

حافظه انجمنی:

راهبردهای پیشرفته یادگیری

10.1: ذخیره سازی الگوهای همبسته

همان طور که در بخش 3.1 و 4.3 بحث کردیم توانایی بیادآوری صحیح خاطره ها اگر عدد p از الگوهای ذخیره شده از یک حد معین بیشتر شود تجزیه می شود (از بین می رود). هنگامی که پیوندهای سیناپسی بر طبق قانون Hebb تعیین شده هستند، این کار در تراکم (چگالی) ذخیره سازی اتفاق می افتد $\alpha = \frac{p}{N} = 0.138$. دلیل این رفتار، تاثیر دیگر الگوهای ذخیره شده که با ضابطه اختلال نوسان در (3.13) بیان می شود، می باشد. همان طور که در انتهای بخش 3.1 اشاره کردیم، اگر الگوها بر یکدیگر متعامد باشند همانند آنچه که در بخش (3.16) بیان شد، این اثر دقیقا ناپدید خواهد شد. به عبارت دیگر توان تجدید خاطره روبه وخامت می رود حتی قبل از آنکه الگوهای ذخیره شده شدیداً با یکدیگر مرتبط شوند. درباره نمایش گرافیکی حروف رومی فکر کنید، جایی که E شباهت نزدیکی به F دارد و C شبیه G است یا درباره لیست معمولی از اعداد کتابچه تلفن، که احتمالا بسیار به هم مرتبط هستند، فکر کنید.



از برنامه ASSO برای یادگیری و بیادآوری 26 حرف حروف الفبا استفاده کنید: A-Z مقادیر زیر را انتخاب کنید: (26/26/0/1) و (1//0/0/1;2) برای مثال آپدیت متوالی، دما و آستانه صفر و آزمایش با مقدار مجاز

اختلال (صدا). آیا حرف خاصی پایدار است؟ آزمایش را با 6 حرف اول الفبا و مطالعه توانایی شبکه در به یاد آوردن حروف مشابه E و F تکرار کنید.

ماهیت مساله از آنچه که در بخش قبلی در ارتباط با شبکه های لایه ای تغذیه مستقیم مطرح شد، فرق نمی کند. ضابطه یادگیری پرسپترون راهبرد یادگیری کامل برای پرسپترون های ساده بدون لایه های مخفی نورون ها فراهم می کند. با این حال، این طرح ها در عمل قابل استفاده نیستند چون حتی در حل مسایل بسیار ساده هم شکست می خورند. به همین دلیل است که مفاهیم پرسپترون برای بیست سال کنار گذاشته شد، اگرچه پرسپترون های چندلایه با لایه های پنهان نورون از چنین مشکلی رنج نمی برد. اما بدون الگوریتم یادگیری عملی، که در ابتدا با خطای پس انتشار در دسترس است، یک جایگزین عملی فراهم نخواهد شد.

به طور مشابه بعضی از مشکلات عملی برخورد شده با شبکه های حافظه انجمنی ماهیت طبیعی ندارند، بلکه در عوض نتیجه نارسایی شکل اولیه قانون (ضابطه) یادگیری Hebb می باشند. قانون هب (Hebb) بر پایه مفاهیم اندازه گیری فاصله هامینگ (Hamming) بین الگوهای مختلف نهاده شده است. در اصطلاح ریاضی اندازه فاصله یا سنجش، متریک اقلیدسی (Euclidean metric) در فضای الگوها نامیده می شود. عموماً بیشتر اندازه گیری ها از فاصله بین الگوهای مختلف امکان پذیر است و اگر الگوها با یکدیگر ارتباط داشته باشند (همبسته باشند) ممکن است مفید باشد. برای مثال در مورد حرف های E و F فاصله اندازه گیری شده منحصر بر پایه بخش پایین حرف است که به راحتی بین الگوها مورد تبعیض واقع می شود. با این حال همان طور که حرف های K و R نشان می دهد، این انتخاب ساده هنوز یک جواب عمومی فراهم نکرده است و نه حتی برای الفبا.

10.1.1 قانون تجسم

با این وجود روشن است که مشکل تبعیض بین الگوهای مرتبط یک جواب ساده مهم دارد که حتی به ذخیره گاه $p=N$ اجازه می دهد به طور اختیاری الگوهای همبسته را انتخاب کند تا زمانی که به طور خطی مستقل باشند*. برای دیدن اینکه چگونه کار می کند ماتریس ضرب اسکالر بین جفت الگوها ($\sigma_i^H = \pm 1$) را تشکیل می دهیم:

$$(10.1) Q_{\mu\nu} = \frac{1}{N} \sum_i \sigma_i^\mu \sigma_i^\nu \quad (1 \leq \mu, \nu \leq p)$$

برای الگوهای مستقل خطی ماتریس $Q_{\mu\nu}$ وارون است، و می توانیم نقاط قوت جفت سیناپسی بهبودیافته را چنین تعریف کنیم:

$$(10.2) \tilde{\omega}_{ij} = \frac{1}{N} \sum_{\mu, \nu} \sigma_i^\mu (Q^{-1})_{\mu\nu} \sigma_j^\nu$$

از دید ریاضی (10.2) متناظر با تکنیک تصویر که همبستگی های بین الگوهای موجود را حذف می کند می باشد، از این رو این قانون یادگیری اغلب قانون تجسم (تصویر) نامیده می شود. با این انتخاب اثر متقابل الگوهای ذخیره شده ناشی از ترم نوسان کننده در (3.13) دقیقاً ناپدید می شود همان طور که به راحتی با محاسبه پتانسیل پس سیناپسی در حضور الگوهای حفظ شونده σ_i^λ به آسانی بدست می آید:

$$\begin{aligned} \tilde{h}_i &= \sum_j \tilde{\omega}_{ij} \sigma_j^\lambda = \frac{1}{N} \sum_{\mu, \nu} \sigma_i^\mu (Q^{-1})_{\mu\nu} \sum_j \sigma_j^\nu \sigma_j^\lambda \\ &= \sum_{\mu, \nu} \sigma_i^\mu (Q^{-1})_{\mu\nu} Q_{\nu\lambda} = \sum_{\mu} \sigma_i^\mu \delta_{\mu\lambda} = \sigma_i^\lambda. \end{aligned} \quad (10.3)$$

نتیجه می گیریم که هر الگوی ذخیره شده یک پیکربندی پایدار شبکه ای مستقل از رابطه بین الگوها، را نشان می دهد. البته شرط p کوچکتر مساوی N به محدود کردن ظرفیت حافظه ادامه میدهد، چون حداکثر N الگوی مستقل خطی از N واحد اطلاعات تشکیل می شوند.

همان طور که این عبارت بیان می کند کاملاً صحیح نیست چون بعضی از الگوها بدون نمایش تلویحی در جفتهای سیناپسی به طور موثری حفظ می شوند. چنین پدیده هایی برای ما جدید نیست. در بخش 3.3 (3.23) دیده ایم که ترکیب خطی از الگوهای ذخیره شده ممکن است حالت های حافظه پایداری باشد. همان طور که آپر (Oppor) نشان داد این کار برای الگوریتم یادگیری تکرار شونده کراث و مزارد (Krauth and Mezard) رخ می دهد (بخش 10.1.3 را ببینید) که به ذخیره گاه اجازه می دهد تا $2N$ الگوی مختلف با روش بهینه ذخیره کند. فقط N تا از این الگوها بر طبق قانون تجسم ذخیره می شوند و بقیه بدون ذخیره ضمنی حفظ می شوند. با این حال برای $p > N$ فرایند یادگیری

به طور آهسته همگرا می شوند. (تکنیک هایی برای سرعت بخشیدن به همگرایی در [Ab89, An89a] ارائه شده است.) ظرفیت ذخیره بهینه یک شبکه عصبی در فصل 20 مفصل بحث خواهد شد.

