

نقشه‌های خودسازمانده کوهنن

Kohonen Self Organizing Maps

منبع: <http://www.ai-junkie.com/ann/som/som1.html>

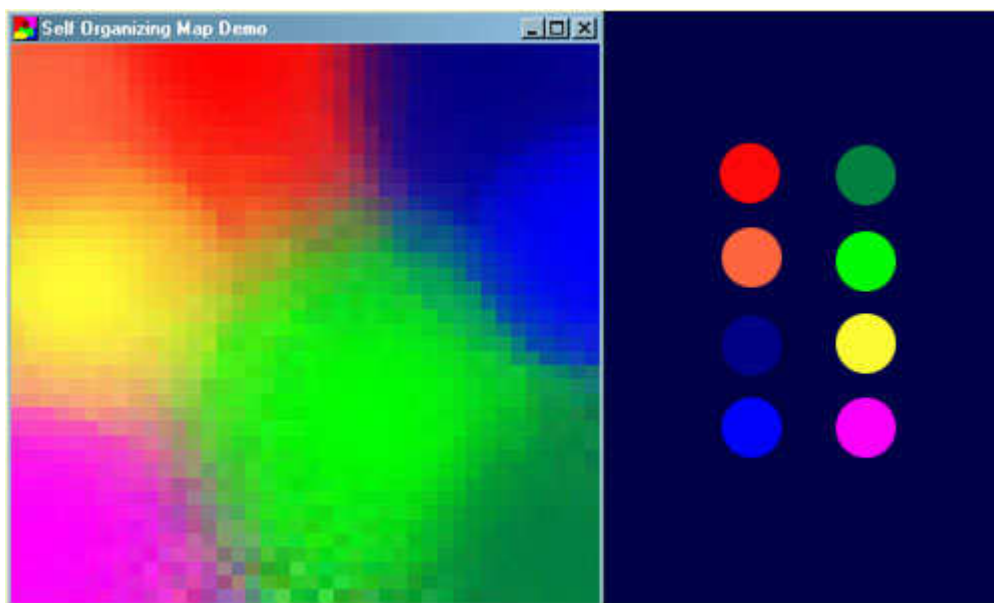
امیرمسعود ایرانی

<http://amib.ir/weblog>

نقشه‌های خودسازمانده نخستین بار توسط پروفسور کوهنن استاد دانشگاه فنلاند ابداع شد. این نقشه‌ها نمایش داده‌های چند بعدی در یک یا چند بعد را ممکن می‌کردند. افزون بر آن در این روش ارتباط مکانی میان داده‌های آزمایشی پابرجا می‌ماند.

یکی از متداول‌ترین مثال‌ها برای آموزش این الگوریتم، نگاشت رنگ‌ها از سه مولفه‌ی آن‌ها یعنی قرمز، سبز و آبی است.

تصویر زیر نمونه‌ای از شبکه‌ای دو بعدی است که برای تشخیص ۸ رنگ موجود در سمت راست آموزش داده شده است.

