روی شبکه می‌شود.

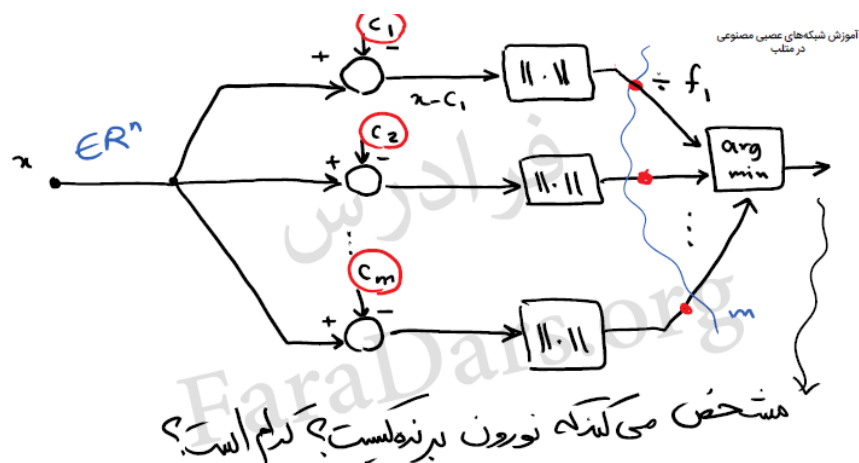
برخلاف شبکه‌های عصبی سنتی، در SOM فرآیند آموزش **بدون نظارت** است و هدف آن کشف ساختارهای پنهان در داده‌ها و نمایش آن‌ها به صورت نگاشتی دوبعدی و قابل تفسیر است.

---

### نکات تکمیلی درباره SOM:

- **کاربردها:** کاهش ابعاد داده‌ها، خوشه‌بندی، تصویرسازی داده‌های پیچیده، تحلیل اکتشافی داده‌ها.
- **ساختار:** معمولاً از یک لایه نورون دوبعدی (ماتریس مربعی یا شش ضلعی) استفاده می‌شود.

- ویژگی مهم SOM: به حفظ توپولوژی داده‌ها معروف است؛ یعنی داده‌هایی که به هم نزدیک‌اند، در نگاشت نهایی نیز به هم نزدیک باقی می‌مانند.



14

در تصویر فوق این تمام کاری است که ما انجام می‌دهیم برای این متد .

### نواحی برودمن (Brodman Areas) و ارتباط آن‌ها با شبکه عصبی SOM

نواحی برودمن مناطقی از قشر مغز انسان (و برخی پستانداران دیگر) هستند که توسط دانشمند آلمانی، گربینیان برودمن (Korbinian Brodmann) در اوایل قرن بیستم تعریف شدند. او بر اساس ساختار سلولی (cytoarchitecture)، یعنی نوع و چیدمان سلول‌های عصبی در بخش‌های مختلف مغز، موفق شد ۵۲ ناحیه متفاوت از قشر مغز را شناسایی و شماره‌گذاری کند (از 1 تا 52). هر یک از این نواحی با عملکردهای خاصی مانند حس، حرکت، زبان، بینایی، حافظه و شناخت مرتبط هستند.

برای مثال:

- ناحیه 17 برودمن: مربوط به بینایی اولیه در لوب پس‌سری (Occipital Lobe)
- نواحی 1، 2 و 3: مربوط به حس‌های بدنی در لوب آهیانه‌ای (Parietal Lobe)
- ناحیه 4: ناحیه حرکتی اصلی در لوب پیشانی (Frontal Lobe)

## ارتباط با شبکه عصبی SOM

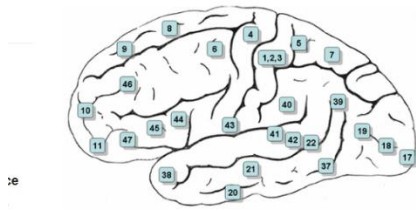
شبکه عصبی (SOM (Self-Organizing Map، یک مدل محاسباتی الهام گرفته از نحوه پردازش اطلاعات در مغز است. این شبکه نیز مانند قشر مغز، اطلاعات را به صورت نقشه‌ای سازمان یافته و توپولوژیکی نمایش می‌دهد؛ به این معنی که داده‌هایی که به هم نزدیک‌ترند، در شبکه نیز در کنار هم قرار می‌گیرند.

شبکه SOM به نوعی شبیه‌سازی ساده‌ای از عملکرد مغز در نواحی‌ای مثل نواحی ورودی است، جایی که سلول‌های عصبی بر اساس وظایف خاص و موقعیت مکانی خود، ساختار منظمی را شکل می‌دهند. همان‌طور که هر ناحیه ورودی عملکرد خاصی دارد، در SOM نیز هر نورون یا خوشه می‌تواند نماینده‌ی یک نوع خاص از داده یا ویژگی باشد.

### خلاصه تشابه:

- نواحی ورودی: سازماندهی زیستی مغز بر اساس ساختار سلولی و عملکرد.
  - SOM: سازماندهی مصنوعی داده‌ها بر اساس شباهت و نزدیکی ویژگی‌ها.
- بنابراین، مطالعه نواحی ورودی به ما درک بهتر ساختارهای مغزی کمک می‌کند و الهام‌بخش طراحی مدل‌هایی مانند SOM در حوزه‌ی یادگیری ماشین و شبکه‌های عصبی شده است.
- شکلی از ناحیه ورودی برای این کار میاورم :

Note: the lateral view, or side view, of the brain is denoted the lateral surface

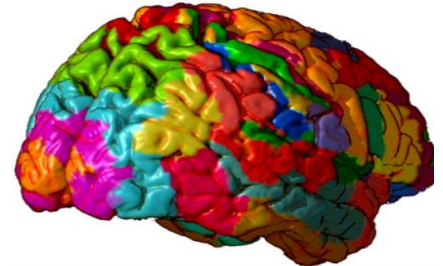
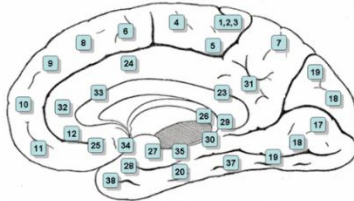


ie

7

Clickable map: medial surface [edit]