بکاربردن (10.2) برای جفت های قوی نیز جواب شبه معکوس نامیده می شود. به طور خاص تبدیل وارونه مجموعه بردارهای الگو که با یک ماتریس با N ستون و p ردیف دیده می شود را انجام می دهد. نقش شبه معکوس بسیار ساده درک خواهد شد اگر با یک نگاه ثابت به (10.3) بدون زیرنویس i داشته باشیم. سپس داریم $\tau^{\mu} = \sum_j \sigma_j^{\mu} \omega_j$ یا در ماتریس علامت گذاری (نوتاسیون): $\tau = \sigma w$. در اینجا τ^{μ} بیان کننده بازده (خروجی) است که وقتی الگوی ورودی σ_j^{μ} به شبکه ارائه می شود بازده برانگیخته می شود. باید معادلات سیستم های خطی p برای بردار w ، N بعدی حل کنیم. چون در کل ماتریس σ مربعی نیست ($p < N$) نمی توانیم مستقیم معکوس آنرا حساب کنیم. با این حال می توانیم ماتریس شبه معکوس را بدست آوریم:

$$\underline{\sigma}^{pi} = \underline{\sigma}^T (\underline{\sigma} \underline{\sigma}^T)^{-1} \quad (10.4)$$

بنابراین $w = \underline{\sigma}^{pi} \tau$ با حل مسئله داریم:

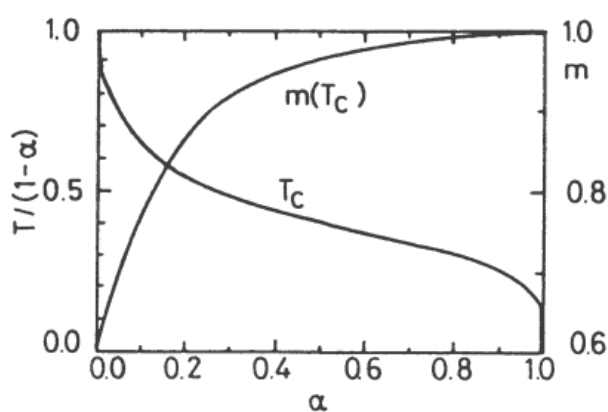
$$\underline{\sigma} w = \underline{\sigma} \underline{\sigma}^{pi} \tau = \underline{\sigma} \underline{\sigma}^T (\underline{\sigma} \underline{\sigma}^T)^{-1} \tau = \tau . \quad (10.5)$$

در حافظه های خودانجمنی τ جهت منطبق شدن با σ اتفاق می افتد اما این کار ضروری نیست، جواب شبه معکوس برای دیگرانجمنی و برای پرسپترون بدون لایه های پنهان نیز صدق می کند.

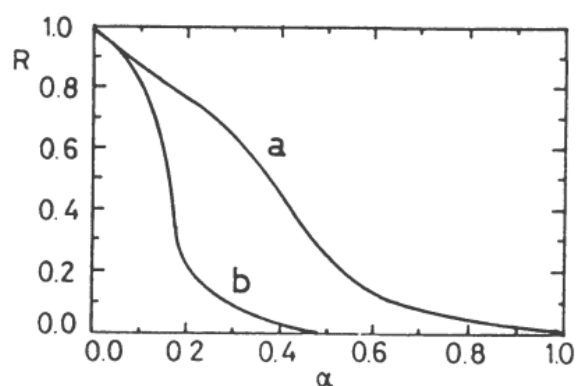
کیفیت به یاد آوری الگو با افزایش دما و استفاده از حافظه $\alpha = p/N$ رو به تحلیل می رود. همانند مورد قانون Hebb کیفیت بازیابی با پارامتر m تعریف شده در (4.15) می باشد. $m=1$ نشان دهنده یادآوری کامل حافظه است با این

¹ توجه داشته باشید که در مدلها جفت های مورب $\bar{\omega}_{ii}$ در صفر تنظیم شده است. اگر شرایط مورب باقی بماند، ظرفیت حافظه بحرانی در $T=0$ ثابت باقی می ماند ($\alpha_c = 1$)، اما در حضور نوسانات کوچک الگوهای ذخیره شده در بالای $\alpha = 0.5$ نمی تواند بیاد آورد (گفته می شود که شعاع جذب الگوهای ذخیره شده صفر است).

حال $m=0$ نشان دهنده فراموشی کامل است. ناحیه کارایی و فراموشی حافظه با مقدار m در مرز فاز در شکل 10.1 نشان داده شده است. شعاع جذب R الگوی ذخیره شده در شکل 10.2 به عنوان تابعی از تراکم ذخیره α برای مدل های کانتر و سامپولینسکی و پرسوناز، نشان داده شده است. در اینجا شعاع جذب با رابطه $R=1-m_0$ تعریف شده است که کمترین همپوشانی الگوی S_i که می تواند با الگوی ذخیره شده σ_i داشته باشد، که به طور قطعی توسط شبکه تشخیص داده می شود. همان طور که می بینید از بین رفتن جفت های مورب ω_{ij} در مدل [Ka87] منحنی (a) اثر موثرتری (بهینه تر) دارد.



شکل 10.1: ناحیه کارایی و سردرگمی کامل حافظه و کیفیت بیادآوری m در مرز فاز نامگذاری شده با T_c



شکل 10.2: متوسط شعاع جذب R به عنوان تابعی از تراکم حافظه در مدل های [Ka87] (منحنی a) و [Pe86b]

(منحنی b)

10.1.2 طرح یادگیری تکرار شونده

کاربردهای عملی قاعده آموزش تصویری برای شبکه های بزرگ حافظه اشباع شده از نیاز مبرم تبدیل $p \times p$ ماتریس $Q_{\mu\nu}$ که مشکل عددی دشواری را دربرخواهد داشت، رنج می برد. خوشبختانه تبدیل معکوس ماتریس فقط یکبار انجام می شود وقتی الگوها در داخل شبکه ذخیره شده اند. حافظه ریشه دوانده می تواند اغلب مطابق انتظار بدون تلاش اضافی یادآوری کند. روش عملی اجرای قاعده (قانون) تصویربر پایه طرح تکرار شونده بنانهاده شده است که پیوندهای سیناپسی صحیح به یکدیگر متصل شده اند تا الگوهای وابسته را در برابر یکدیگر پایدار کنند. به خاطر ساده سازی این روش را فقط برای شبکه قطعی ($T=0$) نشان می دهیم.

از قانون تکامل نورون ها $s_i(t+1) = \text{sgn}[h_i(t)]$ هر الگوی σ_i^μ نمایانگر یک پیکربندی شبکه پایدار است اگر h_i همان علامت σ_i را داشته باشد. یعنی:

$$\sigma_i^\mu h_i = \sum_j w_{ij} \sigma_i^\mu \sigma_j^\mu > 0 \quad (10.6)$$

برای هر نورون i .

