Subject :

Year . Month . Date . (1)

شبکه عصبی

ساعات برگزاری کلاس: یکشنبه ۱۰:۴۵ الی ۱۲ بجز بعضی یکشنبه‌ها

چهارشنبه ۱۶:۴۰ الی ۶:۰۰ بجز ۵ رمضان

معاون اضطراری: حداقل پژوهشی

منابع:

1. Neurocomputing, Robert Hect-Nielsen, 1990
2. Fundamentals of Neural Networks, Laurene Fausett, 1994
3. Neural Networks, A Comprehensive foundation, Simon Haykin, 1994 (اساس بحث)

ارزیابی:

- امتحان پایان ترم
- پروژه (حداکثر ۲ نفر) تحویل در پایان ترم
- گزارش مطالعات
- پیشنهاد موضوع از طرف دانشجو
- موضوع باید اوایل ترم انتخاب شود
- جستجوهای مقالات و انتخاب مقالات
(دانشجو) (استاد)

موضوع درس:

۱- مقدمه: تعریف شبکه‌های عصبی، وجوه متمایز کننده شبکه‌های عصبی، زمینه تاریخی، کاربردها و تکنولوژی کامپیوتری امکانی و...

۲- نورون (مدل‌های آن): ساختار نورون‌های طبیعی، بررسی انواع مدلهای شبکه‌های عصبی مصنوعی

۳- مفاهیم و تعاریف بخش‌های مختلف شبکه‌های عصبی: واحدهای پردازش، اتصالات وزنی و...

۴- واحدهای پردازش گر در سطوح و دسته‌بندی آنها: دسته‌بندی، واحدهای پردازش گر، قواعد یادگیری

PAPCO

Subject:

Year. Month. Date. ( )

۵- تقسیم الگوریتم و شبکه ای ارائه شده ها: شبکه متقارن پس خور خطی،
شبکه متقارن بازگشتی و قضیه کوهن برگراسبرگ و شبکه متقارن دو طرفه

۶- مدل های یادگیری: یادگیری با نظارت، یادگیری بی نظارت، یادگیری ارزشی

* ۷- شبکه های خود سازمانده و یادگیری رقابتی: شبکه های تقابلی و مدل های وزن های ثابت
قانون یادگیری کوهنن و یادگیری لوگانز بی سوپروایزری

* ۸- شبکه های چندلایه پرسپترون و قانون انتشار خطا به عقب: قانون استخراج انتشار خطا
تعریف تعلیم به شبکه، روش های بهینه الگوریتم استخراج انتشار خطا و نسخه های
مختلف و انواع الگوریتم

۹- شبکه های تعلیم یافته شناختی

۱۰- شبکه های GMDH Group Method of Data Handling

۱۱- شبکه های دیگر (در صورت داشتن وقت)

۱ مقدمه

Neural Networks, Neural Nets.
Artificial Neural Networks (ANN)
Parallel Distributed Processing (PDP)
Connectionist Models
Neuromorphic Systems
Neural Computing
Neuro Computing

شبکه های عصبی چیستند؟ تلاش برای شبیه سازی، تقلید از روش عملکرد مغز خود
در انجام کارهای متعدد و مختلف با انجام دهه و از اطلاعات
ناقص، متناقض نیز ادراک و استنتاج برخوردار می باشد.

Subject:

Year. Month. Date. (۱)

مقایسه یادگیری عصبی: انسان از کامپیوتر بهتر است و نه دقیق تر از بسیاری از کارهای خطی
بهتر از کامپیوتر انجام می دهد. چرا؟

← ما هنوز نرم افزار لازم را برای انجام این کار نداریم.

← مغز برای پردازش اطلاعات طبیعی مناسب تر است.

ویژگی پردازش اطلاعات طبیعی: اطلاعات محدودیت های مختلف را بطور همزمان در نظر گرفته شده، اینها خوب سکنر است نه قطعی و متناقض هم باشند.

سیستم ادراک انسان می تواند بدون تصمیم گیری زیاد امکانات مختلف را ارزیابی کند، بسیاری تمام محدودیت ها ارضا شوند. در آخر حال، تصمیم نهایی اتخاذ می شود. برای پرسشنامه کوتاه کامپیوتر احتیاج به ایجاد چنین تعادلهایی در آنها داریم.

مقایسه بین رایانش سنتی (traditional computing) و رایانش عصبی

در ۵۰ سال اخیر	نسبت به الگوریتم نداریم
برنامه یک الگوریتم می نویسیم	اگر الگوریتم مشخص نباشد
الگوریتم پیاده سازی شده	پیاده سازی آن پرهزینه، بغرنج، شکننده است
قابل پیاده سازی است	از رایانش عصبی استفاده می کنیم.
نوشتن نرم افزار	

رایانش عصبی: رایانش معماری، توزیع شده، تطبیقی است که در آن مغز به یک محیط اطلاعاتی قدرت پردازش اطلاعات آن محیط را پیدا می کند.

Subject:

Year. Month. Date. ( )

روش های هوش مصنوعی سنتی

معمولاً دانش بصورت قواعد یک می شود

حافظه و اطلاعات توزیع شده کامل نیست

حافظه متمرکز است

صدمه بر حافظه سیستم باعث از کار افتادن آن می شود

دانش عصبی

سیستم با دیدن نمونه (خود قواعد لازم را برای

اطلاعات در سیستم بصورت کاملاً توزیع شده است

حافظه متمرکز است (سیستم با دیدن مواردی

که قبلاً مشاهده نکرده، واردی نا قص،

نویز دارد، پاسخ مناسب می دهد)

خراب شدن بعضی قسمت (واحد، اتصالات)

عملکرد شبکه را بطور قابل توجهی کم نمی کند.

(Fault Tolorance)

Graceful Degradation

این ویژگی برای کاربردهای حساس بسیار مفید است.

شبکه های عصبی، فازی، ژنتیک و... امروزه جزء AI حساب می شوند اصطلاح دیگری به نام Computational Intelligence

Neural Networks ↵

Fuzzy Systems ↵

Genetic Algorithm ↵

AI همان روش های کلاسیک می باشد.

ساده سازی شبکه های عصبی

۱- شبیه سازی نرم افزاری

۲- شتاب دهنده های سخت افزاری

۳- معماری مجتمع فشرده (مخصوص)

۴- سازماندهی مجدد نوری

Subject:
Year. Month. Date. (۴)

۱- شبیه سازی نرم افزاری

- نرم افزار MATLAB ← جعبه ابزار Neural Networks
- نرم افزار Mathematica
- نرم افزار Neural Works از شرکت Neural Wave

۲- شتاب دهنده های نرم افزاری

روش بالا برای مقیاس کوچک کاربرد دارد ولی امکانات خاص شبکه های عصبی را ارائه نمی دهد.
مثلاً یک پردازنده خاص: یک میلیون واحد پردازشگر، یک میلیارد اتصال، ۱.۵ میلیارد
هنگام سازی وزن های شبکه و ...
این پردازنده ها معمولاً از مجموعه IC های DSP استفاده می کنند.

۳- مدارهای مجتمع مقیاس بزرگ (VLSI)
پیاده سازی شبکه های در یک IC
شرکت AT&T ← Electronic Neural Network (ENN)
۷۵۶ نورون، 100,000 وزن
مدت زمان ساخت IC بارها بیشتر از زمان نرم افزار می شود.
در پیاده سازی های جدیدتر شبکه روی IC امکان یادگیری وزن ها پس از ساخت ایجاد شده.
(On-Chip learning)

۴- پیاده سازی نوری
در پردازش نوری، اطلاعات را از پردازنده ای به پردازنده نوری بصورت موازی با هم کار می کند. حافظه های ارتباط بین نزدیک هم از طریق اطلاعات می شوند. اتصالات می توانند به صورت گسترده تر از حالت الکترونیکی شوند.
مسائل مربوط به تداخل اتصالات و شدت کاهش می یابد.
به وجود این محاسن به سطح تکنولوژی دیگر برای پردازش نوری، بیشتر نیاز به توسعه تکنولوژی پیاده سازی شبکه های عصبی در زمینه های دیگر است.

Subject:

Year. Month. Date. ( )

کاربردی شبکه ی عصبی

فوق العاده متعدد می باشد:

۱- شناسایی الگو

۲- حل مسائل دسته بندی شبکه ای که به درستی آموزش دیده اند.

۳- پردازش تصویر و بینایی کامپیوتری

۴- هوش مصنوعی

۵- پردازش سخنگویان

۶- پیش بینی اقتصادی و مالی

۷- تشخیص پزشکی

کنفرانس های مهم در زمینه شبکه ی عصبی:

- IEEE International Conference on Neural Networks (ICNN)
- International joint conference " " " (IJCNN)

دو سازمان: INNS ، IEEE ← برگزار کنندگان

(International Neural Networks Society)

مجلات علمی

1. IEEE Transactions on Neural Networks
2. Neural Networks (INNS)
3. Neural Computation
4. Neuro Computing (Elsevier)

Subject :

Year . Month . Date . ۱۳

سابقه تاریخی :

۱- Jackson (1869 . 1958) عصب‌شناس
منتقد سیار شدید از نظریات قرن ۱۹ فرنولوژی در خصوص توزیع شدگی
عقیده به توزیع شدگی چند سطحی بودن ادراک در سیستم ادراک طبیعی

۳- W. McCulloch & W. Pitts (1943)
نشان دادند شبکه‌های عصبی ساده می‌توانند محاسبات منطقی انجام دهند.

۴- Hebb (1949) روان‌شناس
مطرح قانون هب : روی تغییرات سیناپسی در سلول‌ها

۶- Lashley (1950) عصب‌شناس
اصرار بر روی توزیع شدگی در سیستم‌های عصبی
There are no special cells reserved for special memories!

۵- Minsky (1951)
اولین کامپیوتر عصبی با نام Snark ساخت. این کامپیوتر می‌توانست اصلاح خودکار وزن‌هایش را تنظیم کند.

۷- Rosenblatt (1957) دانشگاه کرنل
مدل پرسپترون ساخت و دستگاهی به نام Mark 1 perceptron
الگوریتم یادگیری پرسپترون را ارائه کرد و بعدها این تحقیقات در
بحث معادله فهم شدن بسیاری افزود. الان رشته شد.
بسیاری روی قابلیت محاسباتی شبکه‌های عصبی می‌دانند.

Subject:

Year. Month. Date.

۷. فیشکن و بابرت (اواسط دهه ۶۰) آزمایشگاه MIT کتاب در انتقاد از پرسپترون‌ها و NN

۸. ویدرو widrow (1959) Stanford یک واحد پردازشگر به نام Adaline از طریق کدهای قدرت یادگیری زیادی بعد Adaptive Linear Element

* پرسپترون تنها می‌تواند توابع خطی را یاد بگیرد. توابعی ساده‌ای مانند XOR قابل یادگیری نیستند. پس از این مطرح شد.
کار آنها باعث شد که توجهات از شبکه‌ای عصبی به Classic AI سوق پیدا کند.

۹. آندرسون ۱۹۷۲ - ۷۷ (BROWN Uni.)
مدلی به نام Linear Associator مبتنی بر مدل‌های ذخیره یادگیری انسان و شناخت در حافظه انسان.

۱۰. Vlerbos (1974)
روش برای انتشار خطا از خروجی به ورودی‌ها ارائه شد.

۱۱. کوهونن ۱۹۷۲ - ۱۹۸۲ دانشگاه صنعتی هلسینکی
در زمینه یادگیری رقابتی کارهای مهمی انجام داد. شبکه‌ی کوهونن (نقشه سازمانده)

۱۲. گروسبرگ 1978 نوروساینس، ریاضیات (BOSTON)
تئوری ریاضی خواص شبکه‌های عصبی سیستمی انجام داد.

۱۳. ویلشاو Willshaw 1981
تحلیل ریاضی در زمینه ذخیره توزیع شده

۱۴. ساتن و بارتو 1981

P4PCO قاعده اصلاح وزن‌ها به نام قاعده Delta

Subject :

Year . Month . Date . ( )

۱۵. هاسلمو (1982) مدل Crossbar Associative Network - نام شبکه هامینگ هم شناخته شده

= مقایسه شدید مجدد به شبکه ای عصبی بیشتر شد!

۱۶. باروک (1985)

قاعده ای برای اشتباه خطا از خروجی یک شبکه به وزن های ارائه شد

۱۷. ریلمارت و مک کلند (1986) ورلاشتاین

قاعده اشتباه خطا عقب رو، اصلاح قاعده تبادل ارائه نمود. (حق سؤال شبکه را پیدا کنید)

۱۸. از سال 1986 تاکنون گسترش این رشته فوق العاده زیاد بوده است.

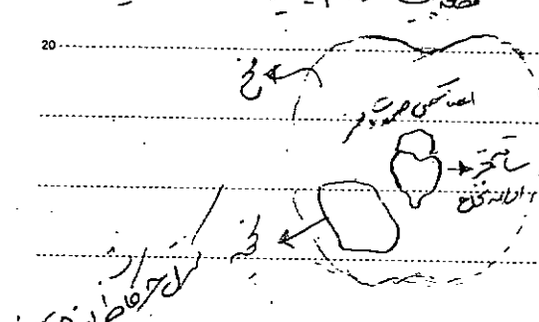
بخش ۲ قاعده ای یادگیری

یادگیری = بزرگترین مسأله در دانش بشر

مغز بیش از دست خاکستری، صورت جیم خورده پیچیده کرده، تقریباً نیم کیلو 1500 گرم

دارای ۲ نیمه، تقارن آینه ای.

قطعات (تکلم، یادگیری، حافظه، حضور، ...)



چین و چروک های مشابه تقریباً

در مغزهای همه در هر حیوان پستاندار دیده می شود.

سطح مغز متشکل از این چین و چروک ها بخش بندی شده است.

Subject:

Year Month Date ( )

(Parietal lobe)

شیار حلو سوی سر شیار جانبی نام دارد. قطعه‌هایی برای بینایی و شنوایی و درد و لمس و ... این حس

قطعه گیجگاهی Temporal lobe : رفتاری، یادگیری، حافظه، تکلم (طرفین)

قطعه پس‌سری (Occipetal lobe) : بینایی

نیمکره راست مغز مربوط به نیمه چپ بدن و بالعکس در اکثر مهره‌داران نیز همین گونه است

کانون فعال برای تکلم عموماً در نیمکره چپ می‌باشد مگر در افراد چپ‌دست

در زیر قشر مغز (مخ) جسمی به نام جسم پینه‌ای وجود دارد. ارتباط میان دو نیمکره را برقرار می‌کند.

اگر سلول‌های مغزی از بین بروند، سلول‌های جدید جایگزین آن‌ها نخواهند شد.

سایر سلول‌های بخش‌های دیگر قشر مخ، وظایف سلول‌های آسیب دیده را بر عهده می‌گیرند.

هر یک از ساختارهای مغز از نیم مستقل می‌باشند و در نقاطی مشخص به یکدیگر متصل شده‌اند، نیم ارتباط

مرتبه. فضای بین مغز، مخزن‌های از مغز است که طی نامیده شده است. این فضا از یک

مایع مغزی نخاعی به نام Cerebrospinal Fluid پر است. کار این مایع رساندن مواد و

تغذیه‌ای مورد نیاز سلول‌های مغز است. مغز از سلول‌های عصبی و الکترون و گلوکز نیازمند

است. 20٪ از اکسیژن مصرفی بدن در مغز صرف می‌شود. اگر جریان خون و اکسیژن و گلوکز به مغز برسد،

فوراً می‌افتد و در صورت ادامه این حالت فرد به کما می‌رود و اگر باز هم ادامه پیدا کند می‌میرد.

PAPCO

Subject :

Year . Month . Date . ( )

مغز و رژیم الکتریکی سلول

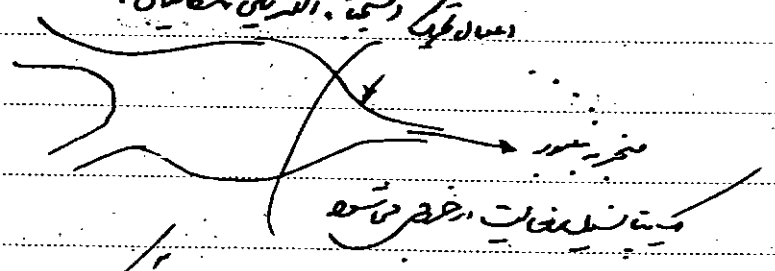
بزرگسالی با 30,000 برابر بزرگی دارد که داخل نورون

تشریح مطالعات از سلول‌های عصبی که به نام نورون (یا یاخته) نامگذاری شده است می‌باشد.
(10^{15} الی 10^{12} عدد) در مجموعه‌ای 10^{7} الی 10^{8} سلول بسته در ۶ لایه قرار دارند.
شکل این نورون‌ها هرمی است یا ستاره‌ای یا لوزی یا گلابی
قطر نورون‌ها 10 الی 100 میکرون اتصالات بین نورون‌ها تقریباً بسیار دقیق و منظم بین آن‌ها وجود دارد. سلول‌های دیگری نیز در مغز وجود دارند به نام سلول‌های گلیا (Glia) از نورون‌ها کوچکترند. تعداد آن‌ها 10 برابر تعداد نورون‌ها می‌باشد
ماده میلین (Myelin) عایق است

در میزبان یک نورون ممکن است بیش از 10^{4} اتصال سیناپسی داشته باشد. به این حساب در مغز انسان 10^{14} الی 10^{16} اتصال وجود دارد.