Note: the view of the section between the right and left hemispheres of the brain is denoted the 'medial surface'

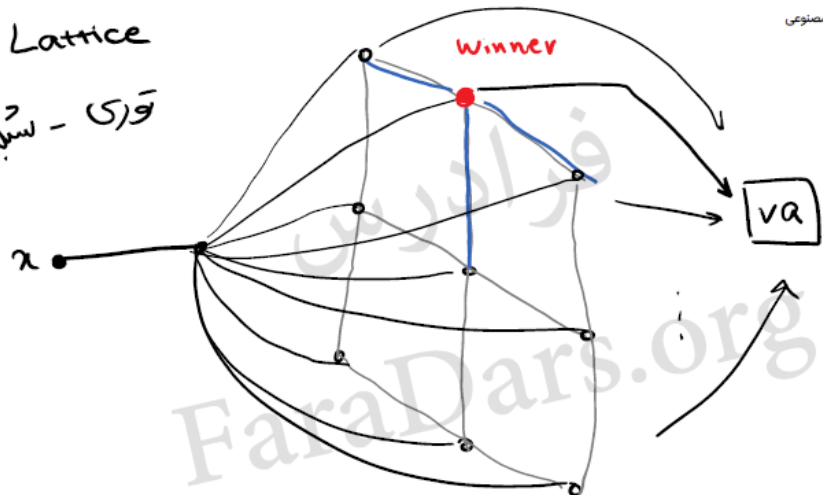


این هم شماتیکی از این روش بردمن است .

در ادامه میخواهیم یک ورودی به چندین نرون بدهیم سپس یگ نرون بیشتر از همگی آنها تحریک میشود بعد از این که تحریک شد، قطعا مراکز کناری که اتصال دارند با این مرکز نرون یا همان خوشه ما است قطعا تحریک می شوند. پس در نهایت یکی بیشتر از همه تاثیر پذیر بیشتری را دارد بر روی ورودی سپس به یک المان برداری داده و یک سری عملیات روی آن انجام داده می شود و بعد از آن خروجی یک عدد ثابت است .

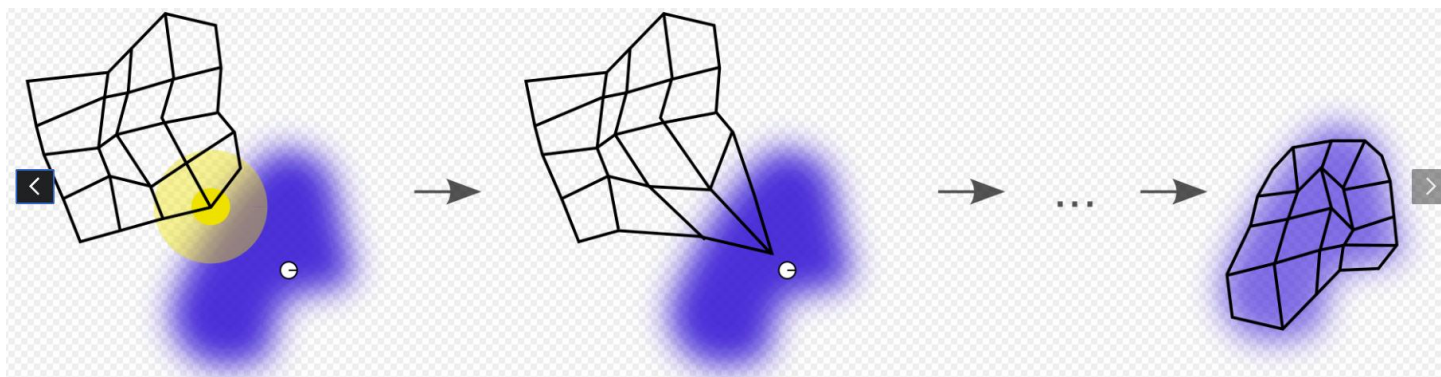
Lattice این همان چیزی است که در بالا

قوری - سبک برای شما توضیح دادم .



ایمپلیمنتی مصنوعی در مطلب

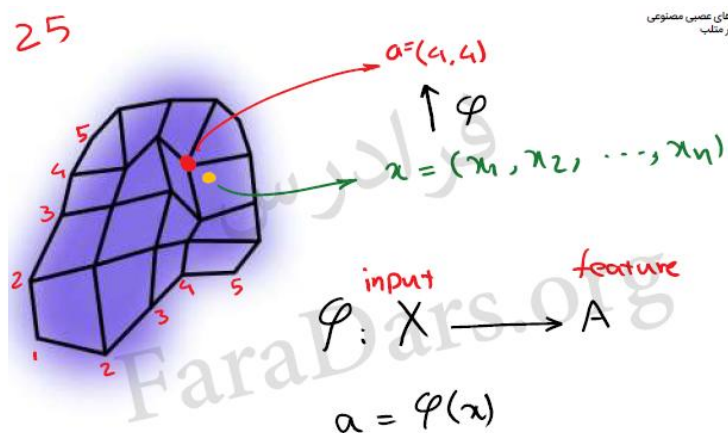
سپس آن ناحیه که در اثر این ورودی تحت شعاع قرار میگیرد انقدر تکرار میشود این روند تکراری که برای سایر نقاط هم باعث میشوند تحت تاثیر روند تغییرات این نرون قرار بگیرند و یک روندی را طی کرده تا شبیه وردی شوند حال با یک تصویر بیشتر این فهم را منتقل میکنم.



این تصویر همان چیزی هست که برای شما توضیح دادم .