اگر عبارت (10.6) فقط کمی مثبت باشد، هر اختلال (انحراف) کوچک، برای مثال $s_i \neq \sigma_i^\mu$ برای تعداد کمی نورون i ، علامت آنرا می تواند تغییر دهد. برای دستیابی به پایداری بیشتر الگوهای حافظه خواسته شده، فرض می کنیم که عبارت (10.6) نه تنها مثبت است بلکه از مقدار آستانه معین نیز بیشتر است ($K > 0$). برای یک الگو منفرد، قاعده Hebb بدست می دهد: $h_i = \sigma_i$. در نتیجه شرط $h_i \sigma_i = 1$ همواره در این مورد صدق می کند. کاملاً طبیعی به نظر می رسد که آستانه پایداری را در $K=1$ بگیریم و همچنین در مورد کلی الگوهای ذخیره شده چندگانه، و فرض کنیم که پیوندهای سیناپسی چنین انتخاب شوند:

$$\sigma_i^\mu h_i = \sum_j w_{ij} \sigma_i^\mu \sigma_j^\mu = 1 \quad (10.7)$$

برای همه نورون های i .

یک روش واضح دستیابی به نتایج دلخواه با انتخاب پیوندهای سیناپسی اولیه بر طبق قاعده Hebb شروع می شود:

$$w_{ij} = \frac{1}{N} \sum_{\mu} \sigma_i^{\mu} \sigma_j^{\mu} . \quad (10.8)$$

در گام بعدی برای تمام الگوها بررسی می کنیم که آیا شزط (10.7) برقرار است یا نه؟ اگر برقرار نبود برطبق دستور زیر سیناپسی ها را تصحیح می کنیم.

$$w_{ij} \rightarrow w'_{ij} = w_{ij} + \delta w_{ij} \quad (10.9)$$

با

$$\delta w_{ij} = \frac{1}{N} (1 - \sigma_i^{\mu} h_i) \sigma_i^{\mu} \sigma_j^{\mu} , \quad (10.10)$$

که μ الگو را تحت ملاحظه نشان می دهد^۲. با این پیوندهای سیناپسی تصحیح شده برای الگوی مشابه μ بدست می آوریم:

$$\begin{aligned} \sigma_i^{\mu} h'_i &= \sigma_i^{\mu} h_i + \sum_j \delta w_{ij} \sigma_i^{\mu} \sigma_j^{\mu} \\ &= \sigma_i^{\mu} h_i + \frac{1}{N} \sum_j (\sigma_i^{\mu})^2 (\sigma_j^{\mu})^2 (1 - \sigma_i^{\mu} h_i) = 1 , \end{aligned} \quad (10.11)$$

چون $(\sigma_i^{\mu})^2 = 1$. همچنین بعد از آپدیت کردن تمام سیناپس ها، شرط آستانه پایداری (10.7) برای الگوی مورد نظر صدق می کند. وقتی به الگوی بعدی ($\mu + 1$) برسیم جفت های سیناپسی دوباره تصحیح می شوند، بنابراین (10.7) برای الگوی بعدی تحت نظر معتبر خواهد شد. با این حال (10.7) ممکن است برای الگوی قبلی μ کار نکند. بعد از یک چرخه کامل بر روی تمام الگوهای ذخیره شده، شرط فقط برای آخرین الگو $\mu = p$ با قطعیت کار می کند، اما برای برای

² در اصل بدون انتخاب سیناپسی های Hebbian اولیه نیز می توانستیم انجام بدهیم. اگر باشبکه کامل قطع شده ($w_{ij} = 0$) شروع کنیم، با ساده لوحانه، اولین کاربرد قانون تصحیح (10.9) پیوندهای سیناپسی با مقدار دقیق نسبت داده شده توسط قانون Hebb نتیجه می داد!

دیگر الگوها ضروری نیست. سوال اساسی این است که آیا این فرایند همگرایی تدریجی یا آیا تداوم بدون محدودیت بدون رسیدن به یک حالت پایدار که شرط آستانه برای تمام الگوها راضی کننده باشد امکان پذیر است؟^۳ برای پاسخ به این سوال، بدنیست بعضی از اختصارها را تعریف کنیم. در اینجا فرض می کنیم که سیناپسی ها به طور متوالی مرتب شده اند که به این معناست بررسی عملکرد هر شبکه برای هر الگوی منفرد، انحراف از آستانه در چرخه ℓ ام بروز شده و مجموع همه انحراف های برخورد شده تا آن نقطه را چنین تعریف می کنیم:

$$\delta x_i^\mu(\ell) = 1 - \sigma_i^\mu h_i, \quad x_i^\mu = \sum_{\ell'=1}^{\ell} \delta x_i^\mu(\ell'). \quad (10.12)$$

سیناپسی ها بر طبق (10.10) برای الگوی ν ام در چرخه ℓ ام تصحیح شده است می توانیم به شکل زیر بنویسیم:

$$\delta w_{ij} \equiv \frac{1}{N} \left(1 - \sum_k w_{ik} \sigma_i^\nu \sigma_k^\nu \right) \sigma_i^\nu \sigma_j^\nu = \frac{1}{N} \delta x_i^\nu(\ell) \sigma_i^\nu \sigma_j^\nu. \quad (10.13)$$

بنابراین بعد از کامل شدن چرخه ℓ ام پیوندهای سیناپسی طبق رابطه زیر بیان می شود:

$$w_{ij} = \frac{1}{N} \sum_{\nu} x_i^\nu(\ell) \sigma_i^\nu \sigma_j^\nu. \quad (10.14)$$

اکنون در نظر بگیرید که در چرخه تکرارشونده $(\ell + 1)$ چه اتفاقی می افتد. اگر به الگوی μ برسیم همه الگوهی قبلی سهم $(\ell + 1)$ برابر در فرایند تصحیح سیناپسی ها دارند، با این حال بقیه آنها شامل μ فقط ℓ سهم دارند. بنابراین تصحیح برای الگوی μ ام توسط رابطه زیر داده می شود:

$$\begin{aligned} \delta x_i^\mu &= x_i^\mu(\ell + 1) - x_i^\mu(\ell) = 1 - \sum_k w_{ik} \sigma_i^\mu \sigma_k^\mu \\ &= 1 - \frac{1}{N} \sum_k \left[\sum_{\nu < \mu} x_i^\nu(\ell + 1) \sigma_i^\nu \sigma_k^\nu \sigma_i^\mu \sigma_k^\mu + \sum_{\nu \geq \mu} x_i^\nu(\ell) \sigma_i^\nu \sigma_k^\nu \sigma_i^\mu \sigma_k^\mu \right]. \end{aligned} \quad (10.15)$$

³ اثبات همگرایی بعد از تئوری همگرایی پرسپترون آمده است .

