Subject:
Year. Month. Date. (۱)

شبکه های عصبی

* استاد:
دکتر رضا صفابخش
safa@aut.ac.ir

* نگارنده:
بردیا شکلی

* مراجع درس:

- Neurocomputing, R Hecht-Nielsen, Addison Wesley, 1990
  قدیمی‌ترین و زبان جذاب‌ترین
- Fundamentals of Neural Networks, L. Fausett, Prentice-Hall, 1994
  آسان، کلاسیک
- Neural Networks, A comprehensive Foundation, S. Haykin, Macmillan College Pub., 1994
  قدیمی‌ترین و جامع‌ترین، دیدگاه آماری و سخت
- Artificial Neural Networks, Theory and Applications, D.W. Petterson, Prentice-Hall, 1996
- Introduction to the Theory of Neural Computation, Hertz, Krogh & Palmer, Addison Wesley, 1991
  موضوعات محدودتر و دید تئوری
- Introduction to Artificial Neural Systems, J.M. Zurada, Info Access and Distribution, 1992

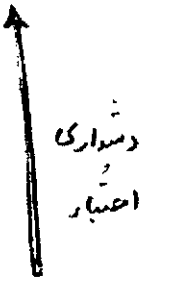
- محمدباقر منهاج، مبانی شبکه‌های عصبی، انتشارات دانشگاه صنعتی امیرکبیر، ۱۳۷۹

PqPCO

Subject:
Year. Month. Date. ( )

* مهمترین ژورنالهای مربوط

- Neural Computation , MIT Press
- IEEE Trans. on Neural Networks , IEEE
- Neural Networks , Elsevier
- Neural Processing Letters , Springer
- Neurocomputing , Elsevier



* مهمترین کنفرانسها

- IJCNN : Intl. Joint Conf on NN
- ICANN : Intl. Conf on ANN
- WCCI : IEEE World Congress on Computational Intelligence
- ESANN : European Symp. on ANN
- ICNC , Intl. Conf. on Neural Computation
- ISNN : Intl. Symp. on NN
- ICIC

* مباحث درس

• مقدمات : مفاهیم ، تاریخچه ، کاربرد

• شبکه‌های عصبی طبیعی : ساختار نورون ، مغز

• شبکه‌های عصبی مصنوعی : شبکه ، پردازنده ، اتصالات

• مسأله دسته‌بندی : معرفی ، حل با واحد پرسپترون

• توابع فعالیت و شبکه‌های توابع ترسیده : پیش خور ، پس خور ، دو طرفه

• یادگیری : با نظارت ، بی نظارت ، با ترکیبی از این دو

• شبکه‌های خود سازمان ده : رقابتی ، رقابتی دارای وزن تابت ، کوانتیزاسیون برداری

• شبکه پرسپترونی خطی و قانون استاندارد خطا به عقب : معرفی ، تقریب ، تعمیم ...

• شبکه‌های توابع پایه شعاعی : چگونگی یادگیری لایه‌ها ، آموزش شبکه

• شبکه‌های ابتکاری : بهینه‌سازی ، شبیه‌سازی تبرید ، ماشین بولتزمان

• شبکه تقریب تشدید تطبیقی : معرفی ART1

• شبکه‌های دیلی : Time Delay NN

Subject :

Year . Month . Date . ( )

* ارزیابی

• امتحان پایان ترم	٪ ۵۰
• گزارش مطالعاتی	٪ ۱۰
• ارائه شفاهی	٪ ۱۰
• پیاده سازی	٪ ۱۵
• پروژه ۴۰	٪ ۱۵

Subject:
Year. Month. Date. (۳)

@ مقدمه

* عناوین شبکه‌های عصبی

Neural Networks , Neural Nets
Artificial Neural Networks (ANN)
Connectionist Model ← فضای روانشناسی
Parallel Distributed Processing Model (PDP)
Neuromorphic Systems
Neurocomputing , Neural Computing

شبکه‌های عصبی ، شبکه‌های نورونها ، شبکه‌های سلولهای عصبی (مصنوعی)

* شبکه‌های عصبی چه هستند؟

راهی برای تقلید از عملکرد مغز انسان
مغز انسان می تواند ادراک را استنتاج هوشمندی از اطلاعات ناقص، مبهم یا متناقض داشته باشد.
انسان نه از کامپیوتر سریعتر است و نه دقیقتر
اما انسان بسیاری از کارها را بهتر از کامپیوتر انجام می دهد.
☑ درک اشیای موجود در یک صحنه و روابط بین آنها
فهم سریع و صحیح زبان
بازیابی سریع اطلاعات مرتبط از حافظه
طراحی رباتی برای انجام کارها : تعیین بدنهای لازم
انجام بهتر کارها با تجربه آن : نسبت به
یادگرفتن

Subject:
Year. Month. Date. ( )

* چرا انسان بهتر از ماشین عمل می کند.

- ما هنوز نرم افزارهای لازم را نداریم.

- معماری مغز با معماری ماشینهای ما متفاوت است.
معماری مغز برای پردازش اطلاعات طبیعی مناسب تر است.

* دیدگاهی پردازش اطلاعات طبیعی

- اطلاعات و محدودیتهای محیطی باید به طور بهینه در نظر گرفته شود.

- این اطلاعات و محدودیتها هم ممکن است ناقص یا مبهم باشد.

☑ وقتی می خواهیم از جای نشسته جسمی را از روی زمین برداریم و آن جای جایی در دست ما است

طرح ریزی حرکات بدن
جایی بنویسد، از روی صندلی پایین بیاییم، دستمان به لبه تیز میز نخورد
وزن جسم در نحوه برداشتن مؤثر است.
محل شئی، وضعیت شئی، اندازه و شکل و وزن مورد انتظار شئی
اشیاء دیگر سر راه ما

☑ یک زبان

معنی روی دستور اثر می گذارد و دستور روی معنی

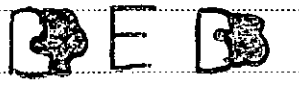
I like the joke.
I like to joke.

I saw the Grad Canyon flying to NY.
I saw the sheep grazing the field.
The boy the man chased shot the girl.

Subject:

Year. Month. Date. ۱۴

تشخیص حروف و کلمات



انسان : BED

BED EISH

برای تشخیص از دانش سطح بالای ما (زبان انگلیسی) استفاده می‌کند.
یعنی می‌کند امکانات مختلف را در نظر بگیرد.

* سیستم ادراک انسان می‌تواند بدون تصمیم‌گیری نهایی امکانات مختلف را ارزیابی کند تا اینکه سبه محدودیت ها ارضا شود. در آنحال تصمیم نهایی اتخاذ می‌شود.

ما علاقمند به ایجاد چنین توانایی‌ای در کامپیوترها هستیم.

* مقایسه رایانش (computing) سنتی و عصبی

رایانش سنتی	رایانش عصبی
— روش حل مسئله ۵۰ سال گذشته	— نیازی به الگوریتم نیست
— بدنبال یک الگوریتم می‌گردیم	— اگر الگوریتم مشخص نباشد یا مشخص باشد
— الگوریتم باید پیدا شود	ولی پیاده‌سازی آن پرهزینه، زمانبر،
— باید قابل پیاده‌سازی باشد (بسیاری)	و یا مشکل باشد می‌توان از رایانش عصبی
— باید نرم‌افزار مناسب نوشته شود.	بهره جست.

* رایانش عصبی

رایانش موازی توزیع شده و تطبیقی که به پاسخ به یک محیط اطلاعاتی قدرت پردازش اطلاعات را پیدا می‌کند.

Subject :
Year. Month. Date. ( )

* مقایسه روشهای هوش مصنوعی سنتی و رایانش عصبی

روشهای هوش مصنوعی سنتی	رایانش عصبی
ـ دانش به شکل قواعد کد می‌شود.	ـ سیستم با دریافت نمونه‌هایی از مسئله خود قواعدی ایجاد می‌کند.
ـ حافظه و اطلاعات تودیع شده کامل هست.	ـ اطلاعات در شبکه به صورت کاملاً توزیع شده است.
ـ حافظه تداعی گر نیست.	ـ حافظه تداعی گر است: به ورودی‌های ناقص، نویزی یا قبلاً مشاهده نشده پاسخ مناسب می‌دهد.
ـ صدمه به حافظه سیستم را از کار می‌اندازد.	ـ با خراب شدن بعضی واحدها و اتصالات رفتار شبکه تنها کمی تغییر می‌کند. با تغییرات بیشتر رفتار بدتر می‌شود و (Graceful Degradation) ... این ویژگی برای کاربردهای حساس بسیار مهم است.

☑ کاربردهای مناسب برای Graceful Degradation:
تشخیص اعتبار، شناسایی‌ها، بورس‌ها، ...
جایگزینی برای سخت افزار معیوب

Subject:
Year.. Month. Date. (۵)

* پیاده سازی شبکه‌های عصبی

- شبیه سازی نرم افزاری
- مدارات مجتمع پرتراکم
- کارتهای شتاب دهنده سخت افزاری
- کامپیوترهای عصبی
- پیاده سازی نوری

* شبیه سازی نرم افزاری
- متداول ترین ← طراحی و حل اولیه مسائل
- شبیه سازی صرف ← قابل ترکیب بیشتر (با نرم افزارهای دیگر)
  - شبیه سازی شبکه‌های رایج
  - مصورسازی فرایند آموزش و ساختار شبکه

SNNS, Emergent, Java NS, Neural Lab, Matlab ☑

- توسط کامپیوترهای متعارف و سیستمهای بزرگ موازی
- محیطهای توسعه ← متداول
  - Component-based
  - ترکیب انواع شبکه
  - قابل ترکیب با نرم افزارهای دیگر

Synapse, NeareSolutions, NeureLaboratory ☑

- عیب: کند

Subject:
Year. Month. Date. ( )

* مدارات مجتمع پردازش (VLSI)

• انجام سخت افزاری عمل جمع و ضرب پرداخته است

• سرعت بسیار بالا ✓ پیشرفت تکنولوژی

• آنالوگ ← سریع تر
↙ فشرده تر
↙ کم مصرف تر
↙ ارتباط مستقیم با سنسور

• دیجیتال ← دقت بیشتر
↙ انعطاف پذیری بیشتر

• پارامترهای مهم ← حداکثر تعداد نرون و اتصال (مهم)
↙ دقت وزن دهی، خروجی و وزنها
↙ دقت مدارات ضرب کننده

☑ چندلایه پیشخور، بازگشتی، تابع شعاعی

• نیازهای شبکه عصبی: ① آموزش: مسئله را بدون نمونه و تعامل با محیط حل کردن
مسئله را یاد می گیرد (تغییر پارامتر)
② استفاده: پارامتر ثابت

• یادگیری در پیاده سازی سخت افزاری ← سودآوری شامل وزنها در هنگام ساخت (نیاز ②)
↙ یادگیری وزنها پس از ساخت ← جدیدتر و سریعتر

On-Chip learning

Subject:
Year . . Month . . Date . 6

* کارتهای شتاب دهنده سخت افزاری

• افزودن شتاب دهنده‌های خاص سخت افزاری برای افزایش کارایی

• انواع کارتها از لحاظ سخت افزاری ← زیر پردازنده‌ها
پردازنده‌های ریسک سریع
کارت DSP
مدارات مجتمع عصبی

* کامپیوترهای عصبی (Neuro computers)

• شامل تعدادی Board و مجهز شدن ← مخصص این کامپیوتر

• معمولاً تحت UNIX + محیط نرم افزاری گسترده

* پیاده سازی بندی

• پیشرفت با تکنولوژی لیزر

• حل مسئله مداخل کانالهای ارتباطی ✓ فشرده سازی اتصالات (x4) ✓
بهبود تلاقی اتصالات ✓

• آرایه‌ای از پردازنده‌های ساده به طور موازی

• دیواری پیاده سازی

Subject:
Year. Month. Date. ( )

* تاریخچه

○ تا دهه ۴۰

- فعالیت های تجربی ← بیولوژیکی ← زیست شناس
  ← علوم اعصاب
  ← روانشناسی
  ← شناخت ساختار و عملکرد ← مدلسازی

- نظریه توزیع شدگی ← چند سطحی بودن سیستم های پردازش طبیعی

○ دهه ۴۰ : تولد ANN

- مدل برای نورون ← یک آستانه دیجیتال
  ← انجام عملیات منطقی و حساب

- قانون هب (Hebb) ← توانایی برای تغییرات تاثیر گذار نورون بعدی نورون دیگر در محل اتصال آنها

○ دهه ۵۰ و ۶۰ : رشد اولیه ANN

- توزیع شدگی حافظه ← شبکه
  There are no special cells reserved for special memories.

- ماشین Snark ← دستگاه الگوریتمی (مدرن + پارامتر)
  ← شبکه

Paper

Subject:
Year. Month. Date. ۷۱

تنظیم اتوماتیک وزنهای انطباقی

- پرسپترون ← واحد پردازشگر (Perception + Electronics)
  - الگوریتم یادگیری ← اثبات همگرایی
  - وزنهای انطباقی ← پدر شبکه‌های عصبی

- آدالاین (Adaptive Linear Element) ← عنصر پردازشگر
  - قانون یادگیری تندترین
  - ترکیب آدالاین‌ها (Madaline) ← ویدرو

- نقدی علیه پرسپترون ← پرسپترون مسائل ساده غیرخطی را نمی‌تواند حل کند.
  - ماشین‌های علائم به شبکه عصبی (۱۹۷۰)

• دهه ۷۰ : سالهای سکوت ANN

- شبکه‌های تداعیگر ← کوهونن

- شبکه‌های تداعیگر خطی ← بر مبنای مدلهای ذخیره، بازیابی و شناسایی حافظه
  - اندرسون

- انتشار خطا به عقب (پس‌خوراند) ← استفاده از شبکه‌ای از پرسپترونها برای حل مشکل تک‌واحدها
  - وربوز ← کمی زودتر از زمان ویدرو

- ساخت مدلهای محاسباتی عصبی ← با استفاده از داده‌های نورولوژی
  - کارهای ریاضی ارزشمند در تحلیل خواص شبکه عصبی
  - گراسبرگ

Subject:
Year. Month. Date. ( )

• دهه ۸۰ : توجه مجدد به ANN

- روش‌های توزیع شده ← تحلیل‌های ریاضی
↙ ویلیامز

- تانزل دلما ← سائل و بارتو
↙ تحلیل

- شبکه‌های پنیلید ← روشی برای شبکه‌های مداومی در بازشناسی
↙ چهره پنیلید

- شبکه نگاشت ویژگی خودسازمانده ← کوهونن
↙ Self organizing Feature Map

- شبکه نظریه تشدید تطبیقی ← کارپنتر + گراسبرگ
↙ نسخه‌های اولیه

- انتشار خطا از خروجی به لایه‌های عقب‌تر ← پارکر ← نرم‌افزار اسپرل
↙ راملهارت، هینتون و ویلیامز
↙ استفاده از آن در MLP
↙ انتشار کتاب
علاقه مجدد شدید به ANN

• پس از ۸۶ : رشد چشمگیر

- اصل جدیدی برای خودسازماندهی ← لینسکر

Subject:

Year. Month. Date. (۸)

- بر مبنای مفاهیمی از نظریه اطلاعات ← اصل حداکثرسازی اطلاعات متقابل

- شبکه‌های شعاعی ← گردهمه ← نسخه اول
  - پولیو و گیروسی ← اضافه کردن نظریه منظم سازی تیخونوف

- ماشینهای بردار پشتیبان ← وپنیک

- گرایش به سمت مدلهای واقعی‌تر شبکه‌های عصبی طبیعی ← ورودی ضربه به جای 0/1

Spiking Neurons

Subject:
Year. Month. Date. ( )

* کاربردهای شبکه‌های عصبی

• شناسایی الگو و بینایی ماشین

دسته‌بندی و خوشه‌بندی
تشخیص تصاویر
شناسایی متن، اعداد و علائم در زبان‌های مختلف
شناسایی اثر انگشت، چهره و امضا
هدایت ربات‌ها
مجاسبه حرکت
ردیابی چند هدف

• پردازش تصویر

بهسازی
بازگردانی
فشرده‌سازی

• هوش مصنوعی کلاسیک

سیستم‌های خبره و تصمیم‌یار
پردازش زبان‌های طبیعی

(Classical) AI ☑ ← سنتی ← جستجو، بازنمایی، ...
Computation Intelligence ← شبکه‌های عصبی، پردازش تکاملی، منطق فازی
و ترکیب‌های یا دگیر

• کنترل سیستم‌ها

شناسایی سیستم‌های خطی و غیرخطی
کنترل مولد سرو، دو بات
کنترل پیچیده، کنترل تطبیقی

2003

Subject:
Year. Month. Date. (9

• پردازش سیگنال

فیلتر
تخمین
شناسایی و تولید گفتار
تبدیل متن به گفتار
پردازش سیگنالهای حیاتی

• پیش بینی سری های زمانی

به صورت خطی
به صورت ترکیبی با روشهای آماری
بازار بورس
زلزله

• بهینه سازی

مسائل هندسی (یافتن یا سخ)
مسائل مدیریتی

• مدلسازی

اقتصادی و مالی
پیش بینی وضعیت اقتصادی
پیش بینی وضعیت سهام

• تشخیص پزشکی

تشخیص بیماری از روی علائم
بیماری های قلب و ریه و کبد و ...

دل هر ذره را که بشکافی
آفتابیش در میان بینی

فروش است ← حضارت دیان مجچه

Subject:
Year. Month. Date. ( )

آشنایی با سلسله های عصبی طبیعی

* مغز انسان
- بزرگترین سلسله زیست شناسی
- بافتی خاکستری ـ حدودی حدود ۱۵۰۰ گرم
- دو نیمه تقریباً مساوی با تقارن آینه ای
- چین و چروک های مشابه و محل های تقریباً مشابه ، شبیه جیغ های کردو
- تقسیم بندی مغز به نواحی مختلف ← بر مبنای همان چین و چروک ها

* بخش های مغز
- قطعه پیشانی ← تعلم ، یادگیری ، حافظه ، حفظ صفات رفتاری
- قطعه جانبی ← هماهنگی و کنترل اعمال پیچیده حسی و تراکم حرکتی ، مسئولیت
- قطعه گیجگاهی ← تعلم ، یادگیری ، حافظه و حفظ صفات رفتاری
- قطعه پس سری ← ناحیه بینایی
- مخچه ← کنترل حرکات ظریف بدن
- ساقه مغزی ← ارتباط مغز به نخاع ، رشته اعصاب اتصال با هم

* مخچه
- متقارن ← تقارن آینه ای
- متصل به ساقه مغز

* مخ و مغز
- دو نیمکره مغز = مخ
- سطح خارجی ← ۲-۴ mm ضخامت ← قشر مخ Cerebral Cortex
- قشر مخ ← مملو از نورون ← سلول عصبی ، بی بافت

Subject:
Year. Month. Date. ۱۵

• تعلم فقط در یک نیمکره ← در هر دو حضور دارد.
کدام نیمکره؟ چپ در اکثر قشری فعال ، در دیگری غیر فعال
در خانم ها برعکس !!!

• اعداد چپ دست ← نیمکره راست ← بالعکس

• کنترل نیمه راست بدن ← نیمکره چپ ← بالعکس

• ناحیه شنوایی مغزی تحریک شود (شیمیایی ، فیزیکی ، ...) فرد چیزی می شنود.

• در زیر تشریح Corpus Callosum ← جسم پینه ای
ارتباط میان دو نیمکره

• ساختارهای مغز از هم مستقلند اما در نقاطی مشخص به هم متصل هستند.
فضای بین ساختارها Ventricle ← زنجیره ای از غار ← بطن
پر از مایع مغزی - نخاعی ← نقش ساپورتی
شستشوی مداوم زائد

• ساقه مغزی ← محل عبور رشته های عصبی
ادامه نخاع

• در صورت از بین رفتن بعضی از مغز بدنش مجددی در کار نیست.
ولی برای جبران خسارت از قسمتهای غیر فعال استفاده می شود.

• مغز به مدت یک الکترون ولت خون داشته است ← چند ثانیه کمبود ← غش
چند دقیقه کمبود ← اغما

Subject:
Year. Month. Date. ( )

* نورون‌ها

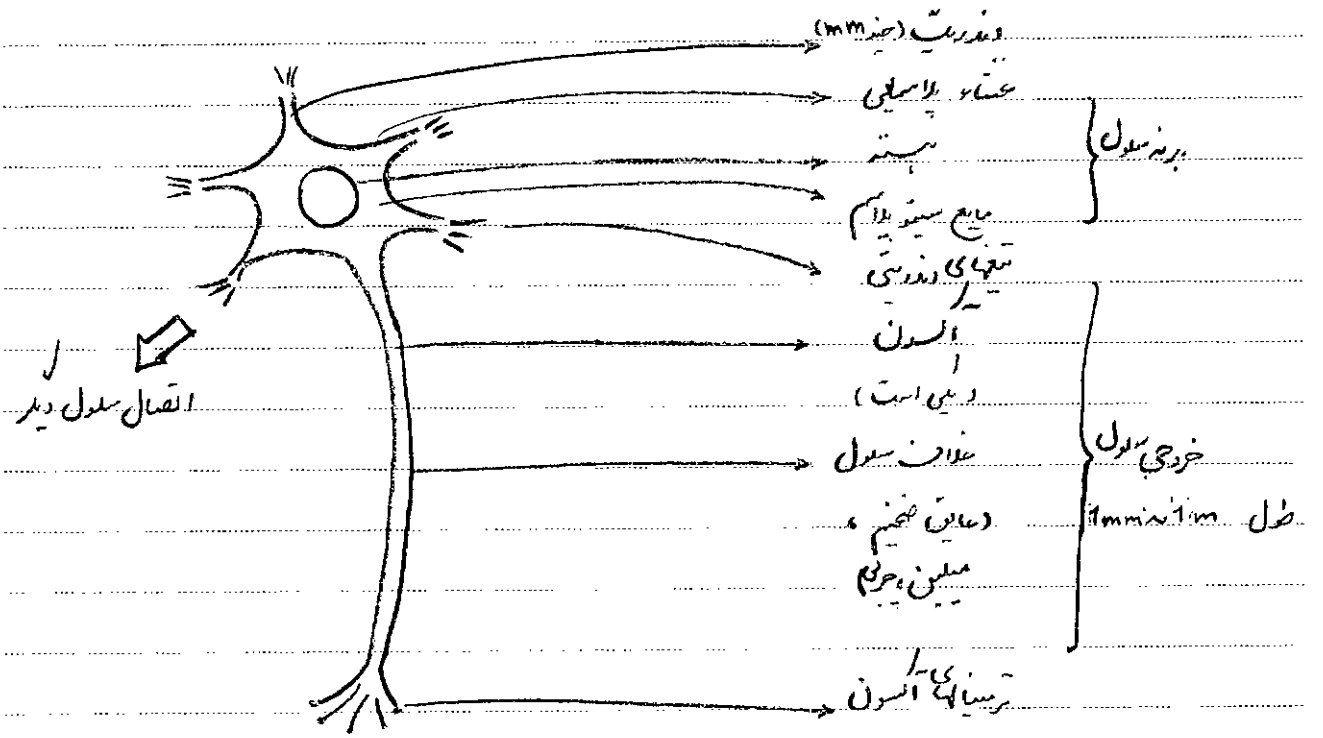
• ساز ماندهی در مجموعه‌های لایه‌ای ← ۶ لایه

• تعداد مجموعه‌های نورونی ← 10^{3} تا 10^{4}

• تعداد نورون در هر گرم از مجموعه‌ها ← 10^{7} تا 10^{8}

• تعداد کل نورون‌ها ← 10^{10} تا 10^{12}

• ساختار نورون



• مشخصات ← تقریبی ۱۰ تا ۱۰۰ میکرون

اجزای اصلی: بدنه + آکسون + دندریت

Dendrite Axon Soma

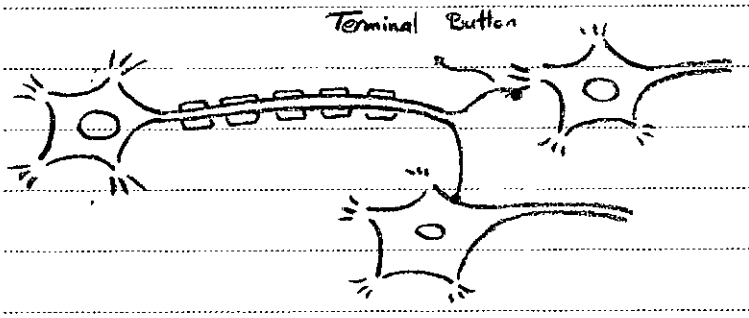
Subject:
Year. Month. Date. ۸۶

اشکال ← هرمی
ستاره ای
لوزی شکل
طناب شکل

• تعداد زیاد دندریت + حداکثر تشعب جانبی

• اتصالات بین نورون‌ها ← غیر انتهایی، دقیق، منظم
سیم کشی کاملاً حساب شده در سطح مغز
تعداد اتصالات
میلیارد $> 10^4$
تریلیون $> 10^{12} \sim 10^{16}$

سیناپس
اتصال آکسون به سلول بعدی
نوع اتصال ← با دندریت بعدی
با بدنه بعدی



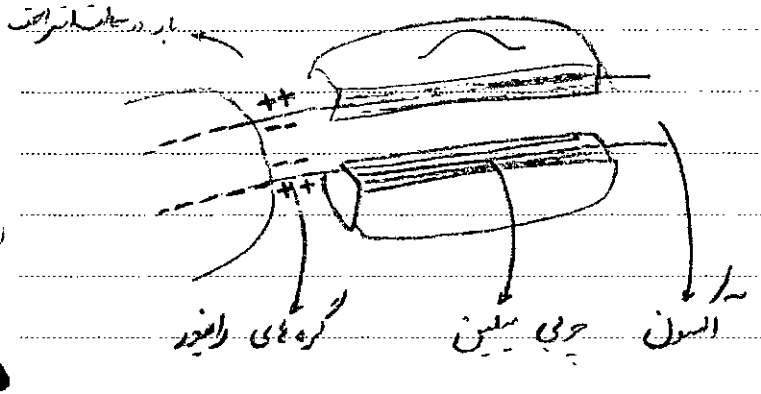
سلول پس سیناپسی سلول‌های پیش سیناپسی

Subject:
Year Month Date ( )

• دردل بدنه سلول ← میتوکندری
↙ ریبوزوم

• علاوه بر نورون‌ها در بافت عصبی سلول‌های دیگری هم هستند ← به نام گلیا Glia
↙ ۱۰ برابر تعداد نورون‌ها
↙ سازنده میلین

• عملکرد میلین
— افزایش سرعت انتقال سیگنال
— مانع نشتی جریان خروجی

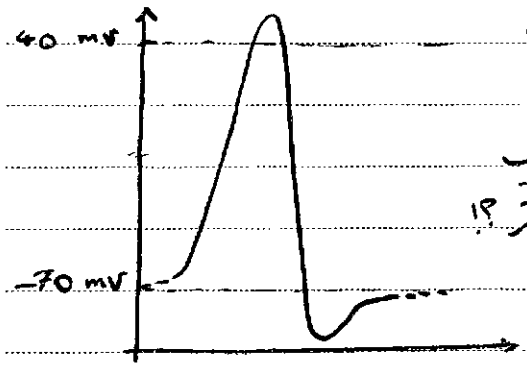


• خصوصیات الکتریکی
— دلایل ← وجود مولکول‌های پروتئین با بار منفی در داخل مایع سلولی
↙ توزیع نابرابر یون‌های Na^+ و K^+ در دو سوی غشاء

— اختلاف پتانسیل ← در طرف غشاء ← 10^{-8} متر = 10^{-6} cm
↙ ← mV $V_c \sim 100$ ← پتانسیل استراحت
↙ معادل ۱۰kV در سر یک cm

• تحریک نورون ← مکانیکی، شیمیایی، الکتریکی
↙ اختلاف پتانسیل ← موج، ضربه
↙ پتانسیل فعالیت یا کار

Subject:
Year. Month. Date. ۸۶



شکل و اندازه تانسیل فعالیت
- ثابت
- مستقل از تحریک
- تحریک قوی‌تر ← ضعیف‌تر ؟؟

تحریک قوی‌تر

← دنباله ضربه ← هیپرپلاریزاسیون طولانی
← قوی‌تر ← طولانی‌تر

• چگونگی ایجاد تانسیل فعالیت

- گیرنده ← دندریت‌ها ← شکل درخت مانند
← ابتدای سلول ← به سمت بدنه سلول

- بدنه سلول ← ابتدای آکسون
← حد آستانه

- رسیدن به حد آستانه ← شلیک ضربه
← انتقال در آکسون ← سریع توسط میلین

- آکسون ← پس از ضربه ← دوره تحریک ناپذیری مطلق
← تا حدود 1ms

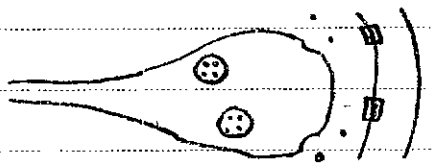
• انتقال تانسیل فعالیت بین نورون‌ها
در سیناپس چه اتفاقی می‌افتد ؟

Subject:
Year. Month. Date. ( )

— شکاف سیناپسی ← در محل اتصال آکسون و دندریت

— مراحل

۱- رسیدن پتانسیل عمل به سیناپس ← کیسه‌های مواد انتقال دهنده به سمت شکاف می‌روند
کیسه‌های مواد حفره را تخلیه می‌کنند.



مواد تخلیه شده به دیواره شکاف نفوذ می‌کنند
این مواد انواع مختلف وجود دارد.
← در سیر فرستادن یک نوع ماده

۲- افزایش کلسیم ورودی ضربه ← نوع آزاد شدن انتقال دهنده ↑

۳- مولکول‌های گیرنده ← در یافت ماده انتقال دهنده ← فعال شدن
ساختار عضله و وضعیت قطبی سلول پس سیناپسی تغییر می‌کند

— مواد انتقال دهنده: تحریک / دارنده / Excitatory

* پتانسیل سرعت سلول دوم ← کاهش می‌یابد.
* کاهش اختلاف پتانسیل ← Depolarization
* آماده‌تر شدن سلول برای شلیک شدن ← کم شدن مانع
* مقدار کم ماده منتقل کننده ← دپلاریزاسیون کم ← پتانسیل دارنده کم

EPSP : Excitatory Post Synaptic Potential

Subject:
Year. Month. Date. ۱۳

* اثر دو عامل سیناپس یا سیناپس‌های مجاور قبل از رسیدن به اثر EPSP اولیه از بین برود
یک EPSP دومی، سومی و... ایجاد شود، اثر آن‌ها با هم جمع می‌شود.
* مجموع به حد آستانه‌ای برسد ← فعال شدن سلول دوم

* نوع ماده منتقل کننده در ناحیه پس سیناپسی ← حالت ثابت سلول ← حد اثر تحریک
پلاریزاسیون ثابت
ماده حل طبیعی
آنزیمی برای شکستن ماده منتقل کننده

← مواد انتقال دهنده: بازدارنده / Inhibitory

* می خواهد حدی فعال شدن سلول بعدی را تغییر دهد. ← ایجاد وقفه در سلول بعدی

* اثر این پلاریزاسیون سلول پس سیناپسی ← Hyper polarization
تابش کادمی برای شلیک

* پتانسیل ایجاد شده: پتانسیل بازدارنده پس سیناپسی جزئی
IPSP: Inhibitory Post Synaptic Potential

۴- شلیک شدن یا شدن سلول دوم ← مجموع عام رخدادهایی که در هر لحظه به سلول می‌رسد.
بالاخص در جایی که به بدنه اکسون وارد می‌شود.

← سهم همه سلول‌های متصل در شلیک شدن یک سلول یکسان نیست.

* یک سلول + اندیت سلول دوم ← هزاران سیناپس

Subject:
Year. Month. Date. ( )

— عوامل مؤثر در تعادل مدل سلول

* تعداد سیناپس‌های ایجاد شده
* فاصله سیناپس نسبت به سلول پس سیناپسی
* طول دندریت‌ها
* محل اتصال دندریت‌ها به بدنه — اتصالات نزدیک آکسون مؤثرترند

— ابتدا ماده منتقل کننده در شیار ویرسیدن برلیزده ۴

تجزیه ماده‌ای سبب منتقل کننده در برلیزده ۴ بدا می‌شوند ولی با آنزیم‌های سلول از بین رود.

غیرفعال یا معدوم شدن آنزیم‌های منهدم کننده

غیر فعال شدن سیناپس

☑ سم کورار ← انسداد گیرنده ← مانع عضلانی ← فلجی

☑ گازهای عصبی ← از بین رفتن فعالیت استراز برلیزده ← مواد از بین نمی‌رود ← تهیج و انقباض دایم عضله ← مرگ در دقایق

☑ نیکوتین، حشیش، LSD ← داخل یا کنار سیناپس

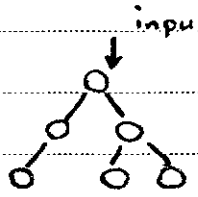
Subject:
Year. Month. Date. ۱۴

* جریان اطلاعات

• مربوط به تئوری و سلولی بهم بندی نرون‌ها ← حالتهای متعدد مغناطیس نرونی

• چهار شکل

۱- مدل واگرا Diverging



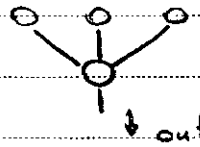
ساختار درختی

جریان رو به پایین

یک نرون پیش سیناپسی باعث تحریک چندین نرون پس سیناپسی می‌شود.

☑ اثر بر سلولهای حسی در مغز

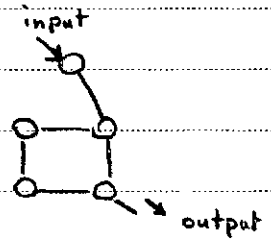
۲- مدل همگرا Converging



چندین نرون پیش سیناپسی باعث تحریک چندین نرون پس سیناپسی می‌شود.

☑ نرون Motor ورودی از دسته‌ای مختلف مغز می‌گیرد

۳- مدل نوسانی Reverberating . Oscillating



تحریک یک نرون پس سیناپسی در چندین نرون

پس سیناپسی اثر می‌گذارد.

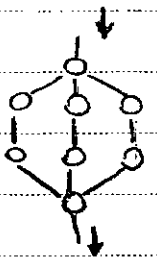
ساختار مراف

☑ حرکتهای عضلانی با هدف کرامی مانند تنفس و ضربان قلب

Subject:
Year. Month. Date. ( )

۳- Parallel After Discharging

مدارات بدازی پس از تخلیه



یک نوبل پیش سینالیسی یک سری نوبل پس سینالیسی به یک نوبل نهایی جهتی می‌شوند با تحریک می‌کند.