ساختار الکتریکی مغز

بین داخل سلول و سطح آن ۷۰ الی ۱۰۰ mv اختلاف پتانسیل برقرار است = پتانسیل استراحت
دلیل: توزیع نابرابر یون‌های سدیم + و پتاسیم - در دو سوی غشاء و وجود سوراخ‌هایی پروتئینی به نام کانال در داخل تابیخ سلول. توجه: ضخامت غشاء 10^{-6} cm است.



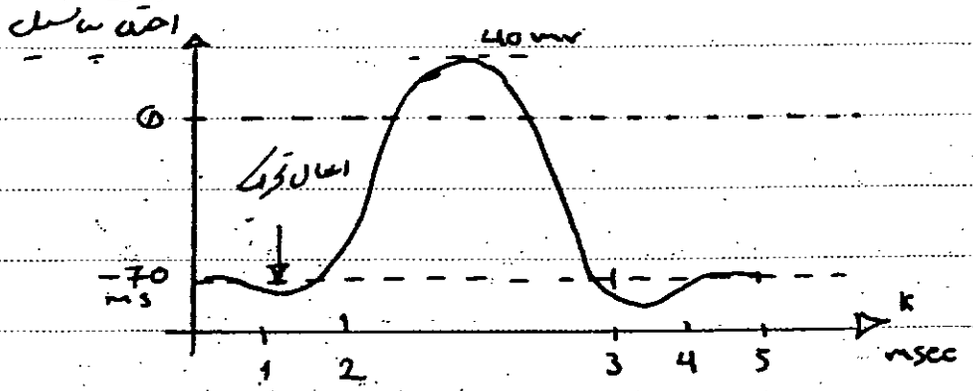
یک سرنخ ضربه‌ای با سرعت 25 m/s روی اکسون

پیشروی می‌کند

Subject:

Year. Month. Date. ( )

آکسون، حدود ۱ ms پس از اعمال محرک، تحریک‌پذیری را نشان می‌دهد

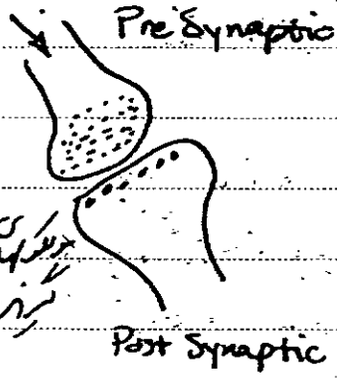


شکل روبرو، پتانسیل کار، رفتار شاخه‌ی وابسته‌ی آزون تحریک‌پذیر

اطلاعات، فقط بصورت پتانسیل عمل از این سرها انتقال پیدا می‌کنند

چگونگی پتانسیل فعالیت در آکسون، شلیک شدن، یا نشدن سلول بعدی چگونه مشخص می‌شود؟

۱- رسیدن پتانسیل کار به سیناپس، یکی از مواد انتقال دهنده به سمت شکاف می‌رود و مولکول‌های خنثی را تخلیه می‌کند تا به سمت شکاف نفوذ کند. انواع مواد انتقال دهنده فعلاً است خیلی زیاد شده اما در تست به نظر می‌رسد تنها یک نوع ماده در نظر باشد.



۲- با افزایش فرکانس و دامنه ضربه، نرخ آزاد شدن مواد انتقال دهنده بصورت (لگاریتمی) افزایش می‌یابد

۳- مولکول‌های گیرنده در سلول پس سیناپسی منتظر دریافت مواد انتقال دهنده نشسته، با دریافت آنها درهم قلاب می‌شوند.

ساختار غشاء و وضعیت تطبیقی سلول پس سیناپسی تغییر می‌کند

- اگر نوع ماده انتقال دهنده تحریک‌کننده (Excitatory) باشد، پتانسیل استراحت سلول پس سیناپسی به سمت صفر می‌رود (دپلاریزاسیون Depolarization) و اختلاف پتانسیل کم می‌شود. اگر این دپلاریزاسیون به بالاتر از یک آستانه برسد، نورون پس سیناپسی شلیک می‌کند

Subject :

Year . Month . Date . ( )

ورود ماده منتقل کننده سبب ایجاد یک دپولاریزاسیون کوچک می‌شود و یک پتانسیل را ایجاد می‌کند

(EPSP) Excitatory Post Synaptic Potential

اگر در پایان سیناپس ۱، سیناپس ۲ی مجاور قبل از آنکه اثر EPSP اول از بین برود، یک EPSP دوم، سوم، ... ایجاد شود اثر آن‌ها با هم جمع می‌شود و اگر این حاصل جمع به یک حد آستانه‌ای برسد سلول AP شلیک می‌کند.

ماده منتقل کننده نباید در شکاف پس سیناپسی تجزیه شود زیرا در این حالت سلول به یک حالت ثابت حداکثر تحریک پذیر دپولاریزاسیون ثابت دوم می‌آید. سلول‌های آنزیمی‌اند که ماده منتقل کننده را می‌شکنند و راه را برای مواد دیگر انتقال تسهیل می‌کنند.

اگر ماده منتقل کننده در شکاف نفوذ شود و فقط از یک گیرنده از گیرنده‌های پس سیناپسی عمل می‌کند.

اگر ماده‌ای منتقل کننده تجویز شود که از یک گیرنده از گیرنده‌های پس سیناپسی جلوگیری کند، از عمل ماده منتقل کننده، از آن گیرنده پس سیناپسی عمل می‌کند.

اگر ماده‌ای منتقل کننده، گیرنده‌های پس سیناپسی را تحریک کند، عمل ماده منتقل کننده را تقلید می‌کند.

نمونه‌ای از این اعمال، وجود مواد (کورار) است که از خوستان گیرنده خود می‌مانند از عمل (ACh) جلوگیری می‌کند حاصله از تأثیر عضلات تنفسی می‌باشد.

(جاذبه‌ی عصبی) (عملکرد سلول کننده) را افزایش می‌دهند سلول از بین نمی‌رود و عضله به یک حالت تهیج و انقباض بیشتر دوام می‌آورد و مرگ ناشی از آن بین از بین در رنگ است.

آثار نیکوتین، حشیش و LSD نیز با اثرات مزبور، خلاص انتقال عصبی در سیناپس نسبت داده شده است.

- سیناپس‌ها می‌توانند در سلول پس سیناپسی (کمی کردن) (وقفه‌ای) ایجاد کنند. اگرچه نحوه عمل یکسان است (با تأثیر سلول در اینجا متفاوت است.

مستقل کننده، باز دارنده (Inhibitory) (پولاریزاسیون) سلول پس سیناپسی را زیاد می‌کند.

این امر به هایپرپولاریزاسیون معروف است و تحت عنوان پتانسیل باز دارنده جزئی شناخته می‌شود (IPSP)

۴- شلیک شدن یا شدن سلول AP به نحو تمام یا هیچ است. خلاصه AP در هر لحظه به سلول پس سیناپسی و بالا خص به

PAPCO

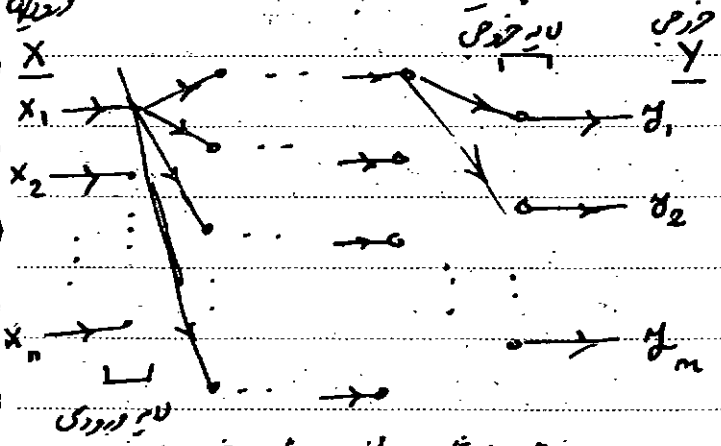
Subject :

Year Month Date ( )

ناحیه‌ای که سلول، پیغام‌اش را بسوی آکسون می‌فرستد، سیناپس نامیده می‌شود. سیناپس محل ارتباط دو سلول است. سیناپس در شلیک شدن یک سلول کاربرد بسزایی دارد. یک سلول متفکر است، زیرا سلول دوم حافظه را بطه سیناپسی ایجاد کند. تعداد این سیناپس‌ها یک عامل موثر است. عامل دیگر فاصله سیناپس‌ها نسبت به سلول پس سیناپسی است. (طول دندریت) عامل دیگر محل دندریت بر روی بدنه است.

۲ ← مفاهیم، تعاریف و انواع شبکه‌های عصبی

یک شبکه عصبی، یک ساختار پردازشی توزیع شده موازی است. فرم یک گراف جهت دار است.



- در این گراف هر گره یک المان پردازشگر (processing Element) یا واحد پردازشگر (processing Unit) یا نرون (Neuron) نامیده می‌شود.

- هر گره می‌تواند به تعداد ورودی داشته باشد اما تنها یک خروجی دارد (خروجی‌های انشعاب یافته عین همان سیگنال خروجی می‌باشد)

خروجی هر گره می‌تواند به هر تعداد گره وصل شود. سیگنال عبور اتصالات یک‌طرفه و برابر سیگنال خروجی می‌باشد.

- بر اتصال گراف وزنی (weight) است که جهت و شدت اثر ورودی تأثیر گره بعدی را مشخص می‌کند.

- گره‌ها به لایه‌ها (Layer) یا (Slab) تقسیم می‌شوند. لایه اول شبکه معمولاً لایه ورودی (Input layer) نامیده می‌شود. هر عنصر این لایه یک سیگنال ورودی از خارج شبکه دریافت می‌کند و آنرا به لایه بعدی می‌فرستد و هیچ کار دیگری روی آن انجام نمی‌دهد. واحدهای این لایه، fanout units نامیده می‌شوند.

Subject:

Year. Month. Date. ( )

- آخرین لایه شبکه، لایه خروجی، Output L. نامیده می‌شود. خروجی‌های (ی) این لایه، خروجی شبکه را تشکیل می‌دهند.

- لایه‌های بین لایه‌های ورودی و خروجی، لایه‌های پنهان، مخفی (Hidden Layers) نامیده می‌شوند.

- هر واحد می‌تواند به واحد(های) لایه خود یا لایه‌های دیگر متصل شود.

- واحد(های) یک لایه معمولاً به طور همزمان در نظر گرفته می‌شوند.

- نحوه پردازش در اتصالات به صورت لحظه‌ای و از لایه‌ای به لایه بعدی در نظر گرفته می‌شود.

- عمل واحد(های) می‌تواند به صورت همزمان (Synchronous ، Episodical) یا غیرهمزمان (ناهمگام) (Asynchronous ، Continuous) باشد. در واحد(های) همزمان در زمان‌های خاصی واحد(های) به‌کار فعال می‌شوند و در نتیجه آن، با استفاده از ورودی‌ها و وضعیت قبلی واحد(ها)، خروجی(های) جدید لایه مشخص می‌شوند. در شبکه‌های غیرهمزمان واحد(ها) به طور پیوسته در حال کار بوده و با دریافت ورودی‌ها خروجی(های) جدید محاسبه می‌شوند.

- کلیه واحدهای یک شبکه معمولاً همزمان در نظر گرفته می‌شوند یا غیرهمزمان.

- روش‌های اتصالات در یک فاز آموزش، یادگیری نامیده می‌شوند. (Training) ، (Learning)

* تنظیم وزن‌ها یک مقایسه یادگیری شبکه و نمونه‌های متعدد ورودی - خروجی انجام می‌شود.

Subject:

Year. Month. Date. ( )

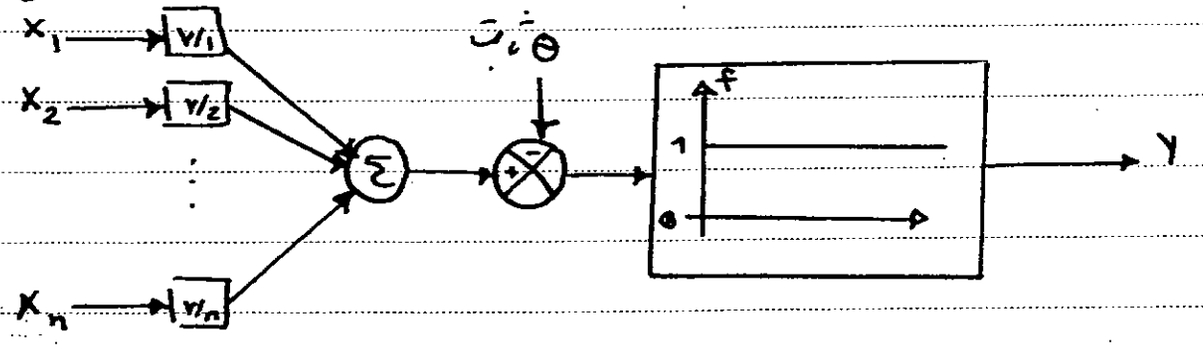
واحدی پردازشگر

1943

McCulloch & Pitts: مدل سلول ای ارائه دادند با توجه به اطلاعات موجود در مورد نحوه عملکرد سلول عصبی و مغز انسان

یک مدل ریاضی برای یک نورون بصورت زیر ارائه دادند.

ورودی‌ها



وزن ورودی

$w_i > 0$: تحریک‌کننده

$w_i < 0$: بازدارنده

$w_i = 0$: عدم اتصال

اثر فرستنده سیناپسی $\leftarrow w_i$

Σ : جمع تحریکات وارد شده به سلول

$\otimes$: آیا جمع تحریکات وارد شده از یک حد θ بالاتر است؟

تابع f $\begin{cases} \rightarrow 1 & \text{فعال شدن سلول} \\ \rightarrow 0 & \text{غیرفعال شدن سلول} \end{cases}$

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

برای سادگی فرض می‌کنیم: $-\theta = w_0 , x_0 = 1$

$$\Rightarrow Y = f\left(\sum_{i=0}^{n} w_i x_i\right)$$

Subject :

Year . Month . Date . ( )

سلول یک واقعی دارای تعدادی کسیدهای بیشتری می باشند :

۱- بردار ورودی سلول به وزن ها بطور پیوسته می باشد نه بصورت گسسته

۲- بسیاری از سلول های واقعی دارای تابع خطی از وزن های آنان را بصورت استاندارد خطا می دهند

مثلاً در روش یادگیری سلول با سلامت خطا تابع خطی انجام می شود

۳- خروجی نورون واقعی یک مقدار پیوسته است نه یک مقدار ثابت .

۴- سلول های واقعی دارای تأخیری با مسافت هستند .

۵- خروجی نورون انتقال داده شده در یک سیناپس ممکن است بصورت غیر قابل پیش بینی تغییر کند .

- واحدی برای شبکه عصبی : با تعمیم مدل سلول ساده به طرق مختلف ، واحدی متفاوت می توان داشت .

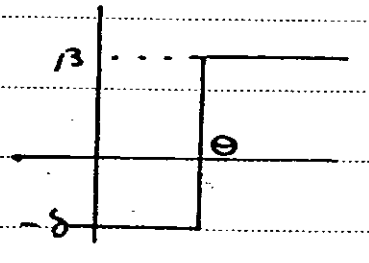
۱- تغییر تابع فعالیت : (Activation, Gain, Tranofer, Squashing Function)

در مدل نورون مبنایی : f تابع پله ای بود

الف) تابع خطی : $f(n) = \alpha x \quad (\alpha > 0)$

ب) تابع حد آستانه ای :

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



در این تابع اگر $\beta = \delta = 1$ ، $\theta = 0$:

$$sgn(n) = f(n) = \begin{cases} 1 & n \geq 0 \\ -1 & n < 0 \end{cases} \quad \text{(Sign Function)}$$

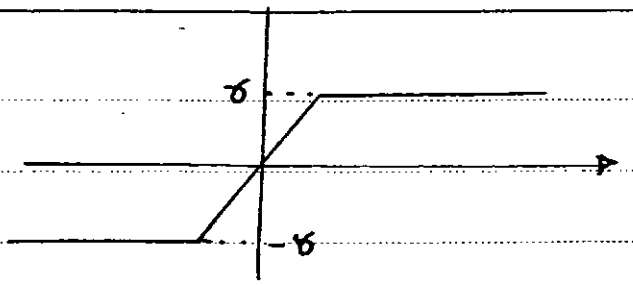
خروجی دوقطبی (دارای ۲ مقدار مختلف غیر صفر)

این اثر شبیه گذاشتن منبع تابع در شبکه الگوریتم می شود . روش Reward & Punish

Subject:

Year. Month. Date. ( )

ج) تابع قطعه قطعه خطی



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

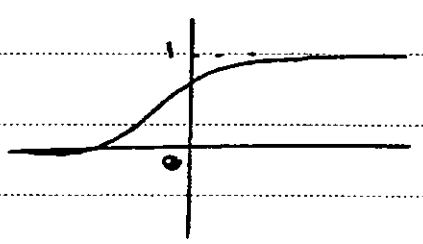
د) تابع سیگموئید (Sigmoid Function)

← تابع سیگموئید دارای (تابع لجستیک (Logistic function)

← دارای پارامتر α

تابع لجستیک:

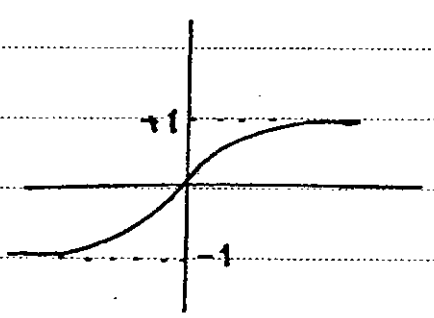
$$f(x) = \frac{1}{1+e^{-\alpha x}} \qquad \alpha > 0$$



مشتق پذیر است

$$f'(x) = \alpha f(x) \cdot [1 - f(x)]$$

$$f(x) = \frac{2}{1+e^{-\alpha x}} - 1 \quad \text{یا} \quad \frac{1-e^{-\alpha x}}{1+e^{-\alpha x}}$$



مشتق پذیر است

$$f'(x) = \frac{\alpha}{2}(1+f(x))(1-f(x))$$

Subject :

Year . Month . Date . ( )

ه) تابع tgh

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

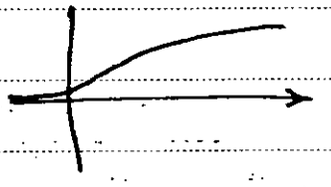
(شیب نقطه‌ای، $\alpha = 2$)

تفاوت در صفر $[-1, 1]$

$$f'(x) = (1 - f(x))(1 + f(x))$$

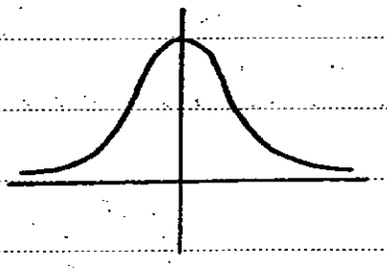
و) تابع نسبت درجه

$$f(x) = \begin{cases} \frac{x^2}{1+x^2} & x > 0 \\ 0 & \text{سوی دیگر} \end{cases}$$



ز) تابع گوسی

$$f(x) = e^{-x^2/\sigma^2}$$ σ^2 واریانس



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

تقسیم دیگر: تابع فعالیت برای واحدی مختلف لزوماً مشابه انتخاب می‌شود.