اکنون N ماتریس $(p \times p)$ را معرفی می کنیم.

$$B_i^{\mu\nu} = \frac{1}{N} \sum_k \sigma_i^\nu \sigma_k^\nu \sigma_i^\mu \sigma_k^\mu = \sigma_i^\nu \sigma_i^\mu Q_{\mu\nu}, \quad (i = 1, \dots, N), \quad (10.16)$$

که $Q_{\mu\nu}$ ماتریس متقارن همپوشانی تعریف شده در (10.1) است. این به ما اجازه می دهد که (10.15) را به فرم ساده تری بنویسیم:

$$x_i^\mu(\ell + 1) - x_i^\mu(\ell) = 1 - \sum_{\nu < \mu} B_i^{\mu\nu} x_i^\nu(\ell + 1) - \sum_{\nu \geq \mu} B_i^{\mu\nu} x_i^\nu(\ell). \quad (10.17)$$

فرض می کنیم که روند تکرارشونده همگرا باشد:

$$\lim_{\ell \rightarrow \infty} x_i^\mu(\ell) = y_i^\mu, \quad (10.18)$$

مقدار حدی باید صدق کند:

$$\sum_\nu B_i^{\mu\nu} y_i^\nu = 1 \quad (10.19)$$

برای همه مقادیر i و μ . این یک سیستم خطی با Np معادله برای مقادیر y_i^ν می باشد. روند تکرارشونده (10.16) همان روش معروف سیدل-گوس برای حل تکرارشونده سیستم معادلات خطی است، در اینجا معادلات (10.19) می توان نشان داد که این روش همیشه همگرا می شود اگر ماتریس $B_i^{\mu\nu}$ فقط مقادیر ویژه مثبت داشته باشد، به عنوان مثال $\sum_{\mu, \nu} B_i^{\mu\nu} z^\mu z^\nu$ به فرم نیمه قطعی مثبت درجه دوم است. این شرایط قطعا در مورد ما صدق می کند، چون به خاطر تعریف (10.16) داریم:

$$\sum_{\mu, \nu} B_i^{\mu\nu} z^\mu z^\nu = \frac{1}{N} \sum_k \left(\sum_\nu \sigma_i^\nu \sigma_k^\nu z^\nu \right)^2 \geq 0. \quad (10.20)$$

نتیجه می گیریم که روند تکرارشونده قطعا همگرا خواهد شد و ایجاد پیوندهای سیناپسی می کند:

$$w_{ij} \rightarrow \bar{w}_{ij} = \frac{1}{N} \sum_\nu y_i^\nu \sigma_i^\nu \sigma_j^\nu. \quad (10.21)$$

با توجه به رابطه (10.16) بین ماتریس های $B_i^{\mu\nu}$ و $Q_{\mu\nu}$ می توانیم معادله (10.19) برای y_i^ν به این فرم نیز بنویسیم:

$$\sum_{\nu} Q_{\mu\nu} \sigma_i^{\nu} y_i^{\nu} = \sigma_i^{\mu} , \quad (10.22)$$

که در ضرب می کنیم و از رابطه مفید $(\sigma_i^{\mu})^2 = 1$ استفاده می کنیم. با ضرب کردن در ماتریس معکوس $Q_{\mu\nu}$ و استفاده از همان رابطه داریم:

$$y_i^{\nu} = \sigma_i^{\nu} \sum_{\mu} (Q^{-1})_{\nu\mu} \sigma_i^{\mu} . \quad (10.23)$$

بعد از وارد کردن عبارت بالا در معادله (10.21) که شدت سیناپسی در انتهای فرایند تکرار شونده را توصیف میکند، نتیجه زیر را بدست می آوریم:

$$\bar{w}_{ij} = \frac{1}{N} \sum_{\mu, \nu} (Q^{-1})_{\mu\nu} \sigma_i^{\mu} \sigma_j^{\nu} \equiv \tilde{w}_{ij} . \quad (10.24)$$

این روابط دقیقاً پیوندهای سیناپسی (10.2) \tilde{w}_{ij} قانون تصویر (تجسم) که در ابتدای این بخش بحث شد، هستند که مسئله الگوهای همبسته ذخیره شده را حل می کرد.

10.1.3 یادگیری مکرر Hebbian

10-1-3: برای بیشتر اهداف عملی لازم نیست که دقیقاً از جفت های سیناپسی بهینه \tilde{w}_{ij} استفاده کنیم یا به عبارت دیگر لازم نیست که سمت چپ معادله (10.7) را دقیقاً برابر با یک بدست آوریم. یادآوری می کنیم که نقطه شروع ملاحظات ما، این خواسته بود که عبارت $\sigma_i^{\mu} h_i$ را به طور چشمگیری بزرگتر از آستانه صفر پایداری-بحرانی بدست آوریم. این شرط نیز اگر پیوند سیناپسی را طوری تصحیح کنیم که سمت چپ (10.7) بزرگتر یا مساوی با آستانه K داده شده باشد که ممکن است با یک برابر باشد یا نباشد، نیز صدق خواهد کرد. این رابطه یک مزیت مهمی دارد که

فرایند تکرارشونده بعد از چند مرحله محدود قطعا به پایان خواهد رسید و بالاترین ظرفیت حافظه را ایجاد خواهد کرد $\alpha_c = 2$.

همان طور که در بالا توضیح داده شد این روند دقیقا کار میکند، مگر اینکه ما همه سیناپسی های یک نورون بحرانی را با مقدار ثابت $1/N$ شدت ببخشیم، به عنوان مثال عبارت (10.10) را جایگزین کنیم:

$$\delta w_{ij} = \frac{1}{N} \sigma_i^\mu \sigma_j^\mu (1 - \delta_{ij}) \theta(\kappa - \gamma_i^\mu), \quad (10.25)$$

با تابع پله ای $\theta(x)$ و اندازه گیری پایداری نرمال شده داریم:⁴

$$\gamma_i^\mu = \sigma_i^\mu h_i / \|w_i\|, \quad \|w_i\| = \left(\sum_{j \neq i} w_{ij}^2 \right)^{1/2}. \quad (10.26)$$

در اینجا به طور صمنی جفت های خود سیناپسی w_{ii} صرف نظر می کنیم که منجر به عملکرد بهتر حافظه خواهد شد همان طور که در بخش های قبلی بحث شد. با محاسبات مشابه درباره رابطه (10.11) داریم:

$$\gamma_i^{\mu'} = \gamma_i^\mu + \theta(\kappa - \gamma_i^\mu). \quad (10.27)$$

وقتی این عبارت بزرگتر از K باشد، روند تکرارشونده همگرا خواهد شد. در غیر این صورت سیناپسی تقویت شده باید تکرار شود.

به یک معنا این روال با یادگیری تکرارشونده بر طبق قاعده Hebb قابل درک است، که سیناپسی ها با مقدار $1/N \sigma_i^\mu \sigma_j^\mu$ افزایش می یابند، همان طور که اغلب به عنوان پایداری لازم برای همه الگوهای ذخیره شده ضروری است. این روند یادآور تجربه یادگیری یک واژه جدید از زبان خارجی است که معمولا لازم است چندین باز آن کلمات را تکرار کنید

⁴ منابع مختلف در تعاریف خود از پایداری اختلاف اندکی دارند. خواننده علاقمند مشتاق کنکاش کردن منابع اصلی برای جزئیات بیشتر خواهد بود. ما همچنین فصل 22 را که اطلاعات بیشتری درباره قانون یادگیری پیدا می شود پیشنهاد می کنیم.

تا وارد حافظه بلندمدت شما بشوند. همان طور که همه می دانید، این روش قطعا کار خواهد کرد، البته بعد از تکرار های زیاد آزاردهنده.