خب حال میخواهیم یک  $x$  در دامنه جدیدی که ایجاد کرده ایم را به یک نقطه که نزدیک است بررسی کنیم . حال ممکن است این ورودی دارای چندین تا  $x$  باشد اما میخواهیم کاری کنیم که این ورودی از یک فضای  $n$  بعدی به فضای دو بعدی نگاشتی کنیم که توسط یک تابع انجام میپذیرد برای فهم بیشتر تصویر زیرا را مشاهده کنید :

این تصویری است از چیزی که توضیح دادم .



شبکه های عصبی مصنوعی  
در کتاب

در این تصویر مینید اثری که نقطه 4و4 x های ورودی ما دارد خیلی بیشتر از همه نقاط است . حال این شبکه عصبی دارای 25 تا نرون یا گره است حال برای این فرض کنید میخواهیم 25 تا

مشکل یا مریضی مثلا حل شود ، حال در هر دامنه یک سری افراد قرار میگیرند که متعلق به آن گره هستند و ادامه کار .

خب در ادامه کار میخوایم قوانین ریاضی برای این مدل بیان کنیم که این متد چگونه میتوان در کامپیوتر به صورت الگوریتمیک پیاده سازی کنیم :

سید مهدی بهمنی منصوری  
در مطلب

سه فاز مهم در طراحی SAM :

۱) فاز رقابتی Competitive Phase

۲) فاز همکاری Cooperative Phase

۳) فاز تطبیق Adaptation

این کار ریاضیاتی است که باید انجام دهیم .

حال در ابتدای کار ما یک بردار از ورودی ها داریم یعنی چی یعنی  $n$  تا  $x$  داریم که میخوایم برای این ها بهترین عملکرد را در شبکه عصبی خودمان پیاده سازی کنیم.

خب در ادامه ما یک شبکه دو بعدی عصبی داریم که داری یک سری نقاط یا گره یا همان نرون است حال بر حسب این که چه مقدار این نقاط به بردار ما نزدیک باشند و تفاوت کمی را دارا باشند باید از یک مفهوم ریاضی استفاده کرد، که در اینجا هر چقدر ضرب داخلی یک بردار در نقاط مش یا گره ها بیشتر باشد آن نقطه برنده به حساب میاید یعنی همان فاز رقابتی است .

در کل مفهوم ریاضیاتی این یک ماکسیمم شدن مقدار ضرب داخلی در هر یک از گره ها یعنی گره  $i$  ما برنده میشود حال برای فهم بیشتر به تصویر زیر دقت کنید :

$$x = (x_1, x_2, \dots, x_n)$$

تورون  $x$  بردار است اگر مقوله  $\langle w_j, x \rangle$  ضرب داخلی Inner product

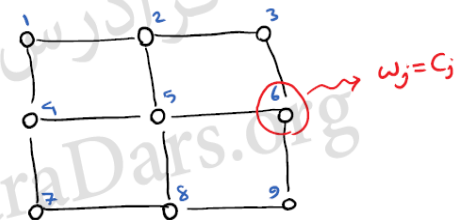
ضرب داخلی Inner product

بسیترین مقدر را در میان همه تورون ها راسته با ستر

فاز رقابت

$$i(x) = \arg \max_j w_j^T x$$

انویس تورون برنده  $x$



این تصویر گویای همه چیز در رابطه با آن چه گفته شد است. حال همین رابطه میتوان بگویم کمترین فاصله ما از مقدار شبیه به  $x$  را به عنوان  $i(x)$  در نظر بگیریم که به صورت زیر است:

$$i(x) = \arg \max_j w_j^T x$$

}

$$i(x) = \arg \min_j \|x - w_j\|$$

## فاز همکاری (Cooperation Phase) در شبکه عصبی SOM

در ادامه فرآیند یادگیری در شبکه SOM، پس از تعیین نورون برنده (BMU)، وارد فاز همکاری می‌شویم. در این مرحله، برای مدل‌سازی تأثیر نورون برنده بر سایر نورون‌های شبکه، یک تابع همسایگی (Neighbor Function) تعریف می‌شود که معمولاً با نماد  $h_{ij}(t)$  نمایش داده می‌شود.

در اینجا:

•  $i$  نشان‌دهنده نورون برنده است. (Best Matching Unit)

•  $j$  نشان‌دهنده نورون همسایه یا هر نورونی است که تحت تأثیر نورون برنده قرار می‌گیرد.

•  $t$  نشان‌دهنده زمان (یا گام آموزشی) است.

این تابع تعیین می‌کند که هر نورون تا چه اندازه باید از ورودی تأثیر بپذیرد. واضح است که:

• نورون برنده بیشترین تأثیر را می‌پذیرد (چون نزدیک‌ترین نورون به ورودی بوده است).

• نورون‌هایی که به نورون برنده نزدیک‌تر هستند (از نظر موقعیت مکانی در شبکه)، تأثیر

بیشتری نسبت به نورون‌های دورتر دریافت می‌کنند.

## نکات مهم:

- در ابتدای آموزش، شعاع همسایگی بزرگ است تا بخش زیادی از شبکه تحت تأثیر قرار گیرد.
- به مرور زمان، این شعاع کاهش می‌یابد تا در نهایت فقط نورون برنده یا همسایه‌های خیلی نزدیکش یادگیری انجام دهند.
- این فاز باعث می‌شود که ساختار توپولوژیک داده‌ها روی شبکه حفظ شود؛ یعنی داده‌های مشابه در مکان‌های نزدیک به هم در نقشه قرار بگیرند.

خب در ادامه یک اسلاید برای شما می‌آورم که توضیحات ساده ایم را برای این کار داده است :

آموزش شبکه‌های عصبی مصنوعی  
در مطلب

ورودی

فاز همکاری

فاصله نورون i در j

نورون دلخواه

انزس برزئو

$$h_{ij}(x) = f(d_{ij})$$

$$f(d) \geq 0$$


---

توسط مهم

$$f(0) \geq f(d) \quad d \neq 0$$

$$\lim_{d \rightarrow \infty} h_{ij}(x) = 0$$

21

خب دو شرط اساسی دارد این هم این است که میبایست یکی اگر فاصله ما صفر شد یعنی خود آن نقطه و در بینهایت هم اثر زیادی ندارد نقطه ای ما در آنجا اثر گذاری داریم .