مثلاً: کلیه واحدی می‌توانند از تابع یک‌سان در نظر گرفته شوند، واحدی لایه‌ای مختلف را متفاوت

تقسیم سوم: تابع خطی در نظر گرفته شود.

$$y = f(\Sigma w_i x_i) \xrightarrow{\text{شبکه}} y = f\left(\Sigma w_i x_i + \sum_{k=1}^{n_i}\sum_{l=1}^{n_i} w_{kl}\, x_k x_l\right)$$

(Second order) مرتبه دوم

این تقسیم از تقسیم درجه ۱ قوی‌تر می‌باشد.

Subject:

Year Month Date ( )

تعمیم چهارم: می‌توان مقادیر ورودی را به طور غیرخطی در نظر گرفت. در این حالت هم در آموزش شبکه با توجه به مقادیر ورودی تغییراتی ایجاد می‌شود.

تعمیم پنجم: استفاده از مقادیر فازی

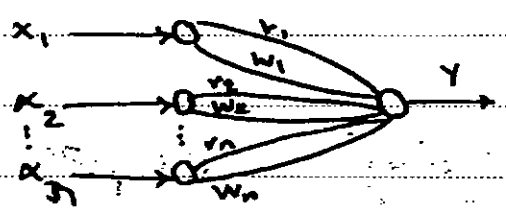
فازی سازی ← دقیق سازی

۱) ورودی‌ها فازی در نظر گرفته شوند.

۲) پارامترهای مقادیر فازی در نظر گرفته شوند.

۳) ورودی‌ها و پارامترهای مقادیر فازی در نظر گرفته شوند.

تعمیم ششم: ارتباطات را به انتخاب داشته باشیم.



یک روش برای یادگیری: مقادیر بردار حداقل و متوسط بردار ورودی‌ها (ورودی‌های) یک دسته ورودی به کار گرفته می‌شود.

$\bar{v}$: بردار مقادیر متوسط

$\bar{V}$: بردار مقادیر بیشینه

$$y = f(x, \bar{v}, \bar{V})$$

در این حالت تابع فعالیت، تابع گوسی انتخاب شده است.

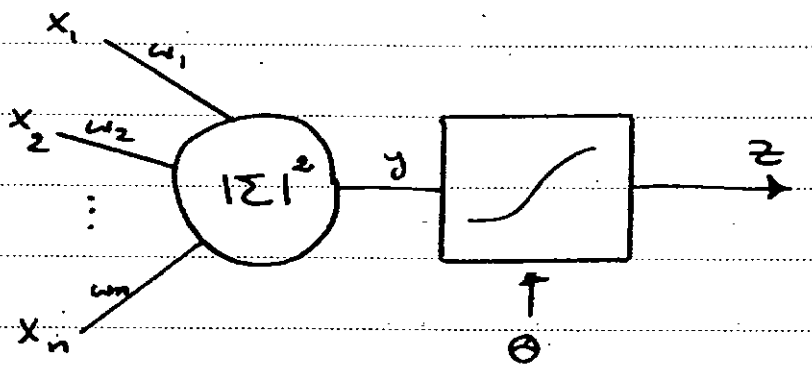
$$y = f\left(\sum_{i=1}^{n}\left(\frac{x_i - w_i}{v_i}\right)^2\right) \qquad f(x) = e^{-x^2/2}$$

مقادیر ورودی را به انتخاب به صورت دیگر نیز به کار گرفته شده‌اند. مثلاً در انتخاب مقادیر حداقل و حداکثر در بعضی از مقالات می‌بینید.

Subject :

Year . Month . Date . ( )

تعمیم ششم : نورون‌های مختلط



$$w_i = a_i + j b_i$$

$$y = \left| x_1 (a_1 + j b_1) + x_2 (a_2 + j b_2) + \cdots + x_n (a_n + j b_n) \right|^2$$

$$= (a_1 x_1 + a_2 x_2 + \cdots + a_n x_n)^2 + (b_1 x_1 + \cdots + b_n x_n)^2$$

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

تعمیم هشتم : نورون‌های سیگما-پی

$$y = \sum_k w_k \prod_{i \in I_k} x_i \qquad I_k = \{1, 2, \ldots, n\} \ ?$$

تعمیم نهم : نورون‌های ضربه‌ای Spiking Neurons

؟ سوال برای شبکه‌های عصبی

۱) نسل اول : پرسپترون مک کلاک و پیتز، هبینیه، پرسپترون، پرسپترون

۱۲ ← دوم : پرسپترون چندلایه و نورون‌های تابع فعالیت پیوسته

پرسپترون چندلایه، الگوریتم پس‌انتشار

PAPCO

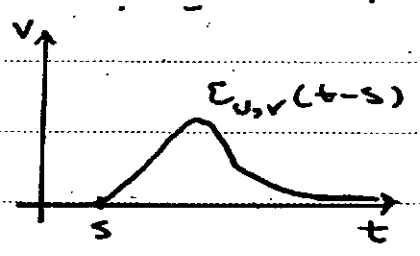
Subject :

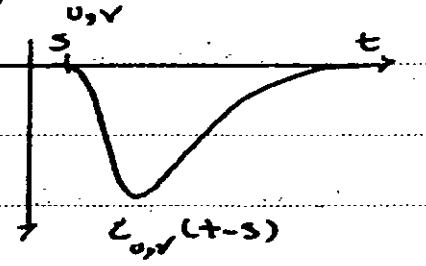
Year . Month . Date . ( )

؟ ۲) نسل سوم : در سال نورون‌ها ضربه‌ای می‌باشند.

در ساده‌ترین مدل ضربه‌ای فرض می‌شود که اگر پتانسیل P_v نورون v به یک حد آستانه‌ای Θ_v می‌رسد شلیک می‌کند. پتانسیل P_v حاوی EPSP و IPSP‌های ورودی به نورون است. شلیک یک نورون پیش‌سیناپسی u در زمان s در زمان t به پتانسیل P_v به اندازه $w_{u,v}\,\varepsilon_{u,v}(t-s)$ به نورون v اثر می‌گذارد.

$\varepsilon_{u,v}(t-s)$ تابع پاسخ u است که به شکل‌های زیر می‌باشد.

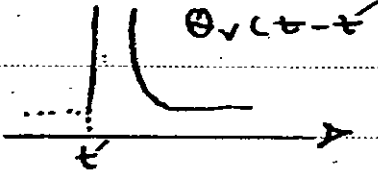




(الف) EPSP

برای سادگی $w_{u,v}$ را ثابت در نظر گرفت.

نورون طبیعی دارای پتانسیل استراحت $-70mv$ و پتانسیل شلیک $-50mv$ است. نورون پیش‌سیناپسی پتانسیل نورون پس‌سیناپسی را در زمان‌هایی که تأثیر جهشی صورت می‌دهد میان دو مقدار تدریجی تغییر می‌دهد. نورونی که در زمان t' شلیک کرده برای مدت کوتاهی قادر به شلیک نیست. این موضوع توسط یک تابع آستانه $\Theta_v(t-t')$ بصورت زیر مدل می‌شود:



در این نوشتار اگر $F_u \subseteq R^+$ مجموعه زمان‌های شلیک نورون u باشد پتانسیل نورون v در زمان t عبارت است از:

$$P_v(t) = \sum_u \sum_{s \in F_u} w_{u,v}\,\varepsilon_{u,v}(t-s)$$

که در آن u به ورودی‌های متصل به

Subject:

Year Month Date ( )

وزن $\theta_V(t+t') \leftarrow P_V(t)$ پیش‌بینی می‌کند

رفتار دینامیکی واحد

برای اینکه نورون مانند یک سلول عصبی رفتار کند، از معادله دیفرانسیل زیر تبعیت کند:

$$\frac{dy}{dt} = -g(y) + \sum_{j=0}^{n} W_j x_j = -g(y) + I$$

$I \rightarrow$ تأثیر ورودی‌ها

تأثیر فیدبک‌ها

در حالت تعادل $\frac{dy}{dt} = 0$

$$\Rightarrow \quad -g(y) + I = 0 \quad \Rightarrow \quad y = g^{-1}(I) = g^{-1}\left(\sum_{j=0}^{n} w_{ij} x_j\right) = f\left(\sum w_{ij} x_j\right)$$

این نقطه همان رابطه‌ای است که برای خروجی نورون نوشته بودیم.

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

اتصالات شبکه:

جهت: جهت جریان اطلاعات را مشخص می‌کند.

وزن: میزان تأثیر سیگنال

وزن‌ها در یک شبکه عصبی، آموزش شبکه تعیین می‌شوند. مقادیر مثبت وزن‌ها نشان‌دهنده اتصالات دارنده هستند و مقادیر منفی نشان‌دهنده اتصالات بازدارنده هستند. ۰ = عدم اتصال

Subject :

Year . Month . Date . ( )

در ترکیب می توانند به نوع لایه ای وابسته باشند : بردار حقیقی، مختلط، ماتریکس و کامل 2 و ...

باید بتوانیم مشخص کنیم که هر اتصال از کجا شروع می شود و در کجا خاتمه می یابد. اگر واحدهای شبکه را از 1 تا n شماره گذاری کنیم می توانیم یک ماتریس اتصالات Interconnection Matrix به صورت زیر تشکیل دهیم :

$M = [M_{ij}]$ $\quad M_{ij} \begin{cases} 1 & \text{اگر اتصال از واحد } i \text{ به } j \text{ وجود دارد} \\ 0 & \end{cases}$

برای ارائه اتصالات وزنهای از ماتریس وزنهای شبکه تعریف شده است (Weight Matrix)

هر ماتریس وزن اتصالات یک لایه به لایه بعد را می دهد.

$W = [w_{ij}]$ $\quad w_{ij}$: وزن واحد i لایه به واحد j لایه بعدی

بعضی مراجع اندیسهای فوق را برعکس می نویسند یعنی w_{ij} را برابر با وزن اتصال از واحد i به واحد j لایه بعد

ماتریس وزن تشکیل شده، در اینحالت transpose ماتریس حالت قبل می باشد.

کتاب Neuro Computing از حالت اول استفاده می کند.

جلسه بعد :

w_{ij} وزن اتصال از واحد i لایه قبل به واحد j لایه فعلی

$\bar{w}_{ij}$ " " ورودی به واحد i لایه فعلی از واحد j لایه قبل

اتصالات معکوس یک به صورت زیر باشد :

1) کامل (Full) : از خروجی به تمام واحدهای لایه بعد وصل می شود.

Subject :

Year . Month . Date . ( )

۲) تابع متقارن (Uniformly Random)

برداری به طول n و دوری از بردارهای R بعدی که تابع آن‌ها با توزیع یکنواخت انتخاب شده از R^n دریافت می‌کند.

۱) تک به تک (One to One)

برداری حداکثر یک لایه، یک الصاق از لایه قبل دریافت می‌کند

معماری شبکه‌های عصبی

۱) شبکه‌ی پیش خور feed forward

۲) " پس خور (پس رو) (بازگشتی) feedback, Recurrent

در شبکه‌ی پیش خور گراف شبکه هیچ مسیر بسته‌ای ندارد.

" " پس خور " " دارای مسیر بسته است.

شبکه‌ی پیش خور عموماً ایستا هستند (تنها یک دسته مقادیر خروجی در هر لحظه به ازای ورودی ای) هیچ رفتاری از حالت قبلی شبکه مستقل است.

شبکه‌ی پس خور دینامیک هستند (وقتی که ورودی اعمال می‌شود، خروجی از طریق مسیرهای بسته مجدداً به ورودی اعمال می‌شود ممکن است خروجی را تغییر دهد)

پیش خور: شبکه پرسپترون چندلایه، شبکه تعلیم یابی رقابتی

پس خور: شبکه هاپفیلد، شبکه (ART) نظریه تشدید تطبیقی

PqPCO

۱۷

Subject:

Year. Month. Date. ( )

Classification

۲- واحدی پردازشگر در سامانه دسته‌بندی:

در سامانه دسته‌بندی، دسته‌بندی کننده، با استفاده از نمونه‌های شناخته شده یاد می‌گیرد چگونه دسته‌ها را از یکدیگر تشخیص دهد. الگوها را به دسته‌ای متناظر دسته‌بندی کند.

پس، در دریافت هر ورودی جدید می‌تواند تصمیم بگیرد ورودی، کدام دسته تعلق دارد.

ساده‌ترین حالت دسته‌بندی: دو دسته

دسته اول: اشیاء متعلق به دسته‌ای خاص

دسته دوم: سایر اشیاء

انجام دسته‌بندی، شبکه مصنوعی:

$\geq 0 \rightarrow y$ { $\Phi(-1)$ دسته دوم ; 1 دسته اول }

مکریلا-پیتز

حالت m دسته‌ای

$\geq 0 \rightarrow y_1$

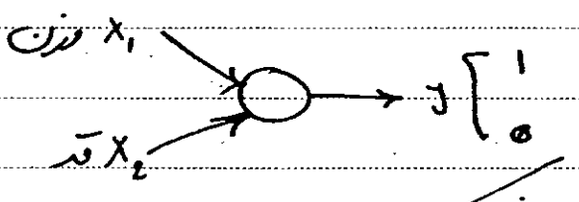
$\geq 0 \rightarrow y_2$

$\geq 0 \rightarrow y_m$

هرکدام از خطی ایجاد شده می‌فهمیم که ورودی متعلق به کدام دسته بوده است.

معمولاً دسته‌بندی در واقع معادل ناحیه‌بندی فضای ورودی توسط شبکه است.

مثال: تشخیص y زن یا مرد از x سکته‌ات



انتخاب ویژگی: قد، وزن

فاز آموزش: تعدادی داده مطلوب سکته‌ات را به سیستم معرفی می‌کنیم

$(\underline{x}, y)$

$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

Subject:

Year Month Date ( )

مدل پرسپترون:

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

$$\underline{X} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \qquad \underline{W} = \begin{pmatrix} -\theta \\ w_1 \\ w_2 \end{pmatrix}$$

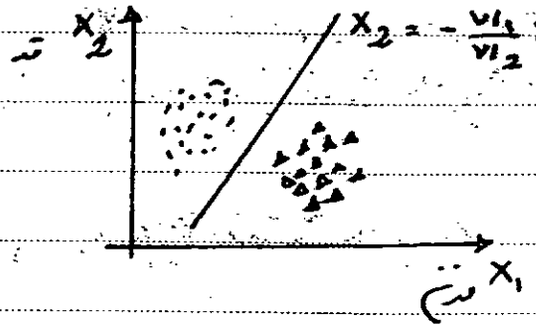
$$I = -\theta + w_1 x_1 + w_2 x_2$$

$$f(I) = \begin{cases} 1 & I \geq 0 \\ -1 & I < 0 \end{cases}$$

در فاز آموزش با استفاده از نمونه ها، وزن ها مشخص می شوند.

متغیرهای (x_1, x_2) که I مقدار مثبت دارند + می شوند از آن مقادیری که I مقدار منفی دارند - می شوند توسط مرز $I = 0$ از هم جدا می شوند.

معادله مرز $w_1 x_1 + w_2 x_2 - \theta = 0$



وزن بردار ▲ کلاس ■

فاز استفاده

اگر مساله بصورتی باشد که نمونه های یک دسته با یک خط فوق و نمونه های دسته دیگر با خط زیر آن جدا شوند مسئله بصورت خطی جدایی پذیر است.

(Linearly Seperable)

PAPCO

Subject:

Year. Month. Date. ( )

قواعد یادگیری هبب و قواعد یادگیری

* ۱) قاعده یادگیری هبب (Hebb)

قاعده هبب ساده ترین و قدیمی ترین قاعده یادگیری است که در سال 1949 توسط Donald Hebb مکتوب شد.