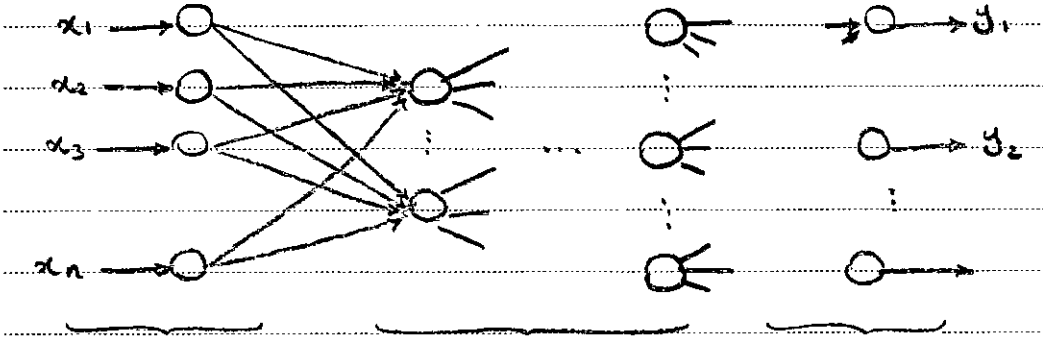
☑ ارسال بدازی بسیوی ضربات سریع

Subject:
Year. Month. Date. ۱۵

معرفی شبکه‌های عصبی مصنوعی

* تعریف : ساختار پردازشی توزیع شده مداری به شکل یک گراف جهتدار

* ساختار کلی



لایه ورودی ($\underline{X}$) لایه‌ی مخفی لایه خروجی ($\underline{Y}$)

• لایه ورودی : وارد شدن داده‌ها به آن ← هیچ کاری روی سیگنال انجام نمی‌دهد

$\underline{X}$: $x_1, \ldots, x_n$ $\longrightarrow$ $\dim(X) = n \times 1$

• لایه خروجی : خروجی‌های شبکه از این لایه گرفته می‌شود

$\underline{Y}$: $y_1, \ldots, y_m$ $\longrightarrow$ $\dim(Y) \neq \dim(X)$

$= m \times 1$

• لایه‌های مخفی : نه از ورودی دیده می‌شوند و نه از خروجی

تعداد آن از صفر به بالا

• هر گراف : نودها ، اتصال پردازشی

هر تعداد ورودی + تنها یک خروجی

Subject:
Year. Month. Date. ( )

خروجی هر گره می تواند به تعداد گره متصل شود.
سیگنال این اتصالات همه یکسان و برابر با سیگنال خروجی

- هر یال گراف : دارای جهت
دارای وزن ← میزان تاثیر گره مبدا روی گره مقصد
- گره ها رابه لایه ها تقسیم می کنند
- هر واحد می تواند به واحدهای لایه خود یا لایه های دیگر وصل شود.
- واحدهای یک لایه عموما" شبیه در نظر گرفته می شوند.
- همه سیگنال ها در اتصالات به صورت لحظه ای و بدون تاخیر در نظر گرفته می شوند.

- عمل واحدها ← همزمان ← در زمانهای خاصی فعال می شوند ← واحد زمانبندی
  - ستردن ← خروجیهای جدید ← ورودیها
  - ایستردن ← وضعیت قبلی واحدها
  - غیرهمزمان ← پیوسته در حال فعالیت
  - آستردن ← خروجیهای جدید ← با دریافت ورودیها
  - پیوسته

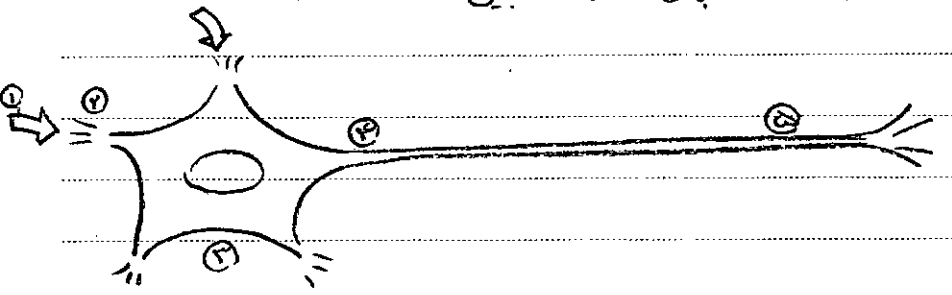
- معمولا" طبقه واحدهای هر شبکه یا همزمان هستند یا غیرهمزمان
- زمانهای اتصالات در یک فاز آموزش یا یادگیری معین می شوند. ← با استفاده از نمونه هایی از مسئله به شبکه نشان می دهیم.

Subject:
Year. Month. Date. ۱۶

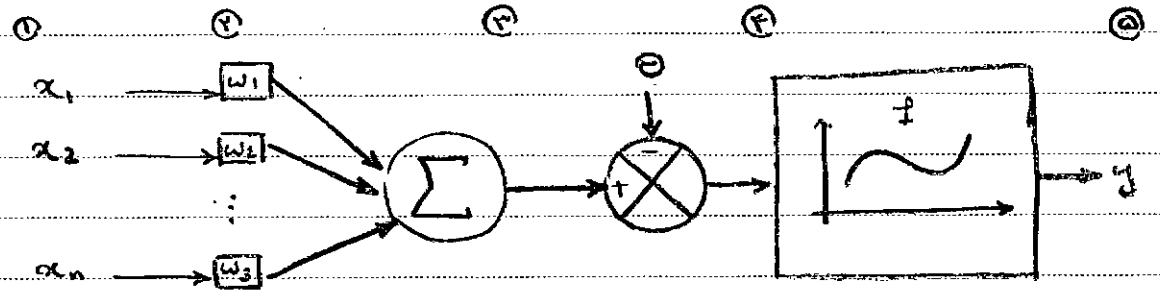
• آموزش ← یک قاعده یادگیری
شبکه‌های متقدم ← ورودی - خروجی
تعیین وزن‌ها

* واحدهای پردازش سلسله

• مدل McCulloch & Pitts برای نورون‌های طبیعی (۱۹۴۳)



ورودی‌ها → وزن سیناپسی → جمع تحریکات ورودی‌ها → تابع فعالیت → خروجی



$w > 0$ تحریک
$w < 0$ بازدارنده
$w = 0$ بدون اتصال

1 بله
0 خیر

$$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right) = f\left(\sum_{i=0}^{n} w_i x_i\right)$$

$x_0 = 1$
$w_0 = -\theta$

Subject:
Year. Month. Date. ( )

ملاحظات شبکه عصبی طبیعی (تفاوت، مدل، پیچیدگی)

— پاسخ پیوسته ← نه بی‌زمان

— مجموع غیر خطی از ورودی‌ها

— خروجی، دنباله ضربه ← نه سطح ثابت

— تأخیر خیلی کم ← میزان داده استفاده شده متغیر است ← غیرقابل پیش‌بینی

به‌هرحال، نشان داده شده است که، انتخاب مناسب وزن‌ها، ترکیبی از نورون‌ها، عملکرد بی‌زمان می‌تواند مسائل محاسباتی خوبی را حل کند.

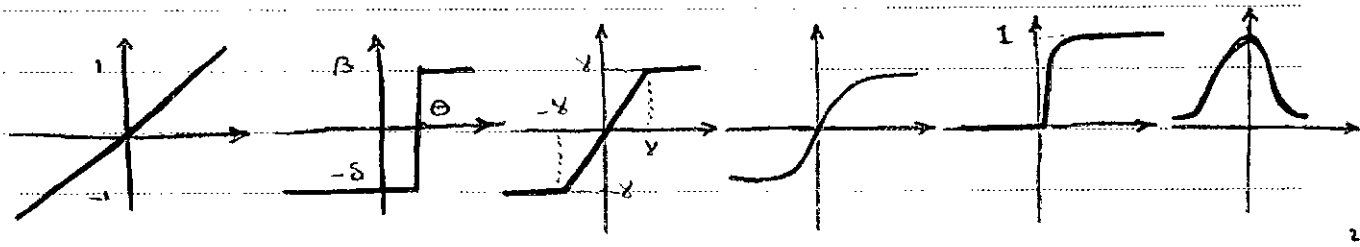
- واحدهای پردازش اطلاعات
- تعمیم مدل نورون به سطح‌های مختلف

۱- تغییر تابع فعال شدن Activation - Squashing - Transfer - Gain

توابع مثال

تابع پله : مک کالاک

خطی	پله‌ای	سیگموئیدی تقریب تکه‌ای خطی	نسبت مربع	هایی



$f(x) = \alpha x \quad \alpha \neq 0$

$f(x) = \begin{cases} \beta & x \geq \theta \\ -\delta & x < \theta \end{cases}$

طایف: 1, 0
سریعتر (دارای مجازات): -1, +1
sgn(x)

$f(x) = \begin{cases} +\gamma & x \geq \gamma \\ x & |x| < \gamma \\ -\gamma & x \leq -\gamma \end{cases}$

$f(x) = \tanh\left(\frac{\alpha x}{2}\right)$

$f(x) = \begin{cases} \frac{x^2}{1+x^2}, & x > 0 \\ 0 & , o.w \end{cases}$

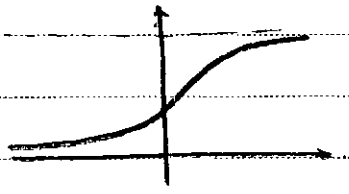
$f(x) = e^{-\frac{x^2}{\sigma^2}}$

سریعتر اشباع می‌شود

☑ تابع سیگموئیدی

$$f(x) = \frac{1}{1+\exp(-\alpha x)}$$

$\alpha > 0$, $\alpha \approx 1$

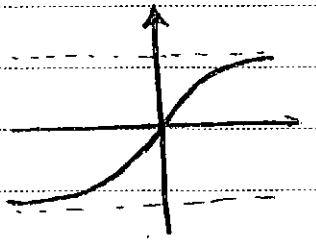


مشتق پذیر : $f'(x) = \alpha f(x)[1-f(x)]$

تابع سیگموئید دوقطبی (لاجستیک)

تابع زوجی

$$f(x) = \frac{2}{1+\exp(-\alpha x)} - 1 = \frac{1-\exp(-\alpha x)}{1+\exp(+\alpha x)} = \tanh(\alpha x/2)$$



$$f'(x) = (\alpha/2)[1+f(x)][1-f(x)]$$

☑ تابع خاص

برخلاف سایر توابع استفاده می شود.

۲- تابع فعالیت متفاوت برای هر لایه

می توان تابع فعالیت برای هر لایه و نیز متفاوت و نظیر ثبت ← رابج مثبت

۳- ترکیب غیر خطی ورودیها

☑ نرونهای درجه بالاتر $y = f\left(\sum_{i=0}^{n} w_i x_i + \sum_{k=1}^{n}\sum_{\ell=1}^{n} w_{k\ell} x_k x_\ell + K\right)$

قویتر از نرونهای خطی

آموزش دشوارتر ← تعداد وزنها ↑ ← محاسبات ↑ ← پیچیدگی ↑

Subject:
Year. Month. Date. ( )

۴ـ واحدهای غیرهمسان

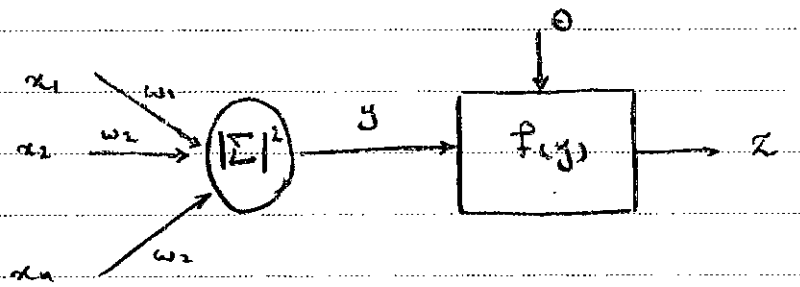
هدف از آموزش واحد ← انتخاب آستانه

شبکه‌های پرسپترون

۵ـ منطق فازی

- ورودیهای فازی
- پارامترهای واحد فازی
- فازی در نظر گرفتن ورودیها و پارامترهای واحد

۶ـ نورونهای دارای وزنهای مختلط



$$\omega_i = a_i + jb_i$$

$$y = |\omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n|^2 = (a_1 x_1 + a_2 x_2 + \dots + a_n x_n)^2 + (b_1 y_1 + \dots + b_n y_n)^2$$

نورون مختلط یک نورون درجه ۲ است.

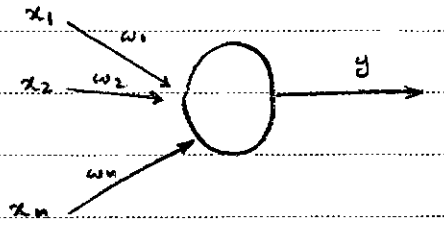
$$z = f(y) = \frac{1}{1 + e^{-\alpha(y-\theta)}}$$

آستانه شلیک ؟

Subject:

Year. Month. Date. ۸۸

۷ - نورون‌های سیگمایی



مجموع حاصلضرب‌های ورودی‌ها

$$y = \sum_k \omega_k \prod_i x_i$$

نورون غیرخطی

۸ - نورون‌های ضربه‌ای

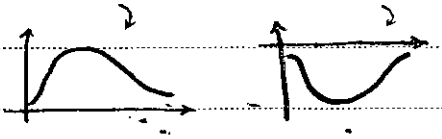
Spiking

پی‌اند مدل‌ها

خروجی وقتی فعال می‌شود که پتانسیل نورون به یک حد آستانه‌ای θ_v برسد.

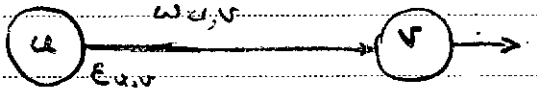
p_v

$$p_v = \sum (EPSP) + \sum (IPSP)$$



شلیک سلول پیش سیناپسی u در زمان s

به پتانسیل p_0 در زمان t به اندازه



$$\omega_{u,v} \times \varepsilon_{u,v}(t-s)$$

اثر می‌گذارد.

$\omega_{u,v}$ می‌تواند متغیر با زمان باشد.

نورون پیش سیناپسی پتانسیل پس‌سیناپسی را در زمان t حداکثر حدود چند mV به طور تدریجی تغییر می‌دهد.

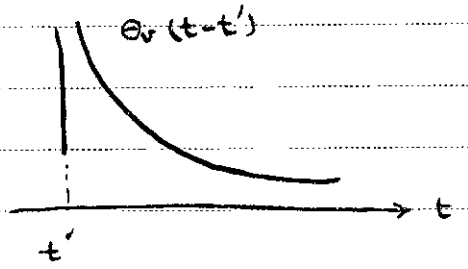
پتانسیل استراحت : -70 mV

آستانه شلیک : $+50$ mV

Subject:
Year. Month. Date. ( )

نورونی که در زمان t' تحریک کرده برای مدت کوتاهی قادر به تحریک نیست.

این مدل با تابع آستانه‌ای $\theta_v(t-t')$



تابع پتانسیل نورون v

$$f_v(t) = \sum_u \sum_{s \in F_u} \omega_{u,v} \cdot \varepsilon_{u,v}(t-s)$$

وابسته به زمان

F_u : مجموعه زمان‌های تحریک نورون u

بررسی $\theta_v(t-t') - f_v(t)$ نورون تحریک می‌شود.

۹- نورون‌های التقاطی Stochastic

فعال شدن یا نشدن : احتمالاتی

y : خروجی نورون ⟵ $P(y)$: احتمال فعال شدن

$$y = \begin{cases} +1 & p(I) \\ -1 & 1-p(I) \end{cases} \qquad P(I) = \frac{1}{1+e^{-I/T}}$$

T : پارامتر کنترل سطح نویز یا میزان عدم قطعیت

I : Net Input ⟵ $I = \sum \omega_i x_i$

Subject:
Year. Month. Date. ۸۹

* دسته بندی شبکه های عصبی به سه نسل

نسل 1
خروجی دو قطبی
بر اساس مدل مک کالاک و پیتز
پرسپترون ، ADALINE ، برانزمان

نسل 2
خروجی پیوسته
توابع منطقی با تعداد واحدی کمتر نسبت به نسل اول پایه سازی می شود.
MLP ، RBF ، بازگشتی

نسل 3
خروجی ضربه
نزدیکتر به نورونهای طبیعی

* رفتار دینامیکی واحد

$$\frac{dy}{dt} = -g(y) + \sum_{j=0}^{n} w_i x_i = -g(y) + I$$ تغییر خروجی با زمان

x_i : سیگنال ورودی سیستم ← مجموعی خطی ضربات وزنیت شده
y_i : متغیر سیستم ← فعالیت نورون

در حالت تعادل :

$$\frac{dy}{dt} = 0 \longrightarrow g(y) = I \longrightarrow y = g^{-1}(I)$$
$$\longrightarrow y = g^{-1}(\sum_{j=0}^{n} w_j x_j) \longrightarrow y = f(\sum w_j x_j)$$

Subject:
Year. Month. Date.

$f(x) = \tanh(\beta x)$ ☑

$g(x) = \frac{-1}{2\beta} \ln \frac{1-x}{1+x}$

× اتصالات شبکه

- یال جهت‌دار ← نقطه ← جهت جریان اطلاعات
- یال وزن‌دار ← وزن = میزان تأثیر ← معین شدن در فاز یادگیری یا آموزش
- طراحی شبکه عصبی: صفر یا یک ← در هر لایه چند نورون ← معین وزن‌ها ← آموزش ...
- نوع داده‌ای وزن‌ها: صحیح، مختلط، کاری، ...
- $\begin{cases} w > 0 & \text{تحریک، دارنده} \\ w = 0 & \text{بدون اتصال، بی‌تأثیر} \\ w < 0 & \text{ممانعت‌کننده، بازدارنده} \end{cases}$
- نمایش اتصالات با ماتریس اتصالات

$M_{n \times n} = [m_{ij}]$

$m_{ij} = \begin{cases} 0 & \text{اتصال از واحد } i \text{ به واحد } j \text{ وجود ندارد.} \\ 1 & \text{وجود دارد.} \end{cases}$

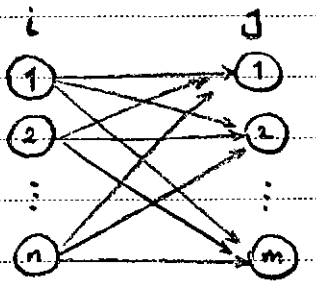
صفحه ← تداوم حافظه

شخص بودن وزن اتصالات ← ماتریس وزن شبکه

Subject:
Year. Month. Date. (۵)

• ماتریس وزن شبکه نوع ۱ : ماتریس ← مورد استفاده در این درس

هر لایه شبکه به یک لایه ← اتصالات یک لایه به لایه بعد را نشان می دهد



$W_1 = [w_{ij}]$ ← w مبدا ← مقصد

w_{ij} = وزن اتصال از واحد i به واحد j

• ماتریس وزن شبکه نوع ۲ : همیلن و کامپت نیلسن

اتصالات یک لایه از لایه قبل را نشان می دهد.

$W_2 = [w_{ji}]$ ← w مبدا ← مقصد $\quad : W_2 = (W_1)^T$

w_{ji} = وزن اتصال به واحد j از واحد i

• انواع اتصالات بین واحدها

۱- اتصالات کامل Full
از هر خروجی به تمام واحدهای لایه قبل

۲- اتصالات تصادفی یکنواخت Uniformly Random
هر واحد تعداد معینی ورودی از لایه قبل
انتخاب اتفاقی و با توزیع یکنواخت ← نحوه انتخاب بهتر در نتیجه

Subject:
Year. Month. Date. ( )

۳- اتصالات یک به یک
هر راحد یک اتصال از لایه قبل

۴- اتصالات با تجمیع خاص (اضافی)
برقراری اتصالات به تجمیع خاصی وابسته اند ← تجمیع گیری

* معماری شبکه‌ی عصبی

Feed Forward
• بدون بازخورد ، پیش‌خور ، جلوسو

- لایه : فاقد سیناپس
- بدون نورون ← اسکالر ← پاسخ به ورودی تنها یک دسته خروجی ایجاد می‌شود (ثابت)
- پاسخ مستقل از حالت قبلی
- بدون حافظه ← حالت قبلی نشتی ندارد

Recurrent
• دارای بازخورد ، بازگشتی

- لایه : دارای سیناپس
- دارای نورون ← دینامیک ← پس از اعمال ورودی ها ، خروجی مستقل می‌شود (پویا) سپس از طریق سیناپس به ورودها اعمال می‌شود و خروجی تغییر می‌کند...
- اعمال ناپایداری ← به علت وجود نورون
- عدم همگرایی خروجی ← به ورودی خوب!
- پیچیده‌تر ← تویر

Subject:
Year Month Date ۲۱

Classification دسته بندی

* مقدمات

• بسیاری از ادراکات انسان ← از طریق دسته بندی
↙ شناسایی بیش

• یادگیری تعلق اشیا با استفاده از ویژگی های تشخیص از دسته ← کار classifier
↙ یادگیری + دسته بندی

• پس از یادگیری، دسته بندی کننده با دریافت هر ورودی جدیدی تواند تصمیم بگیرد که این ورودی به کدام دسته تعلق دارد.

• حل دسته بندی ≡ ناحیه بندی فضای ورودی توسط دسته بندی کننده

* نام های دسته بندی

۱- آموزش / یادگیری
ورودی : ویژگی + برچسب صحیح ویژگی
خروجی : پارامترهای دسته بندی کننده

۲- آزمایش / کار
ورودی : ورودی جدید
خروجی : شناسایی، تعیین دسته
ثابت : پارامترهای دسته بندی کننده

* ورودی ها و خروجی های دسته بندی کننده از روی مسئله تعیین می شود.

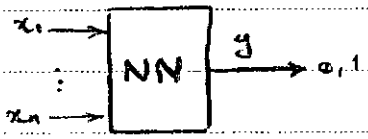
Subject:
Year. Month. Date. ( )

انواع مسائل دسته‌بندی

- دو دسته‌ای

تعلق / عدم تعلق ← نرده

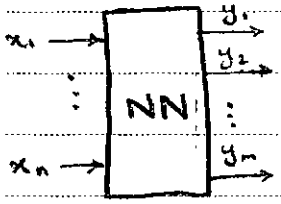
پیاده سازی با شبکه عصبی : یک خروجی



- چند دسته‌ای

m دسته ← تعلق به نرده 1

تعلق به نرده m



پیاده سازی با شبکه عصبی ← m خروجی

یک خروجی یک دیتیه صفر

☑ دوزشکار : وزنه بردار یا بسکتبالیست ؟ (حتماً یکی از این دو)

هدف : ساختن دسته‌هایی برای یکی از این دو

معماری : یک نورون ← یک خروجی (الجبار سبد)

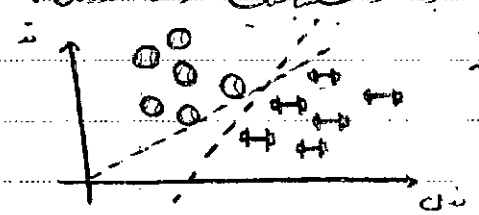
ورودی‌ها : انتخاب دیزنی ← چه چیزی تمایز ایجاد می‌کند ← کدام بهتر است؟

قد ، وزن ، اندازه اورکم ... (الجبار سبد)

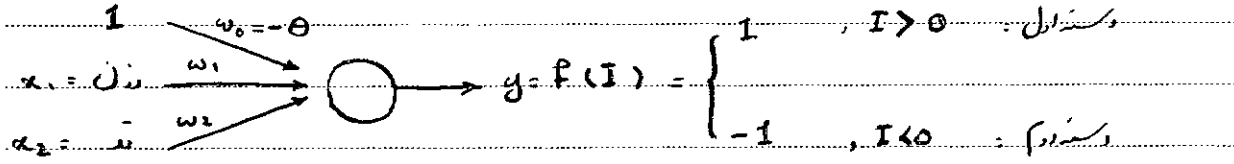
آموزش : اندازه‌گیری قد و وزن تعدادی وزنه بردار و بسکتبالیست (داده ورودی)

تعیین خط (یا مرز) برای تقسیم صفحه

آموزش ← ناحیه‌بندی صفحه



حل مسئله:



$$y = f(I) = f\left(\sum_{i=0}^{2} w_i x_i\right)$$

$$\underline{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \qquad \underline{W} = \begin{bmatrix} -\theta \\ w_1 \\ w_2 \end{bmatrix} \Rightarrow I = -\theta + w_1 x_1 + w_2 x_2$$

$\underline{W}$ در فاز آموزش تعیین می‌شود

خط جداکننده ← مقادیری x_1, x_2 که I را مثبت می‌کنند: دسته اول
مقادیری x_1, x_2 که I را منفی می‌کنند: دسته دوم

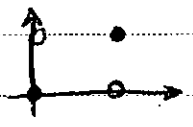
$I = 0$

$-\theta + w_1 x_1 + w_2 x_2 = 0 \longrightarrow x_2 = -\frac{w_1}{w_2} x_1 + \frac{\theta}{w_2}$ (معادله خط)

* خطی جداپذیر: اگر فضای مسئله به گونه‌ای باشد که همه نمونه‌های یک دسته در یک طرف خط و نمونه‌های دسته دیگر در طرف دیگر خط باشند.

Linearly Seperable

مینسکی و پاپرت نشان دادند فقط مسئله‌ای که به صورت خطی جداپذیر باشد با واحد پرسپترون قابل حل است.



XOR ☑

خطی جداناپذیر ← غیر قابل ساده‌سازی با پرسپترون

Subject:
Year. Month. Date. ( )

* واحدهای پرداز نیلر

۱- واحد هب Hebb

- قدیمی ترین - ساده ترین
- نظریه روانشناسی ← هب
- وزنهای طبیعی برای نورون i به طور یکه و بدان که نورون j و نورون i شرکت کند، میزان تاثیر این نورون بر فعال کردن نورون j افزایش می یابد.

← وزن

- قانون یادگیری هب ← فعال شدن نورون j برقرار است با فعال شدن i

Coincidence ← یادگیری هم وقوعی

- الگوریتم

I) مقدار اولیه وزنها تخصیص می شود.

$w_i = 0 \quad , \quad i = 0, \dots, n$

II) برای هر زوج ورودی $\langle s, t \rangle$ مرحله های III و IV انجام شود.

$\langle s, t \rangle$ ← زوج آموزشی

s ← ورودی آموزش شبکه source

t ← مقدار مطلوب خروجی target

} فضای آموزش

III) به واحدهای ورودی مقادیر ورودی داده شود.

$$\begin{cases} x_i = s_i & , i = 1, \dots, n \\ y = t \end{cases}$$

IV) وزنها به صورت زیر اصلاح می شود.

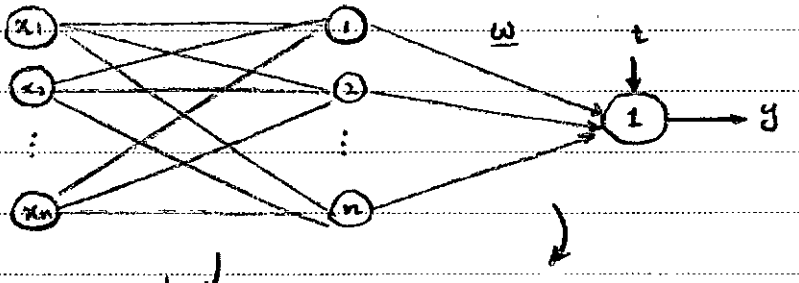
$$w_i(new) = w_i(old) + x_i \cdot y \quad , i = 0, \dots, n$$

Subject:
Year. Month. Date. ۲۳

- برای تغییر وزن ، هر دوی x_n و y باید غیر صفر باشند
- در سال ۹۸ ، مینسکی و پاپرت قانون جدید را ارائه دادند ← پرسپترون خطی بودن یا غیر خطی بودن ارتباط آنها باید تقویت شود.

۲- واحد پرسپترون Perceptron

- دو زبانه
- Perception + Electronics
- اولین دستگاه Mark I Perceptron ←
  - شبیه سازی شبکه ی اعصاب
  - قابلیت برای شناسایی حروف
  - آرایه ۲۰ × ۲۰ از عناصر حسگر ← ۴۰۰ سیگنال
  - شبکه عصبی سه لایه
  - تنظیم وزنها ← ۵۱۲ تا پتانسیومتر متغیر به صورت الکتریکی
  - اتصالات جدید به صورت قابل دسترسی



تنظیم وزنها با قانون پرسپترون وزنها بطور اتفاقی بین یکی از سه مقدار: 0، 1، -1

- شبکه دسته بندی دو دسته ای : دسته A (y=0) ، دسته B (y=1)
- t مقدار مطلوب آستانه ← $\langle s, t \rangle$ ← $s \cdot X$
- چون برای لایه اول مقدار مطلوب مشخص نیست ، مقدار آنها اتفاقی انتخاب می شود.

Subject:
Year. Month. Date. ( )

• قانون یادگیری پرسپترون

$$\underline{W}(new) = \underline{W}(old) + (t-y)\,\underline{X}$$

• اثبات می‌شود که این الگوریتم در تعداد مراحل محدود، برزین دو دسته‌ای که خطی جدایی پذیر هستند را پیدا می‌کند. ← با هر نقطه اولیه

• اصلاح وزن

$$t-y = \begin{cases} 1 \rightarrow & B = \text{صحیح} \quad A = \text{اعلام} \\ 0 \rightarrow & \text{صحیح} = \text{اعلام} \\ -1 \rightarrow & A = \text{صحیح} \quad B = \text{اعلام} \end{cases}$$

• نسخه دیگری از الگوریتم پرسپترون ← میتوانی دارید

$$\underline{W}(new) = \underline{W}(old) + \alpha(t-y)\,\underline{X}$$

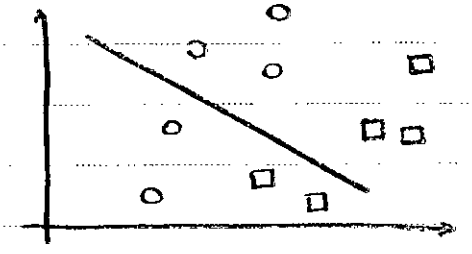
$0 < \alpha < 1$

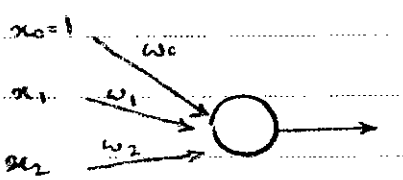
α ☑

$$\alpha = \frac{\underline{X}\cdot\underline{W}}{\|\underline{X}\|^2} \quad \text{یا} \quad \alpha = \frac{1}{\|\underline{X}\|}$$

دقتی که سطح اصلاح برداشته می‌شود مناسب است با ورودی

☑ پرسپترون درجه ۱ : جداسازی



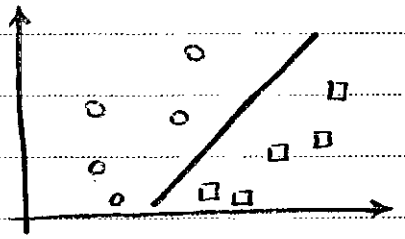


$$I = -\theta + w_1 x_1 + w_2 x_2$$

Subject:

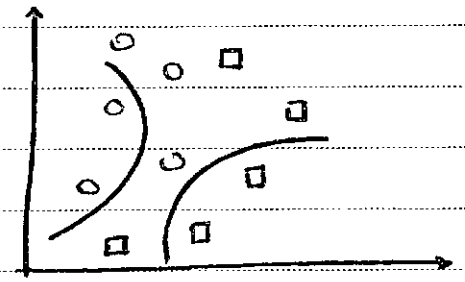
Year. Month. Date. ۲۴

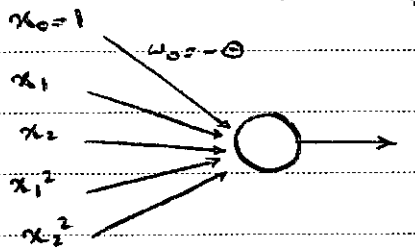
باید برای W مقدار اولیه ای در نظر داد ← خطی در صفحه مشخص می کند.



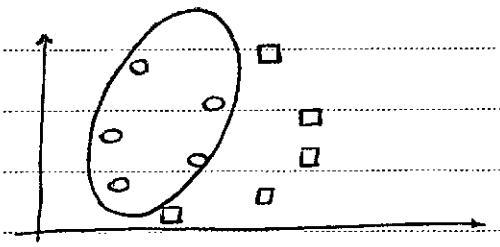
جواب یکی : وقتی برای هیچ مقداری وزنها تغییر نکند.

پرسپترون درجه ۲





$$I=-\theta+w_1x_1+w_2x_2+w_3x_1^2+w_4x_2^2$$



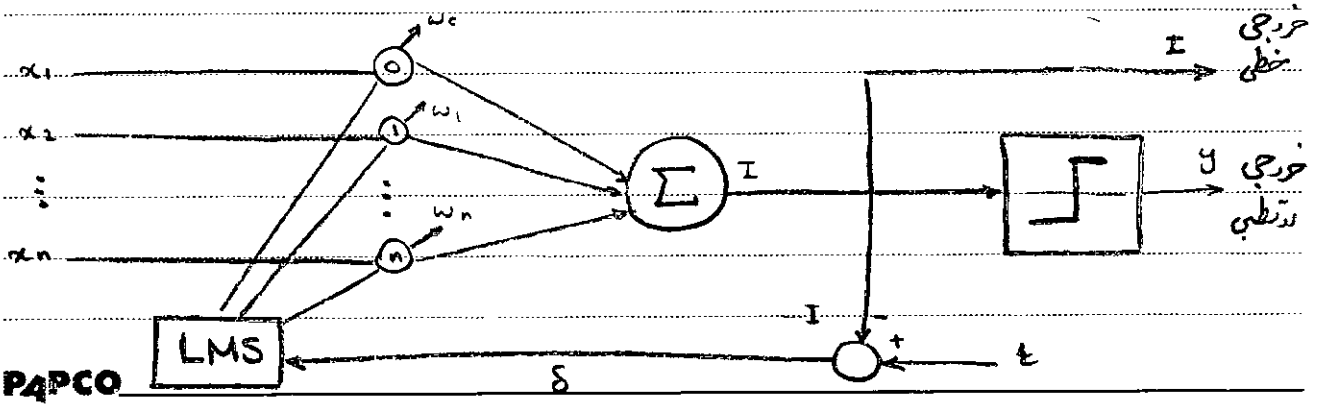
خطی جدا ناپذیر

۳ـ واحد آدالاین

Adaline

• ویدرو ۱۹۵۹

• Adaptive linear



x Least Mean Square W خطای خطی

Subject:
Year. Month. Date. ( )

• پیاده سازی اولیه ← آرایه‌ای از سلول‌ها، تعاریف و دستگاههای ذخیره‌ای

• ورودی‌ها ← بایزی / پیوسته
وزن‌ها ← مثبت / منفی ← حقیقی

• تعیین وزن‌ها ← LMS ← کمینه کردن متوسط خطی
↙ جذاب‌ترین

$$I = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n = \underline{X} \cdot \underline{W} = \underline{X}^T \underline{W} = W^T X$$

$$y = \text{sgn}(I)$$

• تابع ماژیری آدالاین ← تابع‌بی‌بهینه با شروع از یک نقطه
↙ کمینه یا بیشینه کردن تابع هدف
↓ جزء توانین Performance Learning

• فرم ریاضی

$\underline{X}$ ⟶ تعداد : N ⟶ بعدها : p

$\underline{W}$ ⟶ $F(\underline{W}) = \min$

$$F(\underline{W}) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} (t_k - I_k)^2 = E\left[(t_k - I_k)^2\right]$$

$$F(\underline{W}) = E\left[(t_k - I_k)^2\right] = E\left[(t_k - \underline{W}^T \underline{X}_k)^2\right]$$
$$= E\left[t_k^2 - 2 t_k \underline{W}_k^T \underline{X}_k + \underline{W}^T \underline{X}_k \underline{X}_k^T \underline{W}\right]$$
$$= E[t_k]^2 - 2 \underline{W}^T E[t_k \underline{X}_k] + \underline{W}^T E\left[\underline{X}_k \underline{X}_k^T\right] \underline{W}_k$$
$$= P - 2 \underline{W}^T \underline{Q} + \underline{W}^T R \underline{W}$$

P: عدد ثابت ، $\underline{Q}$: بردار ، R: ماتریس

Subject:
Year. Month. Date. ۲۵

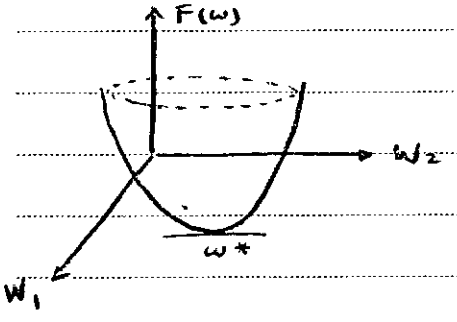
F برای کمینه کردن ← بدنه تعیین w ← P, Q, R از روی داده‌های قابل محاسبه است.