بنابراین بسیار مهم است که سرعت یادگیری را تا حد امکان بهینه کنیم. برای این هدف، آبوت و کپلر (Abbott and Kepler) رابطه (10.25) را با معرفی تابع جدید $f(\gamma)$ تصحیح کردند که اندازه تغییرات سیناپسی را بر طبق انحراف باقی مانده از هدف پایداری مورد نظر تنظیم میکند (تعدیل میکند):

$$\delta w_{ij} = \frac{1}{N} \sigma_i^\mu \sigma_j^\mu (1 - \delta_{ij}) f(\gamma_i^\mu) \|w_i\| \theta(\kappa - \gamma_i^\mu) . \quad (10.28)$$

دو مورد از این تابع مد نظر است، یکی تابع شبه خطی

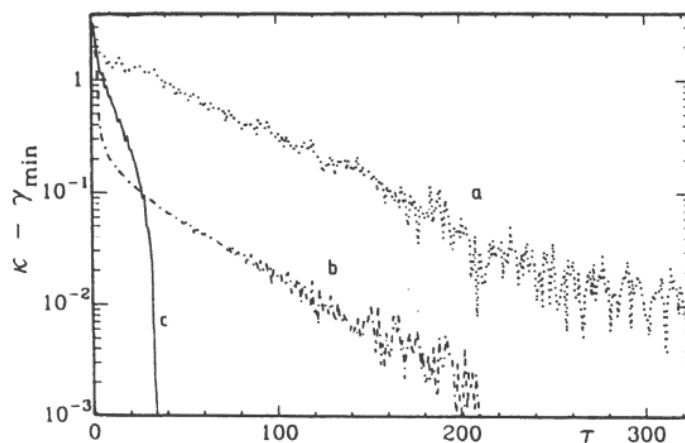
$$f_L(\gamma) = (\kappa + \delta - \gamma)\theta(\kappa + \delta - \gamma) - 2\gamma\theta(-\kappa - \delta - \gamma) \quad (10.29)$$

و دیگری تابع غیرخطی:

$$f_{NL}(\gamma) = (\kappa + \delta - \gamma) + \sqrt{(\kappa + \delta - \gamma)^2 - \delta^2} . \quad (10.30)$$

در اینجا $1 \ll \delta$ پارامتری است که سرعت یادگیری را کنترل می کند. برای تابع شبه خطی (10.29) الگوریتم پس از مدتی کمتر از $2N/\delta^2$ تکرارها همگرا می شود.

سرعت همگرایی برای شبیه سازی نمونه ای با پارامترهای $\kappa=0.43$ و $\delta = 0.01$ و $N=100$ در شکل 10.3 نشان داده شده است. تراکم (چگالی) ذخیره گاه $\alpha = 0.75$ است یعنی 75 الگو توسط شبکه ذخیره شده است. منحنی a به الگوریتم استاندارد (10.28) اشاره دارد در حالی که منحنی های b و c به الگوریتم بهینه شده (10.28) به ترتیب با تابع f_L و f_{NL} اشاره دارد. مزیت الگوریتم های تصحیح شده کاملا واضح است. عبارت تحلیلی برای سرعت همگرایی الگوریتم استاندارد (10.25) با استفاده از [Op88] بدست آمده است، که نشان می دهد همگرایی به طور چشمگیری کند می شود وقتی تراکم حافظه به مقدار بحرانی نزدیک می شود.



شکل 10.3: سرعت همگرایی برای قانون یادگیری تکرارشونده (10.25) منحنی a و (10.28) منحنی b و c. τ تعداد تکرارهاست.

آستانه پایداری K چقدر باید بزرگ انتخاب شود؟ اگر K بسیار بزرگ انتخاب شود هیچ جوابی از شرط $\gamma_i^H > K$ وجود ندارد. در نتیجه الگوریتم همگرا نخواهد شد. اگر K خیلی کوچک باشد الگوریتم (10.25) به سرعت همگرا خواهد شد و الگوهای ذخیره شده پایدار خواهند بود؛ اما محدوده جذب کوچک است و شبکه عصبی لزوماً الگویی که کمی منحرف شده باشد را تشخیص نخواهد داد. مقدار مشخصی از تجربه برای پیدا کردن مجموعه ای بهینه از پارامترهای یادگیری لازم است. وابستگی شعاع متوسط جذب از الگوهای ذخیره شده به انتخاب K توسط آبوت و کیپلر مطالعه شد و مشخص شد که برای شبکه اشباع شده به $R=0.1$ در $K=1$ کاهش می یابد.



از برنامه ASSO استفاده کنید (فصل 22) برای یادگیری همه 26 حرف الفبا با طرح های یادگیری بهبود یافته 2-
5. سرعت همگرایی و پایداری الگوها را در برابر اختلال و نوسانات گرمایی مقایسه کنید.



تمرین بخش 3.4 را با برنامه ASSCOUNT (فصل 23) و استفاده از پروتکل یادگیری Diederich-Opper تا 10 عدد تکرار کنید. آزمایش با پارامترها تاخیر زمانی را کنترل می کند.

10.2 قانون یادگیری خاص

10.2.1 ارتقاء حافظه را فراموش کنید!

همان طور که در بخش 3.3 بحث کردیم قانون یادگیری استاندارد (قانون Hebb) منجر به بروز فراموشی کلی ناخواسته در تابع انرژی $E[s]$ می شد. در عمل به این معناست که تکامل شبکه می تواند گرفتار جعلی بودن، پیکربندی های پایدار محلی، نظیر آنچه که در (3.23) بود؛ بشود. شبکه های بزرگ معمولاً تعداد وسیعی از اعداد نظیر حالت های پایدار جعلی می شود که بسیاری از آنها حتی ترکیب خطی نقاط پایدار مورد نظرهم نیست. قانون یادگیری برای الگوهای همبسته که در بخش 10.1 بحث شد، از این مشکل در امان نیستند. نوسانات حرارتی کمک می کنند تا پیکربندی های جعلی بی ثبات شوند، اما با مصرف ظرفیت حافظه، علاوه بر آن ناپدید کردن همه وضعیت های جعلی در دمای نهایی T تضمین شده نیست.

استراتژی بسیار بهتر، از بین بردن پیکربندی های پایدار ناخواسته با تصحیحات مناسب پیوندهای سیناپسی است. هوپفیلد (Hopfield) پیشنهاد داد تا از این حقیقت که فراموشی جعلی تابع انرژی $E[s]$ معمولاً بسیار کم سطح تر از فراموشی متناظر با الگوهای یادگیری است استفاده کنیم. ایده های قرض گرفته، در مطالعه رویاهای انسان توسعه پیدا کرده است و در بخش 2.3 مورد بحث واقع شده است؛ دنبال کردن این حالت ها با شروع شبکه در یک پیکربندی اولیه تصادفی و اجرای آن تا زمانی که به یک حالت پایدار S_i^∞ متعادل برسد. این امر ممکن است یکی از الگوهای منظم

یادگیری، یا یکی از بسیار حالت های جعلی باشد. فارغ از این که حالت بدست آمده چه باشد، سیناپس ها بر طبق رابطه Hebb به طور جزئی ضعیف می شوند:

$$w_{ij} \rightarrow w_{ij} - \frac{\lambda}{N} s_i^{\infty} s_j^{\infty}, \quad (10.31)$$