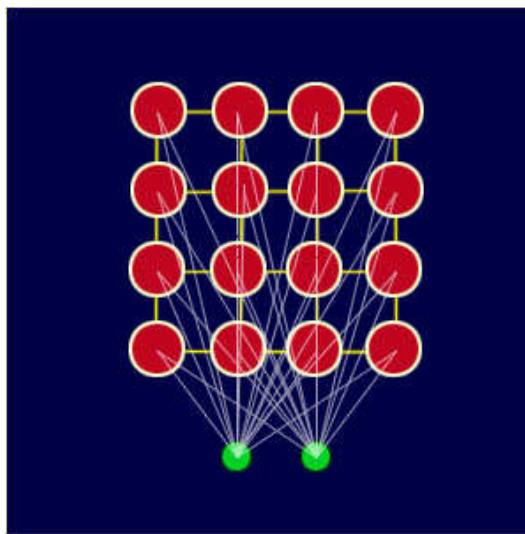


هر رنگ به صورت برداری سه بعدی ارائه شده است. یعنی هر مولفه‌ی رنگ در یک جهت به شبکه ارسال شده است. و شبکه آموخته تا این سه بعد را در فضایی دوبعدی که در تصویر مشخص است به نمایش درآورد. توجه کنید که افزون بر دسته‌بندی، ناحیه‌های مشابه معمولاً در کنار یکدیگر قرار می‌گیرند. این ویژگی شبکه‌های کوهنن کاربردهای مفیدی دارد که در آینده با آن آشنا خواهیم شد.

یکی از جالب‌ترین ویژگی‌های شبکه‌های خودسازمانده عدم نیاز به نظارت در هنگام آموزش است. در شبکه‌های متداول عصبی، داده‌های خروجی با آنچه باید ایجاد شود مقایسه می‌شوند و اگر نتیجه‌ی دلخواه حاصل نشود، وزن گره‌های آنقدر تغییر می‌کنند تا هدف نهایی به دست آید.

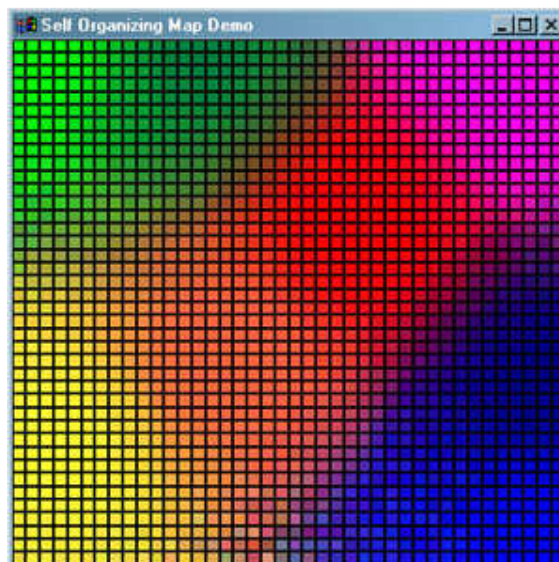
معماری شبکه

برای بررسی معماری شبکه، مثالی دو بعدی را بررسی می‌کنیم. در این نمونه با مجموعه‌ای از گره‌ها در فضایی دوبعدی که هر یک دقیقاً با یک ارتباط به لایه‌ی ورودی متصل است، سر و کار داریم. تصویر زیر نمونه‌ای بسیار کوچک از شبکه‌ی کوهن را به نمایش می‌گذارد. گره‌های سبز رنگ لایه‌ی ورودی و گره‌های قرمز برداری دوبعدی را نشان می‌دهند.



در شبکه‌های خودسازمانده هر گره مکانی مشخص دارد. افزون بر محل فیزیکی، برای هر گره ابعادی مساوی با لایه‌ی ورودی نیز تعریف می‌شود. برای مثال بالا، با توجه به دوبعدی بودن لایه‌ی ورودی، برای هر گره نیز دو بعد مشخص می‌شود.

در نمونه‌ی زیر شبکه‌ای 40×40 را مشاهده می‌کنید. هر گره در این شبکه از سه بعد رنگی تشکیل شده است.



آشنایی با الگوریتم

با توجه به آنچه گفته شد، شبکه‌های خودسازمانده برای آموزش نیازی به خروجی از پیش تعیین شده ندارند. در عوض هر جا که وزن گره‌ها با وزن لایه‌ی ورودی همخوانی داشته باشد، انتخاب می‌شود. سپس در مراحل تکرارشونده وزن گره و گره‌های اطراف آن به آرامی تغییر می‌کند تا هرچه بیشتر شباهت میان گره و لایه‌ی ورودی حاصل شود. پس از تعداد کافی تکرار این مرحله، شبکه به ناحیه‌هایی پایدار مشابه لایه‌ی ورودی تبدیل می‌شود.