F(w) ← نرم درجه 2 از w

R ← متقارن

$\forall x : \sum\sum r_{ij} x_i x_j \geq 0$ ← مثبت نیمه معین

ضرایب منفی نمی‌شوند ← تقریب درجه بالا ← دارای کمینه : w^*



پیدا کردن w^* ← هدف

$\nabla_w F(w) = -2Q + 2Rw = 0 \longrightarrow$

$Rw^* = Q \longrightarrow w^* = R^+ Q$

R^+ ← معکوس تعمیم یافته / شبه معکوس

R معکوس پذیر

$R^+ = R^{-1}$

w^* ~ یکتا $\Longrightarrow$ همواره ~ سطح خطا

R معکوس ناپذیر

ناهموار ~ سطح خطا

w^* ~ بهترین کمینه برای تحقیقات ~

Subject:
Year. Month. Date. ( )

• دشواری پیدا کردن R و Q ← پیشنهاد ویدرو و هاف

w^* را با شروع از یک نقطه اولیه و پایین آمدن روی سطح عملکرد پیدا کنیم.
نزدل در راستای گرادیان در جهت بیشترین کاهش برای رسیدن به مینیمم

$$-\nabla_{\underline{w}} F(\underline{w})$$

$$\nabla_{\underline{w}} F(\underline{w}) = \nabla_{\underline{w}} \left[\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} (t_k - I_k)^2 \right]$$

$$= \lim_{N\to\infty} \left[\nabla_{\underline{w}} \sum_{k=1}^{N} (t_k - \underline{I}_k)^2 \right]$$

$$= \lim_{N\to\infty} \frac{1}{N} \left[\sum_{k=1}^{N} 2(t_k - I_k) \nabla_{\underline{w}} (-I_k) \right]$$

$$= \lim_{N\to\infty} \frac{1}{N} \left[\sum_{k=1}^{N} 2\delta_k \nabla_{\underline{w}} (-\underline{w}^T \underline{X}_k) \right]$$

$$= \lim_{N\to\infty} \frac{1}{N} \left[\sum_{k=1}^{N} 2\delta_k (-\underline{X}_k) \right]$$

$$= -2 E[\delta_k \underline{X}_k]$$

ویدرو و هاف تصمیم گرفتند بجای $E[\delta_k \underline{X}_k]$ از هر $\delta_k \underline{X}_k$ برای تصحیح $\underline{w}$ استفاده کنند.

$$\underline{W}_{k+1} = \underline{W}_k + \alpha \delta_k \underline{X}_k$$

قانون دلتا

$$0 < \alpha < \frac{2}{\text{بزرگترین مقدار ویژه } R}$$

قانون ویدرو هاف ، قانون ویدرو قانون LMS

$R \sim$ ندارم $\sim$ با بعی خطا α

PPCO

Subject:
Year. Month. Date. ۷۹

مقادیر متعارف α : $0.01 \leq \alpha \leq 1.0$

مقدار خوب برای شروع : $\alpha = 0.1$

α بزرگ ← سطح پرش ← روبتدل از روی پایین

α کوچک ← اندی الگوریتم

$\frac{1}{10n} \leq \alpha \leq \frac{1}{n}$ پیشنهاد اردوی ۱۹۸۸ : (n : تعداد ورودی)

• نسخه آموزش دسته‌ای آدالاین

Batching

عدم تغییر $\underline{W}$ با اعمال تمام ورودی $\underline{X}$

$\underline{W}$ پس از تعداد زیادی $\underline{X}$ که $\underline{W}$ ثابت برشبکه اعمال می‌شود، بهنگام در آورده می‌شود

بهنگام سازی ← میانگین برداری $\delta_k \underline{X}_k$ در بهنگام سازی $\underline{W}$

• نسخه استفاده از ممنتم در آدالاین

Momentum

مقاومت در برابر تغییرات زیاد

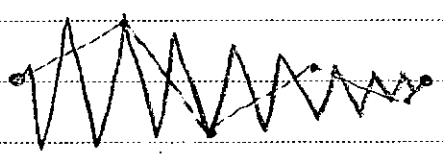
$$\underline{W}_{k+1} = \underline{W}_k + \alpha(1-\mu)\delta_k \underline{X}_k + \mu(\underline{W}_k - \underline{W}_{k-1})$$

$\mu \gg \alpha$ ← ☑ $\mu = 0.9$ & $\alpha = 0.01$

این نسخه ← سرعت بهتر، در سرعت بیشتر

← تمایل عملکرد نه چندان زیاد

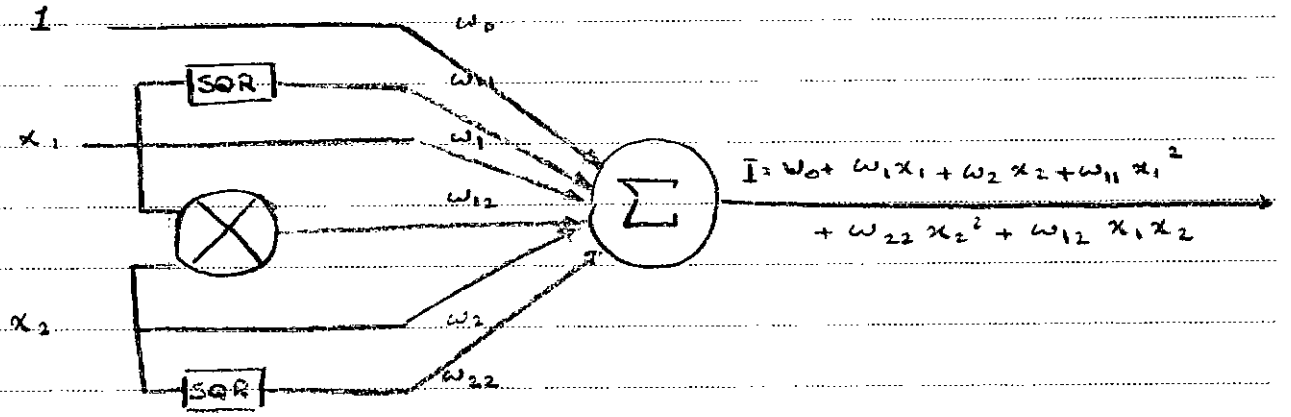
☑ کاربرد ممنتم



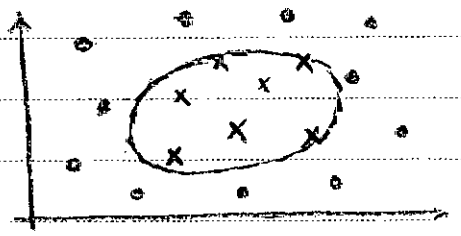
Subject:
Year. Month. Date. ( )

• آدالاین نیز خطی است ← چون ترکیب خطی ورودی‌ها را می‌گیرد
می‌تواند مسائل غیر خطی جدایی پذیر را حل کند.

• آدالاین درجه دو



■ آدالاین درجه دو



Subject:
Year. . Month. Date. ۲۷

◎ تداعی الگوها و شبکه‌های تداعی بر خطی

* مایلیری : تشکیل تداعی بین الگوها

* ارسطو معتقد است که ذهن انسان مطالب ① برخلاف یکدیگر ② از لحاظ زمانی نزدیک به یکدیگر ③ شبیه به هم ④ به طور متوالی واقع شده اند را با هم مرتبط می کند

* الگوی ورودی و تداعی شده ممکن است از یک نوع باشند یا از انواع مختلفی باشند

☑ مثلث متساوی الاضلاع ← مثلث تام الزاویه
مثلث متساوی الساقین
مربع !

☑ گل ← گل دیگر
بو !

* تداعی در انسان
اگر الگوی تحریک شبیه الگوی ذخیره شده باشد هم الگوی پاسخ تداعی می شود .
ورود الگوی دقیق لازم نیست .

☑ شناسایی از روی ناقص

* بد : تداعی علیرغم وجود خطا

* شبکه‌های تداعی بر ، انجمنی ، متداعی ، انتساب بر ، یاد آور
Linear Associative Networks
Associative Networks

Subject:
Year. Month. Date. ( )

* تعریف ریاضی

$$\Psi : \underline{X}_i \longrightarrow \underline{Y}_i \qquad i = 1, 2, 3, \ldots$$

$\underline{X}_i \in \mathbb{R}^n$ ورودی
$\underline{Y}_i \in \mathbb{R}^m$ خروجی
} بردار

$$\underline{Y}_i = \Psi(\underline{X}_i)$$

* دسته بندی اول شبکه های عصبی براساس سیر

Feed Forward ۱- پیش خور : مدل بازخورد ندارد

Recurrent ۲- بازگشتی : دارای بازخورد

الف- خروجی هر لایه تابع ورودی لایه قبل

* دسته بندی دوم شبکه های عصبی براساس یادآوری

Autoassociative ۱- خود عصبی (خود یادآور)

$$\underline{Y}_i = \underline{X}_i \quad , m = n$$

Hetroassociative ۲- غیرخود عصبی (دگر یادآور)

$$\underline{Y}_i \neq \underline{X}_i$$

• در شبکه های خود عصبی بعد از آموزش الگوی ورودی نوعی که قبلاً مدل نوعی را به شبکه آموزش داده بودیم را به شبکه بدهیم، ورودی مدل نوعی را در خروجی تولید می کند.

• در شبکه های غیرخود عصبی بعد از آموزش الگوی ورودی نوعی که قبلاً مدل نوعی را به شبکه آموزش داده بودیم را به شبکه بدهیم، ورودی مدل نوعی را مقداری تغییر داده و در خروجی تولید می کند.

Subject:
Year. Month. Date. ۲۸

* دسته بندی سوم شبکه های تداعی گر از لحاظ کاربردی

۱- کاربردی

Assertive

$$\Psi(\underline{X}_i + \underline{\varepsilon}) = \underline{Y}_i$$

ε : خطای برچسب

۲- درونیاب

Interpolative

$$\Psi(\underline{X}_i + \underline{\varepsilon}) = \underline{Y}_i + \sigma$$

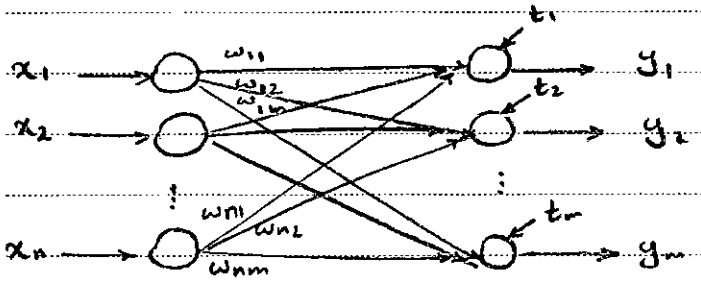
$$|\underline{\varepsilon}| \to 0 \Rightarrow |\sigma| \to 0$$

* شبکه تداعی گر پیش خور

• دولایه
  - لایه اول : به تعداد نرون های ورودی نرون دارد.
  - لایه دوم : به تعداد نرون های خروجی نرون دارد.

• اتصالات کامل
  - هر خروجی = ترکیب خطی از ورودی ها

• معماری



$$\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \underline{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \underline{T} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{bmatrix}$$

$$\underline{W}_i = \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{ni} \end{bmatrix}$$ وزنهایی که به واحد i ام می آیند

$$\underline{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & & & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} = [\underline{W}_1 \; \underline{W}_2 \cdots \underline{W}_m]$$

$\underline{X}$ $\underline{W}$ $\underline{T}$ $\underline{Y}$

$$\underline{Y} = \underline{W}^T \underline{X}$$

Subject:
Year Month Date ( )

• عملکرد ← عملکرد دوردیاب

اگر P زوج $\langle s_i, t_i \rangle$ برای $i = 1, 2, \ldots, P$ برای آموزش

شبکه به کار رود پس از پایان آموزش، ورودی $X = S_k$

به شبکه اعمال شود، خروجی باید برابر $Y = T_k$ شود اگر ورودی

$X = S_k + \varepsilon$ به شبکه داده شود، خروجی باید برابر $Y = T_k + \varepsilon$

شود.

• آموزش شبکه ← تابش سهب

$S_k = \begin{bmatrix} s_{k1} \\ \vdots \\ s_{kn} \end{bmatrix}$ $T_k = \begin{bmatrix} t_{k1} \\ \vdots \\ t_{km} \end{bmatrix}$ k امین زوج آموزشی

$$W_{(new)} = W_{(old)} + S_k T_k^T$$

$$= S_1 T_1^T + S_2 T_2^T + \cdots + S_P T_P \quad (*)$$

$$= \sum_{i=1}^{P} S_i T_i^T$$

(*) مقدار اولیه وزنها برابر صفر ← آموزش با P زوج تمرینه

$$w_{ij}^{new} = w_{ij}^{old} + s_{ki} t_{kj}$$

Subject:
Year. Month. Date. ۲۹)

آیا این آموزش دقیقا درست یا بازیابی خروجی را به خوبی انجام می دهد؟

یک ورودی $\underline{X}_k$ که یکی از نمونه های آموزشی ما بوده است در نظر می گیریم و داریم

$$\underline{W}^T \underline{S}_k = T_1 S_1 S_k + T_2 S_2^T S_k + \cdots + T_k S_k^T S_k + \cdots + T_L S_L^T S_k$$

این باید برابر $\underline{T}_k$ شود (ورودی $\underline{S}_k$ باید خروجی $\underline{T}_k$ بدهد) پس باید:

$$\underline{S}_i^T \underline{S}_j = S_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

یعنی مجموعه ی ورودی $\underline{S}_i$ باید متعامد یکه (orthogonal) باشند.

این شرط بسیار محدود کننده است ← بردارهای ورودی متعامد

بردارهای ورودی یکه

حداکثر تعداد الگوهای قابل ذخیره محدود

تعداد بردارها ← n بعدی در فضای n بعدی

تعداد بردارهای متعامد یکه در فضای n بعدی = n ← در فضای سه بعدی : ۳

برای ما مطلوب است که بتوانیم هر تعداد از الگوهای ورودی و خروجی را داشته باشیم.

با حذف شرط تعامد:

$$\underline{W}^T \underline{S}_k = \underline{T}_k + \sum_{j \neq k} \underline{T}_j \underline{S}_j^T \underline{S}_k = \underline{T}_k + \eta$$

خطای بازیابی ← $\underline{T}_k$

P4PCO

Subject:
Year. Month. Date. ( )

W باید به نحوی انتخاب شود که η کوچک باشد.

که آموزش با حداقل کردن متوسط مربع خطا

ماتریس متشکل از ستونهای S: $S = (\underline{S}_1\ \underline{S}_2\ \dots\ \underline{S}_L)$

$T = (\underline{T}_1\ \underline{T}_2\ \dots\ \underline{T}_L)$

متوسط مربع خطا $F(W) = \frac{1}{L}\sum_{k=1}^{L} |\underline{T}_k - \underline{W}^T \underline{S}_k|^2$

نرم ماتریسی $F(W) = \frac{1}{L}\|T - W^T S\|^2$

برای حداقل کردن $F(W)$

$F(W) = 0 \longrightarrow W = (TS^+)^T$

محدودیت ورودیها ← مستقل خطی بودن ($S^+S = I$)

$S^+S = I \longrightarrow F(W) = 0 \longrightarrow \psi(X_i) = Y_i$

S^+: شبه معکوس یا معکوس تعمیم یافته S

حذف محدودیت مستقل خطی بودن:

$S^+S \neq I \longrightarrow \underline{S}_i = \sum_{i \neq j} \alpha_j \underline{S}_i$

Subject:
Year. Month. Date. ۲۵

$$W^T S_j = \sum_{i \neq j} \alpha_i \; W^T S_i = \sum_{i \neq j} \alpha_i T_i$$

اگر بخواهیم $F(W.) = 0$:

$$F(W.) = 0 \longrightarrow T_j = \sum_{i \neq j} \alpha_i T_i$$

یعنی سمان رابطه خطی S_i باید در T_i نیز وجود داشته باشد. در غیراین صورت

$$W^T S_i \neq T_i \Rightarrow F(W.) > 0$$

Singular Value Decomposition — تجزیه مقدار منفرد

ماتریس $S : n \times h$ ، $n \leq h$ را می‌توان به صورت زیر تجزیه کرد:

$$S = \sum_{i=1}^{h} \sqrt{\lambda_i} \; P_i^T q_i$$

$P_i : n \times 1$ ، $q_i : h \times 1$ ← مجموعه بردارهای ویژه متعامدیله در ماتریس $SS^T : n \times n$ و $S^T S : h \times h$

λ_i : مقادیر ویژه را از بزرگ به کوچک مرتب شده‌اند (اعداد حقیقی، غیرمنفی)

$$S^+ = \sum_{i=1}^{r} \frac{1}{\sqrt{\lambda_i}} \; q_i \; P_i$$

r : مرتبه ماتریس S ← rank ، تعداد ویژه غیرصفر، تعداد سطرهای غیرصفر، تعداد ستونهای غیرصفر

Subject:
Year Month Date ( )

بردارهای ویژه غیرخطی $\Leftarrow S^+$ غیرخطی

$n > h \Leftarrow S^T$ قابل تجزیه

وجود نویز $\Leftarrow$ 2

تابع پایه‌ای وزن‌دار

اثبات با تقریب Greville

$$W(new) = W(old) + \alpha S_k (T_k - W^T(old) S_k)^T$$

$$W_i(new) = W_i(old) + \alpha [(t_k - (W_i^T(old) S_k) S_k]$$

خطا

آموزش به دفعات متوالی به صورت آنلاین یا بلادرنگ

α برای دقت

تغییر عمل کردن با داشتن $\theta \neq 0$ به واحد $\Leftarrow$ تجربه

Subject:
Year. Month. Date. ۳۱

شبکه‌های تداعی گر بازگشتی

Recurrent Associative Networks

* وتنا وابسته‌ماندنی (Assective)

اگر ورودی اندکی تغییر کرد، سعی می‌کند خروجی را ثابت دارند.

* سیر بازگشتی

خروجی به ورودی متصل است.

نقطه پایداری

* این شبکه‌ها عموماً از یک نقطه اولیه شروع می‌کنند به یکی از نقاط پایدار شبکه همگرا می‌شوند.

سعی می‌کنیم پیچیدگی را در این ذره باشد.

* طراحی این شبکه‌ها به دو روش مورد نظر است:

I) شبکه از هر نقطه‌ای که شروع به کار کند باید به یکی از حالات پایدار برسد.

II) حالت پایداری که شبکه به آن می‌رسد باید به طبق نوعی معیار سنجش نزدیکترین حالت پایدار به حالت اولیه باشد.

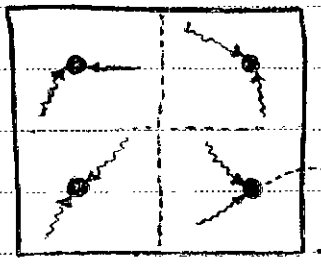
III) باید بتوان هر تعداد حالت پایدار که مورد نظر باشد برای شبکه ایجاد کند.

* عملکرد شبکه بازگشتی تداعی‌گر را می‌توان به یک سیستم جاذبه محدود تشبیه کرد که در آن

Subject:

Year. Month. Date. ( )

هر تعداد گره جذب کننده در هر نقطه دلخواه از فضای n بعدی قرار داده شود. نیروی جاذبه هر گره تنها به فاصله اثر دارد که به آن گره از سایر گره‌ها نزدیک‌تر است. مارکوف به این ترتیب

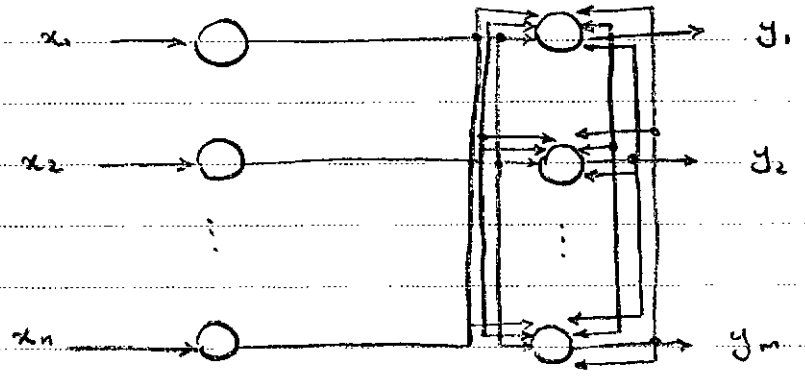


شروع می‌شود که یک جرم آزمایشی در این نقاط قرار داده می‌شود.

جرم به حرکت در آمده به نزدیکترین گره جذب می‌شود.

* طراحی شبکه‌های بازگشتی متداعی ساده، تنها مدت اول تأمین می‌شود
* معماری کلی شبکه‌های بازگشتی متداعی تک لایه

اتصالات کامل



• ماتریس وزنها متقارن

• ماتریس وزنها با قانون هب تعیین می‌شود.

• عناصر قطری: صفر (نزدیک به مقادیر میدانی) یا غیر صفر

هر نورون به خودش فیدبک ندارد

• پاسخ سلول به ورودیها ← بطور نامحدود بزرگ می‌شود ← تابع حظی فعالیت باید محدود شود.

(BSB) Brain State in a Box *

• اندرسون ۱۹۷۷

• n واحد + اتصالات بازگشتی

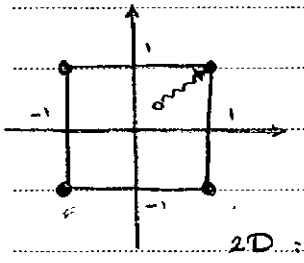
• ورودی x_i تنها به حالت t_0 اعمال می‌شود.

• تابع فعالیت ← تنها مقادیر داخل یک ابر مکعب ← $[-1, +1]^n$

• سیستم در آمدن واحدها: بزمان (سلولها) ← اصلاح وزنها سیستم رسیدن به حالت پایدار

• اتصالات: کامل یا غیر کامل اتانی

• حالات ← پایدار: رئوس ابر مکعب
↙ اولیه: نقطه ای داخل ابر مکعب (فضای جستجو)



• وزنها ← مقادیر اولیه: کوچک
↙ سیستم رسانی: سیستم رسیدن به حالت پایدار

Subject:
Year. Month. Date. ( )

الگوریتم BSB

۰. وزنها = مقادیر کوچک اتفاقی
نرخهای یادگیری α و β = مقادیر اولیه

1. به ازای هر بردار آموزشی S

2. $Y_i = S_i$ (مقدار اولیه خروجی برابر بردار ورودی)

3. تا زمانی که خروجیها پایدار نشده اند

4. محاسبه خروجیهای خطی

$$I_i = y_i + \alpha \sum_{j \neq i} y_i w_{ji}$$

مقدار فعال یا بازخوردی از خروجی خود واحد برابر α است

5. خروجی

$$y_i = \begin{cases} 1 & , I_i > 1 \\ I_i & , -1 \leq I_i \leq 1 \\ -1 & , I_i < -1 \end{cases}$$

6. اصلاح وزنها (قانون هب)

$$w_{ij}(new) = w_{ij}(old) + \beta y_i y_j$$

مربوط به خروجی نه ورودی که آن نیز هستند
ورودیها تطبیق ← اثر ملعب

PAPCO

Subject :

Year . Month . Date . ۴۲

• هر حالت پایدار $\underline{x}$ stable دارای یک زوج اضافی $\underline{x}$ stable - مزاحم دارد.

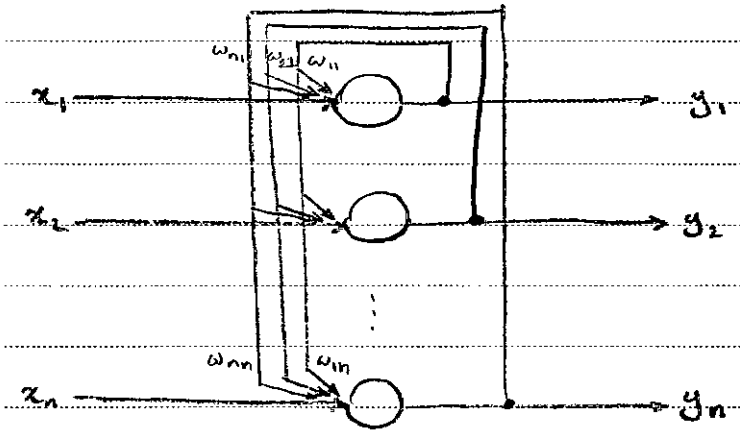
↳ این در واحد ورودی x_0 دارای وزن w_0 ، حذف حالات اضافی

• اصلاح وزن با قانون دیدر $W(new) = W(old) + \eta \underline{S} (\underline{Y}^d - \underline{Y}^a)^T$

* شبکه ی پیلد

• معماری ← اتصالات کامل ← معمولاً وزن نیدیک به صفر برابر صفر

هر ورودی تنها به یک واحد ← ورودیها فاقد وزن



• حال ی پیلد ۱۹۸۲

• اتصالات بازگشتی ← وزنهای متقارن ← $w_{ij} = w_{ji}$

حذف اتصالی ندارد ← $w_{ii} = 0$

• نسخه گسسته ← حالات شبکه در مقداری ← بایپری ← $[0, +1]$

دو قطبی ← $[-1, +1]$

Subject:
Year. Month. Date. ( )

- نسخ پیوسته ← مقادیر در بازه [0, +1]
- سیناپس ها ← آستانه دار ← انتخاب واحد اتفاقی
  ← ساده
- ورودی از خارج به هر واحد پی است ← ثابت تا سیناپس ها
  ← از زمان دهی بعد از اولین سیناپس سازی
- مقدار آستانه ای واحد ← صفر یا غیر صفر
- انتخاب های معمول ← سیناپس سازی نا همزمان ✓
  ← مقادیر حالت ۴ دوقطبی ✓
  ← مقدار صفر برای آستانه ها ✓
- در سیناپس سازی نا همزمان یک تابع انرژی یا لیاپانوف می توان برای شبکه پیدا کرد.
  ← همگرایی اثبات می شود. (روش اثبات در تئوری سیستم های غیرخطی)
- تابع تبدیل واحد
  - دوقطبی

$$y_i(new) = \text{Case}\left(\sum_{j=1}^{n} w_{ij}\, y_j(old)\right) \begin{cases} > \theta_i : +1 \\ = \theta_i : y_i(old) \\ < \theta_i : -1 \end{cases}$$

  - باینری

$\text{Case}(\sim) < \theta_i \longrightarrow y_i(new) = 0$

PAGE

Subject:
Year. Month. Date. ۳۴

• هر واحد : ابر صفحه‌ای در فضای حالت $\langle \underline{w}_i , \theta_i \rangle$

$y_j(old)$ — سمت مثبت ابر صفحه j $y_j(new) = +1$

$y_j(old)$ — سمت چپ ابر صفحه j $y_j(new) = -1 / 0$

$y_j(old)$ — روی ابر صفحه (عبور از حالت پایدار) $y_j(new) = y_j(old)$

برای جلوگیری از عبور ابر صفحه از حالات پایدار، می‌توان بردار وزن $\underline{w}_i$ را به طور جزئی تغییر داد. بر این ترتیب دیگر حالت پایداری پیش نمی‌آید. پس:

bipolar : $y_i(new) = sgn\left(\sum_{j=1}^{n} w_{ji}\, y_j(old) - \theta_i + x_i \right)$

تابع پله

binary : $y_i(new) = u\left(\sum_{j=1}^{n} w_{ji}\, y_j(old) - \theta_i + x_i \right)$

• اگر x_i تنها در لحظه فعال شدن (فعال سازی) $\theta_i = 0$:

$$y_i(new) = sgn\left(\sum_{j=1}^{n} w_{ji}\, y_j(old) \right)$$

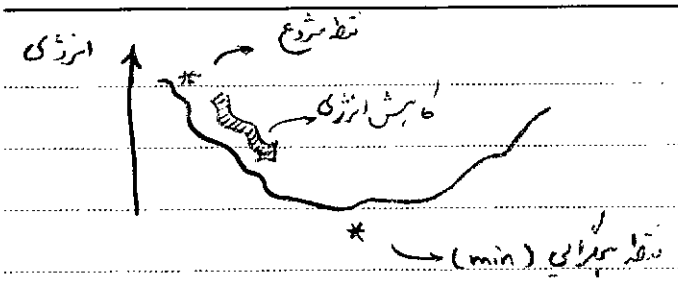
• نمای انرژی (Energy Landscape)

— هدف : اثبات همگرایی شبکه و پیدا کردن تابع انرژی یا تابع لیاپونوف

— تابع لیاپونوف ← تابع همواره کاهشی از پایین کراندار

Subject :

Year . Month . Date . ( )



تابع انرژی ؛ پیلد

$$E = -\frac{1}{2}\sum_{i\neq j}\sum_{j} w_{ij}\, y_i\, y_j - \sum_i x_i\, y_i + \sum_i \theta_i\, y_i$$

تغییر حالت ← آپدیت : نورون واحد مانند k

$$\Delta E = E(new) - E(old)$$

$$\Delta E = -\sum_{j\neq k} w_{kj}\,\underbrace{\left(y_k(new)\, y_j(new) - y_k(new)\, y_j(old)\right)}_{y_j(new) = y_j(old)}$$

1 مرتبه

$w_{ij} = w_{ji}$

$$-\sum_i x_i \underbrace{\left(y_i(new) - y_i(old)\right)}_{i=k \text{ برای } 0}$$

$$+\sum_i \theta_i \underbrace{\left(y_i(new) - y_i(old)\right)}_{i=k \text{ برای } 0}$$

$$= -\left[\sum_{j\neq k} w_{kj}\, y_j(old) + x_k - \theta_k\right]\Delta y_k$$

$$\Delta y_k = y_k(new) - y_k(old)$$

Subject:

Year. Month. Date. ۲۵

$$\Delta E = \underbrace{-}_{①} \underbrace{\left[\sum_{j \neq k} w_{kj}\, y_j(old) + x_k - \theta_k \right]}_{②} \underbrace{\Delta y_k}_{③}$$

این بیان شرط مانع تبدیل است.

a) $y_k(old) = +1$ و $y_k(new) = -1$ $\Rightarrow \sum w_{ji}\, y_j(old) + x_i < \theta_i$

$\Rightarrow$ ② و ③ داخل مرز منفی $\Rightarrow \Delta y_k = -2$

$① \times ② \times ③ \Rightarrow \Delta E < 0 \Rightarrow$ کاهش انرژی

b) $y_k(old) = -1$ و $y_k(new) = +1$ $\Rightarrow \sum w_{ji}\, y_j(old) + x_i > \theta_i$

$\Rightarrow$ ② و ③ داخل مرز مثبت $\Rightarrow \Delta y_k = +2$

$① \times ② \times ③ \Rightarrow \Delta E < 0 \Rightarrow$ کاهش انرژی

پس با تغییر حالت شبکه، انرژی کاهش می‌یابد تا به جایی برسد که دیگر تغییر حالت رخ ندهد ($\Delta E = 0$)، شبکه پایدار می‌شود. لذا در این شبکه همواره $\Delta E \leq 0$

یک دوری شبکه $\Leftarrow$ نقطه معینی در فضای انرژی $\Leftarrow$ حرکت به سمت دره انرژی $\Leftarrow$ دره = حالت پایدار

پس: هر حالت پایدار شبکه = مینیمم محلی در فضای انرژی

برای تضمین پایداری در تعداد کل محدود: $w_{ii} = 0$ و $w_{ij} = w_{ji}$

Subject:
Year. Month. Date. ( )

• پایداری شبکه‌ها

شبکه‌ها باید طوری تعیین شوند که الگوهایی که می‌خواهیم در شبکه ذخیره شوند حالت پایدار شبکه گردند (مقدار انرژی برای این الگوها حداقل شود)

$$E \rightarrow \min \Rightarrow \sum_{i \neq j} \sum_{j} w_{ij} \cdot y_i \cdot y_j \rightarrow \max$$

$$\underline{S}_k = \begin{bmatrix} s_k(1) \\ \vdots \\ s_k(n) \end{bmatrix}$$

$$w_{ij} = s_i \cdot s_j \Rightarrow \sum \sum w_{ij} y_i y_j > 0$$

$$\Downarrow$$

$$\sum \sum w_{ij} \cdot y_i \cdot y_j = \sum \sum s_i^2 s_j^2 > 0$$

$$\Downarrow$$

$$S_k \uparrow \rightarrow E \downarrow$$

اگر شبکه‌ای که انرژی در نقطه‌ی $\underline{S}_k$ را ذخیره کرده، الگوی جدیدی ارائه کنیم که کمتر از نیمی از مولفه‌های آن با مولفه‌های متناظر از $\underline{S}_k$ متفاوت باشد، شبکه به حالت پایدار $\underline{S}_k$ می‌رسد.

اگر بیش از نصف مولفه‌ها متفاوت باشند، الگوی جدید حالت پایدار $-\underline{S}_k$ خواهد شد. حالت $-\underline{S}_k$ را حالت معکوس (Reversed) می‌نامند

Subject:
Year. Month. Date. ۳۶

شبکه خودتداعی : ورودی S_k ← خروجی S_k

اگر با 49٪ هم شبیه S_k باشد جواب S_k است ولی نه $-S_k$ (بسیار حالب)

روش ذخیره بیش از یک الگو تقلبی :

$$w_{ij} = \begin{cases} \sum_{e=1}^{L} s_e(i) \cdot s_e(j) & , i \neq j \\ 0 & , i = j \end{cases}$$

با هر یک ضریب $\frac{1}{n}$ به جمع اضافه می‌شود ← امید پایداری ← تا زمان سهم تقسیم ماتریس

الگوی بایپری :

$$w_{ij} = \sum_{e=1}^{L} (2s_e(i)-1)(2s_e(j)-1)$$

• مزیت ۴ ← سهولت و سادگی نسبی الگوریتم یادگیری ← محاسبه بایپری عمل جمع و ضرب

• تا زمان سهم تقسیم ماتریس شبکه ۴ پتانسیل مرجع قرار گرفتن حالات پایدار و مینیمم های تابع انرژی را دارد

این حالات را حالات بازیابی (Retrieval) می‌نامند ← S_i

• حالات معکوس حالات بازیابی نیز مینیمم های تابع انرژی خواهند بود ← $-S_i$

• حالات مخلوط : ترکیبی خطی از هر تعداد فرد از حالات بازیابی ← S_i^{mix}

$$S_j^{mix}(i) = sgn(\pm S_k(i) \pm S_e(i) \pm S_m(i))$$

PAPCO

Subject:
Year. Month. Date. ( )

• حالات Spin Glass : برای حالتی که تعداد زیادی min ذخیره شوند، مینیمم های محلی دیگری ایجاد می شوند که با هیچ یک از حالات پایدار ارتباط خاصی ندارند.