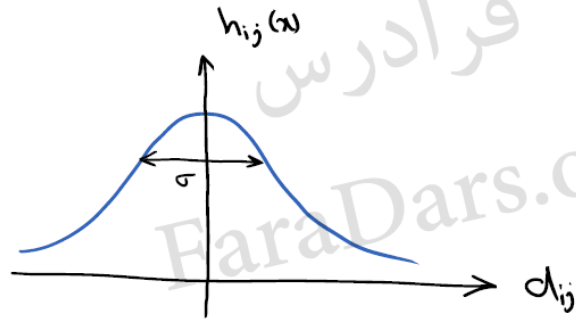
خب به عتلهوان نمونه تابع گوسی را داریم که پارامتر  $d$  به عنوان فاصله میزان دامنه ما است و سیگما میزان باز یا بسته ب.دن دهنه تابع گوسی حال ابتدای امر این تابع دامنه زیادی دارد و سیگما زیاد حال رفتهرفته هر چقدر دقت کار بالا رود این سیگما کوچک و دهنه تابع نیز بسته می‌شود. به اسلاید زیر توجه کنید :



این چیزی است که برای شما توضیح دادم

$$h_{ij}(n) = \exp\left(-\frac{1}{2} \frac{d_{ij}^2}{\sigma^2}\right)$$

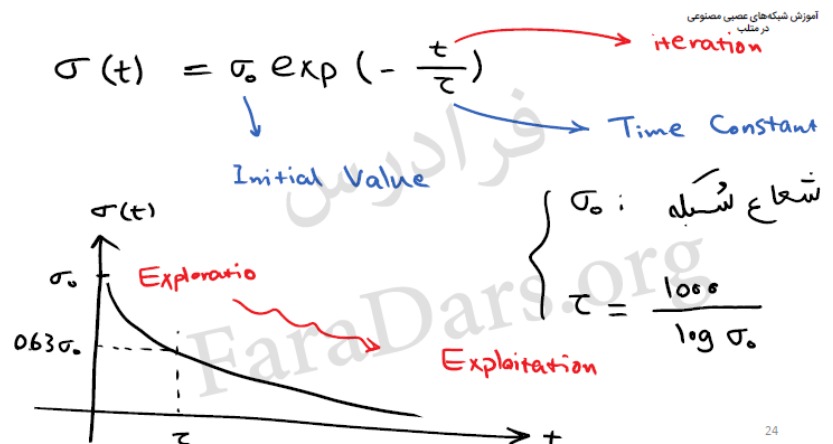
تابع گوسی



22

خب در حالت کلی برای کار ما نتیجه ای که میدهد این میشود کچه یک فرمولی برای کار کردن داریم که می شود به صورت زیر :

این فرمولی برای سیگما که گفتیم در مرور زمان سیر نزولی داریم یعنی کوچک میشود و این همان چیزی است که ما مدنظر داریم یعنی برسیم به هدف خود یا به هدف خود نزدیک تر شویم.



24

حال در نهایت فرمول  $h$  به شرح زیر است :

این همان فرمول در فاز همکاری است / Cooperation

$$h_{ij}(n, t) = \exp\left(-\frac{1}{2} \frac{d_{ij}^2}{\sigma(t)^2}\right)$$

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau}\right)$$

مرحله تطبیق (Adaptation Phase) در شبکه‌های عصبی SOM

بعد از اینکه نورون برنده (و همسایه‌هایش) در مرحله همکاری شناسایی شدند، نوبت به **تطبیق دادن وزن‌های آن‌ها با داده ورودی** می‌رسد. این مرحله را **مرحله تطبیق یا به‌روزرسانی وزن‌ها** می‌نامیم.

در این مرحله، وزن‌های نورون‌ها به سمت داده ورودی حرکت می‌کنند. این یعنی نورون‌هایی که برنده شدند یا به نورون برنده نزدیک‌اند، خودشان را با ویژگی‌های داده ورودی **هماهنگ می‌کنند**.  
**به زبان خیلی ساده:**

- فرض کن نورون‌ها مثل آدم‌هایی هستن که می‌خوان حدس بزنن ورودی (مثلاً یک تصویر یا داده عددی) چیه.
- نورونی که بهترین حدس رو زده (نورون برنده)، بیشترین یادگیری رو انجام می‌ده.
- نورون‌های کناری هم با دیدن حدس خوب نورون برنده، خودشون رو کمی با اون داده تنظیم می‌کنن.
- این باعث می‌شه که نورون‌ها کم‌کم تخصصی بشن: هر نورون نماینده‌ی یک نوع داده بشه.

### نکته مهم:

در طول زمان، هم نرخ یادگیری  $\alpha(t)$  و هم شعاع همسایگی کاهش پیدا می‌کنن. این باعث می‌شه که شبکه در ابتدا با انعطاف زیاد یاد بگیره (کلی یادگیری)، و در ادامه دقیق‌تر و جزئی‌تر یاد بگیره (یادگیری محلی و تخصصی‌تر).  
 خب در ادامه یک تصویر برای فرمول نویسی در فاز تطبیق می‌آورم که از معادله دیفرنس استفاده می‌شود.

این تمام کاری در فاز تطبیق به طور ساده انجام می‌گردد.