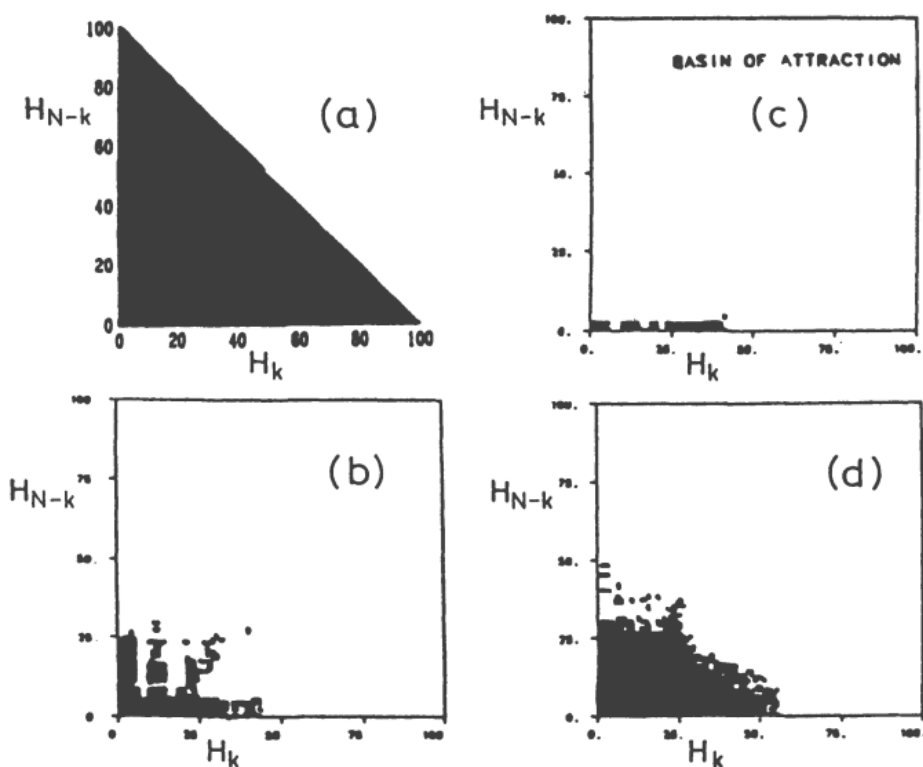
$\lambda \ll 1$ انتخاب شده است. این روال فراموشی دو اثر مثبت دارد. بیشتر حالت های متعادل جعلی از شبکه "فراموشی شدنی" هستند، چون آنها قبلاً با تغییرات کوچک در پیوندهای سیناپسی w_{ij} بی ثبات شده اند. علاوه بر آن ناحیه مختلف پایداری الگوهای ذخیره شده، در اندازه بسیار شبیه هم خواهد شد، چون آنهایی که پایداری بیشتری دارند اغلب بیشتر رخ می دهند نسبت به پیکربندی نهایی و بنابراین بیشتر از بقیه ضعیف تر می شوند.

تأثیر این فراموشی عمدی به خصوص در اندازه حوزه جذب آشکار می شود. این جمله نشان می دهد که مجموعه ای از حالت ها از شبکه های پویا (دینامیک) منجر به الگوی خاصی می شوند. تغییر اندازه حوزه جذب الگوی ذخیره شده، همانند بار کل حافظه افزایش می یابد، که در شکل 10.4 نشان داده شده است. دو محور که H_k و H_{N-k} نام گذاری شده است در این شکل ها نمایانگر اندازه خام فاصله حالت آزمایشی اول s_i از حالت حافظه مورد نظر σ_i^{μ} می باشد. همچنین فاصله جزئی Hamming بین حالت اولیه و حالت حافظه را نشان می دهد که به ترتیب برای اولین K و آخرین $(N - K)$ و تمام 200 نورون ارزیابی شده است.

$$H_k = \frac{1}{4} \sum_{i=1}^k (s_i - \sigma_i^{\mu})^2, \quad H_{N-k} = \frac{1}{4} \sum_{i=k}^{N-k} (s_i - \sigma_i^{\mu})^2. \quad (10.32)$$

در این مورد خاص $k=N/2$ گرفته شده است برای مثال محور اول نمایشگر فاصله Hamming و دیگری نمایشگر نصف نورون های شبکه است. اگر حالت آزمایشی s_i به الگوی ذخیره شده μ توسعه پیدا کند، لکه سیاه رسم می شود. برای حالت حافظه تنها مشخص خواهد شد (شکل 10.4a) نصف حالت های آزمایشی به سمت الگوی ذخیره شده σ_i تکامل پیدا خواهند کرد، نصف دیگر در الگوی مکمل ($-\sigma_i$) منتهی خواهند شد. بنابراین حوزه جذب نمایشگر مثلث سیاه است. برای بیشتر حالت های حافظه این ناحیه به سرعت کاهش می یابد و در محدوده α_c شکل بسیار ناهموار خواهد

گرفت همان طور که به ترتیب در شکل 10.4b,c برای 28 و 32 حالت حافظه ناهمبسته نشان داده شده است. شکل 10.4d نتیجه هزار بار کاربرد الگوریتم فراموشی (10.31) برای شبکه لود شده با 32 الگو را نشان می دهد. (شدت فراموشی برابر است با: $\lambda=0.01$). حوزه جذب به شدت افزایش می یابد (با ضریب 10 یا بیشتر) و یک شکل بسیار منظم به خود می گیرد. احتمال بازیابی الگوهای ذخیره شده بسیار بهبود یافته است.



شکل 10.4: حوزه جذب در شبکه هوپفیلد با 200 نورون برای (a) 1 (b) 28 (c) 32 حالت حافظه. بعد از

فراموشی عمدی حوزه به شدت افزایش می یابد (d).

در بعضی از نسخه های ویرایش شده از این روش، شبکه مجاز است از الگوی های ذخیره شده که با اختلال تصادفی رو به وخامت رفته است، فراتر رود. بنابراین نه تنها پیوندهای سیناپسی توسط حالت های نهایی s_i^∞ ضعیف تر می شوند، بلکه همزمان بازآموزی الگوی صحیح شروع v نیز انجام می شود:

$$w_{ij} \rightarrow w_{ij} - \frac{\lambda}{N} \left(s_i^\infty s_j^\infty - \sigma_i^\nu \sigma_j^\nu \right). \quad (10.33)$$

اگر الگو بدون خطا یادآوری شود، سیناپس ها بر طبق این دستور بدون تغییر می مانند. با این روش ظرفیت ذخیره گاه تا $\alpha = 1$ افزایش می یابد و ذخیره الگوهای وابسته شدید امکان پذیر می شود.