• حالت دروغی یا جعلی (Spurious States)

— حالات مطلوب + خطوط + spin glass ← ناخواسته

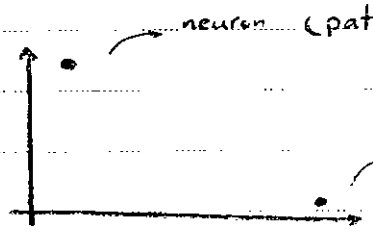
— میدان جاذبه کوچک ← با تعداد بزرگ نمی توان از روی آنها پرید

— راههای حذف این حالات

↳ حالت مطلوب ← علامت برنامه خاصی از بیخ را قبل بدانیم و در حالت شبیه خلاف آن دیده شود ← اصلاح: ضرب تمام برنامه های خروجی در -1

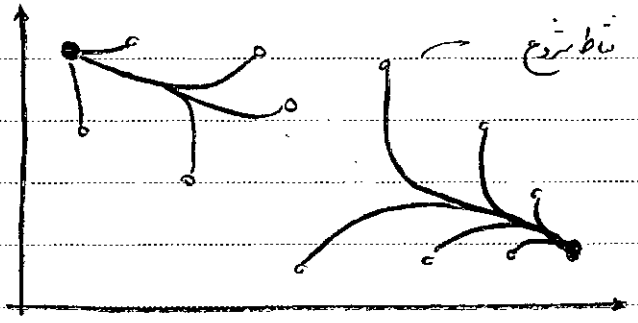
☑ برسیم که نمی خواهیم روحانی از آن min ساخت، بالا بیاید روحانی دیگری از آن نیز min ساخته می شود ← فلسفه وجود حالات ناخواسته

• عملکرد شبکه های پیچیده



Subject:

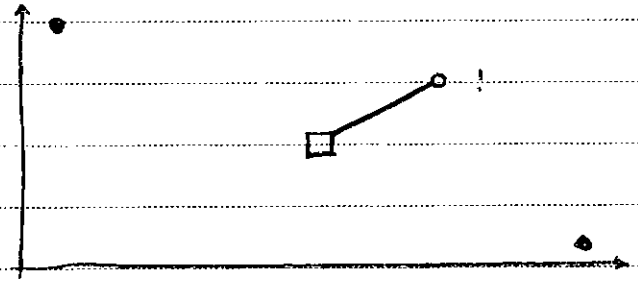
Year. Month. Date. ۲۷



سوال بی جواب:

چه تعداد از این خط داصل بین دو حالت

پایدار متمایند؟



حالت بدی

• الگوریتم شبکه گسسته هاپفیلد

• وزن‌های شبکه را از قانون هب تعمیم یافته محاسبه می‌کنیم.

1. بردار ورودی $\underline{X}$ را از ای

2. متغیرهای حالات برابر ورودی خارجی $\underline{X}$ قرار دهید $y_i(old) = x_i$

3. تا وقتی که پایدار نشده است

4. برای هر واحد i

5. خروجی خالص واحد $I_i = x_i + \sum_j w_{ij} \cdot y_j(old)$

6. محاسبه خروجی

$$y_i(new) = \begin{cases} 1 & I_i > \theta_i \\ y_i(old) & I_i = \theta_i \\ -1 & I_i < \theta_i \end{cases}$$

Subject:
Year. Month. Date. ( )

7. ارسال مقدار y_i (new) به بلوک واحد

8. مقدار y_i را آزمایش می‌کنیم.

• ظرفیت شبکه ← چند الگو قابل ذخیره است؟

↙ برای یک خطای قابل قبول چند الگو؟

ظرفیت $\propto$ تعداد واحدها (n)

ظرفیت $\propto$ تعداد الگوها (L) $\Rightarrow$ $L \uparrow \rightsquigarrow$ خطا $\uparrow$

با اثبات ریاضی: حداکثر تعداد الگوهای قابل ذخیره بدون ایجاد خطای غیرقابل قبول پیش بینی شده

← خطای از پیش معلوم

← نتایج تحلیل اول: تعداد ورودی n

n بیشتر شود مسئله

- خطای کمتر از 1% $\Rightarrow$ $L \leq 0.15\, n$
- خطای کمتر از 5% $\Rightarrow$ $L \leq 0.37\, n$
- خطای کمتر از 10% $\Rightarrow$ $L \leq 0.61\, n$

☐ مسئله با n ورودی ← برای خطای کمتر از 1% الگو نداریم

↙ برای ذخیره الگوی بیشتر ← افزایش نورون‌ها

افزایش نورون‌ها ولی ورودی ثابت ← تقسیم ورودی‌های بالا بدون تعداد برابر ورودی

PAPCO

Subject:

Year. Month. Date. ۳۸

نتایج تحلیل دوم : بازیابی صحیح همهٔ n بیت با دقت ٪۹۹

$$L \leq \frac{n}{2 \ln n}$$

بازیابی کامل همه‌ی n بیت

$$L \leq \frac{n}{4 \ln n}$$

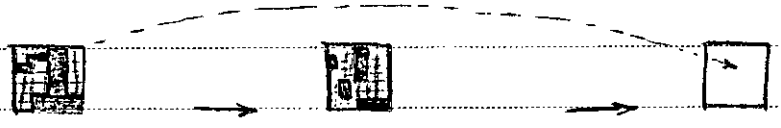
☑ دوردی $n = 100$

تحلیل اول ٪۱ خطا : $L \leq 15$

تحلیل دوم ٪۱ خطا : $L \leq 11$

تحلیل دوم ٪۱۰۰ دقت : $L \leq 5$!!!

☑ شبکه‌ی پیلد



ورودی‌ها — نمونه‌ی بهم ریخته — نمونه بازیابی شده

1 : پیاده‌سازی شبکه‌ی پیلد گسسته

- واسط کاربر ← پذیرش دوردی و خروجی ، تغییر پارامترها
- داده‌های مناسب برای نشان دادن و ژرفیها

Subject:
Year Month Date ( )

* قانون لوپین گراسبرگ

• شبکه عصبی : n پرداشگر ، خروجی‌های y_i ، توابع واحد به فرم

$$\frac{dy_i}{dt} = a_i(y_i)\left[b_i(y_i) - \sum_{j=1}^{n} w_{ij} f_j(y_j)\right]$$

• شرایط

۱- برای وزنها $w_{ij} = w_{ji}$ و $w_{ij} \geq 0$ و $w_{ii} = 0$

۲- تابع $a_i(u)$ ← پیوسته : $\forall u \geq 0$
← مثبت : $\forall u > 0$

۳- تابع $b_i(u)$ ← پیوسته
← پرش ثابت نشده : $\forall u \geq 0$

۴- تابع $f_j(u)$ ← شب پذیر
$f'_j(u) > 0$: $\forall u \geq 0$

۵- $[b_i(u) - w_{ii} f_i(u)] < 0$: $u \to \infty$

۶- یکی از این شرطها

$$\begin{cases} \lim_{n\to\infty} b \to \infty \\ \lim_{n\to\infty} b < \infty \quad \text{AND} \quad \exists u : \int_0^n \frac{ds}{a_i(s)} = \infty \end{cases}$$

Subject:
Year. Month. Date. ۳۹

• این شرایط برقرار باشد آنگاه: اگر حالت شبکه در زمان صفر در بخش مثبت فضای حالت باشد، شبکه همیشه به یک نقطه پایدار در بخش ⊕ فضای حالت همگرا می‌شود. مادامی که اندازه‌های بردار وزن W، همچنین مقدار نقاط پایدار محدود خواهد بود.

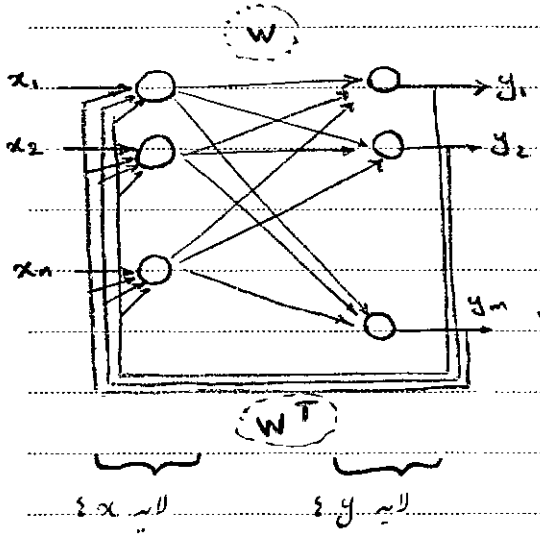
• مدل لیاپانوفی ← معادلات حاکم بر شبکه‌های زیادی حالات خاصی از مدل کوهن-گراسبرگ

$$\dot{y}_i = -A_i y_i - (B_i - C_i y_i)[I_i + f_i(y_i)] - (D_i y_i + E_i)[$$

$$\dot{y}_i = G_i y_i (\ell - \sum_{j=1}^{n} H_{ij} y_j)$$

* شبکه تداعی دوطرفه BAM ≡ Bidirectional Associative Memory

• کوسکو: ۱۹۸۸ تا ۱۹۹۲



• لایه‌ها: لایه x و لایه y

لایه x متصل به لایه y با ماتریس وزن W

لایه y متصل به لایه x با ماتریس وزن W^T

$m \neq n$

• ورودی به لایه x اعمال شود ← خروجی در لایه y

ورودی به لایه y اعمال شود ← خروجی در لایه x

Subject:
Year. Month. Date. ( )

• این شبکه تعمیم شبکه ی پیشنهاد به حالت غیر خود تداعی

• نسخه ی دوتایی : بایپری ، دوقطبی ، پیوسته

← معمولاً عملکرد نسخه ی دوقطبی مقداری بهتر است.

• آموزش شبکه با استفاده از قانون هب

$$\langle S_i , T_i \rangle \quad \begin{cases} \text{دوقطبی} & w_{ij} = \sum_{k=1}^{M} S_k(i)\, T_k(j) \\ \text{بایپری} & w_{ij} = \sum_{k=1}^{M} (2S_k(i)-1)(2T_k(j)-1) \end{cases}$$

• توابع فعالیت واحد ی شبکه ی پیشنهاد

$$x_i(new) = \begin{cases} I_i > \theta_i , & +1 \\ I_i = \theta_i , & x_i(old) \\ I_i < \theta_i , & -1 \end{cases}$$

$$y_j(new) = \begin{cases} I_j > \theta_j , & +1 \\ I_j = \theta_j , & y_j(old) \\ I_j < \theta_j , & -1 \end{cases}$$

• واحد ی های لایه ی x همه باهم بطور موازی بروز می شوند و در قدم بعدی واحد ی های لایه ی y همه باهم

به صورت یکی در میان

۹

Subject:
Year. Month. Date. ۴۵

الگوریتم شبکه

۵. محاسبه وزنهای یادگیری با زدن بردار آموزشی و خروجی واحدها برابر صفر

1. برای هر بردار ورودی آموزشی

2. حالات لایه x با بردار ورودی حاضر (اعمال ورودی S به لایه x)
ورودی I به لایه y

3. تا وقتیکه واحدها پایدار نشده اند

4. بهنگام سازی حالات واحدهای لایه y

$I_j = \sum_i w_{ij} x_i$

$y_j = f(I_j)$

5. بهنگام سازی حالات واحدهای لایه x

$I_i = \sum_j w_{ij} y_j$

$x_i = f(I_i)$

6. آزمایش پایانی: خانه الگوریتم در صورت به تعادل رسیدن هر حالات در دو لایه

شبکه عصبی دولایه پیوسته

ورودی ← آرام پیوسته ← خروجی [0,1]

تابع فعالیت: سیگموئید لاجستیک

$f_i(I_i) = \frac{1}{1+e^{-I_i}}$

مقادیر آستانه‌ای غیرصفر + وزنها از قانون هب

PqPCO

Subject:
Year Month Date ( )

• معادلات حاکم بر دینامیک این شبکه

$$\dot{x}_i = -a_i x_i + \sum_{j=1}^{m} w_{ji} f(y_j) + S_i \quad , a_i, s_i > 0$$

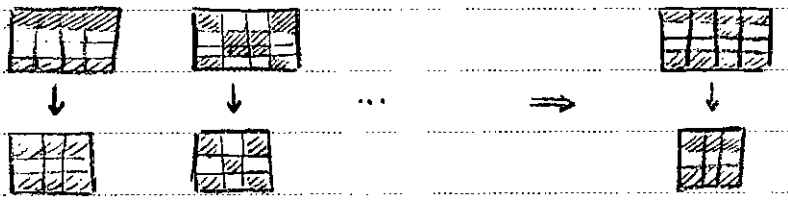
$$\dot{y}_j = -c_j y_j + \sum_{i=1}^{n} w_{ji} f(x_i) + t_j \quad , c_i, t_i > 0$$

• قضیه کوهن-گراسبرگ

اگر شبکه تقاضای دو طرفه از هر حالت اولیه $(x(0), y(0))$ شروع به کار کند، شبکه به حالت پایدار همگرا خواهد شد ← اثبات با استفاده از قضیه لیاپانوف - لاسالل

• کاربرد: شبکه بازیابی تصاویر، کد دیکد کردن اطلاعات

☑ اعداد ۰ تا ۹



Subject:
Year. Month. Date. ۴۱

@ اشکال یادگیری

Learning Paradigms

* انواع مدلهای یادگیری

- یادگیری با نظارت

Supervised l., l. with a Teacher

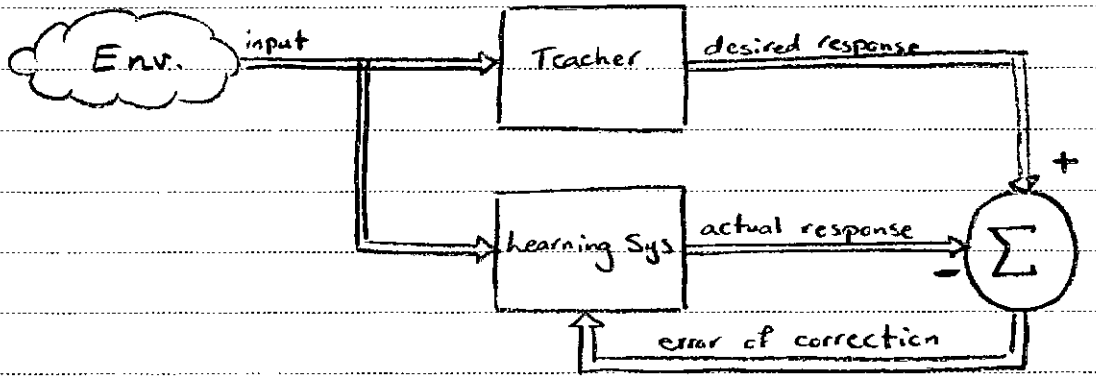
- یادگیری بی نظارت

Unsupervised l.

- یادگیری تقویتی

Reinforcement l., l. with a Critic, Graded l.

* یادگیری با نظارت

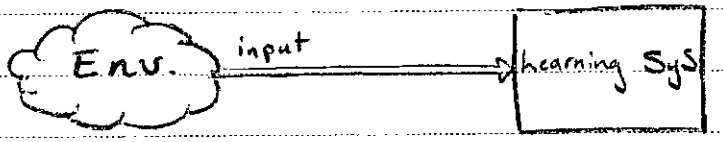


- معلم : دارای دانش محیط ⟵ پاسخ مطلوب برای هر ورودی
- آموزش : ارائه زوجهای < ورودی ، خروجی >
- تنظیم پارامترها : تحت تاثیر حالت محیط و خطای برجود
  ⟵ تدریجی و با تکرار مکرر عمل آموزش
- مجموعه آموزشی : شامل تعدادی نمونه از یک یا چند دسته
- هر نمونه آموزشی : $\langle S, T \rangle$ ⟶ S : feature / attribute
  ↘ T : desired response
  دسته مطلوب ، برچسب ورودی

Subject:
Year. Month. Date. ( )

- یادگیری ← حائل بودن خط (LMS) ☑ دیدرد
  پس انتشار (BP)

* یادگیری بی نظارت



- مدل معلم
- خروجی مطلوب برازای ورودی : ناشنوی ۱۱۶ ☑
- نمونه ها بدون برچسب

- شبکه خود انداز به کشف ویژگیها ، الگو ، نظم ، وابستگی ها یا دسته‌ی مجهد ورودی می پردازد و آنرا در خروجی نشان می دهد.

- نمونه‌ی خروجی :

۱- آشنایی یا شباهت : خروجی پیوسته نشان می‌دهد که الگوی ورودی چقدر به الگوی نمایندهٔ قبلاً مشاهده کردیم شبیه است.
typical
☑ چند درصد شبیه؟

۲- تحلیل مولفه اصلی : کشش بردار نرون به چنین واحد منجر به ایجاد مجموعه محور یا پایه می‌شود که در استاد آن محور شباهت ورودی به الگوی نماینده مربوط سنجیده می‌شود.
Principle Component Analysis
basis

PAPCO

Subject:
Year. Month. Date. ۴۲

۳- خوشه بندی : شبکه دارای m خروجی باینری که هر کدام بیانگر یک خوشه است > در

Clustering هر زمان تنها یکی فعال است > خروجی فعال : اندیس ورودی متعلق به کدام

خوشه > تعداد خوشه : تنظیم در زمان آموزش

۴- نمونه دهی : تشکیل دسته ای (مانند خوشه بندی) > خروجی شبکه : اندیس نماینده خوشه

Prototyping انتخاب شده (میانگین خوشه و...) > مولفه های خروجی : باینری (

کدام خوشه فعال است) یا نماینده شبیه ترین خوشه (میانگین خوشه و...)

۵- کدکردن : تبدیل ورودی به خروجی کدشده > نیاز به شبکه دیگری برای کدگشایی

۶- نگاشت ویژگی : واحدهای خروجی دارای نوع توپولوژی خاص (مثلا آرایه یک بعدی یا دوبعدی

Feature Mapping منظم) > در هر لحظه یک خروجی فعال > شبکه هر اندیس ورودی را

به یک نقطه خاص از آرایه خروجی نگاشت می کند > اندیس های مشابه ورودی،

محل های نزدیک به هم در نقشه را فعال می کند > کار شبکه: فعال سازی آرایه >

محل در آرایه معنادار است نگاشت: با نگاه کردن به خروجی باید بتوان به ویژگی ورودی پی برد.

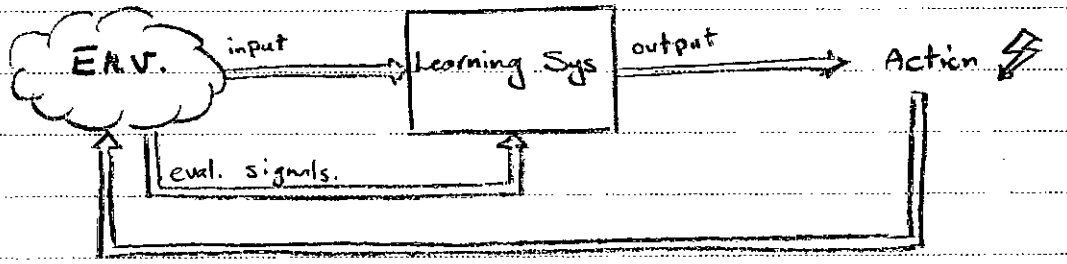
Data Visualization ☑

ترکیبی از این روشها امکانپذیر است.

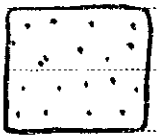
PAPCO

Subject:
Year. Month. Date. ( )

* یادگیری تقویتی



• یادگیری با نظارت ← نمونه‌های ⟨S, T⟩ ← T: معلم



نمونه‌ی ورودی باید کل فضای مسئله را پوشش دهد

بودن خروجی مطلوب برای هر مسئله ؟

عدم پوشش تمام فضای مسئله در نمونه‌ها ؟

• یادگیری تقویتی ← ورودی‌ها را می‌گیرد ولی به معلم نیاز ندارد

برای چندتایی به سیستم می‌دهیم که کارت خوب است یا نه

شبیه بازی «تن تن برو جلو» ← هرچه نزدیک‌تر، صدای تن تن تندتر

• داشتن خروجی مطلوب برای هر نمونه ضروری نیست.

• مسئله پس از طی چندین مرحله یادگیری ارزیابی شده و نتایج آن استفاده می‌شود.

• ارزیابی توسط یک تابع هزینه انجام می‌شود.

• شبکه‌های مبتنی بر یادگیری تقویتی عموماً ضعیف‌تر از شبکه‌های با ناظر هستند، لذا برای حالی مناسبند که

داده‌ی خروجی موجود نباشد.

* یادگیری با نظارت ——(VS)—— یادگیری تقویتی

• معلم: اطلاعات دقیق و گسترده — ناظر: اطلاعات کلی

• فیدبک: موثر در آموزش (مورد به مورد) — فیدبک: ارزیابی

• اطلاعات: کلی و لحظه‌ای — اطلاعات: عمری و دارای تاخیر

• اطلاعات ستیم: سیستم چگونه تطبیق یابد — اطلاعات غیر ستیم: مبتنی بر معنی و محتوا

• سرعت: بیشتر — سرعت: کندتر (تنها جهت داده می‌دهد)

* یادگیری با نظارت ——(VS)—— یادگیری بدون نظارت

• معلم: اطلاعات دقیق و گسترده — بدون معلم: فاقد اطلاعات خاص

• فیدبک: آموزش دهنده — فیدبک: مستقل از مورد

• مقیاس پذیری: ضعیف — مقیاس پذیری: بهتر

Subject:
Year. Month. Date. ( )

شبکه‌های رقابتی Competitive Networks

* شبکه Maxnet

• مدت: انقباض عایز نسبت به ورودیها / انقباض عایز در میزان بعاد بدون واحد

• مبتکر: لیپمن ۱۹۸۷

• کاربرد ← شبکه مستقل / قابل اضافه شدن به سایر شبکه‌ها

مشخص می‌کند که کدام گره آن دارای بزرگترین ورودی است

• معماری ← n واحد

اتصالات قابل و متقارن

وزنهای مثبت

تابع فعالیت خطی رمپ مثبت

$$f(I_i) = \begin{cases} I_i & I_i > 0 \\ 0 & o.w \end{cases}$$

ورودی تنها در شروع شبکه به کار گرفته می‌شود

• الگوریتم

۵. مقادیر اولیه $0 < \varepsilon < \frac{1}{n}$

$y_i(old) = x_i$

$w_{ij} = \begin{cases} 1 & i=j \\ \varepsilon & i \neq j \end{cases}$

هر واحدی در این شبکه می‌تواند دو بیت می‌فرستد، ما هم فرض می‌کنیم چند دم بخار با هم دو بیت‌ران بفرستند. می‌خواهیم چندم بیرم. ارزش چندش مثبت قوی و دو برابر بقیه ساعت می‌کند.

1. شرط خاتمه

2. بدست آوردن مقدار فعالیت واحد j

$y_j(new) = f\left(\sum_k w_{kj}\, y_k(old)\right)$

$= f\left(y_j(old) - \varepsilon \sum_{k \neq j} y_k(old)\right)$

3. ذخیره مقدار فعالیت برای مرحله بعدی

$y_j(old) = y_j(new)$

4. اگر بیش از یک واحد مقدار فعالیت غیر صفر دارد، الگوریتم ادامه می‌یابد در غیر این صورت خاتمه می‌یابد.

در نتیجه تنها نورون برنده مسابقه، مقدار فعالیت غیر صفر دارد ← برنده: در روی max

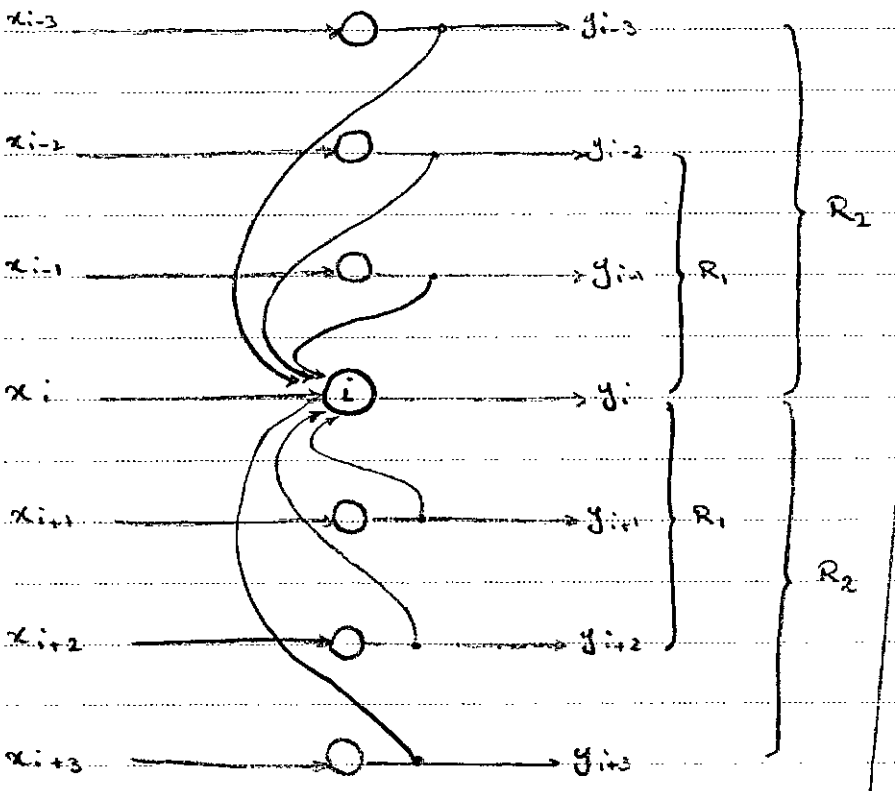
Subject:
Year. Month. Date. ( )

* شبکه لایه ای پرسپترون

• تعریف: ارتباط متقابل بین نورون های شبکه

• مثال: پرسپترون

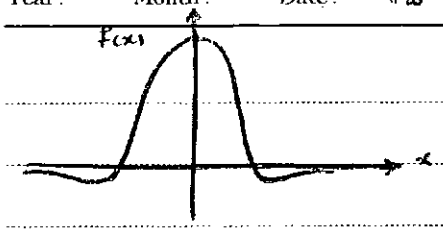
• معماری ← هر واحد با وزن های متصل به سیناپس های همگرای لنگره و با وزن های منفی به سیناپس های رقیب رابط در لایه خود وصل می شود.



وزن های مربوط به واحدهای متقابل نسبت به i یعنی واحدهای $i+1$ و $i-1$ مساوی و تقریباً نزدیک به واحد می باشد و آخر

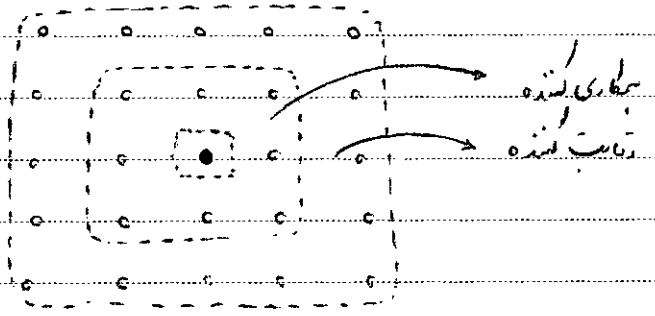
Subject :

Year . Month . Date . ۴۵

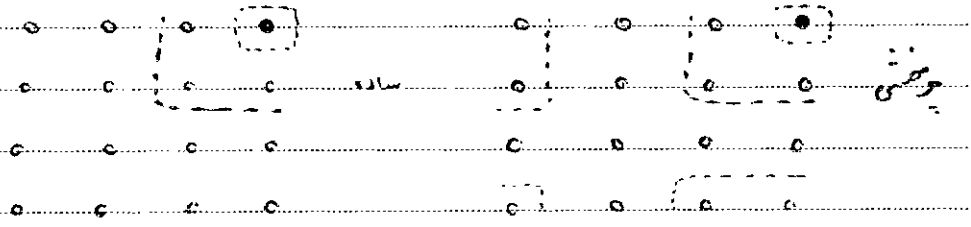


تابع هالهٔ مکزیکی

• چیدمان نورون‌های ردی صفحه



— انواع همسایگی برای کوهنن :



• وزن اتصال ← اتصال واحد i به واحد i ، w_{i+k}

R_2 خارج از

- مثبت : $(|k| \leq R_1)$ واحدهای برانگیزنده
- منفی : $(R_1 < |k| \leq R_2)$ واحدهای بازدارنده

• تابع فعالیت

$$y_i(t) = f\left(\alpha_i(t) + \sum_{k=-R_2} w_{i+k,i} \cdot y_{i+k}(t-1)\right)$$

Subject:
Year Month Date ( )

۰ الگوریتم

۵. مقادیر اولیه ← پارامترها: R_1 ، R_2 ، t_{max}

بخالت واحد: ۵ ← $y_i(old) = 0$

وزنها ←

$$\omega_{i+k,i} = \begin{cases} c_1 > 0 & , |k| \le R_1 \\ c_2 < 0 & , R_1 < |k| \le R_2 \end{cases}$$

۱. اعمال سیگنال خارجی و ذخیره $y_i(old)$

$y_i(old) = x_i$

مقدار اولیه t برابر صفر قرار داده می‌شود.

۲. آزمایش $t < t_{max}$

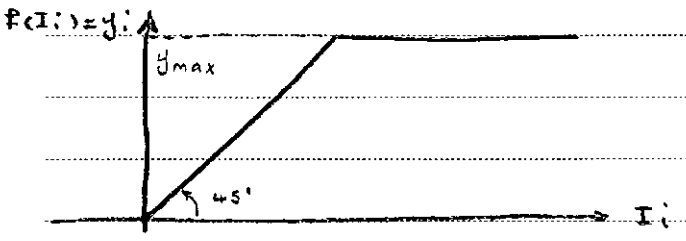
۳. خروجی خطی واحد

$$I_i = \sum_{k=-R_2}^{R_2} \omega_{i+k,i}\, y$$

$$= \sum_{k=-R_1}^{R_1} c_1\, y_{i+k}(old) + \sum_{k=R_1+1}^{R_2} c_2\, y_{i+k}(old) + \sum_{k=-R_2}^{-(R_2+1)} c_2\, y_{i+k}(old)$$

۴. اعمال تابع فعالیت

$y_i = \min(y_{max}, \max(0, I_i))$



$y_i(old) = y_i$ 5. ذخیره خروجی y_i در $Y_i(old)$

$t = t+1$ 6. افزایش شمارنده

7. آزمون شرط خاتمه $t < t_{max}$

* شبکه: Hamming Net

• فاصله همینگ: فاصله همینگ بین دو بردار $\underline{X}$ و $\underline{S_j}$ برابر است با تعداد مؤلفه‌های دو بردار که با هم متفاوت هستند (d_j)

• میزان تشابه دو بردار با n مؤلفه $a_j = n - d_j$

• برای بردارهای دوقطبی

$\underline{S_j} \cdot \underline{X} = a_j - d_j$

[مؤلفه‌های یکسان = 1] و [مؤلفه‌های مخالف = -1] ← مجموع حاصلضرب مؤلفه‌های نظیر ما

⇒ تعداد درایه‌های یکسان − تعداد درایه‌های مخالف

$\underline{S_j} \cdot \underline{X} = a_j - d_j = a_j - (n - a_j) = 2a_j - n \rightarrow a_j = \frac{1}{2}\underline{S_j} \cdot \underline{X} + \frac{1}{2}n$

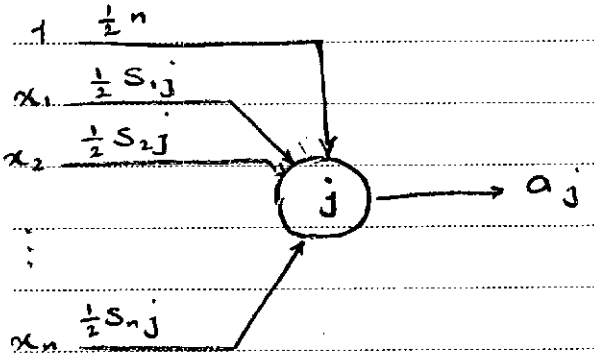
Subject:

Year. Month. Date. ( )

شبیه بالقوه نرون $\rightarrow$ $a_j = \frac{1}{2}\sum_i s_{ij} x_i + \frac{1}{2} n$

می‌توان چنین تعبیر کرد که یک واحد با چنین وظیفه‌ای میزان شباهت $\underline{X}$ و $\underline{S_j}$ را محاسبه می‌کند.

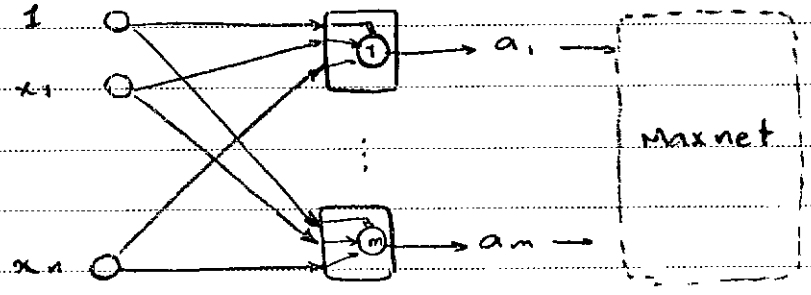
برای چنین عملی بردار $\underline{S_j}$ را در وزن‌ها ذخیره می‌کنیم و هرگاه ورودی اعمال شود، خروجی میزان شباهت با S_j را نشان خواهد داد.



- کاربرد $\leftarrow$ تعدادی واحد مانند

ورودی که وارد می‌شود $\leftarrow$ خروجی: میزان شباهت ورودی به بردارهای ذخیره‌شده

پیدا کردن شبیه‌ترین $\leftarrow$ قرار دادن یک شبکه Maxnet در خروجی



- الگوریتم (n ورودی ، m بردار نمونه S_j)

۰. وزنهای شبکه: $w_{ij} = \frac{1}{2} s_{ij} \quad i = 1,\dots,n \quad j = 1,\dots,m$

۱. برای هر ورودی $\underline{x}$

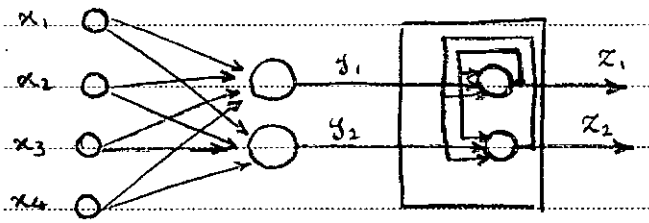
۲. محاسبه خروجی خطی $y_j = \frac{n}{2} + \sum_i w_{ij} x_j \quad j = 1,\dots,m$

۳. تعیین مقادیر اولیه شبکه Maxnet $y_j(0) = y_j$

۴. اجرای الگوریتم Maxnet برای تعیین بیشترین

$e_1 = (1, -1, -1, -1)$ ☑

$e_2 = (-1, -1, -1, 1)$



$W = \begin{bmatrix} \frac{1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & \frac{-1}{2} \\ \frac{-1}{2} & \frac{1}{2} \end{bmatrix}$

تعداد ورودی: ۴ تعداد پرسپترون: ۲ Maxnet (ε = 0.3)

آزمون:

$a = (1, 1, -1, -1)$

$y_1 = 3 \quad , \quad y_2 = 1$

$z_1 > z_2$

* شبکه‌ی حداکثر سازمانده که از نادیری رقابتی استفاده می‌کند، هستی بر شخصه‌ای خاصی از مقرانسان برتری یافته‌اند، سلولها در مقر در نزدیکی محلی طوری سازمان داده شده‌اند که بخشهای

Subject:

Year Month Date ( )

محاسبات حسی یا نقشه‌های حسی مرتب معنی دار (Topologically Ordered Computational Maps)

ارائه می شوند. مثلاً ورودیهای حس لامسه، بینایی، شنوایی و ... هریک با یک ترتیب هندسی

معنی دار به نواحی مختلف مغز مربوط می شوند.