Adaptation

قانون یادگیری

Hebb

نوع ساده

تطبیق

$$\Delta w = w(t+1) - w(t) = \eta yx$$

$$w(t+1) = w(t) + \eta yx$$

↓  
Label

خب در ادامه قانون Kohanen Learning rule است به طور زیر می‌آوریم :

$$\Delta \omega_j = \eta y_j x - \underline{g(y_j) \omega_j}$$

$$g(y_j) = \eta y_j \rightarrow \Delta \omega_j = \eta y_j (x - \omega_j)$$

$$y_j = h_{ij}(x) \rightarrow \Delta \omega_j = \eta h_{ij}(x) (x - \omega_j)$$

Kohonen Learning Rule

$$\omega_j(t+1) = \omega_j(t) + \eta h_{ij}(x, t) (x - \omega_j(t))$$

خب بخواهیم که روند کلی برای الگوریتم خودمان بنویسم و نمایش بگذاریم به صورت زیر است :

$$i = \underset{j}{\operatorname{argmin}} \|x - \omega_j(t)\|$$

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau}\right)$$

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

$$\omega_j(t+1) = \omega_j(t) + \eta h_{ij}(x, t) (x - \omega_j(t))$$

نرخ یادگیری

آموزش شبکه‌های عصبی مصنوعی  
در مطلب

SQM

خب هدف ما این است که به دنبال کاری باشیم که بتوانیم نرخ یادگیری را سیر نزولی کنیم یعنی روی یک نقطه هدف فکوس کرده و دامنه اطلاعاتی خودمان را محدود اما به سرعت بیشتری بتوان روی آنها کار کرد.

که برای نرخ یادگیری ما یک معادله دیفرانسیل در نظر میگیریم که بعد از حل آن بر حسب شرایط اولیه و سایر پارمترها که همان ثابت زمانی و خود زمان هم است این تابه را به صورت زیر تعریف میکنیم :

این همان چیزی است که برای شما توضیح دادم .

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau'}\right)$$

بی مصنوعی **سمازده** **زمان**

↓ **ثابت زمانی**      ↓ **مقدار اولیه**

$$\eta(\tau) \approx 0.63 \eta_0$$

$$\begin{cases} \eta_0 = 0.1 \\ \tau' = 10000 \end{cases}$$

خب در ادامه برای کار کردن و کد نویسی در محیط متلب کار میکنیم که توباکس Competlayer به عنوان کار برای این نوع شبکه عصبی استفاده میشود .

inputs = iris\_dataset; این دیتا ها را به عنوان وردی در نظر گرفتیم که برای اطلاعات بیشتر میتوانید در خود متلب یا در google سرچ کنید.

خب کدی که برای این قسمت نوشته ام به اسم

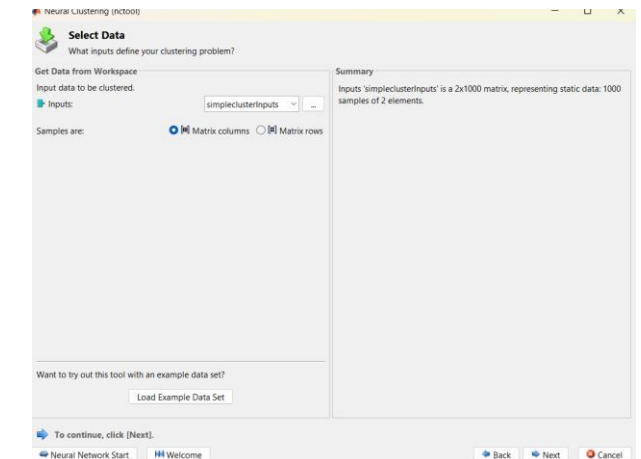
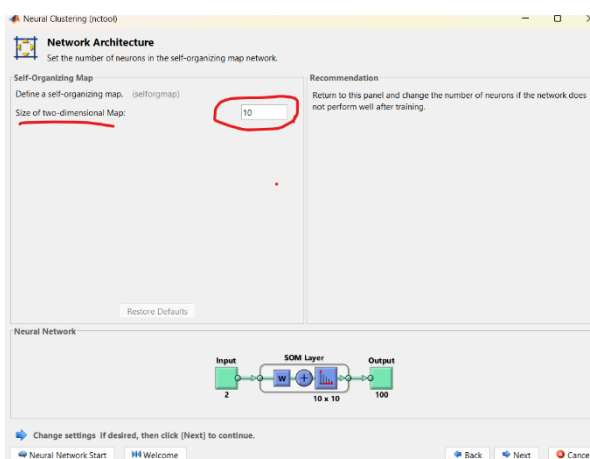
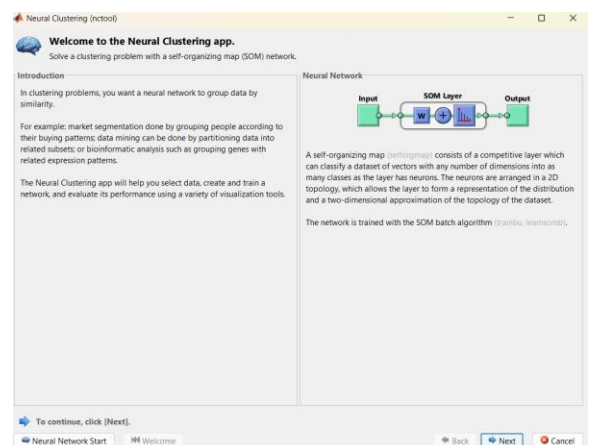
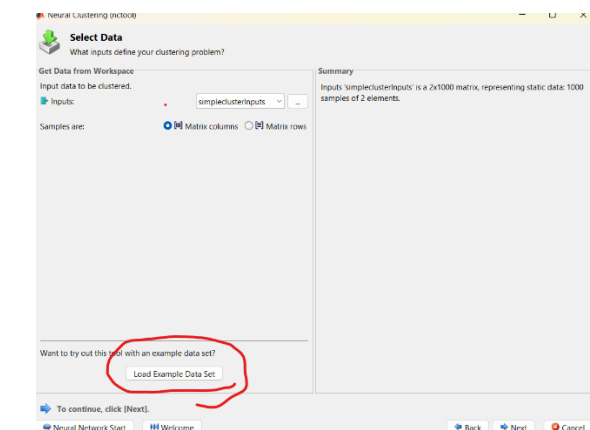
DBZ\_NeuralNetwork\_SOM.m است که جزئیات کامل در آنجا کاملاً میتوانید ببینید.