آموزش در مراحل خاص و در دفعاتی زیاد انجام می‌شود:

۱. وزن هر گره مشخص می‌شود.
۲. گره‌ای تصادفی از داده‌های ورودی انتخاب می‌شود و به شبکه ارائه می‌گردد
۳. تمامی گره‌ها برای شباهت با گره‌ی ورودی بررسی می‌شوند. گره‌ای با بیشترین شباهت به عنوان BMU یا Best Matching Unit برگزیده می‌شود.
۴. همسایگی BMU محاسبه می‌شود. معمولاً این همسایگی با شعاعی برابر اندازه‌ی شبکه شروع و در هر تکرار آنقدر کاهش می‌یابد تا در پایان این همسایگی به صفر برسد.
۵. وزن نقاط قرار گرفته در همسایگی وابسته به فاصله‌ی مکانی با BMU تغییر می‌کند. هرچه فاصله‌ی نقاط با BMU کمتر باشد، مقدار تغییر وزن برای افزایش شباهت به وزن BMU بیشتر خواهد بود.
۶. مرحله‌ی ۲ به دفعات N بار تکرار می‌شود.

جزئیات الگوریتم یادگیری

هم‌اکنون مراحل گفته شده در بخش قبل را با دقت بیشتری مورد بررسی قرار می‌دهیم.

۱. وزن هر گره مشخص می‌شود

در این مرحله وزن هر گره با مقداری تصادفی میان ۰ و ۱ مقداردهی می‌شود. مقداردهی برای هر یک از بعدهای وزن گره جداگانه به انجام می‌رسد.

۲. انتخاب گره‌ای تصادفی از لایه‌ی ورودی

با توجه به تکرار مراحل در دفعات مشخص، تمامی گره‌ها انتخاب خواهند شد.

۳. یافتن BMU

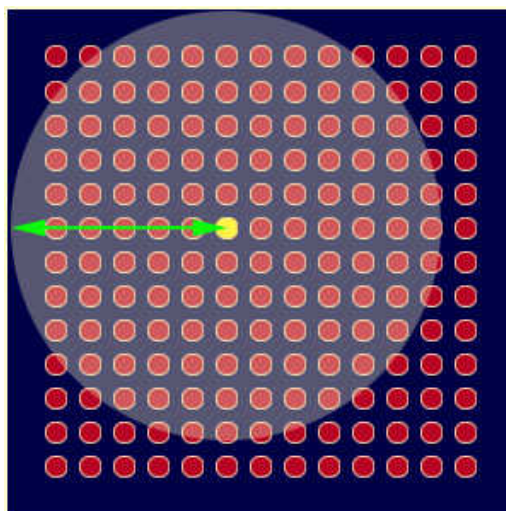
برای یافتن شبیه‌ترین نقطه به گرهی ورودی، وزن هر گرهی موجود در شبکه با وزن گرهی ورودی مقایسه می‌شود. برای مقایسه‌ی وزن گره‌ها در چند بعد اختلاف هر بعد از وزن‌ها را محاسبه می‌کنیم و پس از به توان دوم رساندن اختلاف، آن‌ها را با یکدیگر جمع و در پایان جذر آن را محاسبه می‌کنیم.

$$\sqrt{\sum_{i=1}^n (V_i - W_i)^2}$$

V گرهی ورودی و W گرهی فعلی در شبکه است.

۴. محاسبه‌ی همسایگی

در هر مرحله، پس از یافتن BMU زمان آن فرا می‌رسد که گره‌های همسایگی BMU پیدا شوند. وزن تمامی این نقاط در مرحله‌ی بعدی تغییر خواهد کرد. این مرحله به سادگی انجام می‌شود. نخست لازم است شعاع همسایگی محاسبه شود. تصویر زیر یک شعاع قابل قبول همسایگی را در شروع نمایش می‌دهد.