یک نقشه محاسباتی را می توان متشکل از آرایه‌ای از نورونها دانست که به طور موازی ورودی اطلاعات

حسی ورودی تحلیل می کند. در این مکانیزم اطلاعات ورودی به توزیع احتمال مکانی تبدیل می شود که مقادیر

محاسبه شده پارامترها را با میزان مطابقت نسبی بخشها در نقشه ارائه می دهند.

در یک نقشه ویژگی خودسازمانده (Self Organizing (Feature) Map ≡ SOFM ≡ SOM)

واحدهای پردازشگر در ناحیه‌ای یک شبکه (توری) یک بعدی یا دوبعدی قرار داده می شوند. واحدها در

یک فرآیند یادگیری رقابتی نسبت به زمینه‌های الگوی ورودی تنظیم می شوند. محل واحدهای تنظیم شده در شبکه

به گونه‌ای تنظیم می یابد که بیانگر ویژگی‌های ورودی در یک دستگاه مختصات معنی دار در روی شبکه ایجاد

می شود. لذا یک نقشه خودسازمانده یک نقشه توپولوژیک از الگوهای ورودی بازتشکیل می دهد که در آن

محل قرارگرفتن واحدها بیانگر ویژگیهای الگوهای ورودی است.

یادگیری رقابتی که در این شبکه‌ها بکار گرفته می شود بدین صورت است که در هر قدم یادگیری واحدها برای

بعال متون با یکدیگر رقابت می پردازند.

در پایان یک مرحله رقابت تنها یک واحد برنده می شود. در این حال تغییر وزنهای واحد برنده نسبت به سایر واحدها به طور مقایسه انجام می شود و یادگیری رقابتی می شود.

* تاریخچه شبکه های خود سازمانده

• روشهای تقسیم کردن ← مباحث خوشه بندی یا تقارب (Clustering)

۱۹۹۲ و قبل از آن

۱۹۷۰ مالزبرگ ← تقسیم یادگیری خود سازمانده

(مجموع وزنهای گرفته شده از خروجی باید ثابت باشد)

فرض ثابت بودن میزان مواد شیمیایی در سیناپس

۱۹۷۶ ویلشاو ← و تقسیم مالزبرگ

• ۱۹۷۰ گروسبرگ ← تابع همسایگی در ورودی غیر ساخت: فرض ← تقسیم ادامه گروسبرگ

برپ تقسیم یادگیری باسیتی ساختن یک مجموعه بردار ورودی در ادامه های

هم احتمال یک تابع چگالی احتمال ثابت P و تشکیل می دهند باشد

PAPCO

Subject:

Year. Month. Date. ( )

بعنی بردارهای $\underline{w_i}$ باید خود را طوری تغییر دهند که برای هر بردار $\underline{X}$ ورودی

تابع چگالی احتمال p داشته باشیم.

احتمال اینکه $\underline{X}$ به $\underline{w_i}$ نزدیکترین باشد (m تا داریم) $= \frac{1}{m}$

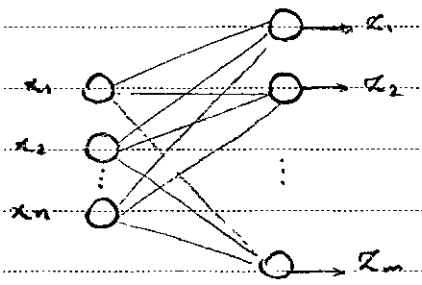
• ۱۹۷۸ وسینو ← حل مشکل نیاز به دانستن بودن تابع توزیع

* شبکه کوهونن

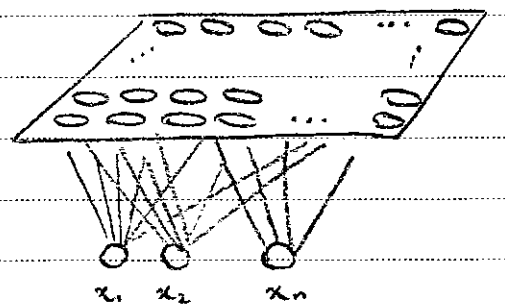
• نمای یک بعدی: یک لایه کوهونن آرایه‌ای از واحدهاست که به صورت یک، دو، یا K بعدی آرایش یافته اند.

• m: تعداد نورونهای لایه کوهونن و n: تعداد ورودی

• انواع نما



1D



2D

• در آموزش این شبکه براساس قانون کوهونن هر یک از نورونها فاصله بردار وزن خود را با بردار

Subject:
Year. Month. Date. ۴۹

- دوری X پیدا می‌کند : با توجه اولیه‌سی ، نمای مردی ($D(w,x) = 10^{-6}$) با الاترین
- هر نورون می‌خواهد نزدیکترین w به x را داشته باشد و وزنهایش به طور خاصی تغییر کند.

به همین منظور با واحدهای دیگر به رقابت می‌پردازد ← نورون برنده : پیروز در رقابت

- تشخیص نورون برنده در شبکه یک بعدی

۱- در بافت اطلاعات سایر نورونها ← خودش می‌فهمد برنده است یا نه

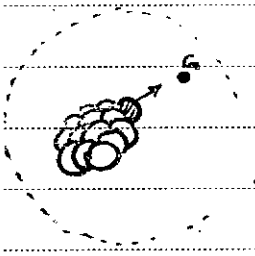
۲- اتصالات جانبی بازدارنده ← تنها واحد برنده فعال می‌ماند.

۳- واحد برنده دنبال ← در بافت I_i ، تعیین برنده

۴- واحد برنده دنبال ← فرستادن پیغام آستانه به همه واحدها ← اولین واحدی که دارای I_i بزرگتر از آستانه باشد برنده است.

- تشخیص نورون برنده در شبکه دو بعدی

برنده = نزدیکترین نورون به ورودی



نورون برنده (و احتمالاً همسایگانش) به بردار ورودی نزدیک می‌شوند.

← اصلاح وزن

Subject:
Year Month Date ( )

• قانون کوهونن

پس از تعیین نورون برنده برای یک ورودی خاص $\underline{X}$ خروجی این نرون (z_i) برابر یک می‌شود

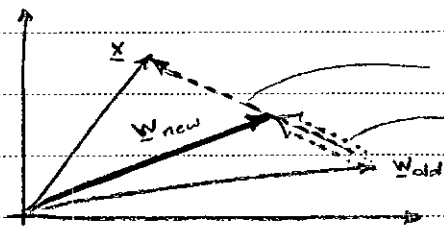
و همه خروجی‌های دیگر برابر صفر می‌شوند.

بردار وزن نورون

$$\underline{W}_i(new) = \begin{cases} \underline{W}_i(old) & z_i = 0 \\ (1-\eta)\,\underline{W}_i(old) + \eta \underline{X} & z_i = 1 \end{cases}$$

در ادامه نمادپذیری ← η مقدار بزرگ

بیشتر شدن نمادپذیری ← $\eta \downarrow$



$\underline{X} - \underline{W}_{old}$

$\eta(\underline{X} - \underline{W}_{old})$

$$\underline{W}_{new} = \underline{W}_{old} + \eta(\underline{X} - \underline{W}_{old})$$

• مشکلات قانون یادگیری کوهونن

هدف: تنظیم خودکار W به طوری که بردارهای ورودی $\underline{X}$ با توابع توزیع $P(\underline{X})$ احتمال برنده شدن تمام نورون‌ها برابر و مساوی $\frac{1}{m}$ باشند.

شرط رسیدن به هدف ← ورودی‌ها از ناحیه متصل دارای توزیع ساده یا تقریباً ساده

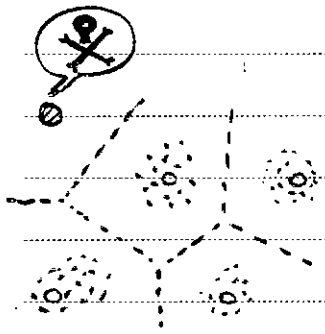
مانع برقراری این شرط ← نورون‌های مرده ← نیازمند بودن توزیع

وجود نورون در همسایگی این نورون

PqPCO

Subject:

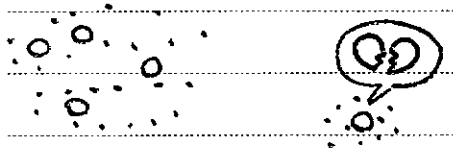
Year. Month. Date. ۵۰



• یکی از مشکلات بزرگ تبدیل ← برخی از نورون‌ها برگشت‌پذیر می‌شوند

← وزن آن‌ها هیچوقت اصلاح نمی‌شود ← نورون مرده

(اصلاح وزن‌ها: ○ ⋯ ○ ⟵ ○ ⋯ ○)



• نورون جدا افتاده هیچگاه نمی‌تواند به سایر نورون‌ها نزدیک شود

• راه حل رشد دادن شعاعی (Radial Sprouting)

— تمام بردارهای وزن را برابر صفر

— بردارهای ورودی را ابتدا ضرب در یک عدد بزرگ ← نزدیک شدن به مبدأ بردارهای وزن

— بلندی زمان متناسب عدد ضرب شده افزایش می‌یابد ← تا جایی برابر 1 شود

— یادگیری کند

• راه حل اضافه کردن نورون تنبیه‌ی ثابت

— اضافه کردن نورون تنبیه‌ی با تنبیه ثابت به بردارهای ورودی

— در ابتدا نورون تنبیه‌ی ← غلبه بر نورون‌ها

Subject:

Year Month Date ( )

- گذشت زمان نیز ضعیف‌تر ← در نهایت صفر

- حتی سریع‌تر از روش پیش

• راه حل اصلاح بیش از یک نورون

- در یک سباقی واحد برنده وزن‌ها اصلاح می‌شوند ← خروج از حالت Winner Takes All

- در ابتدا شعاع همسایگی بزرگ ← اصلاح هم نورون‌ها

- گذشت زمان ← کاهش شعاع همسایگی ← در نهایت همسایگی تنها شامل واحد برنده

- سریع

• راه حل دوم

- تعریف وجدان‌های متفاوت ← هر نورون بیش از حد برنده نشود

- اگر نورونی بیش از حد برنده شد، حذف شود.

- پیاده‌سازی (یک نسخه)

* محاسبه z_i تمام نورون‌های ورودی x

* نسبت‌هایی که دارد تا کنون برنده شده است محاسبه می‌شود.

Subject:
Year Month Date ۵۱

$$f_i(new) = f_i(old) + \beta (\alpha_i - f_i(old))$$

$\beta \cong 10^{-4}$

* محاسبه بایاس

$$b_i = \gamma \left(\frac{1}{m} - f_i(new)\right)$$

$\gamma = cte \longrightarrow$ مرتبه 10

* اصلاح فاصله ← اصلاح نحوه انتخاب نورون برنده

$$D' = D(w_i, x) - b_i$$

نورون کم برنده شده است ← $f_i(new)$ ↓ ← $b_i > 0$ ← D' ↓ ← فاصله ↑

نورون زیاد برنده شده است ← $f_i(new) > \frac{1}{m}$ ← $b_i < 0$ ← D' ↑ ← فاصله ↓

- الگوریتم یادگیری کوهونن

1. مقدار اولیه برای w_{ij}
   پارامترهای نرخ یادگیری
   پارامترهای همسایگی واحد برنده

2. پاسخ به شرط خاتمه ارضا نشده.

Subject :

Year . Month . Date . ( )

3. برای هر بردار ورودی X_i

4. محاسبه فاصله D برای نورون j

5. پیدا کردن نورون j دارای D کمینه

6. بهنگام آوردن وزن نورون و همسایگی نورون j ($N_j(t)$)

$$w_{ij}(new) = (1-\eta)\, w_{ij}(old) + \eta\, x_i$$

7. بهنگام درآوردن نرخ یادگیری η

8. کاهش شعاع همسایگی در فاصل معین

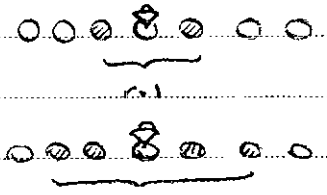
9. آزمودن شرط پایانی (چند بار تکرار ، ...)

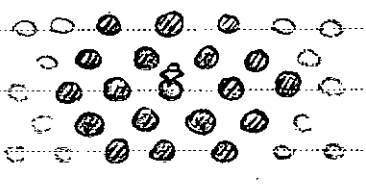
• انتخاب همسایگی

۱- همسایگی مستطیلی پیرامون برنده ← با توجه به توپولوژی ☑ مربع ، دایره ، ...

۲- همسایگی تعریف شده توسط تابع ← تابع موسوم به کلاه مکزیکی

☑ مستطیلی





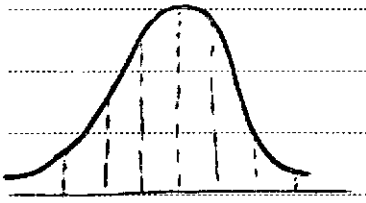
Subject:

Year Month Date ۵۲

☑ تابع ← تشابه بیشتری به عملکرد نورون‌های طبیعی

↓

یک واحد فعال برروی واحدهای همسایه نزدیک‌تر بیشتر تأثیر می‌گذارد تا دورتر



مانند تابع گاوسی

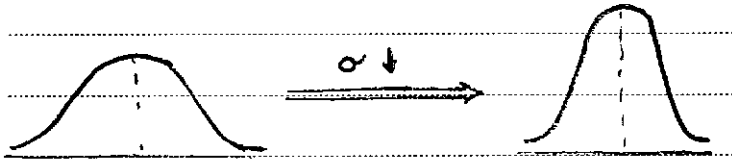
$$h_{j,i}(x) = e^{\left(\frac{-d_{j,i}^2}{2\sigma^2}\right)}$$

○○○●○○○

اندازه همسایگی ← تعیین شده توسط σ

کاهش تدریجی اندازه همسایگی ← σ تابعی نزولی با زمان

$$\sigma(n) = \sigma_0(n) \cdot e^{-\frac{n}{T_1}} \qquad \sigma_0, T_1 = cte$$



• قانون یادگیری اصلاح شده (پس از اعمال σ ، η)

↙ نرخ یادگیری تطبیقی ↘ تابع همسایگی

$$W_j(n+1) = W_j(n) + \eta(n) \cdot h_{j,i}(n) \cdot (X - W_j(n))$$

جذب بیشتر ⇔ همسایه نزدیک‌تر

$$\begin{cases} \sigma(n) = \sigma_0(n) \cdot e^{-\frac{n}{T_1}} \\ \eta(n) = \eta_0(n) \cdot e^{-\frac{n}{T_2}} \end{cases}$$

Subject:

Year. Month. Date. ( )

یادگیری ⟸ تخمین خطایی (دردورها) (نرون‌های اضافی)

I > فاز مرتب شدن Ordering

حالت‌های اساسی

برای حدود ۱۰۰۰ بار

مقدار η

$\eta : 0.1 \longrightarrow 0.01$

$\eta_0 = 0.1$ ، $T_2 = 1000$

فاز خود سازمانده‌ی

همسایگی

نرون برنده و چند واحد اطراف ⟵ تمام نرون‌ها

σ_0 = شعاع نشت ، $T_1 = 1000 / \log(\sigma_0)$

II > فاز همگرایی Convergance

تنظیم دقیق نقشه

برای حدود ۵۰۰۰ بار تعداد واحدی نقشه

مقدار η ⟵ ثابت

کوچک مانند 0.01

همسایگی ⟵ کوچک (شامل نرون برنده و چند نرون اطراف آن)

☑ برای نمایش شبکه می‌توان نرون‌های مجاور را با خط به یکدیگر متصل کرد.
مثلاً برای ساخت دوباره‌سازی پرانتزی سطح می‌توان از شبکه دوبعدی استفاده کرد.

☑ کاربرد: مدل‌سازی سطح، تقطیع تصاویر، بازیابی اطلاعات، طراحی ربات

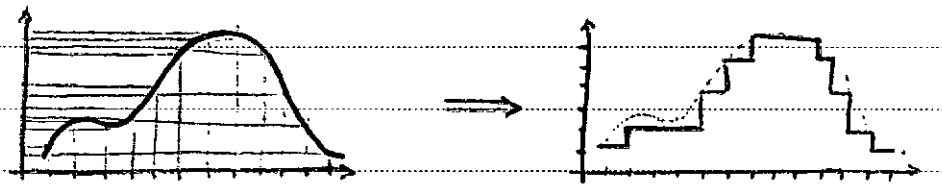
Subject:
Year: Month: Date: ۵۳

* شبکه یادگیری کوانتیزاسیون برداری Learning Vector Quantization (LVQ)

• کوانتیزاسیون (چندی سازی) برداری

برای بهره برداری از ساختار موجود در بردار ورودی ← به منظور فشرده سازی داده ها

☑ اسکالر

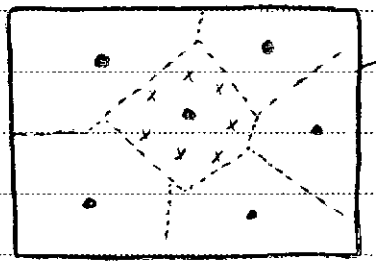


کوانتیزاسیون چند بعدی ← کوانتیزاسیون در هر بعد

کوانتیزاسیون کل بردار ← شامل: فشرده سازی بهتر ✓

برای اسکالر ← کوانتیزاسیون اسکالر : کوانتیزه ای از پیش تعیین شده (عدد)

برداری : " (بردار)



ناحیه بندی نقاط

برای هر ناحیه یک بردار بازسازی تعیین می شود

مجموعه تمام بردهای بازسازی : کتاب کد بازسازی

روش کوانتیزه کننده برداری با حداقل خطا، تقسیم بندی فضا با استفاده از نزدیکترین همسایه مبتنی بر فاصله اقلیدسی است.

کوانتیزه کننده وورونوی (Voronoi)

مجموعه سلولهای وورونوی

Subject:

Year. Month. Date. ( )

- تعیین بردارهای استاندارد به صورت تقریبی با یادگیری رقابتی
- برای دسته بندی انواع از یادگیری کوانتیزاسیون برداری استفاده شده است.

استفاده از یک ملیک برای نظارت:

یک بردار ورودی $\underline{S}$ به صورت اتفاقی از فضای ورودی

تعیین دسته مربوط با نزدیکترین بردار درونی $\underline{W}$

- تطبیق ← $\underline{W}$ جذب به $\underline{S}$
- عدم تطبیق ← $\underline{W}$ دفع از $\underline{S}$

- مراحل LVQ

$\underline{S}$ بردار آموزشی $\quad \underline{S} = (S_1, S_2, \dots, S_n)$

C_s دسته صحیح برای $\underline{S}$

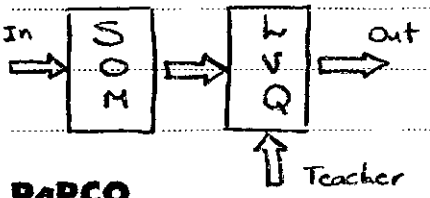
$\underline{W_j}$ بردار وزن واحد خروجی j ام

C_j دسته ارائه شده با واحد خروجی j ام

$\|\underline{S} - \underline{W_j}\|$ فاصله اقلیدسی $\underline{S}$ و $\underline{W_j}$

- خوشه بندی: مرحله‌ی بردار ← اندیس در نقاب آن

- تعیین بردارهای درونی به صورت تقریبی میتواند توسط شبکه SOM که بی نظارت است مشخص شود.

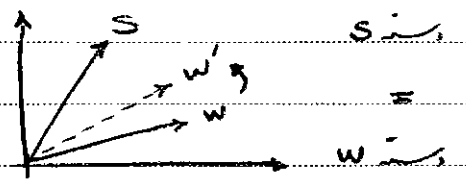


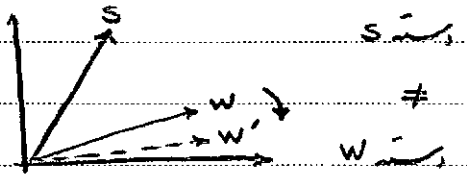
سپس می‌توان برای تنظیم دقیقتر از یادگیری با نظارت استفاده نمود.

PAPCO

Subject:

Year. Month. Date. ۵۴





الگوریتم LVQ

0. بردارهای وزن واحدها و مقدار اولیه α

1. تا زمانیکه شرط خاتمه غلط است.

2. برای هر بردار آموزشی

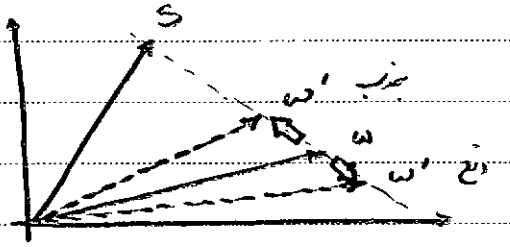
3. فاصله $\|S - W_j\|$ حداقل شود.

4. بهنگام سازی W_j

$$W_j(new) = \begin{cases} W_j(old) + \alpha (S - W_j(old)) & C_s = C_j \\ W_j(old) - \alpha (S - W_j(old)) & C_s \neq C_j \end{cases}$$

5. کاهش نرخ یادگیری α

6. بررسی شرط خاتمه



Subject:
Year Month Date ( )

• تعیین اولیه وزنها

- توسط شبکه SOM

- یک روش خوشه بندی k-means

- m بردار اول دوره ای

- m بردار اتفاقی از دوره ای

• نسخه ای دیگر LVQ

- بجای آنکه تنها بردار برنده که نزدیکترین بردار به بردار ورودی است اصلاح شود،

دومین بردار نزدیک ورودی (نایب برنده) نیز اصلاح می‌شود. (LVQ2)

به شرط ارضای شروط زیر ← بردار برنده و نایب برنده به دو دسته مختلف متعلق باشند

بردار ورودی متعلق به دسته نایب برنده باشد.

فاصله بردار ورودی تا بردار برنده و نایب برنده تقریباً مساوی

$\underline{S}$ بردار ورودی ، $\underline{w}_c$ بردار برنده ، $\underline{w}_r$ بردار نایب برنده

d_c فاصله تا برنده ، d_r فاصله تا نایب برنده

$$\frac{d_c}{d_r} > 1-\varepsilon \qquad \frac{d_r}{d_c} < 1+\varepsilon \qquad (d_c < d_r)$$ } شروط

P4PCO $\underline{S}$ دسته = $\underline{w}_c$ دسته

Subject:
Year. Month. Date. ۵۵

با داشتن این شروط $W_c(new) = W_c(old) \pm \alpha (S - W_c(old))$

- نسخه LVQ 2.1

در نظر گرفتن دو بردار نزدیکترین به ورودی

این بردارها به شرطی اصلاح می‌شوند که یکی از آنها متعلق به کلاس صحیح برای ورودی یعنی S

و دیگری متعلق به آن کلاس نباشد.

$$\begin{cases} \min\left(\frac{dc_1}{dc_2}, \frac{dc_2}{dc_1}\right) > 1-\varepsilon \\ \max\left(\frac{dc_1}{dc_2}, \frac{dc_2}{dc_1}\right) < 1+\varepsilon \end{cases}$$

در این حال برای بردار متعلق به کلاس ورودی S ← جذب

برای برداری که متعلق به کلاس نیست ← دفع

- برای دسته‌بندی الگوها از یادگیری رقابتی استفاده می‌شود.

- با نظارت

- استفاده از اطلاعات مربوط به دسته‌های ورودی

- بردارهای خروجی که به روش برنده تمام شده به طور جزئی تغییر می‌دهد.

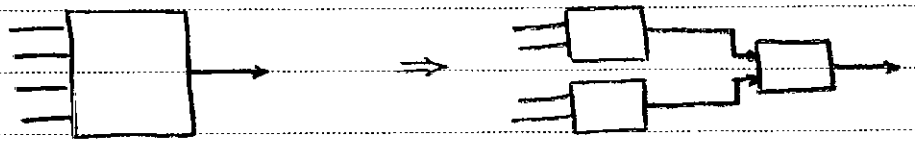
- لایه رقابتی شبکه برداری سلسله مراتبی

Subject:

Year. Month. Date. ( )

کوانتیزاسیون برداری ← محاسبات سنگین ← مقایسه با فاصله از همه کدبوک

کوتاه کردن جستجو در کدبوک ← ساختار سلسله مراتبی برای چند کدبوک کوچکتر



Subject:

Year Month Date ۵۶

شبکه‌ی پرسپترون چندلایه (پیچل) و تابع یادگیری پس‌انتشار خطا (پاخ)

BP MLP

* نسخه استاندارد

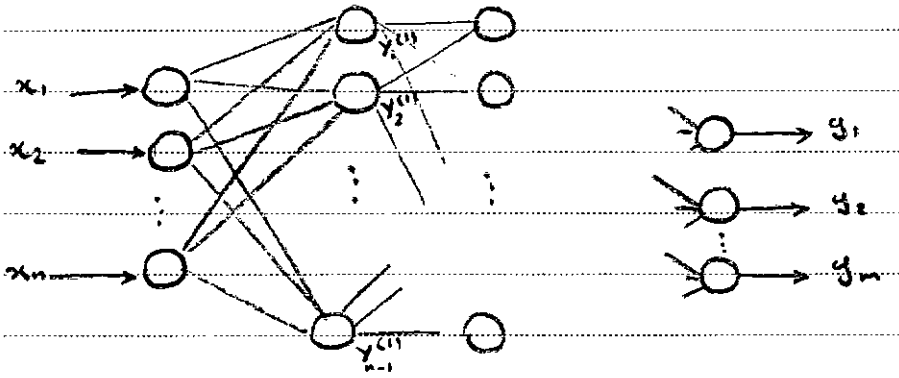
• برای حل مسائلی که به‌صورت خطی جدایذیر نیستند ← حل شدن ایراد وارده به پرسپترون

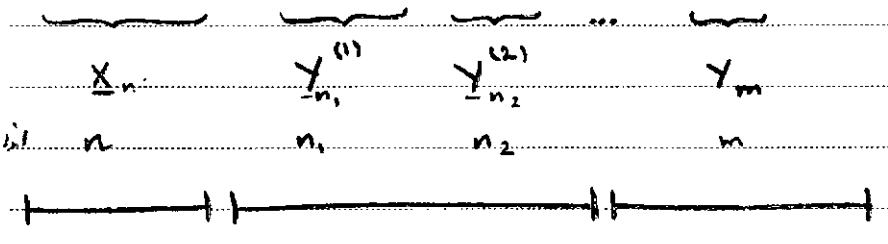
تابع فعالیت شبکه باید پله‌ای باشد تا اطلاعات ترفند از لایه میانی به خروجی انتقال پیدا کنند.

• معماری

دارای ۳ لایه یا بیشتر : لایه ورودی + لایه‌ی خروجی + لایه‌ی مخفی

اتصالات کامل





PAPCO

Subject:
Year. Month. Date. ( )

• قانون یادگیری : پس انتشار خطا / انتشار خطا به عقب / قانون دلتای تعمیم یافته

• مراحل یادگیری

۱- اعمال ورودی آموزشی ← انتشار تا خروجی ← محاسبه خروجی

۲- مقایسه خروجی با مقدار مطلوب ← محاسبه خطا ← انتشار خطا به سمت ورودی

۳- به هنگام در آوردن وزن‌ها با توجه به خطای تعیین شده

• استخراج قانون BP

$$\underline{S}_p = \begin{bmatrix} S_{p_1} \\ S_{p_2} \\ \dots \\ S_{p_n} \end{bmatrix} \qquad \underline{T}_p = \begin{bmatrix} T_{p_1} \\ T_{p_2} \\ \dots \\ T_{p_m} \end{bmatrix}$$ خروجی مطلوب $\underline{S}$ به

$$\underline{Y}_p^{(\ell)} = \begin{bmatrix} y_{p_1}^{(\ell)} \\ y_{p_2}^{(\ell)} \\ \dots \\ y_{p_{n_\ell}}^{(\ell)} \end{bmatrix}$$

$w_{ij}^{(\ell)}$ = وزن اتصال از واحد i در لایه $\ell-1$ به واحد j در لایه ℓ

$$E_p = \frac{1}{2} \sum_j (t_{pj} - y_{pj})^2$$ تابع خطا

$$I_{pj}^{(\ell)} = \sum_k w_{kj}^{(\ell)} \, y_{pk}^{(\ell-1)}$$ فعالیت واحد j در لایه ℓ برای ورودی p ← activation

$$y_{pj}^{(\ell)} = f\left(I_{pj}^{(\ell)}\right)$$ خروجی واحد j در لایه ℓ برای ورودی p

Subject:
Year. Month. Date. ۵۷

حال در تابع زنجیره‌ای داریم :

$$\frac{\partial E_p}{\partial w_{ij}^{(l)}} = \frac{\partial E_p}{\partial I_{pj}^{(l)}} \times \frac{\partial I_{pj}^{(l)}}{\partial w_{ij}^{(l)}}$$

$$\frac{\partial I_{pj}^{(l)}}{\partial w_{ij}^{(l)}} = \frac{\partial}{\partial w_{ij}^{(l)}} \left(\sum_k w_{kj}^{(l)} y_{pk}^{(l-1)} \right) = y_{pi}^{(l-1)}$$

تنها به ازای $k=i$ مقدار داشتن صفر نیست

با فرض

$$\delta_{pj}^{(l)} = -\frac{\partial E_p}{\partial I_{pj}^{(l)}}$$

$$-\frac{\partial E_p}{\partial w_{ij}^{(l)}} = -\nabla_w E_p = \delta_{pj}^{(l)} \times y_{pi}^{(l)}$$

تابع یادگیری را طوری در نظر می‌گیریم که متناسب با ضریب در راستای گرادیان باشد.

• نزول در راستای گرادیان (Gradient Descend)

$$w_{ij}^{(l)}(new) = w_{ij}^{(l)}(old) + \eta \delta_{pj}^{(l)} y_{pi}^{(l-1)}$$

تعیین $\delta_{pj}^{(l)}$ برای واحدهای مختلف

$$\delta_{pj}^{(l)} = -\frac{\partial E_p}{\partial I_{pj}^{(l)}} = -\frac{\partial E_p}{\partial y_{pj}^{(l)}} \times \frac{\partial y_{pj}^{(l)}}{\partial I_{pj}^{(l)}}$$

$$y_{pj}^{(l)} = f_j(\cdot) \longrightarrow \frac{\partial y_{pj}^{(l)}}{\partial I_{pj}^{(l)}} = f_j'(I_{pj}^{(l)} \cdot \alpha_{pj}^{(l)})$$

Subject:

Year. Month. Date. ( )

$$E_p = \frac{1}{2}\sum (t_{pj} - y_{pj})^2 \longrightarrow \frac{\partial E_p}{\partial y_{pj}} = -(t_{pj} - y_{pj})$$

برای واحدهای لایه خروجی (l : لایه آخر)

$$\delta_{pj} = (t_{pj} - y_{pj}) \cdot f'_j(I_{pj} \times \alpha_{pj})$$

برای واحدهای لایه مخفی

$$\frac{\partial E_p}{\partial y_{pj}^{(l)}} = \sum \frac{\partial E_p}{\partial I_{pk}^{(l+1)}} \times \frac{\partial I_{pk}^{(l+1)}}{\partial y_{pj}^{(l)}}$$

① ②

$$① = -\delta_{pk}^{(l+1)}$$

$$② = \frac{\partial}{\partial y_{pj}^{(l)}} \sum_i w_{ik}^{(l+1)} \cdot y_{pi}^{(l)} = w_{jk}^{(l+1)}$$

$$\longrightarrow \frac{\partial E_p}{\partial y_{pj}^{(l)}} = -\sum_k \delta_{pk}^{(l+1)} \cdot w_{jk}^{(l+1)}$$

$$\delta_{pj}^{(l)} = f'_j(I_{pj}^{(l)} \alpha_{pj}^{(l)}) \times \sum_k \delta_{pk}^{(l+1)} \cdot w_{jk}^{(l+1)}$$

Back Propagation ← δ هر لایه به لایه بعدی وابسته است.

تابع f باید مشتق پذیر باشد.

☑ مشتق برای توابع سیگموید

$$f(x) = \frac{1}{1+e^{-kx}} \Longrightarrow f'(x) = k f(x)[1-f(x)]$$

$$f(x) = \frac{2}{1+e^{-kx}} - 1 \Longrightarrow f'(x) = \frac{k}{2}[1+f(x)][1-f(x)]$$

۰ الگوریتم آموزش سیگنال

۵. انتخاب مقادیر اولیه وزنها برابر مقادیر کوچک اتفاقی

1. تا وقتی شرط خاتمه اشتباه است.

2. برای هر زوج بردار آموزشی (S_p, T_p)

3. محاسبه خروجیها از لایه مخفی اول تا خروجیها

$$y_{pj}^{(1)} = f_j\left(\sum_{i=0}^{n} w_{ij} S_{pi}\right)$$

$$y_{pj}^{(\ell)} = f_j\left(\sum_{i=0}^{n} w_{ij} y_{pi}^{(\ell-1)}\right), \quad \ell > 1$$

4. محاسبه δ_{pj} برای واحدهای خروجی

$$\delta_{pj} = (t_{pj} - y_{pj})\, f'(I_{pj} \times a_j)$$

5. محاسبه δ_{pj} برای واحدهای مخفی

Subject:

Year. Month. Date. ( )

$$\delta_{pj}^{(l)} = \left(\sum_k \delta_{pk}^{(l+1)} \, w_{jk}^{(l+1)} \right) . f_j' \left(I_{pj}^{(l)} , \alpha_j^{(l)} \right)$$

6. اصلاح وزن‌ها

$$w_{ij}^{(l)}(new) = w_{ij}^{(l)}(old) + \eta \, \delta_{pj}^{(l)} \, y_{p,i}^{(l-1)}$$

7. آزمون شرط خاتمه (خطا کمتر از حدی / تکرار به اندازه خاص)

برای آموزش شبکه با این الگوریتم مجموعه آموزشی مکرراً به شبکه اعمال شده و اصلاح وزن‌ها انجام می‌گیرد.

هر بار که مجموعه آموزشی به شبکه اعمال می‌شود را یک اپک (Epoch) گویند.

به همین دلیل آموزش MLP بسیار زمان‌بر است.

- الگوریتم استاندارد آموزش:

0. وزن‌ها را برابر مقادیر به‌دست‌آمده شده قرار می‌دهیم.