هنگامی که نورون i به طور مکرر و مداوم در آتش شدن نورون j مشارکت کند میزان تأثیر این نورون در آتش کردن نورون j افزایش می یابد.
تحقیقات بعدی نشان داد که نظریه هبب به طور کلی صحیح است.

قانون هبب بر پایه قانون یادگیری هم وقوعی (Coincidence learning) تعلق دارد.

در یک قاعده ساده، نتایج فعالیت قبلی (از قانون هبب) برای یادگیری استفاده می شود و وزن ها تغییر می یابند.

طبق قانون هبب هرگاه دو نورون به طور همزمان فعال باشند وزن اتصال بین آن ها افزایش یابد.
در سال 1988، دونالد کلمن نتیجه را این طور عنوان کرد که قاعده هبب را می توان به صورت بیان ترجیح برای گسترش دو نورون به طور همزمان خاموش نیز باشند، قدرت اتصال نورون ها افزایش می یابد.

الگوریتم هبب به این صورت خواهد بود:

۵) مقداردهی اولیه وزن ها مشخص شود (مثلا $W_i = 0 \quad i:1..N$)

۱) برای هر زوج نمونه ورودی (s,t) قدم ۲ و ۳ را اجرا می کنیم.
(source, target)

Subject :

Year . Month . Date . ( )

۲) به ورودی و خروجی مقدار نرونه نسبت داده شده

$$\begin{cases} X_i = S_i & i : 1 \dots N \\ y = t \end{cases}$$

۳- وزن ها و مقدار بایس از رابطه زیر اصلاح شده

$$\begin{cases} W_i^{NEW} = W_i^{OLD} + X_i y \\ \theta^{NEW} = \theta^{OLD} + y \end{cases}$$

نکات : چون $X_i y$ داریم پس $(0,0)$ و $(0,1)$ تفاوتی نمی کند در حالیکه آرگومان ها در قطبی داشته باشیم نتایج فرق می کند.

← (استفاده از مقادیر در قطبی باعث عملکرد بهتر این قانون می شود)

* ۲) مطرح پرسپترون

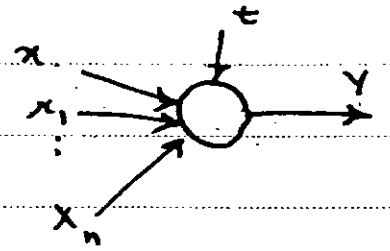
برای شبیه سازی کامپیوتری شکل حروف لاتین از روی تصویر حروف سوراخ شده بر روی یک برگه مقوایی ۲۰×۲۰ از سلولهای حساس متمرکز می گردد. ۴۰۰ سیگنال حاصل به شبکه عصبی پرسپترون داده می شود.

سیستم دارای ۵۱۲ تا نرون پرسپترون بوده که وزن های الکتریکی تغییر داده می شود. تنظیم اتوماتیک وزن ها با استفاده از این تا نرون پرسپترون انجام می شود. تعدادی واحد آستانه گیر برای ایجاد خروجی در نظر گرفته شده و اتصالات بین واحد حسی و یک صفحه فعال دسترسی ای داده می شود. این شبکه پرسپترون بر روی این صفحه بسته می شود. اتصالات بین لایه ای اول و دوم شبکه به طور اتفاقی با وزن های $+1$ و -1 برقرار گردیده و نتایج فعالیت واحدهای لایه دوم بصورت نتایج لایه ای واحد

Subject:

Year. Month. Date. ( )

دسته بندی دیگری از الگوها در نظر گرفته شده. در این حالت خروجی، یک تابع از ورودی است.



مدل خطی پرسپترون

مثال: این مدل خطی خروجی ۲ دسته A و B را از هم تفکیک می‌دهد، بنابراین ورودی X

$$y = \begin{cases} 0 & A \text{ دسته} \\ 1 & B \text{ دسته} \end{cases}$$

در فاز آموزش بردارهای (X, t) مقدار مطلوب خروجی X ورودی

اعمال می‌شود و VI طبق قانون پرسپترون تغییر می‌کند.

$$X = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad VI = \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_n \end{pmatrix}$$

قانون پرسپترون:

$$VI^{NEW} = VI^{OLD} + (t-y)X$$

$$\therefore \Delta VI = (t-y)X$$

$$t-y \begin{cases} 1 & \text{دسته } A \text{ اعلام شده، در حالیکه باید } B \text{ اعلام می‌شد} \\ -1 & \text{دسته } B \text{ اعلام شده، در حالیکه باید } A \text{ اعلام می‌شد} \\ 0 & \text{دسته صحیح اعلام شده است} \end{cases}$$

Subject:

Year. Month. Date. ( )

(دیدن اولین نرونها)

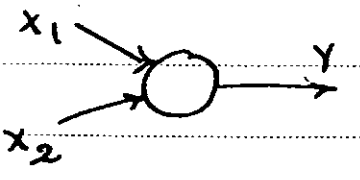
روز نفت ثابت کرد که اگر داده‌ها به صورت خطی جدا پذیر باشند، وزن توجه به مقادیر اولیه وزن‌ها

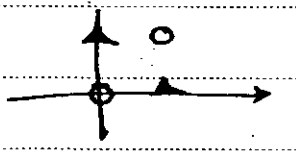
اگر سیستم پرسپترون را یک بار کامل کند و تکرار روی داده‌ها دیگر خطا نداشته باشد [همگرایی الگوریتم]

نقطه ضعف: به صورت خطی جدا پذیر نیست.

مثال: پیاده‌سازی XOR با پرسپترون

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0





دسته اول: $\begin{matrix}0&0\\1&1\end{matrix}$ خروجی = 0

دسته دوم: $\begin{matrix}0&1\\1&0\end{matrix}$ خروجی = 1

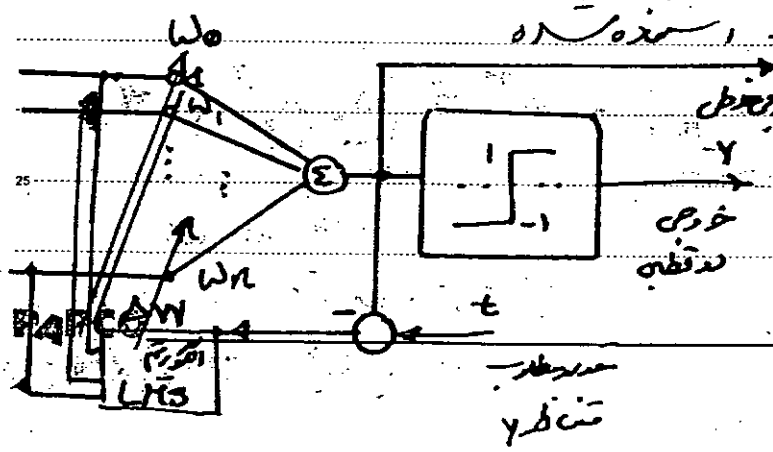
دو نسخه دیگر از قانون یادگیری پرسپترون مطرح شد:

$$\Delta W_i = \alpha (t-y) x_i \qquad \alpha = \frac{1}{\|x\|} \quad \text{یا} \quad \alpha = \frac{|x - W_i|}{\|x\|^2}$$

Adaptive Linear Element — واحد پردازشگر تطبیقی

ویدرو 1959 در استنفورد

در پیاده‌سازی الگوریتم از سیگنال خطا استفاده شده



Subject:

Year. Month. Date. ( )

الگوریتم LMS، ΔW ها را بدست می‌آورد.

$$X = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad W = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$$

$$I = \sum_{i=0}^{n} w_i x_i \quad ; \quad \underline{W} \cdot \underline{X} = W^T X \quad ; \quad X^T W$$

$$y = sgn(I)$$

قانون یادگیری LMS

Least Mean Square

Least Mean Square of Error = LMSE

یادگیری بر پایه خطا که سعی می‌کند میانگین مربع خطا را کمینه کند.

(یا) ترجمه بر پایه تئوری

Performance learning از جمله قوانین

فرض کنید یک دسته از n ورودی $\underline{X}$ ، تابع هدف P در دست نباشد. هدف این است که یک بردار $\underline{W}$ پیدا کنیم تا تابع زیر را حداقل کند:

$$F(\underline{W}) = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} (t_k - I_k)^2 = E\{(t_k - I_k)^2\}$$

$$F(\underline{W}) = E\{(t_k - W^T X_k)^2\}$$

$$= E\{t_k^2 - 2 t_k W^T x_k + W^T X_k X_k^T W\}$$

$$= \underbrace{E\{t_k^2\}}_{P = \text{ثابت}} - 2W^T \underbrace{E\{t_k x_k\}}_{q} + W^T \underbrace{E\{X_k X_k^T\}}_{R_{n+1 \times n+1}} W$$

ماتریس متقارن

شرط نیمه معین مثبت $\sum r_{ij} x_i x_j \geq 0$

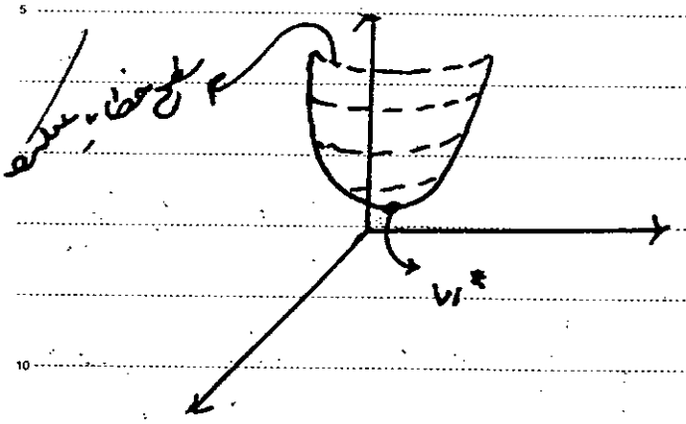
PAPCO

Subject :

Year . Month . Date . ( )

$$F(\underline{w}) = P - 2\underline{w}^T\underline{q} + \underline{w}^T R \underline{w}$$ تابع درجه 2 از $\underline{w}$

واستگی F به $\underline{w}$ بصورت درجه 2 می‌باشد و با توجه به خاصیت فوق R، می‌توان نتیجه گرفت که $F(\underline{w})$ یک سطح سهمی‌گون است. سه ارزش‌های درجه‌ی دوم که مقادیری که نقاط حداقل‌ساز در آن $F(\underline{w})$ حداقل می‌شود. ($\underline{w}^*$)



$$\nabla_{\underline{w}} F(\underline{w}) = \nabla_{\underline{w}} \left(P - 2\underline{w}^T\underline{q} + \underline{w}^T R \underline{w} \right)$$

$$= -2\underline{q} + 2R\underline{w} = \underline{0}$$ شرط لازم برای حداقل

$$\Rightarrow \underline{w}^* = R^+ \underline{q}$$

R^+: معکوس تعمیم یافته Generalized Inverse

Psendo-Inverse

۱۴ آبان

$R^+ = R^{-1}$ اگر R معکوس‌پذیر باشد (non-singular)

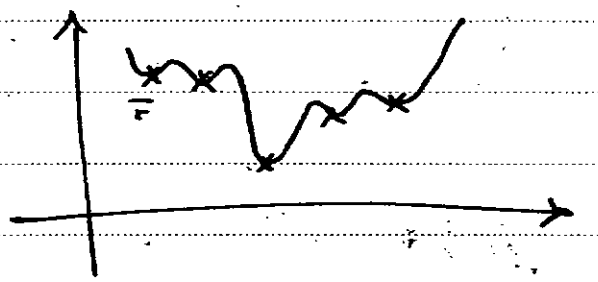
در این حالت $\underline{w}^*$ تنها نقطه‌ی ثابت سطح سهمی‌گون خواهد بود.

Subject:

Year. Month. Date.

اگر R معکوس‌پذیر نباشد (Singular) w^* نزدیک‌ترین مینیمم به مرکز مختصات خواهد بود.
سطح معکوس‌پذیری معکوس‌پذیر خواهد بود (در نتیجه w^* یکتا خواهد شد)

نزدیک‌ترین مینیمم به مبدأ را به ما می‌دهد.



پیاده‌کردن QR کار راحتی نیست. تقریباً دو برابر زمان می‌برد.

در سال 1959 شروع کرد در مورد w^* با شروع از یک نقطه اولیه (w_0) و حرکت در جهت منفی گرادیان سطح معکوس پیش رفت. چون در هر نقطه‌ای سطح معکوس $-\nabla f(w)$ در جهت تندترین کاهش است، با حرکت در این جهت می‌توان مینیمم تابع را پیدا کرد.
لذا $-\nabla f(w_1)$ به دست می‌آید.

$$\nabla_w f(w_1) = \nabla_{w_1}\left[\lim_{N\to\infty} \frac{1}{N}\sum (t_k - I_k)^2\right]$$

$$= \lim_{N\to\infty} \frac{1}{N} \nabla_w \left[\sum (t_k - I_k)^2\right]$$

$$= \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} 2(t_k - I_k) \nabla_{w_1}(-I_k)$$

δ_k $\quad I_k = w^T x_k$

$\quad -x_k$

$$\Rightarrow \nabla = -\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} 2\delta_k x_k = -2E[\delta_k x_k]$$

تقریب زدن δ_k و x_k

Subject:

Year. Month. Date. ( )

در این روش تصحیح وزن‌ها بر اساس مقدار $\delta_k x_k$ از بین $\delta_k x_k$ های مختلف برای تصحیح $\underline{w}$ استفاده می‌شود. آزمایش‌ها نشان داده‌اند که نرخ یادگیری α باید در محدوده $0 < \alpha < \frac{2}{\text{بزرگترین مقدار ویژه } R}$ باشد.

$$\underline{w}_{k+1} = \underline{w}_k + \alpha \delta_k \underline{x}_k \qquad (4.32)$$

قانون یادگیری widrow و LMS نیز نامیده می‌شود.

چون معمولاً تعیین مقادیر ویژه R مشکل است، α را معمولاً با سعی و خطا مشخص می‌کنند. مقادیر محتمل آن در محدوده $0.01 \leq \alpha \leq 10$ و البته در رابطه با $\alpha > 0.1$ توصیه نمی‌شود. اگر α خیلی نزدیک به صفر باشد الگوریتم همگرا می‌شود اگر خیلی کوچک باشد زمان همگرایی بسیار زیاد خواهد شد.

در سال 1988 محدوده پیشنهاد شده برای α به صورت $\frac{0.1}{n} \leq \alpha \leq \frac{1}{n}$ در نظر گرفته شد که $n \to$ تعداد ورودی‌ها

دو نسخه مختلف قانون widrow:

بازگشتی را

Adaline

۱) نسخه Batching:

در این روش پس از تعداد زیادی $\underline{x}$ که بردار وزن‌ها $\underline{w}$ ثابت باقی می‌ماند (معمولاً شامل همه الگوهای آموزشی) برای هر یک از $\underline{x}_k$ ها δ_k محاسبه می‌شود و سپس مجموع $\delta_k \underline{x}_k$ ها در قانون یادگیری به کار گرفته می‌شود.

۲) نسخه Momentom:

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

$$\Delta \underline{w}_{k+1} = \alpha(1-\mu)\delta_k \underline{x}_k + \mu \Delta \underline{w}_k$$

$\mu \gg \alpha$ معمولاً $\mu = 0.9$ (ثابت ممنتم یا تغییرات)

PAPCO

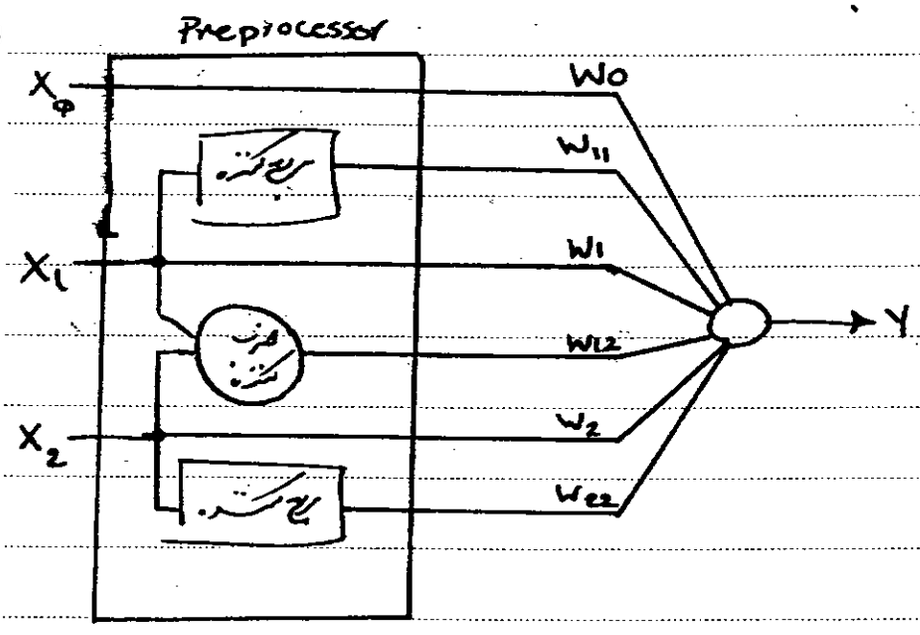
Subject:

Year. Month. Date. ( )

هریک از این مدل‌ها در حالت‌های کلی و در حالت‌های خیر. البته مطلق‌ترین شیوه متفاوت هم نیست.

مرز تصمیم‌گیری این‌ها نیز یک خط است ($I = W^T X$) و مهمترین مسائلی که مربوط خطی جدایی‌پذیر نباشند قابل حل‌اند.