متاسفانه این روش ها حالت های پایدار جعلی متناظر با ترکیب خطی الگوهای ذخیره شده نظیر (3.23) را از بین نمی برند. تصحیح سیناپسی کل برای همه 8 حالت گرفته شده، دقیقا ناپدید می شود چون مجموع تغییرات (10.31) به صفر می رسد. با این حال این روش با کمی اختلاف به خوبی کار میکند که فراموشی با قانون زیر کنترل می شود:

$$w_{ij} \rightarrow w_{ij} - \frac{\eta}{N} (\sigma_i^\mu \sigma_i^\nu \sigma_i^\lambda) (\sigma_j^\mu \sigma_j^\nu \sigma_j^\lambda) . \quad (10.34)$$

در اینجا λ و ν و μ هر سه نشان دهنده الگوهای ذخیره شده هستند. برای $\eta > 1/3$ یک یافته این است که همه هشت حالت جعلی (3.23) قبلا در $T=0$ ناپایدار شده اند. در محدود مقداری از پارامتر دما مقدار کم λ برای بی ثبات کردن کفایت می کند و بالاتر از $T=0.46$ این پیکربندی های جعلی ناپایدار می شوند چون عمل نوسانات حرارتی تنهاست.

10.2.2 قانون یادگیری غیرخطی

اشکال خاص قانون Hebb (3.12) برای الگوهای p :

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^p \sigma_i^\mu \sigma_j^\mu \quad (10.35)$$

برای بسیاری از کاربردها این است که شدت های سیناپسی می تواند محدوده وسیعی داشته باشد ($-p/N \leq \omega_{ij} \leq p/N$). به طور خاص در پیاده سازی سخت افزاری شبکه های عصبی این یک آشفتگی (اختلال) است چون به عناصر سوئیچینگ الکترونیکی با محدوده دینامیکی گسترده نیازمند است. بنابراین طبیعی است که بپرسیم چه محدوده ای از مقادیر مجاز ω_{ij} را می توان با اطمینان محدود کرد. مورد بی نهایت باید فقط بین سیناپس های تحریکی ($\omega_{ij} > 0$) و مهارتی ($\omega_{ij} < 0$) متمایز شود که شدت یکسان مطلق اما با علامت مختلف نسبت داده شده است:

$$w_{ij} = \frac{\sqrt{p}}{N} \operatorname{sgn} \left(\sum_{\mu=1}^p \frac{1}{\sqrt{p}} \sigma_i^\mu \sigma_j^\mu \right) = \pm \frac{\sqrt{p}}{N}. \quad (10.36)$$

این فرایند که قطع سیناپس نامیده می شود یک مورد خاص از قانون Hebb غیر خطی است (بخش 19.3 را ببینید).

$$w_{ij} = \frac{\sqrt{p}}{N} \Phi \left(\frac{1}{\sqrt{p}} \sum_{\mu=1}^p \sigma_i^\mu \sigma_j^\mu \right), \quad (10.37)$$

که $\Phi(x)$ یک تابع دلخواه که به طور یکنواخت افزایش می یابد. انتخاب $\Phi(x) = x$ به همان قانون استاندارد (خطی)

Hebb (3.12) منجر می شود که $\Phi(x) = \operatorname{sgn}(x)$ سیناپس های قطع را توصیف می کند.

سیناپس های قطع به طرز عجیبی تاثیر بسیار کمی بر روی ظرفیت ذخیره گاه شبکه و توانایی بیادآوری الگوهای

ذخیره شده دارد. در مقایسه با قانون خطی Hebb، قانون یادگیری (10.36) مانند یک اختلال اضافی عمل می کند

که تراکم حافظه بحرانی را $\alpha_c = p_{\max}/N$ در $T=0$ از $\alpha_c^{\text{Hebb}} = 0.138$ به $\alpha_c = 0.102$ کاهش می دهد. می

توان نشان داد که در حالت کلی ظرفیت حافظه برای سیناپس های اختیاری همیشه کمتر از مورد خطی Hebb

است ($\alpha_c^\Phi \leq \alpha_c^{\text{Hebb}}$). در حافظه کم تراکم به عنوان مثال برای مقادیر کم α ، نسبت خطا $(1-m)/2$ در الگوی

یادآوری به مقدار ناچیزی سیناپس های قطع بزرگتر از مورد Hebb می باشد. این زوالی توسط نتایج مهم ساده شده

برای پیوندهای سیناپسی ذخیره گاه ω_{ij} در تحقق سخت افزاری جبران می شود، همچون شبیه سازی نرم افزاری

شبکه عصبی در کامپیوترهای دیجیتالی معمولی.

مورد خاص قابل توجه، تابع های شدت سیناپسی محدود شده هستند $|\Phi(x)| \leq \Phi_0$. در اینجا بهتر است تا قانون

یادگیری (10.37) را تصحیح کنیم، به منظور دسترسی بیشتر و بیشتر الگوها به حافظه: اگر سیناپس ها بعد از ذخیره

$(\mu - 1)$ الگو توسط $\omega_{ij}^{(\mu-1)}$ نام گذاری شوند، آنهایی که بعد از اضافه کردن الگوی بعدی به حافظه بدست می آیند

مانند زیر تعریف می شود:

$$w_{ij}^{(\mu)} = \Phi \left(\epsilon \sigma_i^\mu \sigma_j^\mu + w_{ij}^{(\mu-1)} \right). \quad (10.38)$$

اگرچه در مورد قانون خطی Hebb اضافه کردن پیوسته بیشتر حالت های حافظه سرانجام منجر به از کار افتادن توانایی شبکه در بازیابی هر گونه الگوی ذخیره شده می شود، اما عملکرد نزولی حافظه برای قانون یادگیری غیر خطی با سیناپس های شدید محدود با سرعت بسیار آرام تری صورت می گیرد. همان طور که الگوهای بیشتری ذخیره می شود، شبکه به حد ظرفیت ذخیره گاه نزدیک تر می شود α_c . با این حال به جای وارد کردن حالت سردرگمی کلی، شبکه یک ماتی تدریجی از حالت های حافظه قدیمی تر تجربه می کند که به طور آهسته توسط الگوهای تازه یاد گرفته شده، جایگزین می شود.

چون کل حافظه می تواند سرانجام به عنوان ترتیبی از الگوهای تازه قابل بازیابی که بر روی الگوهای قدیمی که وخامتشان افزایش می یابد، دیده شود، چنین ساختار حافظه ای را اغلب پالیمپ سست⁵ (دست نوشته هایی که نوشته روی آن پاک شده باشد ولی دوباره نوشته باشند) می نامند. حافظه هایی از این نوع می توانند به عنوان حافظه های کوتاه مدت موثر به کار گرفته شوند، چون به شبکه اجازه می دهند تا عملکردی مانند وسایل ذخیره اطلاعات به صورت پیوسته، بدون مواجه شدن با محدود شدن حافظه داشته باشد. از طرف دیگر اکثر قریب به اتفاق مدارک فیزیولوژی نشان می دهد که ساختار حافظه کوتاه مدت در مغز وجود دارد، چنین حافظه هایی کاربردهای بسیار مهمی در پردازش اطلاعات الکترونیکی دارند که حافظه های نهان نامیده می شود.