1. با اعمال بردار ورودی

2. $y_i^{(1)} = f\left(\sum_{j=0}^{n} w_{ji}^{(1)} x_j \right)$

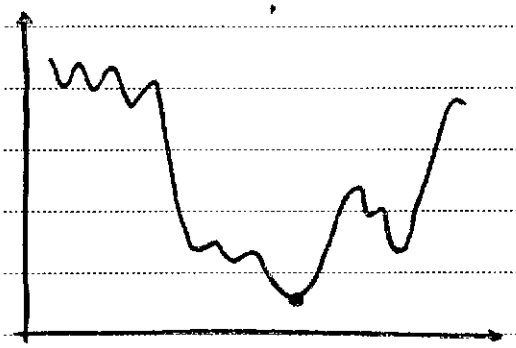
3. $y_i^{(l)} = f\left(\sum_{j=0}^{n} w_{ji}^{(l)} y_j^{(l-1)} \right)$ برای بقیه لایه‌ها

4. بردار خروجی = خروجی‌های آخرین لایه

Subject:

Year Month Date ۵۹

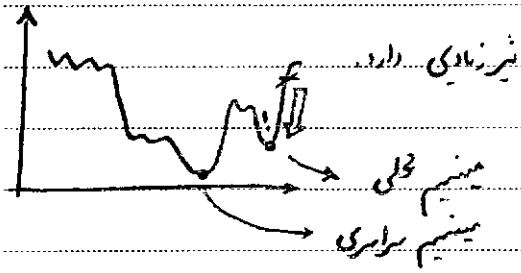
• نمایش عملکرد شبکه



شبکه دارای یک تابع خطا یا هزینه است.

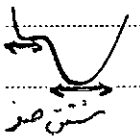
هدف شبکه: حداقل کردن تابع خطا با تنظیم وزن‌ها

در ابتدا وزن‌ها رندوم



انتخاب مناسب نقطه شروع در رسیدن به جواب بهینه تأثیر زیادی دارد.

• مینیمم محلی



وقتی شبکه به نقطه مینیمم می‌رسد از هر طرف که برود با افزایش خطا مواجه می‌شود.

شبکه نمی‌تواند از مینیمم محلی خارج شود ← مشکل روش‌های مبتنی بر شیب و گرادیان

• ناحیه محدب

ناحیه‌ای که اگر هر دو نقطه داخل آن را با یک خط به هم متصل کنیم، تمام آن در ناحیه قرار گیرد.

• عملکرد MLP در دسته‌بندی کننده

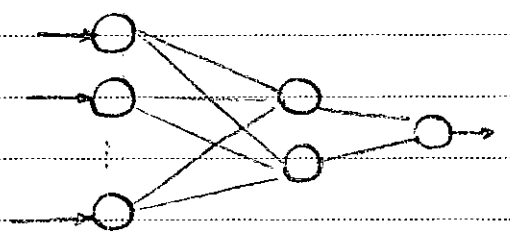
هر نورون یک ابرصفحه در فضای ورودی‌ها رسم می‌کند. (در فضای دو بعدی: خط)

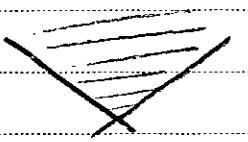
Subject:

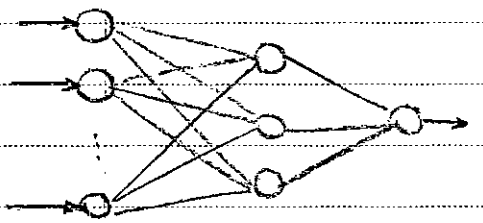
Year Month Date ( )

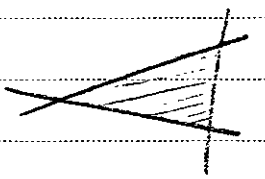
لایه مخفی 1 مجموعه ای از ابر صفحه ها ← لایه بعد : ترکیب ابر صفحه ها ← تشکیل نواحی محدب

تعداد نورون های لایه مخفی اول : تعداد ابر صفحه ها : تعداد لبه های ناحیه





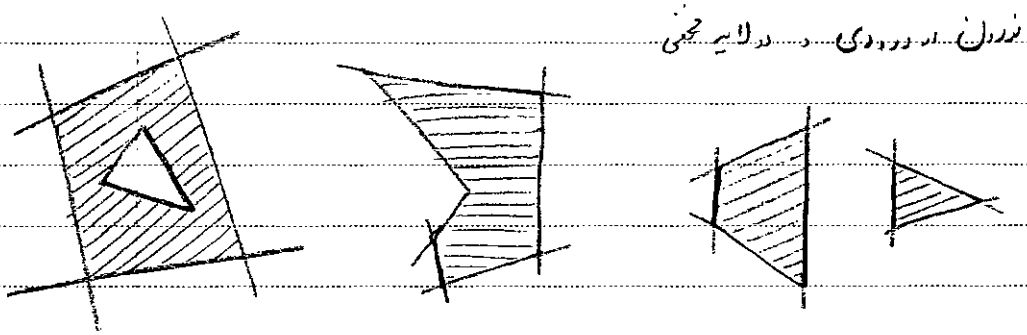




اگر پس از لایه مخفی اول یک لایه مخفی دیگر قرار دهیم ، خروجی های لایه قبل را به نواحی محدب

است ، هم ترکیب می کند و نواحی پیچیده تر تولید می کند ← نواحی غیر محدب ، غیر متصل ...

☑ ۷ نورون ورودی و ۲ لایه مخفی



• با داشتن دو لایه مخفی می توان فضا را به هر شکل دلخواه ناحیه بندی کرد . پس هر مساله دسته بندی با

یک سلول چهار لایه BP حل می شود : 1 لایه ورودی + 2 لایه مخفی + 1 لایه خروجی

Subject :

Year . . . Month . Date . ۹۵

MLP دولایه → → XOR ✗ ☑

MLP سه لایه → → XOR ✓

MLP چهارلایه → → XOR ✓

• قدرت حفظ کردن شبکه (Memorization)

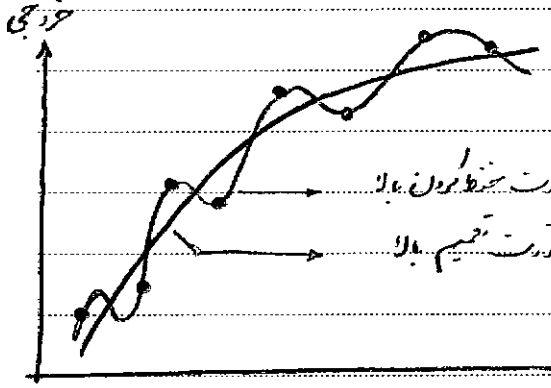
توان شبکه در پاسخ دادن صحیح به نمونه‌های آموزشی پس از خاتمه آموزش

• قدرت تعمیم شبکه (Generalization)

توان شبکه در پاسخ دادن صحیح به نمونه‌های جدید پس از خاتمه آموزش

• قدرت حفظ کردن و قدرت تعمیم شبکه در مقابل هم قرار دارند ← نیاز به برقراری تعادل

• گزینه پادگیری برای توان شبکه عمل برازش منحنی در نظر گرفت ← Curve Fitting



• شبکه: نگاشت غیرخطی از ورودی به خروجی

• تعمیم: درونیابی داده‌های ورودی

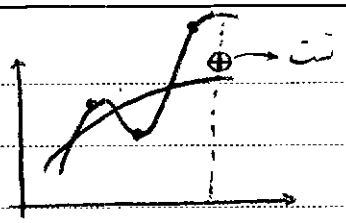
• overfitting: برون‌یابی نقاط

• درونیابی نقاط interpolation

Memorization ⟷ Generalization

PAPCO

Subject:
Year Month Date ( )



☑ باید ارتفاع خطاهای زیادی را متحمل می‌شویم

overFitting

تعداد زیاد زوج آموزشی و یا تعداد دفعات زیاد آموزش

شبکه ← حفظ دقیق تر نمونه ها ← اطمینان بقدرت تعمیم

تعداد واحدهای مخفی ← افزایش ← بیش از حد زیاد

← وجود آبدال مخفی های درخشان

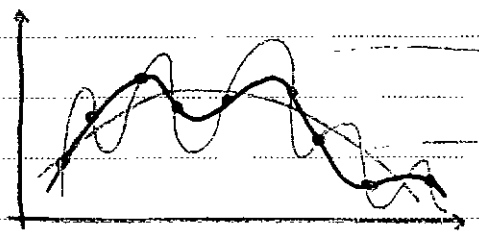
← شکل مخفی ← نرمالی

برازش مخفی ← پارامترهای زیاد ← جزئیات ریز و درشت

← وزن های بایاس بزرگ یا کوچک

خلاصه: زیاد بودن واحدها ← پارامترها ↑ ← تعمیم ↓ ← دنبال کردن جزئیات ↑

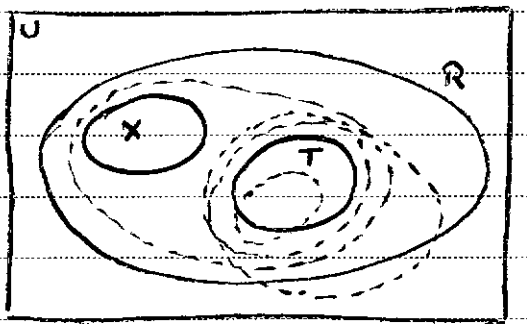
☑ تعداد لایه مخفی



Subject:
Year Month Date 9

• در مسائل یادگیری هدف ما استخراج اطلاعات منطقی در جهت خلاف این جمله در مسائله نباشد، ساده‌ترین و مرتبط‌ترین تابع که یادگیری را با صحت مورد نظر تعیین می‌دهد انتخاب می‌شود ← محاسبات کمتر

• با توجه به تعادل‌های موجود در مسئله و تبدیل اطلاعات پیچیده ← تعمیم همیشه دلخواه نیست.



U : تمام ورودی‌ها و خروجی‌ها

R : تابع مورد نظر

T : مجموعه آموزشی R

X : مجموعه آزمایشی

--- : تعمیم

پس از آموزش مسئله اطلاعاتی در مورد T پیدا کرده، نه X یا R

تعمیم‌های مسئله منطبق با T ← توانایی آن در زیاد است و جواب مورد نظر سخت بدست می‌آید.

① اجباراً تعمیم T را نمی‌پوشاند ② دوست داریم تعمیم X را بپوشاند

* تقریب توابع با HLP

• عمده روشها برمبنای از بین بردن سهل‌ترین مربع خطا ← عدم دانایی HLP ☺

• مسئله‌ی عمومی یادگیری: تقریب تابع

$f: A \subset \mathbb{R}^n \longrightarrow \mathbb{R}^m$

با داشتن نمونه‌های (S_i, T_i) که

$T_k = f(S_k)$

Subject:

Year. Month. Date. ( )

• فاز آموزش : استفاده از نمونه‌ها برای یادگیری تابع

• فاز آزمایش : ارزیابی عملکرد با نمونه‌های جدید آموزش دیده یا آزمایشی

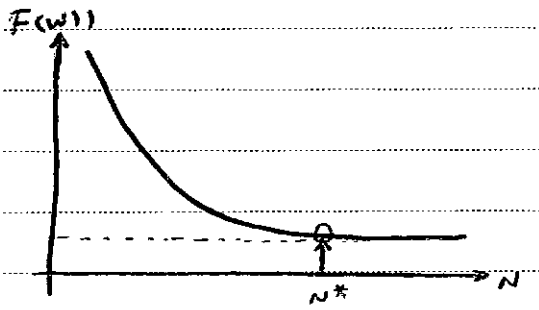
• معیار کار

$F_k(X_k, w) = |f(x_k) - g_k(X_k, w)|^2$: مربع خطای خروجی

$F(w) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} F_k(X_k, w) \geq 0$: میانگین

همگرایی برای یک سیستم خاص، با افزایش تعداد نمونه‌ها، به مقدار حداقلی میل می‌کند

• اندازه مجموعه آزمایشی مناسب (N^*)

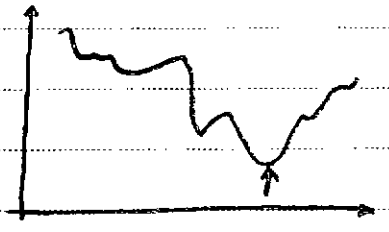


به تدریج اندازه مجموعه نمونه‌های آزمایشی را بزرگ‌تر می‌کنیم

تا جایی که خطا به سمت یک مقدار ثابت میل کند

• همواره افزایش تعداد نمونه‌ها ضروری نیست ← اگر خطا قابل تحمل و مطلوب باشد

• سطح خطای ساده



- محلی در فضای وزن‌ها ← $F(w)$

بدلیل آموزش ← پایین کشیدن سطح

Subject:

Year. Month. Date. ۹۲

• قضیه کولموگروف — Kolmogrov's Theorem

$f : [0,1]^n \longrightarrow \mathbb{R}^m$ تابع پیوسته دلخواه

تابع f را می‌توان به شکل بیش جمله‌ای سه لایه با n واحد ورودی در لایه اول، $2n+1$ واحد مخفی، و m واحد خروجی در لایه سوم پیاده سازی کرد.

واحد‌های لایه مخفی: $z_k = \sum_{j=1}^{n} \lambda^k \psi(x_j + k\varepsilon) + k$

$y_i = \sum_{j=1}^{2n+1} g_i(z_i)$

λ: مقدار حقیقی ثابت، وابسته به n، مستقل از f

ψ: تابع حقیقی پیوسته، اکیداً یکنوا صعودی، وابسته به n، مستقل از f

$\varepsilon = cte \in \mathbb{Q} \quad (0 < \varepsilon < \delta)$

g_i: تابع حقیقی پیوسته، وابسته به f و ε

قضیه کولموگروف تنها وجود سه لایه‌ای پیاده سازی برای تابع f را مطرح می‌کند ← g و ψ چی؟

تابع از $[0,1]^n$ به هر مجموعه فشرده قابل گسترش است. ← فشرده = بسته + کراندار

Subject:

Year. Month. Date. ( )

• اگر تابع $f : [0,1]^n \to \mathbb{R}^m$ در دست باشد، می‌گوییم f در L_2 است یا متعلق به L_2 است، اگر هریک از توابع مختصات f روی این مکعب واحد n-بعدی تابع قابل انتگرال گیری باشد یا: $\int_{[0,1]^n} |f(x)|^2 \, dx$ وجود داشته باشد.

اگر یک تابع برداری از یک متغیر برداری در L_2 باشد، هریک از مولفه‌هایش می‌تواند با هر درجه صحت دلخواه با بسری فوریه‌اش تقریب زده شود.

قضیه: فرض کنید

دلخواه $\varepsilon > 0$ ، L_2 ، $f : [0,1]^n \to \mathbb{R}^m$

یک سیگنال سه لایه با تعداد نورون‌های مخفی وجود دارد که می‌تواند f را با حداکثر متوسط خطای ε تقریب زند. فضای L_2 شامل هر تابعی که روی سطوح بافندی شود. (با درجه صحت دلخواه)

• هر تابع را می‌توان با یک شبکه سه لایه پیاده‌سازی کرد، اما اگر توابع پیچیده باشند:

تعداد نورون‌های لایه مخفی ↑ ← نیاز به لایه مخفی ↑

تعداد سطوح لایه‌های مخفی ↑ ← نیاز به لایه مخفی ↓

• نکته منفی این قضیه: اندازه شبکه

Subject:

Year. Month. Date. ۹۳

۲ . پیاده سازی شبکه

۱) SOM استاندارد

۲) MSOM

الگوریتم آموزش شبکه MSOM ← Modified

1. انتخاب تصادفی یک نمونه از بردارهای ورودی و ارسال آن به شبکه

2. یافتن نورون برنده

3. تغییر بردار وزن نورون برنده و نزدیک همسایگان

4. تکرار مراحل 1 تا 3 به تعداد دفعات اتفاقی

5. محاسبه عملکرد نورون‌ها و تعیین کم کاربردترین نورون و حذف آن از شبکه ← نورون برنده

نرخ : دفعات برنده شدن ÷ تعداد مراحل حاضر

6. اضافه کردن یک نورون با بردار وزن تصادفی به شبکه

7. بازگشت به مرحله 1

پیشنهاد دوم ← یک heuristic برای ساخت مجموعه

ساختن یک الگوریتم تطبیقی Steady State برای ساخت نورون جدید از روی نورون‌های موجود

PAPCO

Subject:
Year. Month. Date. ( )

* عوامل مؤثر در بهبود عملکرد شبکه BP

I. انتخاب اولیه وزنها

• رسیدن یا نرسیدن به مینیمم عمومی

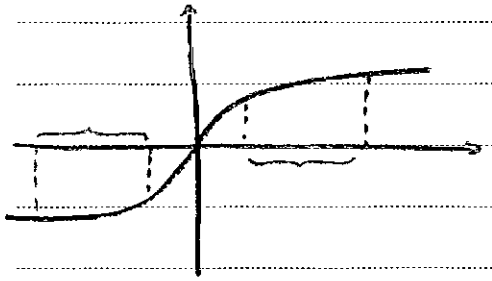
• سرعت رسیدن به مینیمم عمومی

• اگر وزنهای اولیه خیلی بزرگ ← توابع فعالیت لایه آخر: اشباع ← تغییرات کم

← f' کوچک ← اصلاح وزنهای کند

• اگر وزنهای اولیه خیلی کوچک ← ورودیهای خطی لایه آخر: کوچک ← اصلاح

وزنهای کند



• معمولاً وزنهای اولیه

$[-1, +1]$

$[-\frac{1}{2}, +\frac{1}{2}]$

• پیشنهاد ویدرو (widrow & Nguyen)

ابتدا همه وزنها به طور تصادفی از $[-0.5, +0.5]$

برای تعداد واحد n و تعداد واحدهای لایه مخفی p

$\beta = 0.7\sqrt[n]{p}$

P4PCO

Subject:

Year Month Date ۹۵

تغییر وزن‌های لایه‌ی رقابتی

$$w_{ij}(new) = \beta \times \frac{w_{ij}(old)}{\| w_{ij}(old) \|}$$

سپس باید مقدار بایاس اصلاح شود

$$w_{0j} \in [-\beta, +\beta]$$

یادگیری نسبتاً سریع تغییر می‌شود.

III. ارائه داده‌ها

- حل شدن الشدن مسئله ← حل ساده‌تر / دشوارتر
- پیوسته / گسسته ← در محدوده‌های مختلف
- یادگیری در مسئله‌های ساده‌تر متمایز گسسته، ساده‌تر و سریع‌تر انجام می‌شود.
- داده‌ها در نتایج نهایی رقابتی در مقایسه با بایزی ارجح است.

بایزی: داده‌ی غیرمتعادل ← عدم تأثیر در یادگیری

رقابتی: داده‌ی غیرمتعادل ← تأثیر در یادگیری ← عدم همگرایی سریع

III. تعداد واحدهای لایه‌ی خروجی

- افزایش تعداد واحدها ← تصحیح خطاهای بیشتر

کاهش تعداد نمونه‌های لایه‌ی خروجی (دسته‌بندی زشت‌تر)

Subject:

Year Month Date ( )

• اگر دو دسته مشابه به عنوان یک دسته در نظر گرفته شود، میانگین محلی ایجاد خواهد شد

• در واردی تعداد واحدهای نمونه نهایی باید دوردی مرتبط است با برابر تعداد نمونه‌ای آموزشی قطار داده می‌شود.

IV. تعداد نمونه‌های آموزشی لازم

• تئوری Hussler & Beam

اگر ① شبکه Nw وزن داشته باشد و ② خطای قابل قبول دسته‌بندی اندازه‌ی آزمایشی E باشد، با انتخاب Nw/E قوی آموزشی و آموزش شبکه تا جایی خطای اندازه‌ی آموزشی به E/2 برسد، معمولاً، خطای E برای نمونه‌های آزمایشی حاصل خواهد شد

Nw = 80 , E = 10 % = 0.1

یعنی $\frac{80}{0.1} = 800$ ← خطا < 5 %

V. مجموعه‌ی آموزشی، آزمایشی و میزان آموزش شبکه

• شبکه BP در ابتدای یادگیری سعی بر تقریب تابع با یک خرم هموار دارد ← دوردیابی

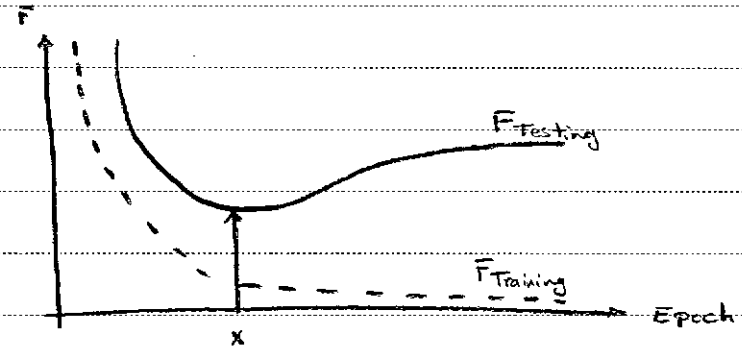
Subject:

Year. Month. Date. 95

• در ادامه آموزش نرم داده ها پیچیده تر شده و از داده ها بهره میبرند. ← عبور از نقاط

• تعادل : دورریزی و عبور از نقاط

• بیدا کردن باسن خطا



x : نقطه توقف آموزش
تقریباً معادل نقطه شروع
شدن خطای آموزش

VI. تشکیل مجموعه های آموزشی و آزمایشی

• داده ی زیاد ← آموزشی : نصف داده ها ← بخوبی به پرسش داده شده
آزمایشی : نصف داده ها

• داده ی کم (n تا) ← 1. آموزشی : همه داده ها ← توقف با نقطه شروع مسطح شدن نمودار F Training

2. n بار مانند با 1-n مجموعه آموزشی دهم و خطا برای مجموعه n ام محاسبه میکنیم ← متوسط مربعی خطاها
سنجش از صحت این شیوه است ← توقف آموزش هر بار در نقطه شروع مسطح شدن نمودار F Training
(leave One Out)

3. حجل داده از روی داده های موجود
که تغییر مناسب در داده ها ← نیروی شدن داده ها

نشانه خطی مناسب نیست!

حسن آن به ازای داده های کم!

داده های دیری نیز موجود است

Subject:

Year. Month. Date. ( )

* برای حل یک مسئله باید معماری آن با آسیب‌پذیری‌های احتمالی را در نظر بگیریم. در واقع یک حالت سعی و خطا برای پیدا کردن روش حل داریم.

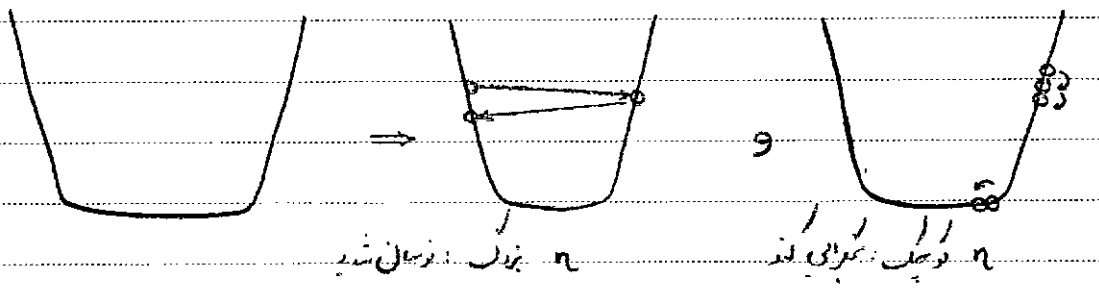
* نسخه‌های مختلف الگوریتم BP

- ضعف الگوریتم: گذر کردن آموزش و گیر افتادن در مینیمم‌های محلی
- چاره‌اندیشی‌ها ← روش‌های مبتنی بر تغییر نحوه اصلاح وزن‌ها
  - روش‌های مبتنی بر تغییر تابع فعالیت واحدها
  - برخی روش‌های ترکیبی دیگر

* ○ روش‌های مبتنی بر تغییر نحوه اصلاح وزن‌ها

- استفاده از مومنتم (اینرسی)

روش تردل را استاد گرادیان برای مسیول ← مینیمم دره‌هایی مانند زیر (چاه) مشکل می‌شود.



برای ایجاد اینرسی در وزن‌ها

؟ بیش برسل

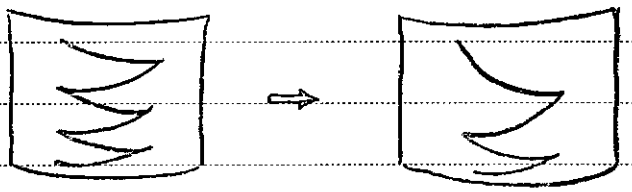
Subject:
Year: Month: Date: ۹۹

$$w_{ij}^{e}(t+1) = w_{ij}^{e}(t) + \alpha \delta_{p_j} y_{p_j}^{e-1} + \mu\left[w_{ij}^{e}(\quad)\right]$$

با افزودن جمله اینرسی ← حرکت با بهای بزرگتر ← احتمال پرش از روی مینیمم های محلی

↓ افزایش سرعت همگرایی

حرکت نقطه را بسمت پایین نقطه قبلی ← دخالت گرادیان نقطه قبلی



• بهنگام آوردن دسته ای وزنها Batch Mode

روش استاندارد یا روش اولیه الگوریتم (Pattern Mode) که وزن های سیناپس

از اعمال هر الگو بهنگام می شود. در حالت دسته ای، وزن های سیناپس از اعمال تعدادی الگو

استفاده می شود. در حالت دسته ای متوسط خطای ایجاد شده در خروجی برای اصلاح وزنها استفاده می شود.

$$E_{avg} = \frac{1}{2L} \sum_{p=1}^{L} \sum_{j \in C} e_{p_j}^2$$ با تعداد نمونه آموزشی دسته

$$e_{p_j} = t_{p_j} - y_{p_j}$$ C: مجموعه واحدهای خروجی سیناپس

Subject:

Year. Month. Date. ( )

• تطبیق کردن نرخ یادگیری

در ابتدا α بزرگ ⟸ در انتها α کوچک

↓ ↓

سرعت بالا — نوسانات کم

عبور از مینیم های محلی — دقت بالا

• تطبیق کردن نرخ یادگیری دلتا ـ باردلتا

هر وزن دارای نرخ یادگیری مخصوص خود ⟸ وزنها به تدریج با زمان تغییر می کنند

قاعده تجربی برای اصلاح نرخ یادگیری : اگر تغییرات وزن در چند دوره متوالی در یک جهت باشد

نرخ یادگیری ↑ (در جهت کاهش وزن)

نرخ یادگیری ↓

$$w_{jk}(t) = w_{jk}(t-1) - \alpha_{jk}(t) \frac{\partial E}{\partial w_{jk}}$$

$$= w_{jk}(t-1) + \alpha_{jk}(t)\, \delta_k\, y_j$$

هر وزن بالسبت متناوبی از دفعه قبل تغییر می کند ⟸ در استاد گرادیان منفی

$$\Delta_{jk} = \frac{\partial E}{\partial w_{jk}(t)} = -\delta_k\, y_j$$ واحد خروجی

$$\Delta_{jk}^{(l)} = -\delta_j^{(l)} \cdot y_j^{(l-1)}$$ واحدهای لایه مخفی l

Subject:

Year Month Date ۹۷

$$\bar{\Delta}_{jk}(t) = (1-\beta)\,\Delta_{jk}(t) + \beta\,\bar{\Delta}_{jk}(t-1)$$ لایه خروجی

$$\bar{\Delta}^{e}_{ij}(t) = (1-\beta)\,\Delta^{e}_{ij}(t) + \beta\,\bar{\Delta}^{e}_{ij}(t-1)$$ لایه مخفی e

↳ ایجاد حافظه ↳ $0 < \beta < 1$

$$\alpha_{jk}(t) = \begin{cases} \alpha_{jk}(t-1) + \chi & , \ \Delta_{jk}(t-1) \times \bar{\Delta}_{jk}(t-2) > 0 \\ \alpha_{jk}(t-1) \times (1-\gamma) & , \ \Delta_{jk}(t-1) \times \bar{\Delta}_{jk}(t-2) < 0 \\ \alpha_{jk}(t-1) & , \ \text{otherwise} \end{cases}$$

سه پارامتر آزاد $0 < \beta, \gamma, \chi < 1$ ← در مسیر !

χ ↓ : راه‌هایی نه چندان قوی در ابتدا
χ ↑ : رد شدن از دره راه‌ها
} ← ترجیح دارد

☑ شبکه XOR با بیش از یک لایه ۱۰ : ۹ : ۶

۲۰ بار شبیه‌سازی ← ادامه تا بهبود شدن

استاندارد > پارامتری متغیر > دلتا بار - دلتا > مومنتم : میانگین مدل دقت بهبودی

☑ شناسایی چهار حرف A, B, C, D با شبکه چهار لایه ۳ : ۲۰ : ۲۰ : ۱۲۸

مومنتم > استاندارد > نرخ یادگیری متغیر > دلتا بار - دلتا

Subject:

Year. Month. Date. ( )

استاندارد

با مختم

۵۵

تعداد دور لازم / تعداد پیدایی

☑ پیاده‌سازی تابع XOR دوبیتی با شبکه سه لایه ۲:۲:۱

اصلاح روش: Batch

مقاله بادلیری: بادسیون مجموع مربع خطا در ۵۰ دور برابر متوسط گیری شده بالاتر از ۰.۰۴

آزمایش با ۲۵ دسته مقادیر اولیه مختلف

مقادیر پارامترها:

	α	μ	χ	γ	β
استاندارد	0.1				
مختم	0.75	0.9			
۵۵	0.8		0.035	0.333	0.75

نتایج:

	تعداد شبیه‌سازی	تعداد پیدایی	تعداد متوسط دور برابر
استاندارد	۲۵	۲۴	۱۶۸۴۰
مختم	۲۵	۲۵	۲۰۵۶
۵۵	۲۵	۲۲	۴۴۷

از پیدایی شود سریع است

* ادعای مبنی بر تغییر تابع مخالفت واحد

• اگر تابع مخالفت یادمتقارن باشد ← سرعت یادگیری ↑

$f(-x) = -f(x)$

Subject:

Year Month Date ۹۸

۱

• تابع سیگموید باینری

$\frac{1}{1+e^{-\alpha x}}$

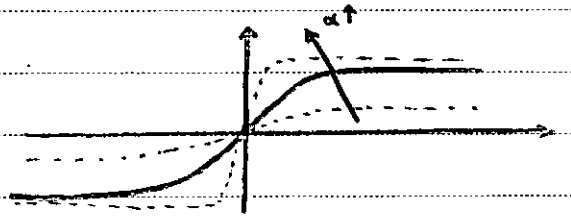
تابع سیگموید دوقطبی

$(b-a)\frac{1}{1+e^{-\alpha x}}+a$

تابع سیگموید با محدوده دلخواه $[a,b]$

$(b-a)\frac{1}{1+e^{-\alpha(t)x}}+a$

تابع سیگموید با شیب متغیر



α متغیر ← بهبود عملکرد شبکه

مانند وزنها

↑ شیب ← ↑ α

$X = \alpha_k^{(l)} I_{pk}^{(l)}$

$y_{pk}^{(l)} = f(X)$

$\delta_{pj} = [t_{pj} - y_{pj}] f'(y_{pj})$ لایه خروجی

$\delta_{pk}^{(l)} = -\sum_k \delta_{pk}^{(l+1)} \alpha_k^{(l+1)} w_{jk}^{(l+1)} f'(y_{pj}^{(l)})$ لایه مخفی

$\Delta w_{jk}^{(l)} = \eta \delta_{pk}^{(l)} \alpha_k^{(l)} y_{pj}^{(l-1)}$

$\Delta \alpha_k^{(l)} = \eta \delta_{pk}^{(l)} I_{pk}^{(l)}$

• استفاده از الگوریتم‌های ژنتیک، سایر روشهای تطابقی، مدل‌های مقاوم

Subject:

Year. Month. Date. ( )

* هر کدام از روش‌ها بسته به نوع مسئله بهتر یا بدتر عمل می‌کند ☐ روش مهم در دنیای فیزیک

Subject:
Year. Month. Date. ۹۹

شبکه‌های توابع پایه شعاعی Radial Basis Function Networks

* جدایی‌پذیری الگوها

اگر در یک مسئله بعد فضای ورودی را بالا ببرد، می‌تواند منجر به جدایی‌پذیری الگوها شود

$$X = \{\underline{X}_i\} \Rightarrow \forall \underline{X}_i : \underline{X}_i \in X^+ \vee \underline{X}_i \in X^-$$

$$\underline{\Phi}(\underline{X}) : \mathbb{R}^n \longrightarrow \mathbb{R}^m \quad , \quad m \geq n$$

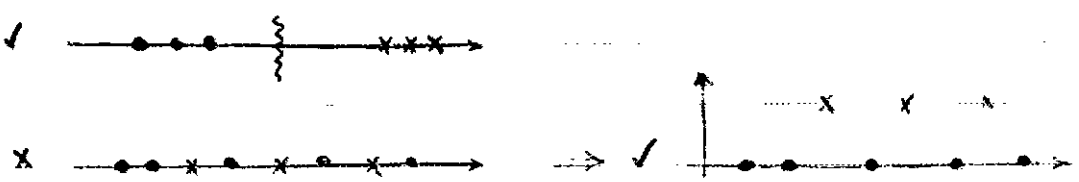
$$\underline{\Phi}(\underline{X}) = [\Phi_1(\underline{X}), \Phi_2(\underline{X}), \ldots \Phi_m(\underline{X})]^T$$

اگر مجموعه $\{X^+, X^-\}$ از X با Φ جدایی‌پذیر، آنگاه بردار m بعدی W وجود خواهد داشت:

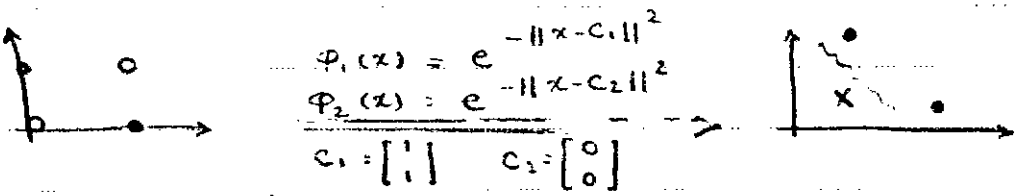
$$\begin{cases} \underline{W}^T \underline{\Phi}(\underline{X}) > 0 & \underline{X} \in X^+ \\ \underline{W}^T \underline{\Phi}(\underline{X}) < 0 & \underline{X} \in X^- \end{cases}$$

این صفحه سطح جدا کننده در فضای Φ

$$\underline{W}^T \underline{\Phi}(\underline{X}) = 0$$



XOR



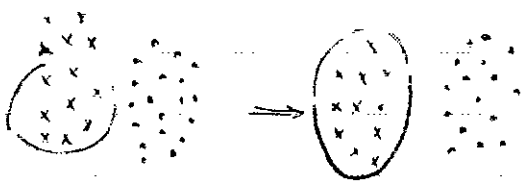
بدون افزایش بعد هم می‌توان مسئله جدایی‌ناپذیر را به جدایی‌پذیر تبدیل کرد

Subject :
Year . Month . Date . ( )

- وزنهای لایه خروجی با الگوریتم LMS قابل تعیین است.
- می‌توان پس از اتمام آموزش شبکه، آنها را مجدداً آموزش داده و تنظیم دقیق‌تر کرد.