درجات غیرخطی مرتبط از این‌ها درجه 2 استفاده کنیم Polynomial



$$I = W_0 + W_1 X_1 + W_2 X_2 + W_{11} X_1^2 + W_{12} X_1 X_2 + W_{22} X_2^2$$

یک مرز بیضی شکل داریم.

مسائل بیشتری را حل می‌کند و تعداد وزن‌ها بیشتر که یادگیری بهتر برای آن انجام شود نیز زیادتر شد.

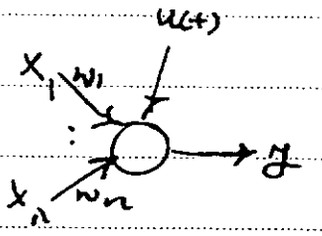
Subject :

Year . Month . Date . ( )

واحد پردازش عنصر اصلی شبکه

Liistan

توسط گرانبرگ معرفی شده.



در این w_i معیار متوسط وزنی $u(+)$ (قدرت تحریک ورودی) x_i ، ضریب x_i را به دست می‌دهد. فرض بر آنست که اگر x_i صفر هستند آن وقتی که تحریک نباشد مقدار ثابت مثال تحریکی تعیین می‌کند.

$$w_i = \overline{x_i : u}$$

این قانون، قانون یادگیری گروسبرگ نامیده شده و جزو قوانین یادگیری فیلتر است Filter Learning

$$w_i^{NEW} = w_i^{OLD} + a\left(x_i u - w_i^{OLD}\right) U(x_i)$$

a: $0 < a < 1$

$U(x_i)$: تابع پله

اگر فرض کنیم مجموعه‌ای داشته باشیم که در آن زمان‌های متفاوت x_i صفر نشده باشد قانون یادگیری به صورت زیر خلاصه می‌شود:

$$w_i^{NEW} = w_i^{OLD} + a\left(x_i u - w_i^{OLD}\right)$$

می‌شود نشان داد که بردار وزن‌ها در قانون فوق، در نهایت به $w_i = \overline{x_i u}$ همگرا خواهد کرد.

در قانون فیلتر، هر یک از جمله‌های حاصل سیگنال مطلوب را ارائه می‌دهد و مقدار خروجی که فیلتر در سیگنال ورودی اعمال شده باشد، بردار وزن اختصاص داده می‌شود.

Subject :

Year . Month . Date . ( )

بخش پنجم — تداعی الگو و شبکه‌های تداعی‌گر

یادگیری ≡ فرآیند تشکیل تداعی بین الگوها

از نظر حافظه انسان خاطراتی که با هم شباهت دارند، در خاطره یکدیگر، یا در کنار هم آن را می‌آفرینند، به‌صورت ساده با یکدیگر مرتبط می‌شوند.

الگوها که با هم مرتبط می‌شوند ممکن است از یک نوع باشند، از انواع مختلف

- رنگ یک گل ← تداعی یک گل دیگر
- بوی یک گل ← شکل یک گل

حفظ کردن الگو ≡ تداعی کردن، به خودش

در حافظه انسان اگر الگوی تحریکی شبیه یک الگوی مشخص فعال شود، بدون نیاز به اینکه الگوی ورودی (یعنی شبیه الگوی قبلی (یادگرفته) باشد الگوی مربوط تداعی می‌شود.

در کامپیوتر مسأله به این شکل است.

در این بخش شبکه‌های ساده‌ای معرفی می‌شوند که می‌توانند مجموعه‌ای از زوج الگوها را یاد بگیرند و تداعی کنند. این شبکه‌ها را شبکه‌های Associative Networks می‌نامند.

(شبکه‌های تداعی‌گر، تداعی، انجمن، شبکه‌های برگشتی، مرتبط‌کننده، شبکه‌های حافظه، انجمنی)

شبکه تداعی‌گر تک لایه، مدل نگاشتی به صورت زیر انجام می‌دهد:

$$\Psi : X_i \rightarrow Y_i \qquad i = 1, 2, \ldots$$

P4PCO $X_i \in \mathbb{R}^n \qquad Y_i \in \mathbb{R}^n \qquad Y_i = \Psi(X_i)$

Subject :

Year . Month . Date . ( )

انواع شبکه‌های متداول

۱) شبکه‌های پیشرو (پیش‌خور)

۲) " " بازگشتی

در شبکه‌های پیشرو اطلاعات تنها از ورودی ← خروجی حرکت می‌کند. با اعمال ورودی X تنها برای یک خروجی Y متناظر آن در خروجی می‌شود.

در شبکه‌های بازگشتی اطلاعات از خروجی هم به ورودی حرکت می‌کند. با اعمال ورودی X، حلقه‌ای از بازگشت ها فعال می‌شود و شبکه به تعادل پایداری می‌رسد.

دسته‌بندی دیگر:

۱) شبکه‌های خودمتداعی — Auto Associative

۲) " غیرخودمتداعی — Hetero Associative

در شبکه‌های خودمتداعی $m=n$ ، $y_i = x_i$

در " غیرخودمتداعی $y_i \neq x_i$

دسته‌بندی سوم:

۱) شبکه‌های تقریب‌زن (Assentive)

۲) " درون‌یاب (Interpolative)

در روش‌های حذف خطا با کمک ε:

$$\psi(x_i + \varepsilon) = y_i$$

$$\psi(x_i + \varepsilon) = y_i + \delta$$

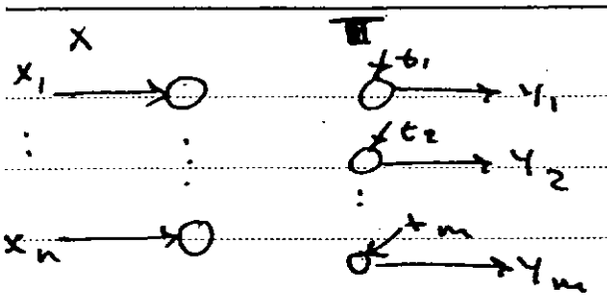
$$|\varepsilon| \to 0 \Rightarrow \delta \to 0$$

PAPCO (خطای خروجی متناسب با خطای ورودی)

Subject:

Year. Month. Date. ( )

شبکه عصبی مبتنی بر خوشه‌بندی



$\underline{X} = [x_1 \dots x_n]^T$ $n\times 1$

$\underline{Y} = [y_1 \dots y_m]^T$ $m\times 1$

$T = [t_1 \dots t_m]^T$ $m\times 1$

$\underline{W}_i = [w_{1i}\ w_{2i} \dots w_{ni}]^T$ $n\times m$

$Y = W^T X$ $\quad W = (\underline{W}_1\ \underline{W}_2 \dots \underline{W}_m)$

$m\times n \times n\times 1$ $\quad 1\times n \times n\times 1$

$\omega = (TX^T)^T = XT^T$

عملکرد حالت استفاده از شبکه

اگر با بردار الگوی (S_i, I_i) برای آموزش شبکه کار کرده شده و پس از آموزش، دوباره $i = 1, \dots, n$

$\underline{X} = \underline{S}_k$ به شبکه اعمال کنیم خروجی را برابر $\underline{T}_k$ می‌دهد.

اما اگر $\underline{X} = \underline{S}_k + \underline{\varepsilon}$ به شبکه بدهیم خروجی را برابر $\underline{T}_k + \underline{\delta}$ می‌دهد یعنی مقدار دقیق را با خطا از خروجی می‌دهد.

آموزش شبکه

الف) استفاده از قانون هب

$W_{ij}^{NEW} = W_{ij}^{OLD} + S_{ki}\, t_{kj}$ برای k امین نمونه آموزش

$\underline{S}_k = (S_{k1} \dots S_{kn})^T$ $\quad \underline{T}_k = [t_{k1} \dots t_{kn}]^T$

$W^{NEW} = W^{OLD} + S_k T_k$ (بدون نرخ یادگیری)

Subject:

Year. Month. Date. ( )

بردار وزن W^{NEW} بعد از یک مرحله آموزش برای W^{OLD} فرض کنید. فرض کنید: $W = 0$

بردار L نمونه آموزشی: $W = S_1 T_1^T + S_2 T_2^T + \dots + S_L T_L^T = \sum_{i=1}^{L} S_i T_i^T$

حال با انتخاب تابع سیگنوم، شبکه چقدر در پاسخ‌دهی از خود خطا نشان خواهد داد؟

فرض کنید ورودی $X_k = S_k$ به شبکه اعمال شده. در این حال داریم:

$$y = W^T X = W^T S_k$$
$$= (T_1 S_1^T + T_2 S_2^T + \dots + T_L S_L^T) S_k$$
$$= T_1 S_1^T S_k + \dots + T_k S_k^T S_k + \dots + T_L S_L^T S_k$$

اگر کلیه بردارهای ورودی S_i متعامد یکدیگر باشند:

$$S_i^T S_j = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$\Rightarrow y = T_k$ برابر T_k خواهد شد.

این شرط فوق، شرط بسیار محدودکننده‌ای است. حداکثر تعداد بردارهای متعامد یکدیگر در فضای n بعدی برابر n است. از سوی دیگر تعداد الگوهای قابل ذخیره در شبکه نیز برابر n خواهد بود. بنابراین شبکه‌ای با n نرون در لایه ورودی، حداکثر n الگوی دودویی برای یادگیری در اختیار دارد.

برای یادگیری الگوهای (S_k, T_k) را تشکیل داده و به شبکه اعمال می‌کنیم و سپس از روابط فوق استفاده می‌کنیم. در شبکه ذخیره می‌شود. اما این کار برای شبکه‌های بیش از چند نرون هیچ صرفه‌ای ندارد.

اگر الگوها متعامد نباشند، آنگاه:

$$\Rightarrow y = T_k + \sum_{j \neq k} T_j S_j^T S_k = T_k + \eta$$

خطای بازده

Subject:
Year. Month. Date.

ب) استفاده از قانون LMS

$$F(W) = \frac{1}{L}\sum_{k=1}^{L} |\underline{T}_k - \underbrace{W^T \underline{S}_k}_{Y_k}|^2$$

فرض کنید

$$S = [\underline{S}_1 \cdots \underline{S}_L]_{n\times L}$$ ماتریس الگو

$$T = [\underline{T}_1 \cdots \underline{T}_L]_{m\times L}$$

می‌توانیم نشان دهیم $W = (TS^+)^T$ انتخاب شبه‌معکوس S

$F(W)$ را حداقل می‌کند

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

چگونگی عملکرد شبکه:

$$T - \underbrace{T S^+ S}_{I} = \underbrace{W^T}_{\text{بایاس}} \quad \text{در برابر Correlate}$$

شرط $S^+S = I$ وقتی برقرار خواهد بود که بردارهای ورودی به صورت خطی مستقل باشند.

شرط حاصل شدن مقدار مطلوب در خروجی به صورت خطی مستقل بودن بردارهای ورودی

(این شرط، شرط حفظ خواهد بود.)

(اگر مجموعه ورودی‌ها به صورت خطی وابسته باشند $S_j = \sum_{i=j} \alpha_i S_i$)

$$\Rightarrow W^T S_j = \sum_{i\neq j} \alpha_i W^T S_i = \sum_{i\neq j} \alpha_i \underline{T}_i$$

در این حالت برای اینکه $f(W)=0$ شود باید $T_j = \sum_{i\neq j} \alpha_i \underline{T}_i$ باشد.

اگر $W^T S_j \neq T_j$ شود، $F(W)\neq 0$ خواهد بود و بنابراین صحیح انجام نگرفته است.

$y = W^T S$

$(1\times 1) = (1\times N)(N\times 1)$

PAPCO

Subject :

Year . Month . Date . ( )

تجزیه مقدار منفرد Singular Value Decomposition

هر ماتریس $S : n \times L$ را می‌توان به صورت زیر تجزیه نمود: $n \le L$

$$S = \sum_{i=1}^{L} \sqrt{\lambda_i}\, P_i q_i^T$$

که $\{P_i\}$ و $\{q_i\}$ به ترتیب بردارهای ویژه متعامدیکه ماتریس‌های $S^T S$ و $S S^T$ می‌باشند. λ_iها مقادیر ویژه از بزرگترین به کوچکترین مرتب شده می‌باشند.

شبه معکوس ماتریس S که

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

r عبارتست از مرتبه ماتریس S

$\text{rank}(S)$ = تعداد سطرهای مستقل S

= " مقادیر ویژه غیر صفر $S^T S$

(چون بردارهای ویژه یک مجموعه متعامد تشکیل می‌دهند پس تجزیه مقدار منفرد یکتاست)

پروژه ۱ : شبیه‌سازی مسلک پرسپترون (در مسئله دسته‌بندی ۲ دسته‌ای)

" ۲ : " " آدالاین (" " ")

توسط کاربر: امکان تغییر پارامترها، انتخاب از منوها

گرفتن نمونه‌ها: خواندن از فایل، ایجاد گوسی‌تصادفی

ذخیره نمونه‌ها

نشان دادن قدم به قدم مراحل آموزش

Subject:

Year. Month. Date. ( )

مطلوب است محاسبه خطی برای روش دی نویزی

می خواهیم بسم جلوه می تلفن از شبکه متداول خطی، با بالاترین دقت، پیچ یک ورودی دارای نویز را بدست آوریم. در این حال شبکه بردار $Z_i = S_i + n$ را به بردارهای چه نزدیک به بردار T_i آن گذشت کند.

فرض: نویز دارای میانگین صفر، مولفه‌ای غیر وابسته و انحراف معیار یک در امتداد بهم محور اصلی باشد.

می تلفن نشان داده که انتخاب $VI = (TS^+)^T$ خطا عبارت خواهد بود از:

$$F(W) = \underbrace{\sum_{i=1}^{L} |Tq_i|^2}_{\text{بواسطه وجود نویز}} + \underbrace{\sum_{i=1}^{r} \frac{L\sigma^2}{\lambda_i} |Tq_i|^2}_{\text{بواسطه وابستگی خطی}}$$

برای حداقل کردن $F(W)$، W را به ازای مقادیر ویژه‌ای که کمترین مقدار متغیر S تغییر دهیم. بعبارت دیگر

$$VI = (T\tilde{S}^+)^T$$

$$\tilde{S}^+ = \sum_{i=1}^{s} \frac{1}{\sqrt{\lambda_i}} q_i p_i^T \qquad (s \text{ تعداد بزرگترین مقادیر ویژه نگه داشته می‌شوند})$$

$$\frac{L\sigma^2}{\lambda_s} \le 1 \le \frac{L\sigma^2}{\lambda_{s+1}} \qquad s \le r \text{ ، } s \text{ طوری انتخاب می‌شود که}$$

بعبارت دیگر، با حذف جملات از r دوم رابطه F به r اول متقل می‌شود با این کار $F(W)$ کاهش می‌یابد.

Subject:

Year. Month. Date. ( )

پس، حذف جمله از تجزیه مقدار منفرد S^+ در صورتی که شبکه در حال آموزش نیز باشد می‌باشد.

اگر شبکه خطی باشد $W_1 = (SS^+)^T$ ، $F(W) = L\sigma^2$ (۲)

و برای $W_1 = (S\tilde{S}^+)^T$

$$F(W) = L\sigma^2 S + \sum_{i=S+1}^{r} \lambda_i$$

بازگردیم W_1 ، قانون وید رو

نرمی از قانون وید رو طوری تغییر برای حالتی که $W_1 = (TS^+)^T$ بصورت بازگشتی به کار رود. این کار به استناد از نتیجه Grerille میرسد.

نتیجه کلی :

$$W_1^{NEW} = W_1^{OLD} + \alpha S_k (T_k - (W^{OLD})^T S_k)^T$$

در هر واحد:

$$W_i^{NEW} = W_i^{OLD} + \alpha (t_{ki} - (W_i^{OLD})^T S_k) S_k$$

این قانون برای آموزش سیستم با مجموعه آموزش را که در آن بردارهای ورودی متعدد و به صورت اتفاقی انتخاب می‌شوند به کار گرفت.

در یک فوق العاده $\theta \neq 0$ ، سلول اسمی می‌باشد.

رفتار دینامیکی شبکه‌ی تک لایه پیش خور

$$\frac{dy_i}{dt} = -g(y_i) + I_i$$

$$I_i = \sum_{j=0}^{n_i} w_{ij} x_j \qquad i = 1, 2, \ldots, m$$

Subject:

Year. Month. Date. ( )

در حالت تعادل $\frac{dy_i}{dt} = 0$ برای $i = 1, 2, \ldots, n$

$$g(y_i) = I_i \qquad y_i = g^{-1}(I_i) = f(I_i)$$

می‌توان واحد را بصورت درجه 2 زیر در نظر گرفت:

$$I_i = \sum_{k=1}^{n_i} \sum_{l=1}^{n_i} w_{ikl} x_k x_l$$

مسئله در این حال پیچیده می‌باشد.

شبکه‌های عصبی بازگشتی تداعی

Recurrent Associative Networks

رفتار: نگهداری

نرون‌ها نقطه اولیه شروع به کار می‌کنند و به یکی از حالات پایدار شبکه همگرا می‌شوند.

در طراحی این شبکه (۲ حرف مهم) قرار است:

۱- شبکه از هر نقطه اولیه شروع به کار کند به یکی از حالات پایدار خود برسد.

۲- حالت پایداری که شبکه به آن می‌رسد باید هر طوری نزدیک‌ترین حالت پایدار به حالت اولیه باشد.

۳- بیشترین تعداد حالت پایدار و حوزه جذب شده برای شبکه ایجاد نماید.