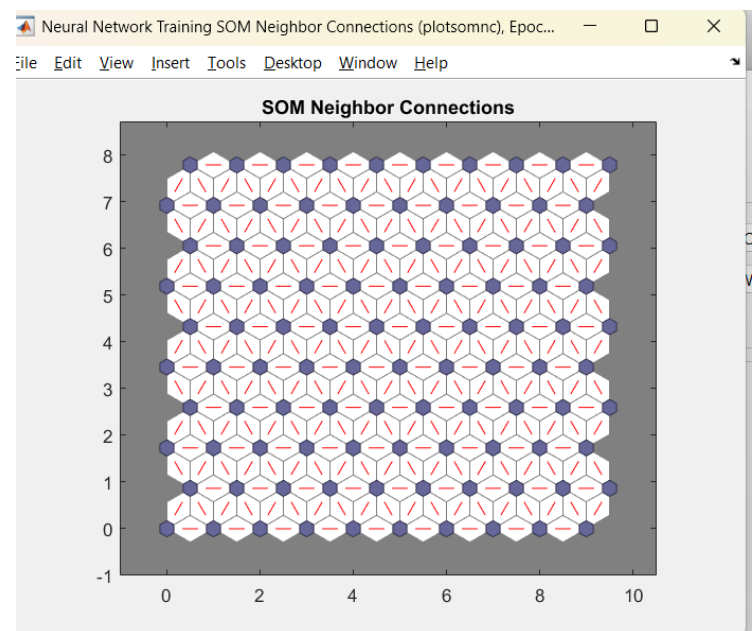
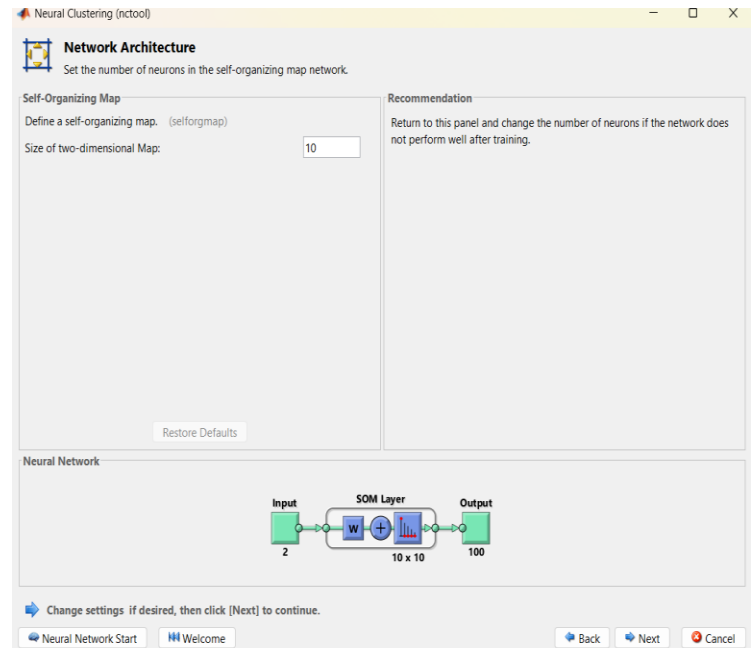
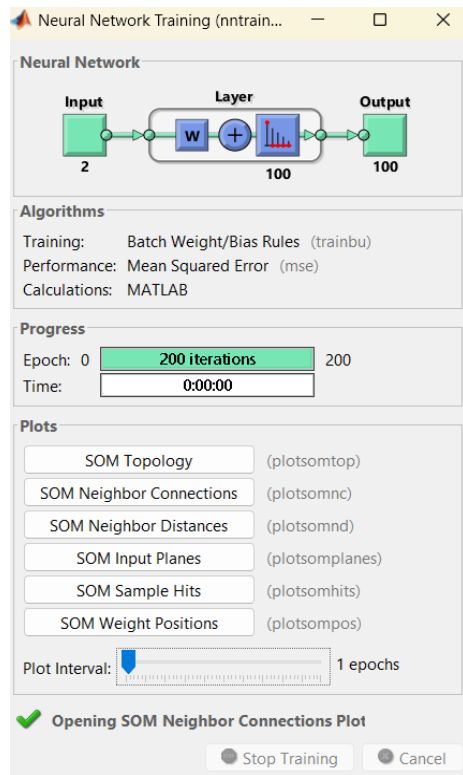
در ادامه روند کاری ما می‌خواهیم SOM را در محیط متلب با nctool که یک اپلیکیشن ساده و آسان برای کار با شبکه عصبی SOM است در محیط متلب کار کنیم .

روند کار ابتدای کار دستور nctool را در ترمینال متلب زده تا محیط آن بالا بیاید سپس در آنجا به شکل زیر عمل میکنیم ما از دیتا های آماده خودت متلب بر ای کار استفاده کرده ایم .

```
Command Window
>> nctool
fx>> |
```

بعد از این محیط کاری ما بالا میاید :



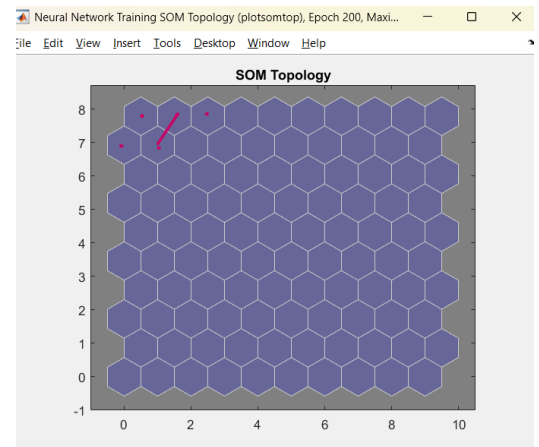


خب مراحل کار کردن با این اپلیکیشن آورده شده است شما میتوانید با این روند کاری کار کنید.