* تعمیم نرم تابع از تقارن شعاعی به تقارن بیضوی Radial Sym → Elliptical Sym

- جایگزینی $(x_i - c_j)$ با $(x_i - c_j) / \beta_j$
- می‌توان تعداد واحدهای لایه مخفی را کاهش داد.
- توانایی آن در کاربردهای دسته‌بندی بالاتر می‌رود ← دایره به بیضی
- عملکرد شبکه در کاربردهای آنالوگ و ترکیب توابع بهبود می‌یابد ← شبکه RBF تعمیم یافته
- تبدیل فاصله اقلیدسی به فاصله ماهالانوبیس



* یادگیری با انتخاب یا نظارت مراکز

- مراکز RBF و همه پارامترهای دیگر تحت یک فرآیند یادگیری با نظارت انتخاب می‌شوند.
- استفاده از روش نزول در امتداد گرادیان

Subject:

Year. Month. Date. ۷۲

$$e_j = d_j - \sum_{i=1}^{M} w_i \underbrace{G(\|x_j - c_i\|_{D_i})}_{\text{نرم } D_i}$$

$$= (x_j - c_i)^T D_i^T D_i (x_j - c_i)$$

$$G(\|x_j - c_i\|_{D_i}) = e$$

$$D_i^T D_i = \frac{1}{2} S_i^{-1}$$

$$E_0 = \frac{1}{2} \sum_{j=1}^{L} e_j^2$$

هدف تعیین w_i و c_i و S_i^{-1} به طریقه خطا حداقل شود.

با استفاده از روش نزول گرادیان

- وزن‌های لایه خروجی

$$w_i(n+1) = w_i(n) - \eta_1 \frac{\partial E(n)}{\partial w_i(n)}$$

$$\frac{\partial E(n)}{\partial w_i(n)} = \sum_{j=1}^{L} e_j(n) \cdot G(\|x_j - c_i(n)\|_{D_i})$$

- مراکز توابع

$$c_i(n+1) = c_i(n) - \eta_2 \frac{\partial E(n)}{\partial c_i(n)}$$

$$\frac{\partial E(n)}{\partial c_i(n)} = 2 w_i(n)$$

- توابع توزیع

$$S_i^{-1}(n+1) = S_i^{-1}(n) - \eta_3 \frac{\partial E(n)}{\partial S_i^{-1}(n)}$$

Subject :

Year . Month . Date . ( )

$$\frac{\partial E(n)}{\partial S_i^{-1}(n)} = -\omega_i(n) \sum_{j=1}^{L} e_j(n)\, G'(\|\underline{x}_j - \underline{c}_i(n)\|_{D_i})\, Q_{ji}(n)$$

$$Q_{ij}(n) = [\underline{x}_j - \underline{c}_i(n)][\underline{x}_j - \underline{c}_i(n)]^T$$

به دلیل نزدیک در استاد کلاس این محاسبه است در میتوانیم آن را نادیده بگیریم

* تقریب توابع با شبکه RBF

• مجموعه نقاط X_i و متناظر آنها d_i ← تابع f را تقریب بزنید

$$\underline{X}_i \in \mathbb{R}^n \quad , \quad d_i \in \mathbb{R} \quad , \quad i = 1, \dots, L$$

$$f(\underline{X}_i) = d_i \quad , \quad i = 1, \dots, L$$

$$f(\underline{X}_i) = \sum_{j=1}^{L} \omega_j \varphi(\|x_i - c_j\|)$$

$$= \sum_{j=1}^{M} \omega_j \varphi_{ij}$$

φ_{ij} توابع پایه شعاعی دلخواه

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{1L} \\ \varphi_{21} & \varphi_{22} & \varphi_{2L} \\ \varphi_{L1} & \varphi_{L2} & \varphi_{LL} \end{bmatrix} \times \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_L \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_L \end{bmatrix}$$

Φ : ماتریس درونیابی

$$\underline{w} = \Phi^{-1} d$$

Subject:

Year Month Date ۷۲

• چند نمونه‌ای از توابع φ

$$\varphi(x) = e^{-\frac{x^2}{2\sigma^2}}$$

σ : پارامتر شاع اثر تابع

$$\varphi(x) = \frac{x}{\sigma^2} \log \frac{x}{\sigma}$$

$$\varphi(x) = \sqrt{x^2 + \sigma^2}$$

$$\varphi(x) = \frac{1}{\sqrt{x^2 + \sigma^2}}$$

• مقایسه RBF و MLP

	RBF	MLP
سرعت	↑↑	↓↓
یادگیری تضمین شده	✓	X
زمان یادگیری	↑	↓
حافظه	↑↑	↓
تعمیم	↑	↓

۳ : پیاده سازی شبکه RBF و RBF تعمیم یافته

- دسته بندی
- تقریب تابع

Subject:

Year. Month. Date. ( )

◎ بهینه سازی Optimization

* در مسئله بهینه سازی تابعی به نام تابع هزینه وجود دارد. هدف پیدا کردن حداقل آن است.

* روش های سنتی بهینه سازی ← ضرایب خطی لاگرانژ
برنامه ریزی خطی
برنامه ریزی پویا

* بهینه سازی با شبکه های عصبی

• الگوریتم خوب عمل می کنند

• در مواردی که روشهای سنتی خوب عمل می کنند، شبکه های عصبی جوابی نزدیک به بهینه می دهند.

☑ بهینه سازی تحت محدودیتهای متناقض

• انواع شبکه های عصبی ← قطعی (deterministic)
مثل برنامه ریزی خطی
احتمالی (stochastic)

* تابکاری فلزات Annealing

• گرم کردن فلزات تا نزدیک ذوب ← سپس سرد کردن تدریجی

رفع تراکم دسته ای شبکه های ژنتیکی ← استدلالات جدید ایجاد می شود.

Subject:

Year. Month. Date. ۷۸

• مجموعه حالات ممکن یک ذره $\{\alpha_i\}$

E_{α_i} انرژی کل سیستم در حالت α_i

T درجه حرارت کل سیستم

← اگر T بزرگتر از صفر مطلق باشد سیستم دارای نوسانات حرارتی خواهد بود.

• نوسانات حرارتی: انتقال از حالتی به حالت دیگر که انرژی را به طور جزئی کم یا زیاد می‌کند

• حالت تعادل حرارتی: نوسانات سیستم حول مقادیر متوسط ثابتی و از این مقادیر دور نمی‌شود.

• احتمال رفتن سیستم به حالت α_i از حال تعادل

$$P_{\alpha_i} = \frac{1}{Z} e^{-\frac{E_{\alpha_i}}{k_B T}}$$

(تابع توزیع بولتزمن - گیبس)

• تابع احتمال بولتزمن

$$Z = \sum_i e^{-\frac{E_{\alpha_i}}{k_B T}}$$

• فرض $k_B = 1$

$$\frac{P_{\alpha_i}}{P_{\alpha_j}} = e^{-\frac{(E_{\alpha_i} - E_{\alpha_j})}{T}}$$

• $T \to \infty \Rightarrow \frac{P_{\alpha_i}}{P_{\alpha_j}} \to 1 \Rightarrow P_{\alpha_i} \simeq P_{\alpha_j}$

یعنی در حرارت بالا، احتمال رفتن سیستم به حالت با انرژی بالاتر مساوی احتمال رفتن سیستم به حالت با انرژی پایین‌تر ← انرژی هم بالا می‌تواند برود و هم پایین ← فرار از سیستم محلی

Subject:

Year. Month. Date. ( )

* شبیه‌سازی تبلوری — Simulated Annealing (SA)

• Analogy ⟵ تابع انرژی بهینه‌سازی ⟵ تابع انرژی فلز

متغیرهای مسئله ⟵ ذرات فلز

بدل نقطه استقراری که تابع هزینه را کاهش می‌دهد ⟵ کاهش سریع دما

گیر افتادن در بهینه محلی ⟵ تکان‌دادن و احتمالات پرشی

اجازه بالارفتن از بهینه محلی ⟵

• روش SA یک جنبه انتزاعی به روش بدل در استار کردن اضافه می‌کند که پارامتر آنیل می‌شود

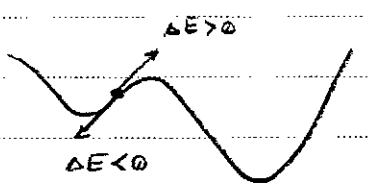
• الگوریتم هم اجازه انتقال به انرژی بالاتر را می‌دهد هم پارامتر

• $\alpha_i \longrightarrow \alpha_j$

$E\alpha_i \longrightarrow E\alpha_j$

$\Delta E = E\alpha_j - E\alpha_i$

$$\Delta E \begin{cases} \text{احتمال پذیرفته‌شدن انتقال} = 1 & , \Delta E \leq 0 \\ \text{احتمال پذیرفته‌شدن انتقال} = e^{-\frac{\Delta E}{T}} & , \Delta E > 0 \end{cases}$$



$$P(\Delta E) = \begin{cases} e^{-\Delta E/T} & , \Delta E > 0 \\ 1 & , \Delta E \leq 0 \end{cases}$$

Subject:

Year Month Date ۷۵

* حداقل کردن $f(\underline{x})$ با $\underline{x}$ باینری

1. انتخاب تصادفی بردار $\underline{x}$

انتخاب مقدار اولیه بزرگ T بطوریکه $e^{-\frac{\Delta E}{T}} > 0.999$

2. انتخاب مؤلفه x_i از $\underline{x}$ به طور تصادفی یکنواخت

3. تعیین $\underline{x}'$ ← همه مؤلفه‌ای $\underline{x}'$ برابر $\underline{x}$ بجز مؤلفه i ام $(x_i' = \bar{x}_i)$

$\Delta E = f(\underline{x}') - f(\underline{x})$

4. اگر $\Delta E < 0$: $\underline{x} = \underline{x}'$

5. اگر $\Delta E > 0$: $\underline{x}$ را با احتمال $e^{-\frac{\Delta E}{T}}$ برابر $\underline{x}'$ قرار دهیم

• انتخاب تصادفی یکنواخت ξ از $[0,1]$

• اگر $\xi < e^{-\frac{\Delta E}{T}}$ باشد $\underline{x} = \underline{x}'$ در غیراین صورت تغییر نمی‌کند.

6. اگر M بار تغییر موفق در $\underline{x}$ باشد یا N بار تغییر $\underline{x}$ از زمان آخرین تغییر در درجه حرارت آنگاه

$T = \alpha \times T \quad , \quad 0.8 < \alpha < 0.999$

7. اگر مقدار حداقل f در L مقدار آخر بیش از ε تغییر نیافته ← خاتمه

در غیراین صورت ← قدم 2

$N \ll L$

Subject:

Year Month Date ( )

• بهینه‌یابی در بی‌نهایت ← جواب‌های خوب با زمان محدود

• توابع چندین رفتار ← تعداد بسیار زیاد نقاط بهینه محلی

جواب‌های بدست آمده نزدیک به بهینه سراسری

* اعمال روش SA به سلول پیلبد است

• تغییر توابع تبدیل

$$y_i(new) = \begin{cases} y_i(old) & \text{otherwise} \\ -y_i(old) & \Delta E < 0 \\ -y_i(old) & \Delta E \geq 0 \ \wedge \ \xi < e^{-\frac{\Delta E}{T}} \end{cases}$$

$$\Delta E = 4\, y_i(old) \left[\sum_{j=1}^{n} w_{ij}\, y_i(old) - \theta_i \right]$$

• پاسخ در دمای بالا : تعداد زیاد پاسخ‌های قابل قبول / پاسخ در دمای پایین

• پاسخ حتی در دمای پایین کامل نیست معمولاً نزدیک پاسخ بهینه است.

• بعضی از این سلول برلزمان می‌گویند (Hopfield + SA)

• مانند سلول پیلبد تنها یک واحد بهینه می‌شود ← انتخاب تصادفی

Subject:
Year Month Date ٧٩

* ماشین بولتزمان Boltzman Machine

• در سال ۱۹۸۳ توسط هینتون و سجنوفسکی

• واحدها دارای حالت باینری و احتمال حالات احتمالاتی

• وزنها دارای مقادیر ثابت ← آموزش ندارند

• برون زا: نشانه‌ای از میزان مطلوبیت فعال بودن هر یک از واحدهای نرون

• مجموعه‌ای از واحدها ، ارتباطات دوطرفه ، وزنهای متقارن ، ارتباط خود با خود

• هدف: حداکثر کردن تابع آرا (Consensus Function)

$$C = \sum_{i} \sum_{j<i} w_{ji}\, y_i\, y_j$$

• حداکثر کردن تابع مزبور توسط SA

• اگر واحد i اجازه تغییر حالت داده شود،

$$\Delta C(i) = [1 - 2x_i]\left[w_{ii} + \sum_{j \neq i} w_{ij}\, x_j\right]$$

• شانس این تغییر حالت را با احتمال زیر بپذیرد.

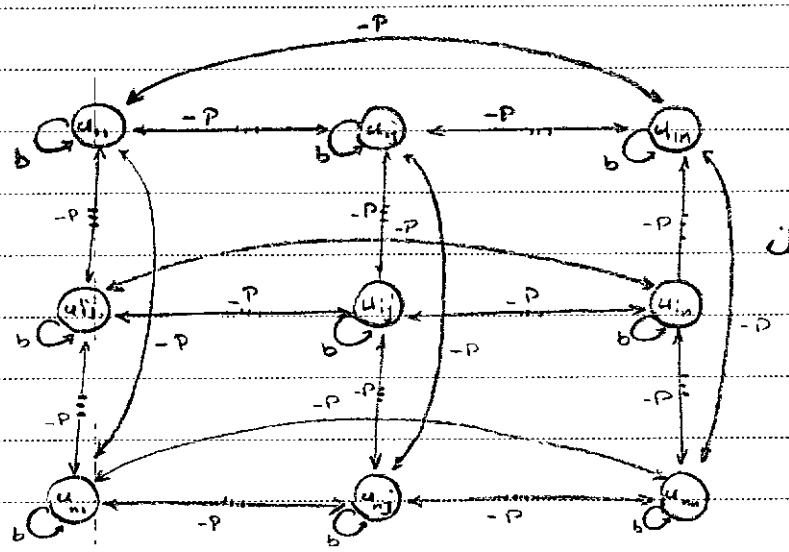
$$A(i,T) = \frac{1}{1 + e^{-\frac{\Delta C(i)}{T}}}$$

• T پارامتر تدریجی

Subject:

Year . . . Month . . . Date . . . ( )

• معماری وابسته به کاربرد ← تعداد دوده‌ها ، وزنه ، ...



آرایه دوبعدی
اتصالات کامل بطرفین
وزن b ثابت
(-P برای سیاه ،
b برای بقیه)

وزنه بین واحد (i,j) ، (I,J)

$w(i,j;I,J)$ ⟶ $w(i,j;i,j) = b$

$w(i,j;I,J) = -P$

الگوریتم:

1. مقادیر اولیه وزنه ← مجموعیت های سلف را در برگیرد

مقدار اولیه دما ← T_0

مقدار اولیه واحد ← بردی یا تصادفی

2. تا ارضا شدن شرط خاتمه

!

Subject:

Year. Month. Date. ۷۷

3. $1\ epoch = n^2$ بار انجام دهید

4. انتخاب متغیر صحیح I,J بین 1 و n

5. محاسبه تغییر تابع آرا

$$\Delta C = [1 - 2\,U(i,j)]\left[b + \sum\sum_{i,j \neq I,J} w(i,j,I,J)\,U(i,j)\right]$$

6. احتمال تغییر حالت

$$A(T) = \left(1 + e^{-\frac{\Delta C}{T}}\right)^{-1}$$

تصمیم‌گیری در مورد قبولی یا رد تغییر حالت

قبولی: $U_{ij} = 1 - U_{ij}$

رد: $U_{ij} = U_{ij}$

8. کاهش دما $T(new) = 0.95 \times T(old)$

9. شرط خاتمه

- دمای حداقل؟
- عدم تغییر برای مدت طولانی؟

Subject:

Year ____ Month ____ Date ____ ( )

☑ حل مسئله فروشنده دوره گرد (TSP) با این روش

n شهر ← نشان دهنده شهر

ن ترتیب بازدید شهر

$U_{ن,i}$ ← واحدی که فرض کند در شهر i ام در مرحله ن ام از سفر

فعالیت آن : $U_{ن,i} = 1$ ← فرض $U_{ن,i}$ صحیح

$U_{ن,i} = 0$ ← فرض $U_{ن,i}$ نادرست

d_{ik} ← فاصله بین شهر i و k

بزرگترین فاصله = d

	1	2	...	n
A	U_{A1}	U_{A2}		U_{An}
B				
:				
J	U_{J1}			U_{Jn}

جدول : آرایه دوبعدی

هر سطر یک شهر

هر ستون ترتیب عبور

در هر ستون فقط یک بار و در هر سطر فقط یک بار فعال خواهد بود (شرایط سؤال)

معماری : خط انتقالی با وزن b ← ظرفیت عبور از شهر i

انتقال به سایر واحدهای سطر i = تابع عبور از هر شهر ← P

انتقال به سایر واحدهای ستون = آن راحد در یک شهر

انتقال واحد $U_{ن,i}$ به واحد $U_{ن+1,k}$ با وزن d_{ik}

← انتقال فاصله طی شده از شهر i تا شهر k در مرحله ن+1

$p > d$: تابع آن برای یک راه حل تا ثبل ستون مقدار بزرگتری دارد تا راه حل غیر تا ثبل ستون

$p > 2d$: تابع آن برای یک راه حل تا ثبل ستون مقدار بزرگتری دارد تا راه حل تا ثبل ستون طولانی تر

Subject:

Year. Month. Date. ۷۸

۱۰ شبکه GMDH

Group Method of Data Handling

* توسط آیواخنکو ، ۱۹۶۸ ، برای پیش‌بینی جمعیت ماهی‌ها در رودخانه‌ها و اقیانوس‌ها

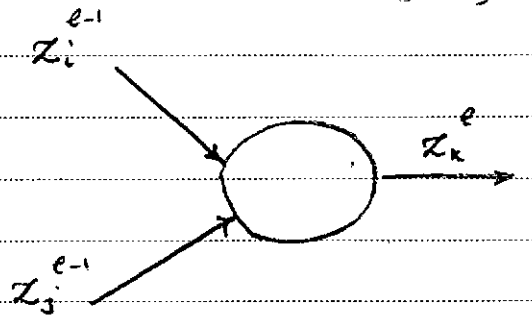
* کاربرد آن عموماً برای تقریب تابع پیوسته

$f: \mathbb{R}^n \longrightarrow \mathbb{R}$

برای تقریب توابع برداری باید برای هر مولفه بردار خروجی چنین شبکه‌ای داشت.

* معماری

• هر واحد دارای تنها دو ورودی به جز واحدهای لایه ورودی



• تابع تبدیل

$$z_k^e = a_k^e (z_i^{e-1})^2 + b_k^e (z_i^{e-1} z_j^{e-1}) +$$

$$c_k^e (z_j^{e-1})^2 + d_k^e (z_i^{e-1}) + e_k^e (z_j^{e-1}) + f_k^e$$

• با زیاد شدن هر لایه در شبکه، درجه چندجمله‌ای ایجاد شده در 2 ضرب می‌شود در شبکه به جز لایه ورودی h سطح داشته باشیم، چندجمله‌ای از درجه 2^h خواهد بود ← چندجمله‌ای آیواخنکو Ivakhnenko

Subject:

Year Month Date ( )

* ساخت شبکه

• کل ساخت لایه به لایه

• در هر لایه‌ی هر زوج خروجی لایه قبل یک واحد قرار می‌دهیم

• تعداد واحدها در لایه l برابر M_l

تعداد واحدها در لایه $l+1$ برابر $\binom{M_l}{2}$

• هر لایه اضافه می‌شود به طور مستقل تطبیق داده و ثابت می‌شود.

• هر واحد در هر لایه (به جز ورودی) تلاش می‌کند خروجی خودش را به بهترین حالت برساند.

• تنظیم θ پارامتر هر واحد با استفاده از تعداد زیادی نمونه ورودی - خروجی انجام می‌شود.

• هر لایه که اضافه می‌شود، به طور مستقل از سایر لایه‌ها تطبیق داده می‌شود و هر لایه به طور مستقل آموزش داده می‌شود. این موضوع باعث می‌شود برای دیپ‌لرنینگ‌ها هم لایه‌ها در آموزش سریعتر می‌شوند.

• اعمال ورودی S_p خروجی لایه‌ی $l-1$ تولید شده و به l می‌رود.

• هر واحدی خروجی مثلاً مطلوب t_p را تولید کند ← آموزش نیاز ندارد.

• برای هر واحد در هر لحظه آموزشی $\langle S_p, t_p \rangle$ ← یک معادله، θ مجهول

• هر لایه با تجدید آموزش مجدد آموزش داده می‌شود.

Subject:
Year Month Date ۷۹

• بدست آوردن پارامترهای : معادلات حاصل از آموزش یک واحد k در لایه l

$$\begin{bmatrix} (z_i^{l-1})^2 & (z_i^{l-1} \times z_j^{l-1}) & (z_j^{l-1})^2 & z_i^{l-1} & z_j^{l-1} & 1 \\ \vdots & & & & & \end{bmatrix} \times \begin{bmatrix} a_k^l \\ b_k^l \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ \\ t_e \end{bmatrix}$$

$$T, \; z_k^{l-1} \Rightarrow z_k^{l-1} \times P_k^l = T$$

• این معادله دارای پاسخ تقریبی خطی بود، چون غیرممکن است هر وزنه آموزشی خروجی مطلوب تولید کند.

• حداقل کردن خطا (MSE)

$$P_k^l = \underbrace{(z_k^{l-1})^+}_{\text{Generalized Inverse}} \cdot T$$

• با تعیین پارامترهای هر واحد در سطح l عملکرد هر واحد در رسیدن به هدف حداقل کردن MSE بایک مجموعه آزمایشی ارزیابی می‌شود ← r : تعداد نمونه‌های مجموعه آزمایشی

$$e = \frac{1}{r} \| z_k^{l-1} P_k^l - T \|^2$$

• اگر این خطا (e) برای واحدی از یک مدل حد آستانه بزرگتر باشد، واحد نامطلوب را حذف می‌کنیم.

با این روش واحدهای بعدی پس از دیگری به لایه‌ها ایجاد می‌شوند تا سرانجام معیار توقف ارضا شود.

Subject:

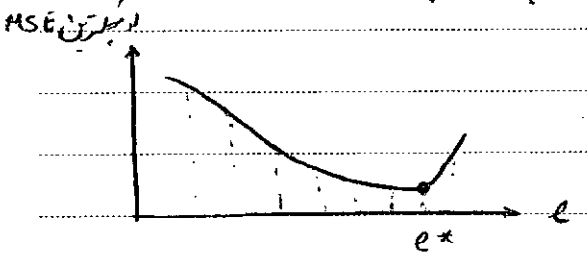
Year Month Date ( )

• روند: ترکیب هر جفت نرون ← آموزش

ورودی‌ها ← ساختن لایه

آستانه‌گیری ← می‌سنجیم خطا

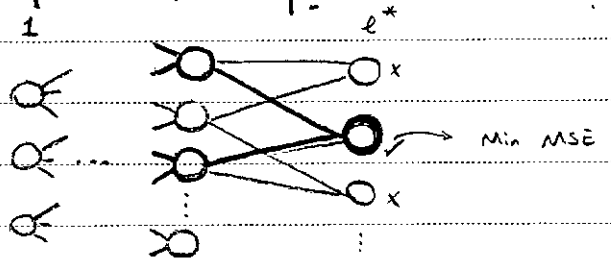
• چندلایی بودن: در هر لایه کوچکترین MSE را پیدا می‌کنیم



e*: همین تعداد لایه کافی است

نرون دارای کوچکترین e را خروجی در نظر می‌گیریم و بقیه نرون‌های لایه e* را حذف می‌کنیم

سپس اتصالات نرون‌هایی که منتهی به خروجی نمی‌شوند را حذف می‌کنیم ← این روش نظم شبکه



• نحوه تعیین اندازه مجموعه آموزشی

- باید مجموعه پارامترها را محاسبه کنید.
- اندازه این مجموعه را زیاد کرده و مجدداً محاسبه کنید.
- اگر تغییر پارامتر زیاد نباشد، اندازه مجموعه آموزشی کافی بوده است.

GMDH ... ۳

Subject:
Year Month Date ۸۵

Adaptive Resonance Theory — نظریه تشدید تطبیقی

* ۱۹۸۷ ، گراسبرگ

* منجر به توسعه سه شبکه
  - ART 1 : خوشه‌بندی برداری باینری
  - ART 2 : خوشه‌بندی برداری پیوسته
  - ART 3 : اصلاح شده ART
  - Fuzzy ART MAP , ART MAP : بعدی

* مسائل مطرح شده
  - پایداری شبکه
  - یادگیری از نمونه‌های جدید برای یک سیستم آموزش دیده ← Incremental learning
* بروز بدون نرخ یادگیری ← یادگیری رقابتی
  - تضمینی وجود ندارد که دسته‌بندی تشکیل شده پایدار بماند
  - دسته‌بندی‌های نمونه ممکن است بدون پایان تغییر کنند
  - راه حل: کاهش تدریجی نرخ یادگیری ← از دست دادن توان شبکه برای یادگیری نمونه‌های جدید
  - مسئله پایداری و تأثیرپذیری یا مسئله Stability - Plasticity Dilemma

Subject:

Year Month Date ( )

• حتی اگر مسئله پایداری ـ تاثیر پذیری حل شود باز هم تعداد دسته‌ها یا خوشه‌ها که توسط تعداد نورون‌های لایه آخر مشخص می‌شود ← عدم امکان تولید خوشه جدید

↙ مشکل عدم تعلق داده جدید به هیچ دسته‌ای

• راه حل دیگر: با ورود داده جدید، تمام داده‌ها از اول آموزش داده شود ← هزینه بر

* شبکه ART 1:

• دیتای ورودی ← دسته‌بندی داده‌های باینری

↙ توانایی تغییر حالت بین دو حالت تاثیر پذیری و پایداری

• در حالت تاثیر پذیری: یادگیری و تغییر پارامترهای شبکه

• در حالت پایداری: بدون صدمه به یادگیری قبلی شبکه، دسته‌بندی ثابت

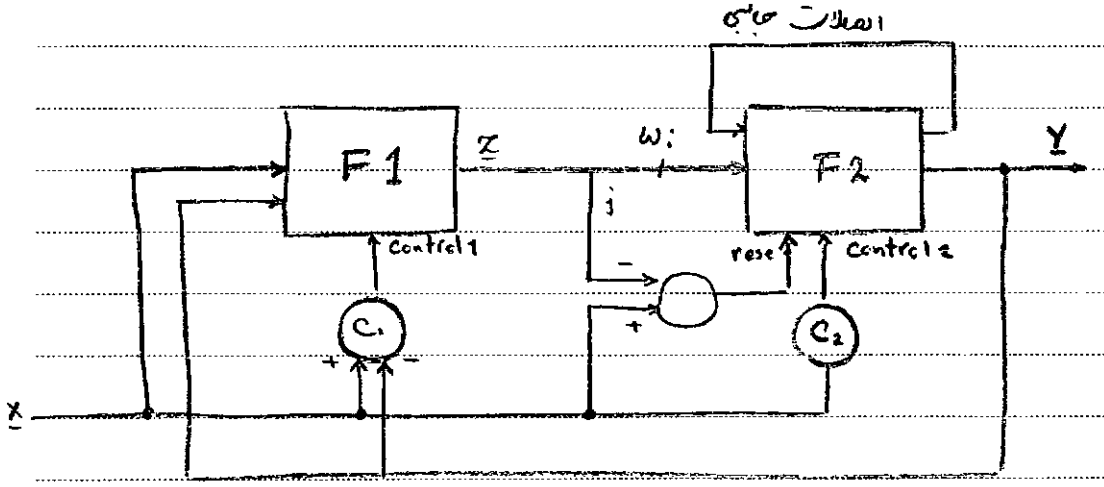
• معماری

- دو لایه اصلی { لایه ورودی، مقایسه، F1
  { لایه خروجی، خوشه‌بندی، F2

- تعدادی سیگنال اضافه برای کنترل چگونگی داده‌ها در هر مرحله از سیگنال دهی شبکه

Subject:

Year. Month. Date. ۸۱



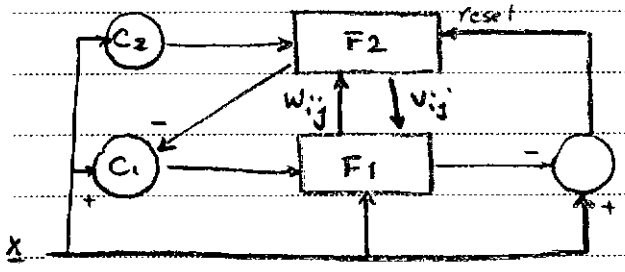
۱- شبکه دارای مکانیسم reset ← reset کردن واحدی خروجی

ورودی قرار یک مقدار آستانه برابری (Vigilance) مقایسه می‌کند و تعیین می‌کند که آیا بهای الگوی جدید ورودی با یک دسته جدید ایجاد شود یا خیر

۲- دو لایه شبکه با ارتباطات کامل بین هر دو بین خود متصلند

۳- واحدی خروجی دارای ارتباطات جانبی

۴- نامین نیز



Subject:

Year Month Date ( )

مرحله آماده سازی:

I. تعیین مقادیر اولیه

- جریان داده از لایه ورودی توسط control1 تعیین می‌شود.
- مقدار این سیگنال باینری است ← حالت ورودی : 1
  حالت مقایسه : 0
- c1 برابر یک است اگر ورودی غیر صفر به سلول اعمال شود
- دقیقاً یکی از واحدهای لایه خروجی فعال می‌شود، c1 اجباراً صفر می‌شود
- سیگنال c2 لایه خروجی را فعال یا غیرفعال می‌کند.
- برای هر الگوی ورودی قابل قبول، c2 برابر 1
- پس از آزمون مراقبت، c2 برابر 0 — Vigilance Test
- مقدار صفر برای c2 واحدهای لایه خروجی را غیرفعال می‌کند و خروجی‌های آنها صفر

- مقادیر اولیه Vij برابر
- مقادیر اولیه Wij برابر 1/(n+1) — n تعداد ورودی
- مقدار آستانه نزدیکی ρ — $0 < \rho \leq 1$

II. شناسایی

- ورودی لایه ورودی ← X
  c1
  خروجی لایه دوم (Y)

- فعال شدن هر ورودی از این لایه : فعال $\frac{2}{3}$ — اگر دو تا یا بیشتر از این ورودی‌ها

Subject:

Year Month Date ۸۲

فعال باشد، از واحد مذکور یک خروجی 1 در غیر این صورت 0 صادر می شود.

— شناسایی نشانه‌ها با شبکه رسپتن دارد.

— هر بردار وزن W در هر واحد لایه خروجی یک الگوی ذخیره شده می باشد.

— بردار ورودی، با هر کدام از الگوهای ذخیره شده مقایسه می شود (با W) و بهترین تطبیق مشخص می شود.

— این کار با محاسبه حاصلضرب داخلی X و بردارهای وزن مشخص می شود.

$|X||W| \cos\theta$

$\theta \downarrow$ ← بردارها نزدیک‌تر

— ممکن است چند واحد خروجی در پاسخ به ورودی دارای فعالیت قوی باشند

← انتشارات بازدارنده جانبی بین واحدها فعالیت کرده و فقط واحدی که دارای بهترین پاسخ است بقیه خاموش می شوند.

— فقط واحد برنده الگوی وزن ذخیره شده را به لایه ورودی ارسال می کند.

— این الگو به صورت بایری در وزن های اتصالات بازگشتی ذخیره می شود.

III. فاز متالیه

— لایه خروجی دارای یک واحد فعال ← $c_1 = 0$

— ورودی X_i و خروجی لایه دوم که دیگر ورودی های واحدهای خروجی لایه اول را تشکیل می دهند

← استفاده از تقابل $\frac{y}{x}$ خروجی لایه مشخص می شود.

← معادل AND بردن بردارهای ورودی خارجی و داده ذخیره شده قبلی

← بردار متالیه $\frac{y}{x}$

— این بردار و بردار X به بخش reset داده می شود.

Subject:

Year. Month. Date. ( )

- در این بخش reset میزان r به عنوان معیاری برای محاسبه با مقدار آستانه مقایسه می‌شود.

$$r = \frac{\text{تعداد 1های ستون } X_j \text{ در } V_i} {\text{تعداد 1های } X} = \frac{\sum_i V_i X_i}{\sum_i X_i}$$

- $r > \rho$: مغایرت دسته‌بندی ← واحد مقابل لایه F2 : دسته X

- $r < \rho$: شباهت دارد ← بهترین الگوی مذکور برای X پیدا شده است ← ناز جستجو

IV. ناز جستجو

- شبکه سعی خواهد کرد با الگوی ذخیره شده در حافظه تطبیق را برای ورودی پیدا کند

  - ابتدا واحد مقابل خروجی با غیرمقابل خروجی آن را صفر می‌کنیم (خروج از مسابقه)

  - این واحد در نهایت بهترین تطبیق‌ای بعدی برای این ورودی پیدا نمی‌کند

- $e_1 = 0$

- دوباره ورودی به لایه F1 اعمال می‌شود و بهترین تطبیق بدست می‌آید (برنده بعدی)

- شبکه دوباره وارد ناز مقایسه می‌شود ← آزمون برای ارضای معیار آستانه

  - تکرار تا پیدا شدن واحد مناسب

- اگر چنین واحدی پیدا نشود، شبکه ورودی جدید را به یک واحد جدید نسبت می‌دهد

  - یعنی در لایه F2 واحد جدیدی به وجود می‌آورد ← اضافه شدن دسته جدید به شبکه

Subject:
Year. Month. Date. ۸۳

• وجه تسمیه (تشدید) : داده ها مکرراً از ورودی به خروجی دایملس حرکت می کنند تا سرانجام به نتیجه برسند.

• الگوریتم ART1

1. تعیین مقادیر اولیه

$\rho = 0.9$

$v_{ij}(0) = 1$ $\quad 0 \le j \le m-1$

$w_{ij}(0) = \frac{1}{1+n}$ $\quad 0 \le i \le n-1$

2. اعمال ورودی جدید به شبکه

3. محاسبه میزان تطبیق با الگوی عنده w

$\mu_j = \sum_{i=0}^{n-1} w_{ij} x_i$

4. تعیین بهترین تطبیق

$J = \text{Arg Max}_j (\mu_j)$

5. آزمون درایت

$r = \frac{\sum_{i=0}^{n-1} v_{ij} x_i}{\sum_{i=0}^{n-1} x_i}$

$r > \rho$ ← مرحله 7

$r \le \rho$ ← مرحله 6

6. واحد J را غیر فعال می کنیم

$y_J = 0$

آیا همه واحدهای خروجی غیرفعال شده اند؟ خیر: مرحله 3

بلی: افزودن یک واحد جدید

PAPCO

مرحله 7 و بعد به مرحله 2

Subject:
Year. Month. Date. ( )

۶. اصلاح وزن‌های مربوط به بهترین تطبیق

$$w_{ji}(t+1) = \frac{v_{ji}(t)\, x_i}{\frac{1}{2} + \sum_{i=0}^{n-1} v_{ji}(t)\, x_i}$$

$$v_{ji}(t+1) = v_{ji}(t) \cdot x_i$$

۸. واحدهای غیرفعال را فعال نموده و به مرحله ۲ برمی‌گردیم.