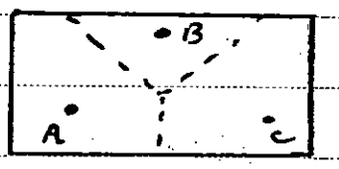
از ۲ حرف فوق، تنها به اولی توسط شبکه‌های بازگشتی متداول پرداخته می‌شود.

Subject :

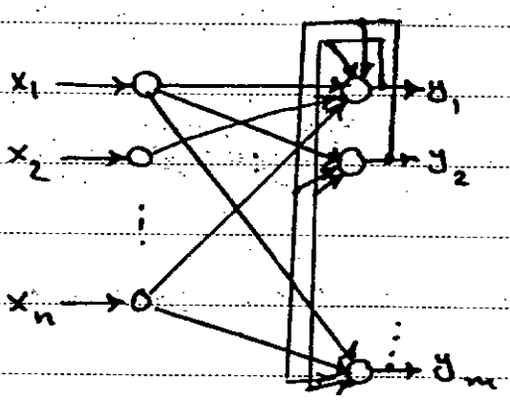
Year . Month . Date . ( )

عملکرد شبکه بدین صورت است که بردار ورودی را به یک سیستم حافظه محدود تشبیه می‌کند که در آن هر نقطه (حالت پایدار) در یک فضای n بعدی قرار داده شده است. نیروی جاذبه هر نقطه تنها بر نقاط نزدیک اثر دارد که به آن نزدیک‌تر هستند.

کار شبکه به این ترتیب شروع می‌شود که یک بردار آزمایشی در این فضا قرار داده می‌شود. جرم حرکت در آن نقطه به سمت نزدیک‌ترین نقطه جذب حرکت می‌کند.



معماری شبکه برای بازگشتی به شکل زیر است:



$f(x) = x$

تابع تبدیل در لایه خروجی یک تابع خطی است.

یک بردار ورودی به شبکه داده می‌شود.

بعضی محققین اتصالات خود بازگشتی را حذف کرده‌اند.

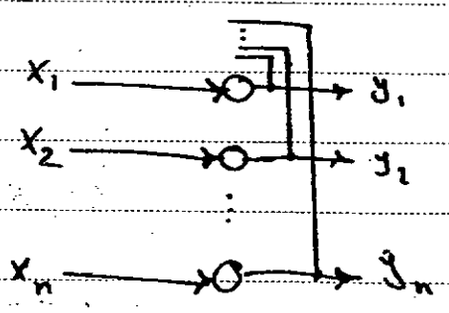
(یعنی خروجی y_i دوباره به y_i برنگردد)

* پاسخ این شبکه به ورودی (به خصوص ورودی $\frac{1}{2}$) که در آموزش به کار گرفته شده است بسیار بزرگ شده از این رو به پاسخ شبکه برخوردی کمتر دقیق می‌شود.

شبکه Brain-State-in-a-Box (BSB)

در سال 1977 توسط جیمز اندرسن معرفی شد.

هر ورودی متناظر با یک نورون



تابع فعالیت خطی (تنها ستاده درخل یک از جمله محدود را احتیاط می‌کند)

Subject:

Year Month Date ( )

ابر مکعب: $[-1,1]^n = \{ \underline{y} = (y_1, y_2 \cdots y_n)^T \in \mathbb{R}^n \mid -1 \le y_i \le 1 \quad \forall \ 1 \le i \le n \}$

اسکام در آوردن نقطه : بیمزن

حالت 2 شبکه می‌تواند از هر نقطه‌ای داخل $[-1,1]^n$ شروع شود و بایستی در حلقه شبکه یکی از رئوس مکعب تولید شود در آن راه‌اندازی شود.
مقادیر اولیه وزن‌ها بسیار کوچک در نظر گرفته می‌شوند.

الگوریتم آموزش شبکه BSB

برای وزن‌ها مقادیر اولیه بسیار کوچک انتخاب می‌کنیم. برای نرخ‌های یادگیری α و β مقادیری انتخاب می‌کنیم.

1- برای هر بردار آموزش S قدم‌های 2 الی 6 را طی می‌کنیم.

2- مقدار اولیه خروجی‌ها برابر بردار ورودی S در نظر می‌گیریم. $y_i = s_i$

3- تا زمانی که خروجی‌ها همگرا شده‌اند قدم‌های 4 و 5 را ادامه می‌دهیم

4- محاسبه خروجی‌های شبکه

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

5- محاسبه خروجی‌ها

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

6- وزن‌ها را اصلاح می‌کنیم.

$$w_{ij}^{NEW} = w_{ij}^{OLD} + \beta y_i y_j$$

P4PCO

Subject:

Year. Month. Date. ( )

منبع دیگری از این الگوریتم برجسته است که در آن فرض شده بردارهای ورودی دوقطبی هستند و تنها متعامد هستند. اتصال خروجی خود حذف شده و تابع فعالیت تابع علامت در نظر گرفته شده

* توجه نشان داد که در این شبکه هر حالت پایدار $\underline{X}^{stable}$ دارای یک زوج $-\underline{X}^{stable}$ است و اگر $\theta \neq 0$ در نظر گرفته شود $-\underline{X}^{stable}$ حتماً حذف می‌شود.

* تغییر وزن‌ها در شبکه BSB به صورت طبق قانون هبب نیز انجام شود یعنی:

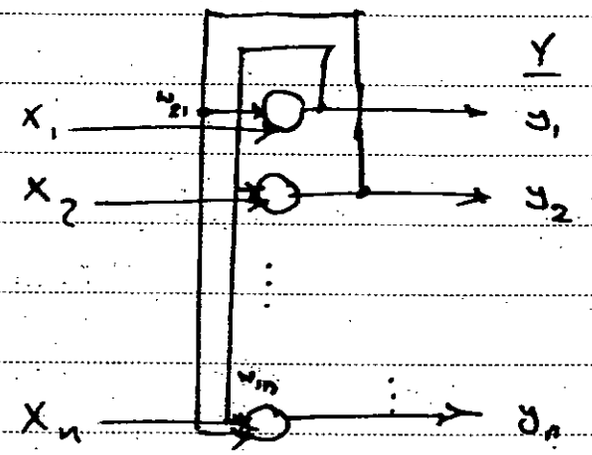
$$W^{NEW} = W^{OLD} + \eta S^{input} (y^{desired} - y^{actual}) T$$

یکی از مشکلات روش‌های بازگشتی این است که علاوه بر جواب‌های خروجی که ما می‌خواهیم، جواب‌های دیگر هم ممکن است تولید شود (مثل همین $-X^{stable}$ در بالا) که در این پس از آموزش وزن‌ها ما نمی‌خواهیم بکنیم است شبکه به آن حالت هنگام شود.

۱۸,۳۴

شبکه انفیلد

اولین نسخه در سال 1982 توسط هاپفیلد (فیزیکدان) معرفی شد.



اتصالات خود بازگشتی وجود ندارد ($W_{ii}=0$)

وزن‌های اتصالات بازگشتی متقارن هستند

$$W_{ji} = W_{ij}$$

Subject :

Year . Month . Date . ( )

پس از ارائه نسخه اولیه شبکه، نسخه‌های متعدد دیگری نیز معرفی شدند.
در شبکه گسسته هاپفیلد، حالات شبکه در حالات اولیه Φ مقداری میگیرند.
(مثلاً، برای Φ_i ها Φ_i و برای Φ_i ها -1 و 1)

در شبکه هاپفیلد پیوسته، تعداد نامحدودی از توابع فعالیت واحدها می‌توانند همه مقادیر پیوسته بین دو مقدار مانند -1 و 1 را اختیار کنند.
بهنگام در آوردن واحد (سلول) است بطور همزمان (Synchronous) یا ناهمزمان (Asynchronous) باشد.
در حالت ناهمزمان واحد (به صورت اتفاقی) انتخاب می‌شود.
در این شبکه بجز اتصالات بازگشتی تنها یک وزن از خروج هر واحد، واحدهای شبکه را متصل است تا همگرایی شبکه بصورت ثابت نماند. بعد از اولین بهنگام سازی از بین رفته، مقدار دست‌نخورده واحد (سلول) است Φ یا هنوز Φ باشد.
از انتخاب‌های فوق معمولاً بهنگام سازی ناهمزمان، مقادیر حالات Φ تعادلی و مقدار آستانه‌ای Φ برای واحد (سلول) مطمئن‌تر و دست‌یافتنی‌تر است.

تابع تبدیل واحد (i) به صورت زیر تعریف می‌شود:

$$Y_i^{NEW} = \begin{cases} 1 & \sum_{j=0}^{N} W_{ji} Y_j^{OLD} + X_i > \Theta_i \\ Y_i^{OLD} & = \Theta_i \\ -1 & < \Theta_i \end{cases}$$

Θ برای حالات دوتایی

در واحد i، از صفحه‌ای از وزن‌های حالات W_{ji} خواهد بود که Θ_i و W_{ji} مشخص خواهد شد.

اگر Y_i^{OLD} در سمت مثبت از صفحه J واقع شده $Y_i^{NEW} = 1$ و اگر در سمت منفی آن باشد $Y_i^{NEW} = -1$ ولی اگر دقیقاً روی از صفحه باشد $Y_i^{NEW} = Y_i^{OLD}$ خواهد بود از یکی از حالات قبلی واحد صفحه خواهد گرفت.

Subject:

Year. Month. Date. ( )

برای جلوگیری از این حالت می‌توان تابع y_i را به صورت جزئی تغییر داد.

در این حالت تابع تبدیل به صورت زیر خواهد بود:

$$y_i^{NEW} = sgn\left(\sum w_{ji} y_j^{OLD} - \theta_i + x_i\right)$$

برای حالت پایدار،

$$y_i^{NEW} = \underset{\text{Step}}{U}\left(\quad \sim \quad\right)$$

Energy Landscape نمای انرژی

هاپفیلد در سال 1984 با در نظر گرفتن یک تابع انرژی، تابع لیاپانوف به صورت زیر را ثابت کرد که شبکه با شروع از یک نقطه اولیه به حالت پایداری نگاشت می‌شود.

تابع لیاپانوف تابعی است غیر افزایشی از حالات سیستم که از پایین محدود است.

تابع انرژی داریم:

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

وقتی شبکه تغییر حالت می‌دهد (نقطه‌ای مطابق شبکه Kنام) تغییر انرژی به صورت خواهد بود از:

$$\Delta E = E_2 - E_1 = -\frac{1}{2}\sum_{i\neq j}\sum_{j} w_{ij} y_i^{NEW} y_j^{NEW} - \sum_i x_i y_i^{NEW} + \sum_i \theta_i y_i^{NEW}$$

$$+\frac{1}{2}\sum\sum w_{ij} y_i^{OLD} y_j^{OLD} + \sum_i x_i y_i^{OLD} - \sum_i \theta_i y_i^{OLD}$$

برای همه iها بجز $i=k$ داریم: $y_i^{NEW} = y_i^{OLD}$

$$\Delta E = -\sum_{i\neq k} w_{kj}\left(y_k^{NEW} y_j^{NEW} - y_k^{OLD} y_j^{OLD}\right) - \sum_i x_i\left(y_i^{NEW} - y_i^{OLD}\right)$$

$$+\sum \theta_i \left(y_i^{NEW} - y_i^{OLD}\right)$$

$i=k$: $= 0$; $i\neq k$: $= \Delta y_k$

Subject:

Year. Month. Date. ( )

$$\Delta E = -\left[\sum_{j\neq k} w_{kj}\, y_j^{OLD} + x_k - \theta_k\right] \Delta y_k$$

دو حالت

۱) اگر $y_k^{OLD} = 1$ $\Leftarrow$ $y_k^{NEW} = -1$ و با توجه به تابع تشدید داریم:

$$\sum w_{jk} y_j^{OLD} + x_k < \theta_k$$

ولذا داخل [ ] در رابطه ΔE منفی خواهد بود

$$\Delta y_k = -2 \quad \therefore \quad \Delta E < 0$$

۲) اگر $y_k^{OLD} = -1$ $\Leftarrow$ $y_k^{NEW} = 1$ باز هم $\Delta E < 0$

(چون واحد k در حالتی بوده و تغییر کرده پس نقطه ضعیف E حالت پیش می‌آید)

از مطلب فوق نتیجه می‌شود که با تغییر حالت شبکه انرژی (E) کاهش می‌یابد تا جایی که شبکه دیگر تغییر حالت ندهد $\Leftarrow$ $\Delta E = 0$ خواهد بود ولذا همیشه می‌توان گفت که

$$\Delta E \leq 0$$

با این توضیح که شبکه تغییرات همیشه E را کاهش می‌دهد. سؤال: چگونه اثبات می‌شود که حدی دارد؟ یعنی این کاهش بالاخره جایی متوقف می‌شود؟

Subject :

Year . Month . Date . ( )

بر حالت پایدار شبکه تنها ظرف مینیمم کلی در سطح انرژی است. شروع کار شبکه از هر نقطه اولیه ای به یک مینیمم محلی در یک تعداد تکرار محدود به یک حالت پایدار می رسد اما حالت مینیمم الزاماً نزدیکترین حالت به نقطه اولیه نیست.

به شرط $V_{ij} = V_{ji}$ و $V_{ii} = 0$ باید به طور دقیق رعایت شود تا شبکه پایدار باشد.

(خواندن چرخش از این شرط می تواند منجر به ناپایداری این شبکه گردد.

یادگیری وزنها

وزنهای شبکه (ماتریس) به طوری تعیین شوند که الگوهای مورد نظر ما حالات پایدار شبکه گردند. این تابع انرژی درقتی حداقل می شود که $\sum\sum_{i \neq j} V_{ij} y_i y_j$ حداکثر شود.

برای ذخیره الگوی k قطبی مانند $S_k = [S_k(1) \cdots S_k(n)]^T$ ماتریس V_{ij} را بصورت زیر انتخاب می کنیم:

$$V_{ij} = \sum_k S_k(i) S_k(j)$$

با این انتخاب انرژی حداقل می شود.

چرخش در حالت پایدار $y_i = S_k(i)$ و $y_j = S_k(j)$ و S_k ها هم $+1$ و -1 هستند $\sum\sum$ برابر $n^2 - n$ خواهد شد.

اگر به شبکه ای که الگوی S_k را ذخیره نموده الگوی جدیدی اعمال کنیم که کمتر از نیمی از مولفه های آن با مولفه های متناظر S_k متفاوت باشد، شبکه به حالت پایدار S_k خواهد رسید.

اما اگر بیش از نصف مولفه ها متفاوت باشند، الگوی جدید به حالت $-S_k$ پایدار خواهد شد.

حالت $-S_k$ را حالت معکوس (Reversal State) می گویند.

Subject :

Year Month Date ( )

برای ذخیره چند الگوی دوقطبی (L الگو)

$$W_{ij} = \sum_{l=1}^{L} S_l(i) S_l(j) \qquad W_{ij} = 0$$

$$i \neq j$$

گاهی اوقات ضرب $\frac{1}{n}$ را نیز در ستال تحریم قلمداد می‌دهند. این کار در شرایط بهداری را اصلاح می‌کند.
قانون یادگیری مبتنی قانون هب تعمیم یافته نامیده شده است.

اگر الگوها بایری لزوم قانون هب تعمیم یافته،

$$W_{ij} = \sum_{l=1}^{L} (2S_l(i) - 1)(2S_l(j) - 1)$$

$$i \neq j$$

حالات بازیابی : S_k Retrieval States

حالات معکوس : $-S_k$ Reversed States

حالات مخلوط : S_i^{Mix} Mixture States

ترکیب خطی یا تعداد فرد از حالات بازیابی می‌باشد.

$$S_j^{Mix}(i) = Sgn(\pm S_k(i) \pm S_l(i) \pm S_m(i)) \qquad \text{(در بیش ۳ تا)}$$

حالت Spin Glass

یک سری حالات پیچیده ارتباطی دارند.

حالات معکوس و مخلوط را حالات اضافی نیز می‌نامند. (Spurious States)
میدان جاذبه کمی دارند.

اصلاح حالات معکوس : بهتر است تعداد حالات ذخیره شده ضرب پیچ در -1 در صورت ضرورت

Subject:

Year. Month. Date. ( )

روش $\frac{1}{2}$ برای کاهش، حرف حالت بردی پیش‌بینی شده است.

الگوریتم شبکه گسسته (هیلد

Θ: وزن‌های شبکه را از قالب هب تعیم یافته‌ی سه می‌گیریم.

1: برای هر بردار ورودی X قدم‌های 2 تا 8 را انجام می‌دهیم:

2: مقادیر اولیه حالات را برابر ورودی جدید X قرار می‌دهیم.

$$y_i^{OLD} = x_i \qquad i = 1 \dots n$$

3: تا زمانی که حالات شبکه ثابت نشده، قدم‌های 4 تا 8 را انجام می‌دهیم:

4: قدم‌های 5 تا 7 را برای هر واحد انجام می‌دهیم:

(واحدها به صورت تصادفی انتخاب می‌شوند)

5: خروجی خالص واحدها را محاسبه می‌کنیم

$$I_i = x_i + \sum_j w_{ji}\, y_j^{OLD}$$

$$\underset{n\times 1}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}} = \underset{n\times n}{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}} \times \underset{n\times 1}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}$$

6: خروجی واحد را محاسبه می‌کنیم

$$y_i^{NEW} = \begin{cases} 1 & I_i > \theta_i \\ y_i^{OLD} & I_i = \theta_i \\ -1 & I_i < \theta_i \end{cases}$$

7: مقدار y_i^{NEW} را به ورودی همه واحدها ارسال می‌کنیم.