خب در ادامه میخواهیم با نمودار های این اپلیکیشن آشنا شویم .

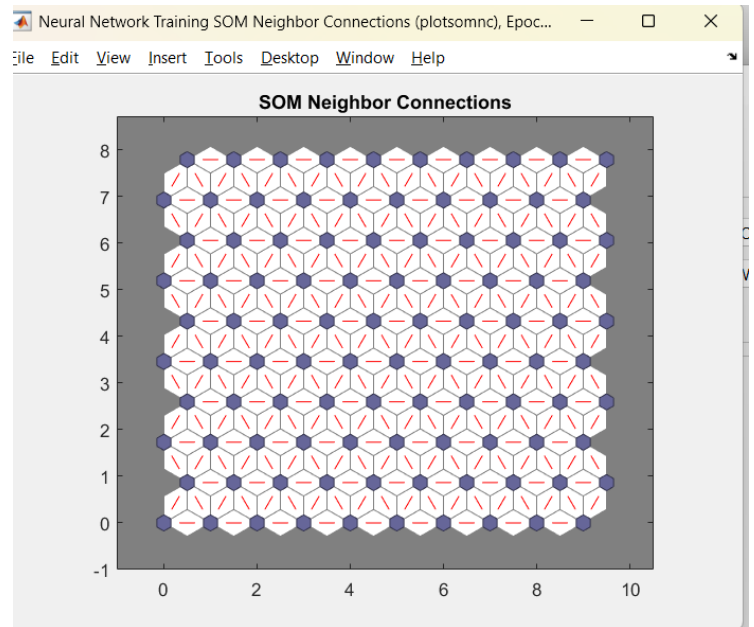
خب ابتدای کار با SOM Topology کار داریم که برای این منظور شکل زیر را برایتان میاورم و صحبت میکنیم :

خب همان طور که میبینید ساختار کاری ما 10 در 10 بوده سپس دلیل این شکل شدن لانه زنبوری یا ساختاری شش ضلعی است که ارتباط بین همسایه ها را میخوایم نمایش دهیم به نحوی که ثابت باشد . خب میبینید که مراکز این شش ضلعی ها همان نقاط گره ها یا همان مراکز تاثیر پذیر هست .

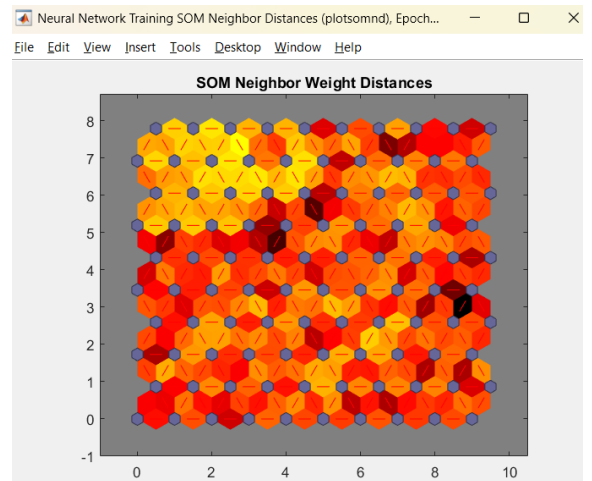


خب نمودار بعد به شکل زیر است :

این نمودار هما طور که میبینید دارد فواصل میان مراکز و همسایه ها را نمایش میدهد .

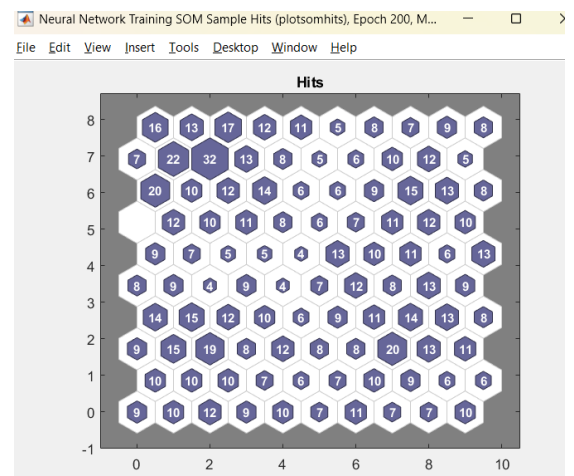


نمودار بعد فواصل را به دقت بالاتری نمایش میدهد که هرچه رنگ آن تیره تر باشد فاصله آن دوتا مرکز شبکه عصبی کم تر است . این شکل در زیر آورده شده است .

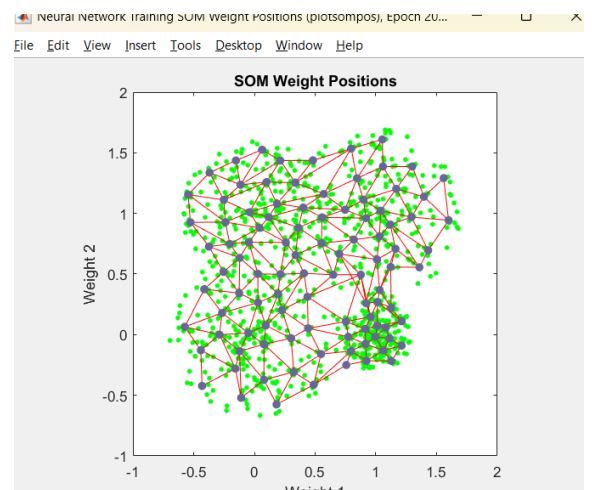


در ادامه نمودار بعدی را می‌خواهیم توضیح دهیم که نشان دهنده این است که از ورودی که میاد کدام شبکه عصبی یا مرکز اثر بیشتری را می‌گیرد که در شکل زیر می‌آورم :

بله دقیقا مشاهده میکنید که کدام شبکه عصبی بیشترین مقدار یا اثر را گرفته است نسبت به ورودی که 32 تا است و کم ترین هم 4 تا است .