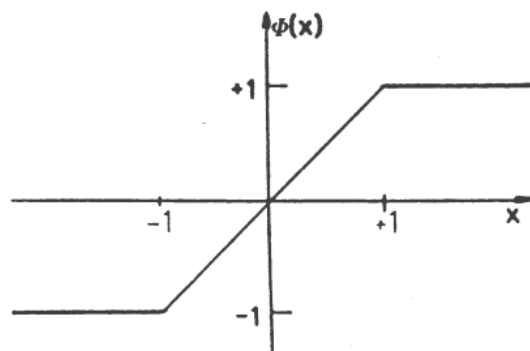
قانون یادگیری غیرخطی (10.38) توسط پارسی (Parisi) برای توابع محدود مورد مطالعه قرار گرفت:

$$\Phi(x) = \begin{cases} x & \text{for } |x| \leq 1 \\ \text{sgn}(x) & \text{for } |x| > 1 \end{cases} \quad (10.39)$$

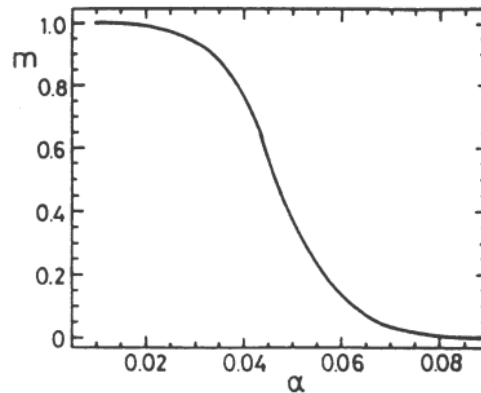
که در شکل 10.5 نشان داده شده است.

⁵ اصطلاح پالیمپ سست از تمرین استفاده شده در قرون وسطی گرفته شده است، وقتی نسخه خطی روی پوست آهو بسیار خاص بود که چندین بار بر روی آن نوشته می شد؛ نوشته های پیشین برای استفاده بعدی به طور کامل یا جزئی پاک می شدند.

همان طور که انتظار می رفت، کیفیت یادآوری m الگوی ذخیره شده بعد از اضافه شدن دیگر الگوها به حافظه، رو به زوال رفت همان طور که در شکل 10.6 نشان داده شده است. اگرچه ظرفیت کل حافظه در هر زمانی محدود باقی می ماند (یافته پارسی: $\alpha_{\max} = p_{\max}/N \approx 0.04$)، شبکه هرگز توانایی یادگیری الگوهای جدید را از دست نمی دهد. مدل های مشابه توسط نادال (Nadal) و دیگر نویسندگان مورد بررسی قرار گرفت. بحث کلی از ویژگی های آنها در [Mo88]، پیدا می شود.



شکل 10.5: تابع محدود شده شدید سیناپسی برای شبکه های حافظه پالیمپست به کار می رود.



شکل 10.6: کیفیت یادآوری m از حالت حافظه قریبی تر بعد از اضافه کردن $p = \alpha N$ حالت جدید، کم کم ناپدید می شود.



از برنامه ASSO استفاده کنید برای یادگیری حداقل 10 حرف از حروف الفبا با یکی از طرح های یادگیری 5-2، از مقادیر استاندارد برای پارامترها استفاده کنید. سپس مورد m را در منو جستجو انتخاب کنید تا شدت سیناپسی را محدود کنید یا فقط سیناپس های دودویی را انتخاب کنید.



از برنامه ASSO استفاده کنید برای یادگیری حداقل 10 حرف از حروف الفبا با یکی از طرح های یادگیری 5-2، از مقادیر استاندارد برای پارامترها استفاده کنید به جز در انتخاب سیناپس های مثبت (محرک) یا منفی در صفحه نمایش.

10.2.3 رقت سیناپس

فرض اینکه همه نورون ها به یکدیگر متصل هستند واقعی نیست، به خصوص اگر کسی به مدل شبکه عصبی بیولوژیکی علاقمند باشد. ممکن است بپرسید آیا شبکه عصبی به عنوان یک حافظه انجمنی موثر عمل کند اگر کسر بزرگی از پیوندهای سیناپسی قطع شوند یا کم شدت شوند؟ اتصال سیناپسی معمولا با حذف پیوندها (اتصالات) به طور تصادفی رقت (کم شدت) می شوند و فقط کسر $d < 1$ از همه سیناپس ها نگه داشته می شود. اگر این کار انجام شود در حالیکه تقارن ماتریس سیناپسی نیز حفظ شود ظرفیت حافظه α_c یک شبکه آموزش دیده با قانون Hebb تقریبا به طور خطی با d کاهش پیدا می کند همان طور که در شکل 10.7 نشان داده شده است که α_c و ضریب کیفیت یادآوری بحرانی $m_c = m(\alpha_c)$ به عنوان تابعی از کسر پیوندهای سیناپسی تخریب شده نشان داده شده است.⁶ این نتایج

⁶ به طور ضمنی بیان می کند که کارامدی ذخیره گاه شبکه واقعا کاهش نمی یابد چون سیناپس های کمی در تناسب لازم هستند. چیزی که باید در خاطر داشته باشید این است که پیچیدگی سیستم عصبی با تعداد N نورون های آن تعریف نمی شود بلکه با تعداد کل پیوندهای سیناپسی $1/2dN(N-1)$ مشخص می شود، بنابراین کارامدی واقعی حافظه توسط این رابطه داده می شود: α_c/d .

به وضوح مقاومت در برابر خطا شبکه های عصبی را نشان می دهد. حتی بعد از حذف کسر بزرگی از سیناپس ها شبکه تلاش می کند تا کاملا قابل اعتماد عمل کند. رهیافت دیگر رقت کردن سیناپس های غیر متقارن است برای مثال تنظیم $\omega_{ji} = 0$ اما $\omega_{ij} = 0$ دیگر لازم نیست. این مورد می تواند در حد $d \rightarrow 0$ به طور تحلیلی حل شود. یکی از یافته ها این است که ظرفیت حافظه در سیناپس های باقیمانده تقریبا چهار برابر $\alpha_c = 2d/\pi \approx 0.64d$ برای شبکه کامل متصل هستند.

ویراسورو (Virasoro) اشاره کرده است که تخریب تصادفی سیناپس ها می تواند منجر به اثرات جالبی شود وقتی الگوهای ذخیره شده ساختار مشابه سلسله مراتبی دارند. این امر به این معناست که الگوها به چندین دسته و کلاس قطعی تقسیم بندی میشوند، الگوهایی که به کلاس عمومی تعلق دارند شدیداً همبسته هستند. مجموعه ای از الگوها با این ویژگی را می توان به این روش تولید کرد. ابتدا الگوهای کلاس ناهمبسته را انتخاب کنید ξ_I^α , ($i = 1, \dots, N$; $\alpha = 1, \dots, Q$) که $\xi_I^\alpha = \pm 1$ با احتمال برابر برای هر دسته α هر کدام p_α الگوی همبسته تولید می کند $\sigma_i^{\mu\alpha} = \pm \xi_I^\alpha$ مقدار $\sigma_i^{\mu\alpha}$, ($i = 1, \dots, N$; $\mu = 1, \dots, p_\alpha$) در حد $m \rightarrow 1$ احتمال $(1 \pm m)/2$ را بگیرد با احتمال $m \rightarrow 1$ الگو در دسته مشابه کوچک و کوچکتر می شود. وقتی توانایی های حافظه شبکه رو به زوال می رود مثلا به خاطر بار اضافی، حضور اختلال حرارتی یا رقت سیناپسی، شبکه ممکن است به مرحله ای برسد که الگوی فردی $\sigma_i^{\mu\alpha}$ دیگر قابل بازیابی نباشد، اما به یادآوری کلاس متناظر ξ_I^α همچنان امکان پذیر است.⁷ چنین رفتارهایی بسیار جالب هستند چون شبکه می تواند عمل طبقه بندی را انجام دهد.