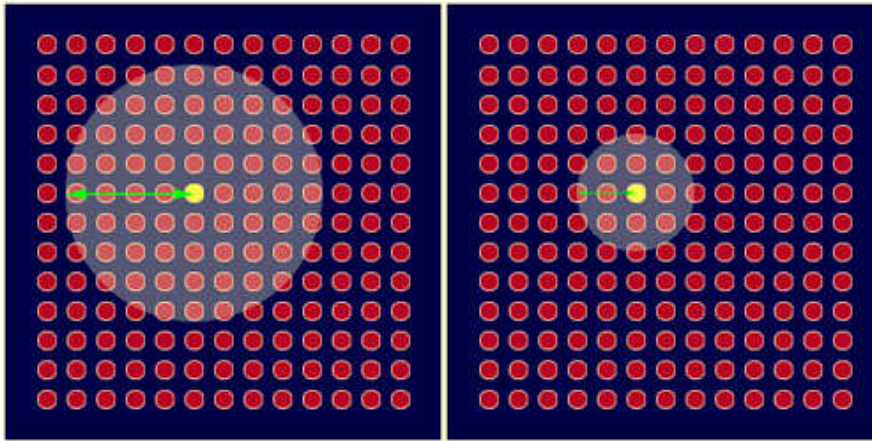


همان‌گونه که گفته شد، در الگوریتم کوهنن، شعاع همسایگی در هر مرحله از پردازش کاهش می‌یابد. این کاهش می‌تواند با فرمولی نمایی به انجام برسد.

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

سیگما « σ_0 » نشان‌دهنده‌ی شعاع شبکه در زمان « t_0 » و حرف لامبدا « λ » ثابتی زمانی است. تابع \exp نیز تابع نمایی است که ثابت نپر را به توان عدد مقابلش « e^x » می‌رساند.

تصویر زیر نمایی از کاهش شعاع را در طی زمان به نمایش در می‌آورد. البته توجه کنید که BMU در هر مرحله ممکن است تغییر کند. برای ساده‌تر شدن نمایش، در مثال زیر BMU بدون تغییر فرض شده است.



با سپری شدن زمان، شعاع همسایگی آنقدر کاهش می‌یابد تا تنها BMU در شعاع همسایگی باقی بماند.

۵. تغییر وزن گره‌ها

در هر مرحله وزن تمامی گره‌های قرار گرفته در همسایگی BMU بر اساس معادله‌ی زیر تغییر می‌کند.

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t))$$

در این معادله t نماینده‌ی زمان است، L متغیری کوچک و کاهش‌یابنده در طی زمان و نماینده‌ی عبارت «Learning Rate» است.

معادله‌ی Θ فرمولی است که فاصله‌ی گره‌های قرار گرفته در همسایگی با BMU را به نسبتی از همسایگی تبدیل می‌کند، ضرب تتا در L سبب می‌شود مقدار اثربخشی L با افزایش فاصله، کاهش یابد. این معادل وظیفه دارد وزن جدید گره را به اندازه‌ی نسبتی « L » به وزن گره‌ی ورودی نزدیک‌تر کند. کاهش مقدار L در هر واحد زمان با کمک معادله‌ی زیر محاسبه می‌شود:

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

برای محاسبه‌ی « Θ » از فرمول زیر استفاده می‌کنیم.

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

در این معادله « $dist$ » فاصله‌ی گره از BMU، و « σ » شعاع همسایگی است که پیشتر آن را محاسبه نمودیم. دقت کنید که مقدار تتا نیز با گذشت زمان کاهش می‌یابد.

همان گونه که مشاهده می‌کنید، این تابع همانند تابع کاهش شعاع همسایگی که در مرحله‌ی ۴ شرح داده شد، عمل می‌کند.

مقدار « L_0 » در این فرمول عددی کوچک و اختیاری است. برای مثال این عدد می‌تواند « $0,1$ » تعیین شود.