در نمودار آخر هم وضعیت یا موقعیت نرون ها همراه با وزن ها را در دیتا نشان میدهد:





خب در ادامه چندتا از توابع مهم برای کار های شبکه عصبی مصنوعی SOM برای شما میاورم که دانش آن و یادگیری آن خالی از لطف نیست که به صورت زیر است :

یکی از آن توابع topology functions است که به سه دسته

(randtop,gridtop,hextop,tritop) دسته بندی میشوند که هر کدام برای شبکه عصبی یک

کاری انجام میدهد برای اطلاعات بیشتر و نحوه کار کردن به help Matlab مراجعه کنید.

یکی دیگر از توابع مهم برای کار با شبکه عصبی dist functions است که یکی دیگر از توابع مهم

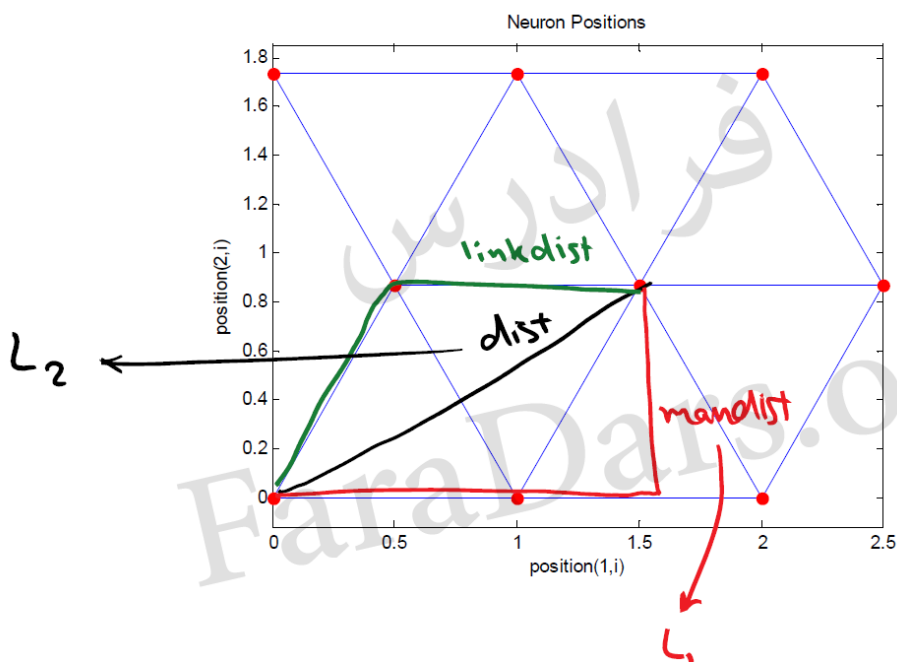
از linkdist است که کار آن برای اندازه گیری میان فواصل شبکه عای عصبی کار برد دارد.

Mandist هم یکی دیگر از توابع برای فوال در شبکه عصبی میباشد .

حال در شکل زیر برای شما تفاوت بین این اندازه گیری فواصل را میاورم که ببینید به چه نحوی عمل کرد دارد:

در این شکل به خوبی مشاهده میکنید که هر کدام از فواصل به چه شکلی عمل میکند .

تابع آخری که برای اندازه گیری فواصل کار میکنیم boxdist است که برای این کار استفاده میشود برای



اطلاعات بیشتر میتوان به help Matlab مراجعه کرد .

یکی از دیگر توابع مهم در شبکه عصبی Som میتواند به selforgmap پرداخت که خیلی مهم میباشد چون تمامی تنظیماتن شبکه عصبی SOM را در خود جای داده است که داکيومنت آن به شرح زیر است :

### selforgmap

Self-organizing map

#### Syntax

`selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn)`

#### Description

Self-organizing maps learn to cluster data based on similarity, topology, with a preference (but no guarantee) of assigning the same number of instances to each class.

Self-organizing maps are used both to cluster data and to reduce the dimensionality of data. They are inspired by the sensory and motor mappings in the mammal brain, which also appear to automatically organizing information topologically.

`selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn)` takes these arguments,

dimensions	Row vector of dimension sizes (default = [8 8])
coverSteps	Number of training steps for initial covering of the input space (default = 100)
initNeighbor	Initial neighborhood size (default = 3)
topologyFcn	Layer topology function (default = 'hextop')
distanceFcn	Neuron distance function (default = 'linkdist')

این هم تصویری از نحوه کاری و help آن میباشد که شما میتوانید برای اطلاعات بیشتر به آ» رجوع کنید .

خب برای نمونه کار کردن قسمتی از کد خودم که تحت عنوان

DBZ\_NeuralNetwork\_SOM2.m است که در گیت هاب هم میتوانید مشاهده کنید آورده

ام که نحوه کار کردن با آن به چه صورتی است :

```
%% Create a Self-Organizing Map
Latticsize = [10 10] ;
CoverSteps = 20;
InitNeighbor = 3;
TopologyFcn = 'gridtop';
DistanceFcn = 'linkdist';
net = selforgmap(Latticsize,CoverSteps,InitNeighbor,TopologyFcn,DistanceFcn) ;
```

این تمام کاغری از که انجام داده ایم .

حال میخواهیم یک الگوریتم traning برایاین شبکه عصبی معرفی کنیم بغیر حالت دیفالت خودش که به اسم trainbu شناخته میشود.

برای اطلاعات بیشتر میتوانید به `learning functions` های متلب مراجعه کنید در `help` Matlab که اطلاعات خیلی خوبی و نحوه استفاده از کدام الگوریتم برای کدام شبکه عصبی منطقی تر و بهتر است .