- این شبکه در برخورد با داده آموزش دیده پاسخ می‌دهد.
- شبکه ART در طی آموزش نسبت به پارامترها بسیار حساس است.
- ρ: حساس‌ترین پارامتر

$\rho \downarrow$ (حدود 0.1) ← دسته‌بندی درشت‌دانه با تعداد دسته کم

$\rho \uparrow$ (حدود 1) ← دسته‌بندی ریزدانه با تغییرات کم منجر به ایجاد طبقه‌های جدیدی می‌شود.

- ρ بزرگ ← حساسیت بیش از حد شبکه به تغییرات بین نمونه‌های هر دسته

ایراد ART ← عملکرد ضعیف در رویارویی با داده‌های نویزی

راه حل: ρ متغیر ← ابتدا ρ کوچک: خوشه‌بندی خام اولیه
افزایش ρ: افزایش دقت دسته‌ها

Subject:

Year Month Date ۱۶

* واحدهای پردازش تکی

۱- مدل مک کالاک و پیتز

$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$

$= f\left(\sum_{i=0}^{n} w_i x_i\right) \qquad x_0 = 1 \;,\; w_0 = -\theta$

ورودی‌ها: پاسخ پیوسته، مجموع غیرخطی از ورودی‌ها، دنباله‌ی زمانی به عنوان خروجی، تاخیر غیرآسان

۲- تغییر تابع مقایسه

خطی، پله‌ای، دقطه‌ای خطی، سیگموئیدی، تاسی، نسبت برج $\left(\frac{x^2}{1+x^2}\right)$

۳- تابع مقایسه متفاوت برای هر لایه

۴- ترکیب غیرخطی ورودی‌ها

تعداد وزن ↑ ← آموزش دشوارتر ↑ ← محاسبات ↑ ← پیچیدگی ↑ ← دقت ↑

$y = f\left(\sum_{i}^{n} w_i x_i + \sum_{i}^{n}\sum_{k}^{n} w_{ik} x_i x_k + k\right)$

۵- واحدهای غیرهمزمان

۶- منطق فازی ← ورودی‌ها / پارامترها / بردار

۷- ورودی‌های دارای وزن‌های مختلط

$w_i = a_i + b_i$

$y = |w_1 x_1 + \dots + w_n x_n|^2 = (a_1 x_1 + \dots + a_n x_n)^2 + (b_1 y_1 + \dots + b_n y_n)^2$

$x = f(y) = \frac{1}{1 + e^{-\alpha(y-\theta)}}$

نرون درجه ۲

Subject:

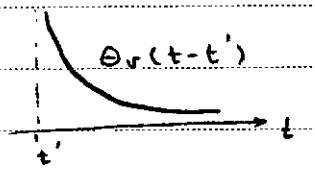
Year Month Date ( )

۸- نورون‌های سیگما-پی ← غیرخطی $y = \sum_k \omega_k \prod_i x_i$

۹- نورون‌های ضربه‌ای

پتانسیل نورون : $P_v = \sum EPSP + \sum IPSP$

آستانه تحریک : $P_v > \Theta_v$



قدرت تحریک : $\omega_{u,v} \times \varepsilon_{u,v}(t-s)$

نورون‌های تحریک F_u : $P_v(t) = \sum_u \sum_{s \in F_u} \omega_{u,v} \times \varepsilon_{u,v}(t-s)$

۱۰- نورون‌های احتمالاتی

$$y = \begin{cases} +1 & P(I) \\ -1 & 1-P(I) \end{cases} \qquad P(I) = \frac{1}{1+e^{-I/T}}$$

* نسل‌های نورون‌ها

۱- نسل اول : مدل مک کالاک پیتز ، خروجی دوقطبی ← پرسپترون ، آدالاین ، بولتزمان

۲- نسل دوم : خروجی پیوسته ← MLP ، RBF ، بازگشتی

۳- نسل سوم : خروجی ضربه ، نزدیک به نورون‌های طبیعی

* رفتار دینامیکی واحد

$$\frac{dy}{dt} = -g(y) + I$$

تعادل : $g(y) = I \longrightarrow y = g^{-1}(I) = f(I) \longrightarrow g = f^{-1}$

* انواع اتصالات

۱- اتصالات کامل

۲- اتصالات تقاطعی پیازاهت

۳- اتصالات تحریک

۴- اتصالات با توزیع خاص

Subject:
Year. Month. Date. ح۹

* دسته بندی

۱- ماژ آموزش ← ورودی: نمونه ها + پاسخ آنها خروجی: پارامترهای دسته بندی کننده

۲- ماژ آزمایش ← ورودی: داده جدید خروجی: تعیین دسته (پارامترها ثابت)

* واحدهای دسته بندی کننده

۱- واحد هب ← هب:

در سلولهای طبیعی هرگاه دندون i به طور مکرر در فعال کردن دندون j شرکت کند، میزان تاثیر این دندون بر فعال کردن دندون j افزایش می یابد.

یادگیری هم وقوعی Coincidence learning

قانون هب $w_i(new) = w_i(old) + \alpha i \times y$

تعمیم هب ← طبیعی و واقعیات ← در دندون خاموش یا منفی نیز ارتباط آنها باید تقویت شوند.

۲- واحد پرسپترون ← زنبلات

قانون یادگیری پرسپترون $w_i(new) = w_i(old) + (t-y)\,X$

تعمیم ← منطقی و واقعیت $w_i(new) = w_i(old) + \alpha(t-y)\,X$

اصلاح مناسب با ورودی $\alpha = \frac{X \cdot w}{11 \times 11^2}$ یا $\alpha = \frac{1}{11 \times 11}$

پرسپترون درجه ۲ $I = -\theta + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2$

۳- واحد آدالاین ← دیود

Performance learning

کمینه کردن MMS ← تیپ دیودی

Subject:

Year Month Date ( )

$$F(w) = \lim_{N\to\infty} \frac{1}{N} \sum^{N} (t_k - I_k)^2 = E\left[(t_k - I_k)^2\right]$$

$$\nabla_w F(w) = -2Q + 2Rw = 0 \longrightarrow w^* = R^+ Q$$

$$\nabla_w F(w) = \nabla\left[\lim \ldots\right] = -2E\left[\underbrace{(t_k - I_k)}_{\delta_k} X_k\right] = -2E\left[\delta_k X_k\right]$$

نداریم ← شیب نزولی

قانون دلتا

$$w(new) = w(old) + \alpha \delta X$$

$$0.01 \leq \alpha \leq 10 \longrightarrow \alpha = 0.1$$ مقدار مرجع

پیشنهاد دیگر:

$$\frac{1}{10n} \leq \alpha \leq \frac{1}{n}$$

پیشنهاد دیگر: آموزش دسته‌ای ← پس از تعداد زیادی X ، w ثابت ← میانگین‌گیری $\delta_k X_k$

پیشنهاد دیگر: استفاده از ممنتم

$$w_{k+1} = w_k + \alpha(1-\mu)\delta_k x_k + \mu(w_k - w_{k-1})$$

$$\mu \gg \alpha \longrightarrow \mu = 0.9 \;\&\; \alpha = 0.01$$

آرایش درجه ۲

$$I = w_0 + w_1 x_1 + w_2 x_2 + w_{11} x_1^2 + w_{22} x_2^2 + w_{12} x_1 x_2$$

Subject:
Year. Month. Date. ۳ع

* تداعی در مغز انسان (ارسطو) ① مطالب شبیه به هم ② برخلاف یکدیگر
③ از لحاظ زمانی نزدیک به هم ④ به طور متوالی

* دسته بندی های شبکه های تداعی

۱- پیش خور	vs	بازگشتی	براساس سیگنال
۲- خود تداعی ($y_i = x_i$)	vs	غیر خود تداعی ($y_i \neq x_i$)	عملکرد مقابل نرمیز
۳- نگهدارنده ($\psi(x+\varepsilon) = y$)	vs	دروغیاب ($\psi(x+\varepsilon) = y+\sigma$)	از لحاظ نگهداری

* شبکه تداعی گر پیش خور

• دوگانه $(n : m)$ — اتصال کامل — برخوردی ترکیب خطی دردوها

• دروغیاب ←
$$\begin{cases} X_k = S_k \Rightarrow Y_k = T_k \\ X_k = S_k + \varepsilon \Rightarrow Y_k = T_k + \delta \end{cases}$$

• خروجی: $\underline{Y} = \underline{W}^T \underline{X}$

• آموزش: قانون هب $W(new) = W(old) + S.T$

• بدون خطا: S_i متعامد ← تعداد الگو $= n$ ← $X_i = S \Rightarrow Y = WS = T$

• دارای خطای η: حداقل کردن MS

$$F(W) = \frac{1}{n} \| T - WS \|^2$$
$$F(W) = 0 \longrightarrow W = (TS^+)^T$$

حالت خطی: $S^+S = I \rightarrow F(W) = 0 \rightarrow \psi(X_i) = Y_i$

حالت غیر خطی: $S^+S \neq I \rightarrow S_i = \sum \alpha_j S_j \rightarrow W^T S_i = \sum \alpha_j T_j$

برای $F(W) = 0$ باید رابطه خطی بین S_i ها بین T_i ها نیز باشد

محاسبه S^+ به کمک تجزیه مقدار منفرد
$$S = \sum^{L} \sqrt{\lambda_i}\, p_i^T q_i \rightarrow S^+ = \sum^{rank} \frac{1}{\sqrt{\lambda_i}} q_i p_i$$
Singular value decomposition

P4PCO

Subject:

Year Month Date ( )

• آموزش با ناظر وجود دارد

$$w(new) = w(old) + \alpha \left[\underbrace{(t_k - (w(old)\, S_k))}_{\delta \text{ خطا}} S_k \right]$$

آموزش به دفعات متعدد به صورت آماری پیاده شده ـ α کوچک و + ـ $\theta \neq 0$ بهتر است

* شبکه ادالاین یک لایه بازگشتی ساده

• ادالاین ← با شروع از هر نقطه به یک حالت پایدار برسد ✓

بهترین حالت پایدار برسد ✗

ایجاد هر تعداد حالت پایدار برای شبکه

• تک لایه ـ ماتریس وزنی متقارن ـ عناصر قطری صفر (پیوندی) یا غیر صفر

• آموزش: ناظر هب

• پاسخ شبکه به ورودی‌ها به طور نامحدود بزرگ می‌شود ← تابع خطی فعالیت

* شبکه BSB (Brain State in a Box) ← اندرسن

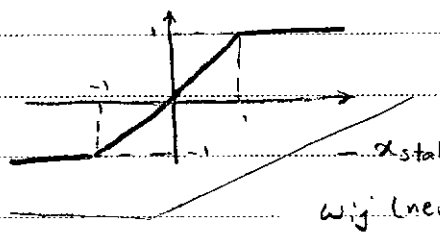
• ورودی تنها یک بار شبکه منتقل می‌شود.

• اتصالات: کامل یا غیر کامل (آماری) و $w_{ii} = 1$

• حالات: پایدار ← درون هایپرلوب اولیه ← نقطه‌ای داخل هایپرلوب

• وزنها: اولیه ← کوچک بروزرسانی ← هبزنان ← هنگام رسیدن به حالت پایدار

• تابع فعالیت



• آموزش: ناظر هب ← حالت پایدار دلخواه نزدیک α stable

$$w_{ij}(new) = w_{ij}(old) + \beta\, y_i\, y_j$$

ناظر دیدرو ← حل مشکل نرن

$$\underline{w}(new) = \underline{w}(old) + \eta\, \underline{S}\, (\underline{Y}^d - \underline{Y}^o)^T$$

PqPCO

Subject :
Year . Month . Date . ۴۲

• الگوریتم BSB

۰. وزنها = مقادیر ویژه اتفاقی و نرخهای یادگیری α و β

۱. $\forall S$

۲. $Y_i = S_i$

۳. تا زمانیکه پایدار نشده‌اند

۴. $I_i = y_i + \alpha \sum_{j \neq i} y_j w_{ji}$

۵. $y_i = f(I)$

۶. اصلاح وزنها : هب

* شبکه ۴ نورونی ← ۴ نورون !

• اتصالات : کامل و $w_{ii} = 0$ و $w_{ij} = w_{ji}$

• آستانه : نا هزمان ـ دو قطبی ـ $\theta_i = 0$

• تابع فعالیت :

$$y_i(new) = \begin{cases} +1 & \sum w_{ji} y_j(old) > \theta \\ y_i(old) & \sum w_{ji} y_j(old) = \theta \\ -1 & \sum w_{ji} y_j(old) < \theta \end{cases}$$

برای حالتهای حالت تساوی (روی برچسب) می‌توان به طور تصادفی تغییر داد.

$y_i(new) = Sgn(\sum w_{ji} y_j(old) - \theta_i + x_i)$

$= Sgn(\sum w_{ji} y_j(old))$ x_i بردار لحظه ورودی و $\theta_i = 0$

• سیستم سازی ناهزمان ← تابع لیاپانف (انرژی) ← اثبات همگرایی

• تابع انرژی

$$E = -\frac{1}{2} \sum_{i \neq j} \sum_j w_{ij} y_i y_j - \sum_i x_i y_i + \sum_i \theta_i y_i$$

$$\Delta E = -[\sum_{j \neq k} w_{kj} y_j(old) + x_k - \theta_k] \Delta y_k \xrightarrow{\forall k} \Delta E \leq 0$$

مینیمم محلی تابع انرژی = حالت پایدار ← تضمین پایداری : $w_{ij} = w_{ji}$ و $w_{ii} = 0$

P4PCO

پایداری $\uparrow \Rightarrow E \downarrow \Rightarrow \sum\sum w_{ij} y_i y_j \uparrow \Rightarrow S \uparrow \Leftrightarrow E \downarrow$

Subject:

Year. Month. Date. ( )

- حذف سادگی ← بردار ورودی تا 99٪ شبیه S_k ← خروجی S_k ← برعکس $-S_k$

- ذخیره الگو: $w_{ij} = \begin{cases} 0 & i=j \\ \sum_{\mu=1}^{L} S_\mu(i) S_\mu(j) & j \neq i \end{cases}$

- قانون هب تعمیم یافته ← اضافه کردن $\frac{1}{n}$ جهت محدود ← بهبود پایداری

- مزایا ← سهولت یادگیری ← محاسبات میزان عمل جمع و ضرب

- انواع حالات ← Retrieval : بازیابی = S_i

$-S_i$: Inverse معکوس حالات بازیابی

$S_{j(i)}^{mix}$: Mix ترکیبی خطی از تعداد فرد از حالات بازیابی

Spin Glass : تعداد زیاد الگو ← min محلی‌ای بی‌ربط به حالات پایدار

Spurious : ترکیب سه حالت خواسته نادانسته

- حذف حالت معکوس: حالت برنامه خاصی از پاسخ را از قبل داشتیم ← وحالت شبه خلاف آن دیده شد ← $x=-1$

- الگوریتم

محاسبه وزنها: سپ تعمیم یافته

1. $\forall x$
2. $y_i(old) = x_i$
3. تا زمانی که پایدار نشده است
4. $\forall i$
5. $I_i = x_i + \sum_j w_{ij} y_j(old)$
6. $y_i(new) = f(I)$
7. ارسال $y_i(new)$ به همه واحدها
8. آزمون همگرایی

- حداکثر الگوهای قابل ذخیره:

تحلیل اول: $L \leq 0.61n$ / $err<5\%$: $L \leq 0.37n$ / $err<1\%$: $L \leq 0.15n$

تحلیل دوم: بازیابی کامل / $h \leq \frac{n}{4 \ln n}$: $h \leq \frac{n}{2 \ln n}$: رتبه 99٪

افزایش بردار ورودی ثابت ← کاهش ورودی برای بالا بردن تعداد برنامه

P4PCO

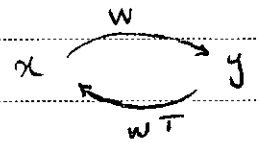
Subject:
Year. Month. Date. ۵۶

* قانون لیپن - لراسپرک

به ازای شرایط خیلی عمومی (!) ثابت می شود که از حالت شبکه و زمان صفر به بخش مثبت فضای حتمی $\mathcal{H}$ تغییر می یابد، شبکه همیشه به یک نقطه پایدار در این بخش همگرا می شود.

* شبکه متداعی دوطرفه (BAM) ← کاندل

• لایه x و لایه y



واحدهای لایه x هر کدام با هم لایه y ارتباط دارند و y با هم

• تعمیم شبکه ع پیلید به حالت غیرخود متداعی

• آموزش: هب

• تابع مقایسه: شبیه ع پیلید

• الگوریتم

۵. محاسبه وزنها و خروجی واحدها برابر صفر

۱. $\forall x$

۲. $X_i = S_i$ (و یا) $Y_i = T_i$

۳. ... پایدار نشده است.

۴. $I_j = \sum_i w_{ij} x_i \longrightarrow y_j = f(I_j)$

۵. $I_i = \sum_j w_{ij} y_j \longrightarrow x_i = f(I_i)$

۶. آزمون همگرایی: به حاصل رسیدن حالات پایدار در لایه

• شبکه متداعی دوطرفه بیوسته

$f_i(I_i) = \frac{1}{1+e^{-I_i}}$ و تابع هب و $\theta_i \neq 0$

* قضیه کاندل

اگر شبکه متداعی دوطرفه از هر حالت اولیه شروع به کار کند، شبکه به حالت پایدار همگرا خواهد شد.

← اثبات با قضیه لیپن - لراسپرک

ص ۱۰۲ ← حفظی جات!

Subject:

Year. Month. Date. ( )

سال ۸۷ در یک نگاه

فروردین ربیع الاول - ربیع الثانی ۱۴۲۹ Mar. - Apr. 2008			اردیبهشت ربیع الثانی - جمادی الاول ۱۴۲۹ Apr. - May 2008			خرداد جمادی الاول - جمادی الثانی ۱۴۲۹ May - Jun. 2008			تیر جمادی الثانی - رجب ۱۴۲۹ Jun. - Jul. 2008			مرداد رجب - شعبان ۱۴۲۹ Jul. - Aug. 2008			شهریور شعبان - رمضان ۱۴۲۹ Aug. - Sep. 2008		
پنج شنبه	۱	۱۲ 20	یکشنبه	۱	۱۳ 20	چهارشنبه	۱	۱۵ 21	شنبه	۱	۱۷ 21	سه شنبه	۱	۱۹ 22	جمعه	۱	۲۰ 22
جمعه	۲	۱۳ 21	دوشنبه	۲	۱۴ 21	پنج شنبه	۲	۱۶ 22	یکشنبه	۲	۱۸ 22	چهارشنبه	۲	۲۰ 23	شنبه	۲	۲۱ 23
شنبه	۳	۱۴ 22	سه شنبه	۳	۱۵ 22	جمعه	۳	۱۷ 23	دوشنبه	۳	۱۹ 23	پنج شنبه	۳	۲۱ 24	یکشنبه	۳	۲۲ 24
یکشنبه	۴	۱۵ 23	چهارشنبه	۴	۱۶ 23	شنبه	۴	۱۸ 24	سه شنبه	۴	۲۰ 24	جمعه	۴	۲۲ 25	دوشنبه	۴	۲۳ 25
دوشنبه	۵	۱۶ 24	پنج شنبه	۵	۱۷ 24	یکشنبه	۵	۱۹ 25	چهارشنبه	۵	۲۱ 25	شنبه	۵	۲۳ 26	سه شنبه	۵	۲۴ 26
سه شنبه	۶	۱۷ 25	جمعه	۶	۱۸ 25	دوشنبه	۶	۲۰ 26	پنج شنبه	۶	۲۲ 26	یکشنبه	۶	۲۴ 27	چهارشنبه	۶	۲۵ 27
چهارشنبه	۷	۱۸ 26	شنبه	۷	۱۹ 26	سه شنبه	۷	۲۱ 27	جمعه	۷	۲۳ 27	دوشنبه	۷	۲۵ 28	پنج شنبه	۷	۲۶ 28
پنج شنبه	۸	۱۹ 27	یکشنبه	۸	۲۰ 27	چهارشنبه	۸	۲۲ 28	شنبه	۸	۲۴ 28	سه شنبه	۸	۲۶ 29	جمعه	۸	۲۷ 29
جمعه	۹	۲۰ 28	دوشنبه	۹	۲۱ 28	پنج شنبه	۹	۲۳ 29	یکشنبه	۹	۲۵ 29	چهارشنبه	۹	۲۷ 30	شنبه	۹	۲۸ 30
شنبه	۱۰	۲۱ 29	سه شنبه	۱۰	۲۲ 29	جمعه	۱۰	۲۴ 30	دوشنبه	۱۰	۲۶ 30	پنج شنبه	۱۰	۲۸ 31	یکشنبه	۱۰	۲۹ 31
یکشنبه	۱۱	۲۲ 30	چهارشنبه	۱۱	۲۳ 30	شنبه	۱۱	۲۵ 31	سه شنبه	۱۱	۲۷ 1	جمعه	۱۱	۲۹ 1	دوشنبه	۱۱	۳۰ 1
دوشنبه	۱۲	۲۳ 31	پنج شنبه	۱۲	۲۴ 1	یکشنبه	۱۲	۲۶ 1	چهارشنبه	۱۲	۲۸ 2	شنبه	۱۲	۳۰ 2	سه شنبه	۱۲	۱ 2
سه شنبه	۱۳	۲۴ 1	جمعه	۱۳	۲۵ 2	دوشنبه	۱۳	۲۷ 2	پنج شنبه	۱۳	۲۹ 3	یکشنبه	۱۳	۱ 3	چهارشنبه	۱۳	۲ 3
چهارشنبه	۱۴	۲۵ 2	شنبه	۱۴	۲۶ 3	سه شنبه	۱۴	۲۸ 3	جمعه	۱۴	۱ 4	دوشنبه	۱۴	۲ 4	پنج شنبه	۱۴	۳ 4
پنج شنبه	۱۵	۲۶ 3	یکشنبه	۱۵	۲۷ 4	چهارشنبه	۱۵	۲۹ 4	شنبه	۱۵	۲ 5	سه شنبه	۱۵	۳ 5	جمعه	۱۵	۴ 5
جمعه	۱۶	۲۷ 4	دوشنبه	۱۶	۲۸ 5	پنج شنبه	۱۶	۱ 5	یکشنبه	۱۶	۳ 6	چهارشنبه	۱۶	۴ 6	شنبه	۱۶	۵ 6
شنبه	۱۷	۲۸ 5	سه شنبه	۱۷	۲۹ 6	جمعه	۱۷	۲ 6	دوشنبه	۱۷	۴ 7	پنج شنبه	۱۷	۵ 7	یکشنبه	۱۷	۶ 7
یکشنبه	۱۸	۲۹ 6	چهارشنبه	۱۸	۱ 7	شنبه	۱۸	۳ 7	سه شنبه	۱۸	۵ 8	جمعه	۱۸	۶ 8	دوشنبه	۱۸	۷ 8
دوشنبه	۱۹	۳۰ 7	پنج شنبه	۱۹	۲ 8	یکشنبه	۱۹	۴ 8	چهارشنبه	۱۹	۶ 9	شنبه	۱۹	۷ 9	سه شنبه	۱۹	۸ 9
سه شنبه	۲۰	۱ 8	جمعه	۲۰	۳ 9	دوشنبه	۲۰	۵ 9	پنج شنبه	۲۰	۷ 10	یکشنبه	۲۰	۸ 10	چهارشنبه	۲۰	۹ 10
چهارشنبه	۲۱	۲ 9	شنبه	۲۱	۴ 10	سه شنبه	۲۱	۶ 10	جمعه	۲۱	۸ 11	دوشنبه	۲۱	۹ 11	پنج شنبه	۲۱	۱۰ 11
پنج شنبه	۲۲	۳ 10	یکشنبه	۲۲	۵ 11	چهارشنبه	۲۲	۷ 11	شنبه	۲۲	۹ 12	سه شنبه	۲۲	۱۰ 12	جمعه	۲۲	۱۱ 12
جمعه	۲۳	۴ 11	دوشنبه	۲۳	۶ 12	پنج شنبه	۲۳	۸ 12	یکشنبه	۲۳	۱۰ 13	چهارشنبه	۲۳	۱۱ 13	شنبه	۲۳	۱۲ 13
شنبه	۲۴	۵ 12	سه شنبه	۲۴	۷ 13	جمعه	۲۴	۹ 13	دوشنبه	۲۴	۱۱ 14	پنج شنبه	۲۴	۱۲ 14	یکشنبه	۲۴	۱۳ 14
یکشنبه	۲۵	۶ 13	چهارشنبه	۲۵	۸ 14	شنبه	۲۵	۱۰ 14	سه شنبه	۲۵	۱۲ 15	جمعه	۲۵	۱۳ 15	دوشنبه	۲۵	۱۴ 15
دوشنبه	۲۶	۷ 14	پنج شنبه	۲۶	۹ 15	یکشنبه	۲۶	۱۱ 15	چهارشنبه	۲۶	۱۳ 16	شنبه	۲۶	۱۴ 16	سه شنبه	۲۶	۱۵ 16
سه شنبه	۲۷	۸ 15	جمعه	۲۷	۱۰ 16	دوشنبه	۲۷	۱۲ 16	پنج شنبه	۲۷	۱۴ 17	یکشنبه	۲۷	۱۵ 17	چهارشنبه	۲۷	۱۶ 17
چهارشنبه	۲۸	۹ 16	شنبه	۲۸	۱۱ 17	سه شنبه	۲۸	۱۳ 17	جمعه	۲۸	۱۵ 18	دوشنبه	۲۸	۱۶ 18	پنج شنبه	۲۸	۱۷ 18
پنج شنبه	۲۹	۱۰ 17	یکشنبه	۲۹	۱۲ 18	چهارشنبه	۲۹	۱۴ 18	شنبه	۲۹	۱۶ 19	سه شنبه	۲۹	۱۷ 19	جمعه	۲۹	۱۸ 19
جمعه	۳۰	۱۱ 18	دوشنبه	۳۰	۱۳ 19	پنج شنبه	۳۰	۱۵ 19	یکشنبه	۳۰	۱۷ 20	چهارشنبه	۳۰	۱۸ 20	شنبه	۳۰	۱۹ 20
شنبه	۳۱	۱۲ 19	سه شنبه	۳۱	۱۴ 20	جمعه	۳۱	۱۶ 20	دوشنبه	۳۱	۱۸ 21	پنج شنبه	۳۱	۱۹ 21	یکشنبه	۳۱	۲۰ 21

www.papcoiran.com

محصولات زیبا برازنده انسانهای خوب

سال ۸۷ در یک نگاه

مهر	آبان	آذر	دی	بهمن	اسفند
رمضان - شوال ۱۴۲۹ هـ.ق	شوال - ذی القعده ۱۴۲۹	ذی القعده - ذی الحجه ۱۴۲۹	ذی الحجه ۱۴۲۹ - محرم ۱۴۳۰	محرم - صفر ۱۴۳۰	صفر - ربیع الاول ۱۴۳۰
Sep. - Oct. 2008	Oct. - Nov. 2008	Nov. - Dec. 2008	Dec. 2008 - Jan. 2009	Jan. - Feb. 2009	Feb. - Mar. 2009

دوشنبه	۱	۲۱ 22	چهارشنبه	۱	۲۲ 22	جمعه	۱	۲۲ 21	یکشنبه	۱	۲۲ 21	سه شنبه	۱	۲۳ 20	پنج شنبه	۱	۲۳ 19
سه شنبه	۲	۲۲ 23	پنج شنبه	۲	۲۳ 23	شنبه	۲	۲۳ 22	دوشنبه	۲	۲۳ 22	چهارشنبه	۲	۲۴ 21	جمعه	۲	۲۴ 20
چهارشنبه	۳	۲۳ 24	جمعه	۳	۲۴ 24	یکشنبه	۳	۲۴ 23	سه شنبه	۳	۲۴ 23	پنج شنبه	۳	۲۵ 22	شنبه	۳	۲۵ 21
پنج شنبه	۴	۲۴ 25	شنبه	۴	۲۵ 25	دوشنبه	۴	۲۵ 24	چهارشنبه	۴	۲۵ 24	جمعه	۴	۲۶ 23	یکشنبه	۴	۲۶ 22
جمعه	۵	۲۵ 26	یکشنبه	۵	۲۶ 26	سه شنبه	۵	۲۶ 25	پنج شنبه	۵	۲۶ 25	شنبه	۵	۲۷ 24	دوشنبه	۵	۲۷ 23
شنبه	۶	۲۶ 27	دوشنبه	۶	۲۷ 27	چهارشنبه	۶	۲۷ 26	جمعه	۶	۲۷ 26	یکشنبه	۶	۲۸ 25	سه شنبه	۶	۲۸ 24
یکشنبه	۷	۲۷ 28	سه شنبه	۷	۲۸ 28	پنج شنبه	۷	۲۸ 27	شنبه	۷	۲۸ 27	دوشنبه	۷	۲۹ 26	چهارشنبه	۷	۲۹ 25
دوشنبه	۸	۲۸ 29	چهارشنبه	۸	۲۹ 29	جمعه	۸	۲۹ 28	یکشنبه	۸	۲۹ 28	سه شنبه	۸	۳۰ 27	پنج شنبه	۸	۳۰ 26
سه شنبه	۹	۲۹ 30	پنج شنبه	۹	۳۰ 30	شنبه	۹	۳۰ 29	دوشنبه	۹	۱ 29	چهارشنبه	۹	۱ 28	جمعه	۹	۱ 27
چهارشنبه	۱۰	۱ 1	جمعه	۱۰	۱ 31	یکشنبه	۱۰	۱ 30	سه شنبه	۱۰	۲ 30	پنج شنبه	۱۰	۲ 29	شنبه	۱۰	۲ 28
پنج شنبه	۱۱	۲ 2	شنبه	۱۱	۲ 1	دوشنبه	۱۱	۲ 1	چهارشنبه	۱۱	۳ 31	جمعه	۱۱	۳ 30	یکشنبه	۱۱	۳ 1
جمعه	۱۲	۳ 3	یکشنبه	۱۲	۳ 2	سه شنبه	۱۲	۳ 2	پنج شنبه	۱۲	۴ 1	شنبه	۱۲	۴ 31	دوشنبه	۱۲	۴ 2
شنبه	۱۳	۴ 4	دوشنبه	۱۳	۴ 3	چهارشنبه	۱۳	۴ 3	جمعه	۱۳	۵ 2	یکشنبه	۱۳	۵ 1	سه شنبه	۱۳	۵ 3
یکشنبه	۱۴	۵ 5	سه شنبه	۱۴	۵ 4	پنج شنبه	۱۴	۵ 4	شنبه	۱۴	۶ 3	دوشنبه	۱۴	۶ 2	چهارشنبه	۱۴	۶ 4
دوشنبه	۱۵	۶ 6	چهارشنبه	۱۵	۶ 5	جمعه	۱۵	۶ 5	یکشنبه	۱۵	۷ 4	سه شنبه	۱۵	۷ 3	پنج شنبه	۱۵	۷ 5
سه شنبه	۱۶	۷ 7	پنج شنبه	۱۶	۷ 6	شنبه	۱۶	۷ 6	دوشنبه	۱۶	۸ 5	چهارشنبه	۱۶	۸ 4	جمعه	۱۶	۸ 6
چهارشنبه	۱۷	۸ 8	جمعه	۱۷	۸ 7	یکشنبه	۱۷	۸ 7	سه شنبه	۱۷	۹ 6	پنج شنبه	۱۷	۹ 5	شنبه	۱۷	۹ 7
پنج شنبه	۱۸	۹ 9	شنبه	۱۸	۹ 8	دوشنبه	۱۸	۹ 8	چهارشنبه	۱۸	۱۰ 7	جمعه	۱۸	۱۰ 6	یکشنبه	۱۸	۱۰ 8
جمعه	۱۹	۱۰ 10	یکشنبه	۱۹	۱۰ 9	سه شنبه	۱۹	۱۰ 9	پنج شنبه	۱۹	۱۱ 8	شنبه	۱۹	۱۱ 7	دوشنبه	۱۹	۱۱ 9
شنبه	۲۰	۱۱ 11	دوشنبه	۲۰	۱۱ 10	چهارشنبه	۲۰	۱۱ 10	جمعه	۲۰	۱۲ 9	یکشنبه	۲۰	۱۲ 8	سه شنبه	۲۰	۱۲ 10
یکشنبه	۲۱	۱۲ 12	سه شنبه	۲۱	۱۲ 11	پنج شنبه	۲۱	۱۲ 11	شنبه	۲۱	۱۳ 10	دوشنبه	۲۱	۱۳ 9	چهارشنبه	۲۱	۱۳ 11
دوشنبه	۲۲	۱۳ 13	چهارشنبه	۲۲	۱۳ 12	جمعه	۲۲	۱۳ 12	یکشنبه	۲۲	۱۴ 11	سه شنبه	۲۲	۱۴ 10	پنج شنبه	۲۲	۱۴ 12
سه شنبه	۲۳	۱۴ 14	پنج شنبه	۲۳	۱۴ 13	شنبه	۲۳	۱۴ 13	دوشنبه	۲۳	۱۵ 12	چهارشنبه	۲۳	۱۵ 11	جمعه	۲۳	۱۵ 13
چهارشنبه	۲۴	۱۵ 15	جمعه	۲۴	۱۵ 14	یکشنبه	۲۴	۱۵ 14	سه شنبه	۲۴	۱۶ 13	پنج شنبه	۲۴	۱۶ 12	شنبه	۲۴	۱۶ 14
پنج شنبه	۲۵	۱۶ 16	شنبه	۲۵	۱۶ 15	دوشنبه	۲۵	۱۶ 15	چهارشنبه	۲۵	۱۷ 14	جمعه	۲۵	۱۷ 13	یکشنبه	۲۵	۱۷ 15
جمعه	۲۶	۱۷ 17	یکشنبه	۲۶	۱۷ 16	سه شنبه	۲۶	۱۷ 16	پنج شنبه	۲۶	۱۸ 15	شنبه	۲۶	۱۸ 14	دوشنبه	۲۶	۱۸ 16
شنبه	۲۷	۱۸ 18	دوشنبه	۲۷	۱۸ 17	چهارشنبه	۲۷	۱۸ 17	جمعه	۲۷	۱۹ 16	یکشنبه	۲۷	۱۹ 15	سه شنبه	۲۷	۱۹ 17
یکشنبه	۲۸	۱۹ 19	سه شنبه	۲۸	۱۹ 18	پنج شنبه	۲۸	۱۹ 18	شنبه	۲۸	۲۰ 17	دوشنبه	۲۸	۲۰ 16	چهارشنبه	۲۸	۲۰ 18
دوشنبه	۲۹	۲۰ 20	چهارشنبه	۲۹	۲۰ 19	جمعه	۲۹	۲۰ 19	یکشنبه	۲۹	۲۱ 18	سه شنبه	۲۹	۲۱ 17	پنج شنبه	۲۹	۲۱ 19
سه شنبه	۳۰	۲۱ 21	پنج شنبه	۳۰	۲۱ 20	شنبه	۳۰	۲۱ 20	دوشنبه	۳۰	۲۲ 19	چهارشنبه	۳۰	۲۲ 18	جمعه	۳۰	۲۲ 20

www.papcoiran.com

با محصولات پاپکو اسناد و مدارک تحت کنترل شماست