8: آزمایش کنید.

Subject:

Year. Month. Date. ( )

ظرفیت شبکه (پیچیده نگفته):

با روش‌های مختلف سعی در تخمین ظرفیت این شبکه شده است.

ظرفیت شبکه (پیچیده) برای یک خطای قابل قبول بستگی دارد به (و تعداد الگوها که می‌خواهیم ذخیره نماییم (L) وابسته است. افزایش L احتمال خطای بازیابی را افزایش می‌دهد. اگر بخواهیم برای سطحی قابل قبول شعاع انتخاب کنیم می‌توانیم حداکثر تعداد الگوهایی که می‌توانند ذخیره شوند را بدست آوریم که خطای غیرقابل قبول پیش نیاید. دسته‌ای روش تحلیل فرمول‌های مختلفی مرتبط شده است از جمله:

برای خطای کمتر از ۰.۰۱ یا ۱٪ $L \leq 0.15n$

" " ۵٪ " " $L \leq 0.37n$

" " ۱۰٪ " " $L \leq 0.61n$

در تحلیل دیگری برای اینکه همه n بیت را با احتمال 99 از بیت صحیح بازیابی کنیم بستگی داشته باشیم:

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

برای اینکه همه الگوها بطور کامل بازیابی شوند باید

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

$n = 100$

$L \leq 0.15n \rightarrow 15$

$\frac{n}{2\log_e n} \rightarrow 11$

$\frac{n}{4\log_e n} \rightarrow 5$

پرده ۳: شبکه همدیگر را می‌شناسند

پرده ۴: شبکه‌ها فیلم کمک از الگوی تصویری استفاده کنید که چگونه آن را بدست آورید

P4PCO

Subject:

Year. Month. Date. ( )

قضیه کوهن گراسبرگ

اگر در یک شبکه عصبی متقارن، n واحد پردازشگر خروجی‌های y_i و با تابع تبدیل

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

در نظر گیریم داریم:

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

۲- تابع $a_i(u)$ برای $u \geq 0$ پیوسته باشد و برای $u > 0$ مثبت

۳- تابع $b_i(u)$ پیوسته باشد برای $u \geq 0$ هرگز نابود نشود

۴. تابع $f_j(u)$ مشتق‌پذیر باشد برای $u \geq 0$ و $f_j'(u) > 0$

۵. اگر بین ترمهای قطری یک رابطه داشته باشیم $[b_i(u) - w_{ii} f_i(u)] < 0$

۶. یا داشته باشیم $\lim_{u \to 0^+} b_i(u) \to \infty$ و یا داشته باشیم

$\left(\lim_{u \to 0^+} b_i(u) \not\to \infty\right)$ ، $\left(\int_0^u \frac{ds}{a_i(s)} \to \infty \quad u \text{ برای یک}\right)$

آنگاه اگر حالت شبکه در زمان Φ در بخش مثبت فضای n بعدی حقیقی باشد، شبکه همیشه به یک نقطه تعادل در بخش مثبت فضای حقیقی همگرا می‌شود مگر برای تعدادی از شرایط اولیه که تعادل نباشد. ۷۱

P4PCO

Subject:

Year. Month. Date. ( )

مدل کوهن گروسبرگ قدرتمند عمومی برای شبکه‌ها است. این تقریب نشان داده که تقریب زیر نیز دارای شرایط صعود تقویت هستند:

$$\dot{y}_i = -A_i y_i + (B_i - C_i y_i)[I_i + f_i(y_i)] - (D_i y_i + E_i)\left[J_i + \sum w_{ji} g_j(y_j)\right]$$

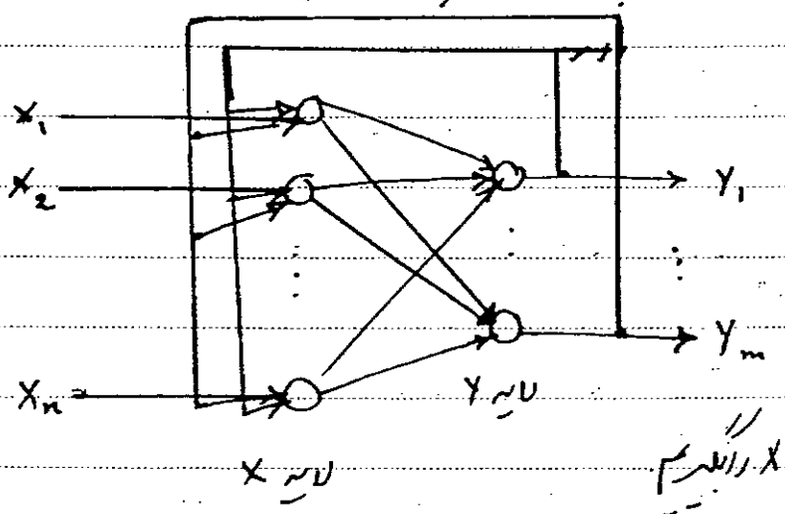
$$\dot{v}_i = G_i y_i \left(1 - \sum H_{ij} y_j\right)$$

شبکه حافظه دوسویه (Bidirectional Associative Memory) (BAM)

حافظه دوسویه

در سالهای ۱۹۸۸ الی 1992 توسط کوزکو معرفی شد. نسخه‌های مختلف دارد.

(در عمل ساده‌ترین آنها دو لایه‌ای هستند)



شبکه

در شبکه پیشنهادی X می‌دهیم انتظار داریم Y را بگیریم.

و در این Y می‌دهیم انتظار X داریم.

این مدل تعمیم مدل‌های پیشین به حالت غیرخودتداعی است.

Subject:

Year. Month. Date. ( )

شبکه حافظه ستقل است از هر طرف ورودی بدهیم ، از آن طرف خروجی می‌گیریم. $\begin{pmatrix} X \to Y \\ Y \to X \end{pmatrix}$

اسم علت : این شبکه از طرفین نیز می‌گیرد

ورودی دارد X و به Y (برای ماتریس VI باشد، ورودی انتظار بازگشت VI^T خواهد داشت.

نسخه‌ی بایپولر ، دقیق و پیوسته

متعلق از طرفه گسسته Discrete BAM

بایپولر } عملکرد دقیقا مثل دقیق می‌باشد
دوقطبی }

آموزش : براساس آموزش : (S_i, T_i) $i = 1..L$

$$VI_{ij} = \sum_{k=1}^{L} S_k(i)\, T_k(j)$$ برای دقیق

$$VI_{ij} = \sum_{k=1}^{L} (2S_k(i)-1)(2T_k(j)-1)$$ برای حالت بایپولر

توابع فعالیت واحد :

$$X_i^{NEW} = \begin{cases} 1 & I_i > \theta_i \\ X_i^{OLD} & I_i = \theta_i \\ -1 & I_i < \theta_i \end{cases} \qquad Y_j^{NEW} = \begin{cases} 1 & > \\ Y_j^{OLD} & = \\ -1 & < \end{cases}$$

در حالت بایپولر = 0 خواهد شد

P4PCO

Subject:

Year. Month. Date. ( )

واحدی لایه X را به هم به هنگام در آنها می‌شنود و واحدی لایه Y را به ...

الگوریتم پس‌انتشار خطا در 3 طبقه شبکه

0 : وزن‌های اولیه را با اعداد تصادفی کوچک انتخاب می‌کنیم.

فرض می‌کنیم واحد i لایه 0 قرار می‌گیرد.

1: برای هر زوج آموزشی ورودی و خروجی 2 الی 9 را انجام می‌دهیم.

2- الف) حالت لایه X را برای ورودی حاضر قرار می‌دهیم.

2- ب) حالت لایه Y را برای مقادیر ورودی حاضر قرار می‌دهیم.

3- تا زمانی که واحد پنهان شده از لایه‌ی 4 الی 6 را انجام می‌دهیم.

4- حالت لایه Y را به صورت هم‌زمان بدست می‌آوریم. $I_j = \sum_i v_{ij} x_j \qquad y_j = f(I_j)$

5- حالت لایه X را به صورت هم‌زمان بدست می‌آوریم. $I_i = \sum_j w_{ij} y_j \qquad x_i = f(I_i)$

6- آزمون پایانی: اگر همه X و Y در حالت پایدار قرار گرفتند الگوریتم خاتمه یافته در غیر این صورت ادامه می‌دهیم.

* شبکه‌ی لایه‌ی 1 خروجی ورودی X خروجی O

ورودی را همانند هم با 2-الف وارد هم با 2-ب

Subject:

Year. Month. Date. ( )

Continuous BAM — شبکه متداوم BAM دوطرفه پیوسته

در این حالت پیوسته در زمان - خروجی (نورون در فاصله $[0,1]$) تغییر می‌کند. تمام واحدها در این شبکه از تابع سیگموئید لجستیک استفاده می‌کنند و دارای $\theta \neq 0$ هستند.

$$f(I_i) = \frac{1}{1+e^{-I_i}}$$ سیگموئید لجستیک

نمودی دیگری نیز موجود است.

$$\dot{x}_i = -a_i x_i + \sum_{j=1}^{m} v_{ji} f(y_j) + s_i \qquad \begin{matrix} i = 1 \dots n \\ a_i, s_i > 0 \end{matrix}$$ معادلات حاکم بر حرکت در این شبکه

$$\dot{y}_j = -c_j y_j + \sum_{i=1}^{n} v_{ji} f(x_i) + t_j \qquad \begin{matrix} j = 1 \dots m \\ c_j, t_j > 0 \end{matrix}$$

قضیه کاربرد: شبکه متداوم فوق از هر حالت اولیه که شروع به کار کند، حالت پایداری همگرا می‌شود.

اثبات: با استفاده از قضیه کوهن - گراسبرگ

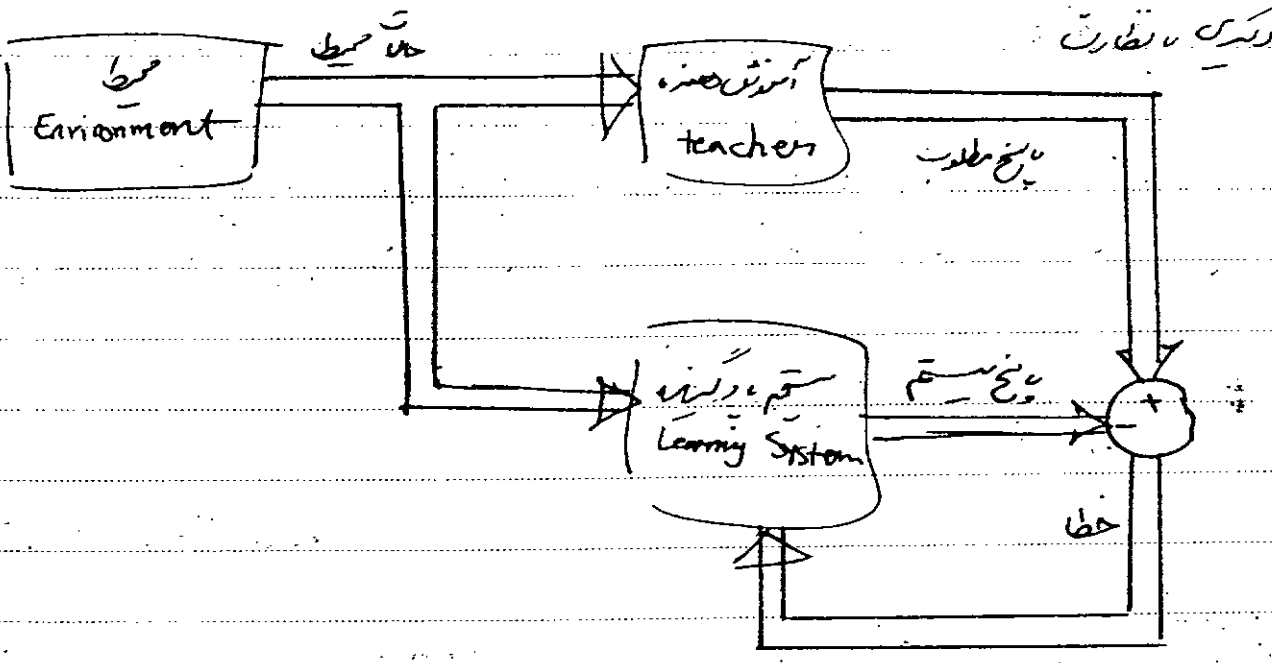
۶ - مدل‌های یادگیری

1. مدل یادگیری با نظارت (Supervised Learning)
2. " بی نظارت (Unsupervised Learning)
3. " تقویتی یا پاداش - امتیازی Reinforcement Learning
   Graded Learning

Subject:

Year. Month. Date. ( )

یادگیری با نظارت

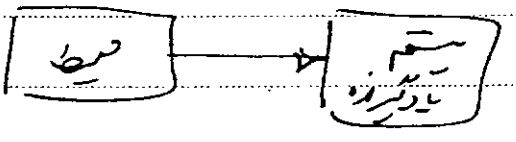


نکته اصلی حالت آموزش دهنده این است که برای هر حالت پاسخ مطلوب را مشخص می کند.

این آموزش می تواند با ارائه مجموعه ای از نمونه های ورودی پاسخ مطلوب به سیستم انجام شود.

این نوع یادگیری را یادگیری با معلم هم می گویند. (Learning with a teacher)

یادگیری بی نظارت



در این گونه مواقع سعی می شود که برای هر حالت خروجی چه باید باشد.

و به خروجی صحیح بستگی به نقاط

شبکه خودسازمان یادگیری (الگوریتم) با استفاده از یک سری نمونه های ورودی ویژگی های آنرا کشف کرده و خروجی ارائه دهد.

نمونه ای از این گونه شبکه ها را می توانید در جلسه ی ارائه ببینید.

Subject:
Year. Month. Date. ( )

۱- آشنایی : یک خروجی بوسته می تواند نشان دهد که الگوی ورودی چقدر آشنا یا شبیه به یک الگوی نمونه
یا ترکیبی از الگوهای که تاکنون مشاهده کرده است.

۲- تحلیل مؤلفه اصلی : Principal Component Analysis PCA
گسترش سری اول - چند ملاصد - مخروبه ای از یک مجموعه مخروبه به چند مؤلفه ای می شود که در
اغلب آن مخروبه - شباهت به مؤلفه ای دیده شده سنجیده می شود.

۳- خوشه بندی Clustering
اگر یک مجموعه از خروجی را داشته باشیم که در مورد آن تنها یکی از آنها فعال باشد خروجی می تواند
مشخص کند که یک الگوی ورودی متعلق به کدام خوشه آمده است.

۴- نمونه دهی Prototyping
شبکه می تواند مانند حالت قبل دسته ای تشکیل دهد اما در خروجی به جای ارائه شماره خوشه الگوی
نمونه ای از خوشه را ارائه دهد. (شبیه شبکه های لبه ای)

۵- کدگذاری Encoding
خروجی عملکرد است نسخه فشرده شده ای از ورودی باشد که حتی الامکان اطلاعات موجود در ورودی را
حفظ نموده آن را برای تعداد بیت کمتری ارائه دهد. عمل عکس (کدبرداری) نیز میسر می باشد.

۶- نقشه سازی از ویژگی ها (Feature Mapping)
اگر واحدهای خروجی در یک بردار آرایش خاصی داشته باشند مثلاً در مخروبه ای تنها یک واحد فعال
باشد شبکه الگوی ورودی را به یک نقطه خاص از خروجی نگاشت می کند. در این حالت مانند آن
است که روی نقشه ای محل یابی کنند.
الگوهای مشابه به محل های نزدیک هم روی نقشه فعال خواهند شد.

Subject:

Year. Month. Date. ( )

کار کمودکی ترکیبی نیز میتوانیم از دو مدل فوق تشکیل شده باشد

PCA + Encoding

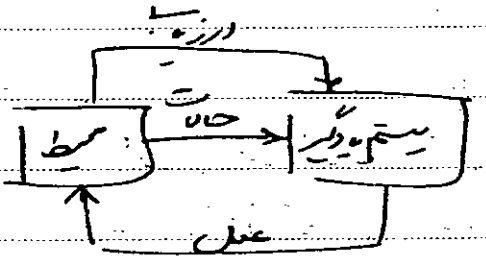
ANN + Clustering

شبکه‌ی عصبی مصنوعی با این مدل بسیار ساده هستند (مجموعه‌ی ANN)

یادگیری با ارزیابی

یادگیری با نظارت، برای نمونه در شبکه‌ی انجام می‌شود. انتخاب جمع نمونه مهم است. برای هر ورودی باید خروجی را داشته باشیم. یادگیری حداقل در بخشی بطور مناسب انجام نخواهد شد.

یادگیری با ارزیابی در دو مدل فوق قابل استفاده است.



این نوع یادگیری، یادگیری با یک ناقد

Learning with a critic

نامیده شده است.

ضرورت نظارت در یادگیری بر ورودی، بدین معنی است که خروجی صحیح مشخص باشد. به جای آن شبکه پس از عمل حرکت مرحله یادگیری، ارزیابی شده به عملکرد آن امتیاز یا نمره‌ای داده می‌شود. این نمره توسط یک تابع (cost function) تولید می‌شود.

شبکه‌ی عصبی مبتنی بر الگوریتم ژنتیک، یادگیری با ارزیابی عموماً ضعیف‌تر از شبکه‌ی با نظارت هستند؛ اما کاربرد وسیعی در روش‌های بدون نظارت نیز دارند بسیار کاربردی است.

Subject:
Year. Month. Date. ( )

۷ - شبکه‌های خودسازمان‌ده و یادگیری رقابتی

شبکه‌های خودسازمان‌ده از یادگیری رقابتی استفاده می‌کنند و هدف آن‌ها نگاشت ویژگی‌های خاصی از متغیرهای توسعه یافته از سیگنال ورودی آن‌ها در نقاط مختلف خروجی سازمان داده شده از لحاظ بخش‌های حسی مختلف با نقشه‌های توپولوژیک مرتب معنی‌دار ارائه می‌شوند.

Topologically Ordered Computational Maps

مثلاً ورودی‌های حسی لامسه، حس بینایی و حس شنوایی با یک ترتیب توپولوژی معنی‌دار به نقاط مختلف مغز مرتبط می‌شوند.

SOM : Self Organizing (feature) Map خودسازمان‌ده (ویژگی) نقشه

SOFM واحدهای پردازشگر در یک شبکه (lattice) یک‌بعدی، دوبعدی یا بیشتر قرار داده می‌شوند.

واحدها در یک فرآیند یادگیری رقابتی نسبت به دسته‌های ورودی تنظیم می‌شوند. محل واحدهای تنظیم شده در شبکه بگونه‌ای منظم می‌باشد که برای توزیع‌های ورودی یک دستگاه، مختصات معنی‌دار در دامنه شبکه ایجاد می‌شود. در یک نقشه خودسازمان‌ده یک نقشه توپولوژیک از الگوهای ورودی با شبکه می‌دهد که در آن محل قطعه‌ای از شبکه واحد، شاخص ویژگی‌های الگوهای ورودی است.

یادگیری رقابتی که در این شبکه‌ها بکار می‌رود به این صورت است که در هر قدم واحد برنده یا فعال شده یا یادگیری رقابت می‌پردازند. در این یک مرحله رقابت تنها یک واحد برنده می‌شود. (این واحد را واحد برنده همه‌چیز (winner-take-all) نامیده‌اند) در اینجا تغییر وزن‌های واحد برنده نسبت به تغییر وزن سایر واحدها بطور متفاوت انجام می‌شود. یادگیری از این نوع را یادگیری مشابه می‌نامند که یادگیری رقابتی از نوع آن نظارت است.

Subject:

Year. Month. Date. ( )

شبکه های رقابتی بطوری که ذکر شد :

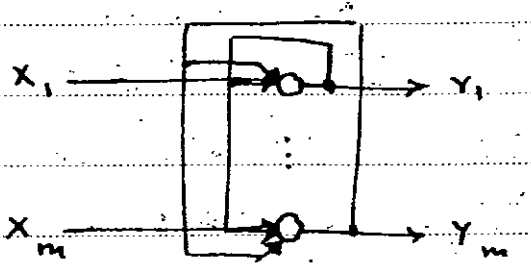
برای افزایش تاثیر بین نورون فعال و نورون های اطراف آن از روش رقابت در بین آنها استفاده می شود.

چند نمونه :

1- شبکه ماکسنت Maxnet

در سال 1987 توسط لیپمن (Lippman) طرح گردیده است.

می توان در شبکه های زیر شبکه ای به کار رفته مشخص می کند کدام یک از آن دارای بزرگترین ورودی است.



مشخصات :

اتصالات فعل و متقابل،

وزنهای ثابت

تابع فعالیت واحد به این صورت هست.

$$f(I_i) = \begin{cases} I_i & I_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

الگوریتم شبکه :

0- تعیین مقادیر اولیه

ε در نامساوی $0 < \varepsilon < \frac{1}{m}$

$Y_i^{OLD}(0) = X_i$ مقادیر اولیه نورونها (ورودی) برابر ورودیها تعریف می شود.

تعیین وزنها

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

Subject:

Year. Month. Date ( )

۱- تا زمانی که شرط خاتمه برقرار نشده، دست مرحله‌ی 2 تا 4 را تکرار می‌کنیم.

2- مقادیر فعالیت جدید را محاسبه کرده و ذخیره می‌کنیم.

$$y_j^{NEW} = f\left[y_j^{OLD} - \varepsilon \sum_{k \neq j} y_k^{OLD} \right] \qquad j = 1, 2, \ldots, m$$

3- مقادیر فعالیت جدید را برای تکرار بعدی ذخیره می‌کنیم.

$$y_j^{OLD} = y_j^{NEW}$$

4- اگر بیش از یک واحد مقدار فعالیت غیر صفر دارد، تا زمانی که فعالیت خاتمه یابد

* این مقادیر در تعیین W_{ij} هاست. هر واحد میگوید وزن خودش (بزرگتی به خودش) 1 باشد بقیه خودش را تقویت میکند طبق مرحله ۴- میگذارد بقیه تضعیف میکند

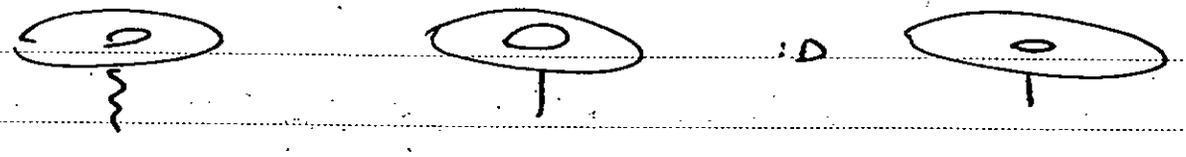
مثال:) $m=4 \qquad X = [0.2 \quad 0.4 \quad 0.6 \quad 0.8]^T$

$\varepsilon = 0.2$

، 5 بار تکرار الگوریتم، Y برابر خواهد شد:

$Y = (0.0 \quad 0.0 \quad 0.0 \quad 0.421)$

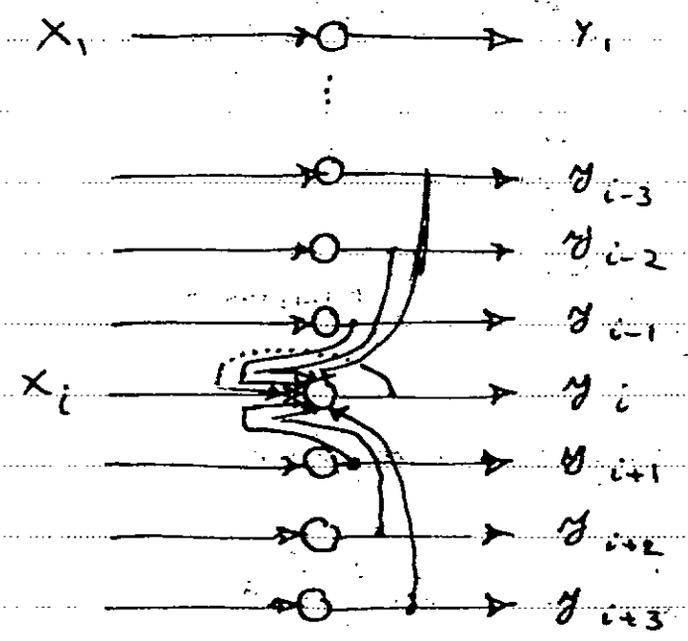
۳- شبکه‌ی مکزیکی Mexican Hat Network



بدان سان ۱۹۸۹ ترسط کوهنن پیشنهاد شد. برای افزایش تباین بین دورها

Subject:

Year. Month. Date. ( )



ساختمان:

در این شبکه هر واحد با دو ورودی
ثابت و خروجی یک سه ای تک کانال
و با وزن های صفر، یک و منفی یک به
صورت تصادفی به سوی واحدهای
دیگر وصل می شود.

شامل y_{i-2} تا y_{i+2}
واحدهای همکار برای واحد i ام هستند.
(در شعاع R_1)

شعاع دیگری مطرح می شود به نام R_2.
شعاع اتصال یعنی فراتر از این دایره
واحد مربوطه متصل نیست (no feedback)
مثلاً اگر شعاع ما 3 باشد ورودی های y_{i-3} و y_{i+3} را نخواهیم داشت.

اتصالات در شکل بالا فقط برای واحد i ام کشیده شده است. سایر واحدها نیز اتصالاتی این
چنین با وزن های مربوطه به واحدهای همسایه (در شعاع) نسبت به i ام برقرار می شود.

نحوه اتصالات واحد به واحدها به صورت $W_{i,i+k}$ که نشان دهنده وزن یال
$R_1 < |k| \le R_2$ وزن صفر است و $|k| \le R_1$ منفی یک است.

فعالیت هر واحد در این شبکه از رابطه زیر و با تابع + محاسبه می شود:

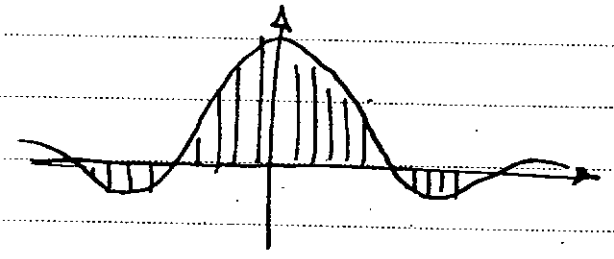
$$y_i(t) = f\left[x_i(t) + \sum_{k=-R_2}^{R_2} W_{i,i+k}\, y_{i+k}(t-1)\right]$$

Subject:

Year. Month. Date. ( )

Why Mexican Hat ?

there's a special function in Math which is called Mexican Hat. The characteristics of this network is to somehow like that function.



الگوریتم:

۱- مقادیر اولیه پارامترهای $R_2 - R_1$ و t_{Max} را مشخص می‌کنیم.

مقادیر اولیه فعالیت واحدها را برابر ۰ قرار می‌دهیم.

مقادیر وزن‌ها بصورت زیر مشخص می‌شوند:

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

۲- ورودی‌ها را به شبکه می‌دهیم و مقادیر اولیه فعالیت را در y^{OLD} ذخیره می‌کنیم.

$$\underline{Y} = \underline{X} \qquad y_i^{OLD} = X_i$$

مقدار t را یک واحد زیاد می‌کنیم. $t = t + 1$

۳- تا زمانی که $t < t_{Max}$ است مراحل 3 تا 7 را تکرار می‌کنیم.

۴- خروجی‌های تمام واحدها را محاسبه می‌کنیم.

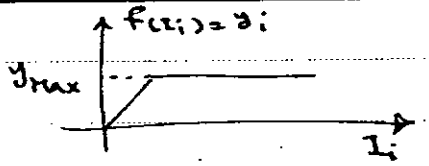
$$I_i = \sum_{k=R_2}^{R_2} w_{i+k,i}\, y_{i+k}^{OLD}$$

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

PAPCO

Subject:

Year. Month. Date. ( )



4- تابع فعالیت زیر را اعمال می‌کنیم

$$y_i = \text{Min}\{y_{Max}, \max(0, I_i)\}$$

5- خروجی را یک بار y_i^{OLD} ذخیره می‌کنیم

$$y_i^{OLD} = y_i \quad i = 1..N$$

6- افزایش شمارنده

$$t = t + 1$$

7- آزمایش شرط خاتمه: اگر $t < t_{Max}$ ادامه می‌دهیم.

در غیر این صورت: خاتمه

شبکه همینگ Hamming Network

فاصله همینگ (d_j): تعداد بیت‌هایی که در بردار $\underline{X}$ و بردار $\underline{S_j}$ در موقعیت‌های متناظر با هم متفاوت هستند

اگر $\underline{X}$ و $\underline{S_j}$ دو رشته دودویی با طول n باشند رابطه $a_j = n - d_j$ برقرار است.

اگر هر دو بردار $\underline{S_j}$ و $\underline{X}$ قطبی باشند

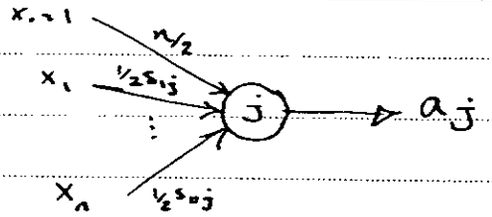
$$\underline{S_j} \cdot \underline{X} = a_j - d_j \quad (?)$$

از رابطه بالا داریم: $d_j = n - a_j$

$$\Rightarrow \underline{S_j} \cdot \underline{X} = 2a_j - n \quad ; \quad a_j = \frac{1}{2}\underline{S_j} \cdot \underline{X} + \frac{1}{2}n$$

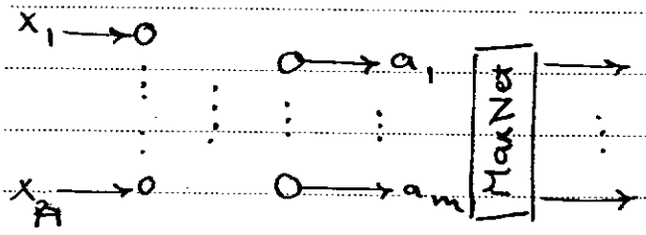
$$\frac{1}{2}\sum_i S_{ij} X_i + \frac{1}{2}n$$

Subject:
Year. Month. Date. ( )



در نوع یک واحد بصورت زیر در نظر گرفت

همینه ای که طرح می شود این است که بعد از شبکه ای ساخته شدیم از آن در ورودی x بردار متقابل شده،



با شبکه هیمنگ می توانیم نمونه ای که از شبکه
بردار نمونه (S_j) شبیه ترین بردار به
بردار ورودی می باشد.

می توانیم خروجی شبکه را به یک لایه شبکه Max Net متصل نمود، شبیه ترین بردار را که
مشخص شده

الگوریتم:

زمانی که n تعداد مشخصه ها، m تعداد نمونه های ذخیره شده S_j بردارهای نمونه باشند

۱. وزن های شبکه را تعیین می کنیم

$(V_{ij} = \frac{1}{2} S_{ij})$

۱. برای هر بردار x تکرار ۲ الی ۴ را انجام می دهیم

۲. خروجی خطی را محاسبه می کنیم $y_j = n/2 + \sum V_{ij} x_i$

۳. مقادیر اولیه شبکه Max Net را مشخص می کنیم $y_j(0) = y_j$

۴. الگوریتم Max Net را اجرا می کنیم. شبیه ترین نمونه مشخص شده

لایه (شبکه) برگشتی

ریشه فلسفی برگشتی ۱۹۸۲ قبل از آن مبحث خودش مبنای این نظریات

Subject:

Year. Month. Date. ( )

مجموعه داده‌های مربوط به بروزی (یا مشاهدات مختلف یک‌ازیک خروجی) آنقدر زیاد باشد که بتوان یکی از آنها را حذف کرد بدون اینکه تغییری در اطلاعات بدست آمده از آن داده‌ها بوجود آید. (بسته محدود یعنی مجموعه‌ای از مشاهدات)

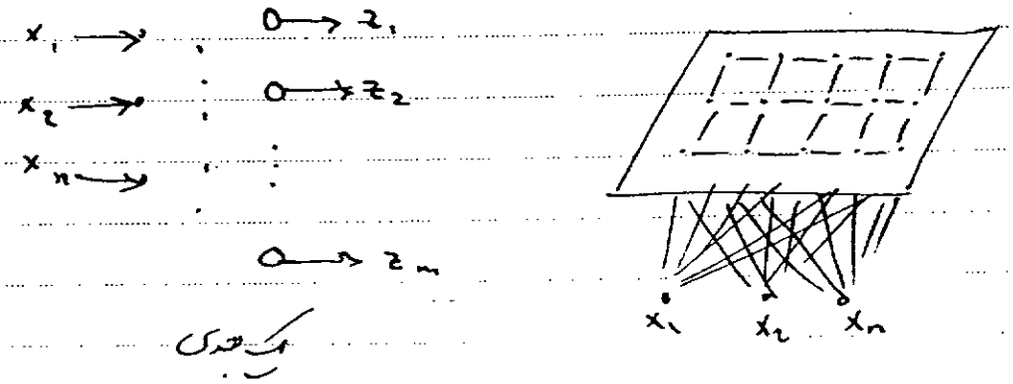
در سال 1976 به سبب این ایده بود که وقتی که در این نقل سعی بر این شد که مطرح شود.

در همین زمان بازرگ ورلت تحقیقی از این قبیل را ارائه داد و اما در حدود 1970 کشف این نتیجه را در پی داشت که هدف این قبیل پژوهش‌ها چیست که مجموعه W_i های داده‌های اندازه‌گیری هم احتمال یک نتیجه حاصل احتمال ثابت p را تشکیل می‌دهند. یعنی برای هر W_i باید خروجی تغییر هندسی برای هر مجموعه X مورد یک حاصل احتمال p داشته باشیم.

$$P\{\sum_{i=1}^{n} W_i \cdot X\} = \frac{1}{m!}$$

کشف این نتیجه مبنی بود بر این ایده که اندازه‌گیری‌های تصادفی روی توان حاصل احتمال یکنواخت به طور مطلوب کاری که در سال 1987 دوین سیز تغییر در قبل از کشف بوجود آمد (یا حداقل) مشکل بزرگ داشتن که این بود که حال این قبیل به نام قبل کشف شناخته می‌شود.

ایده کشف اولیه‌ای از تئوری‌ها که بصورت یک بعدی، دو بعدی یا بیشتر ارائه شده است.



یک بعدی

Subject:
Year. Month. Date. ( )

بردار یادگیری یک از واحد (i) فاصله مطابق بردار X تا مرکز خوشه‌ای محاسبه می‌کند

$$I_i = D(X, W_i)$$

↑ بخش مقایسه

تابع فاصله $D(u,v) = |u - v|$ فاصله اقلیدسی

$D(u,v) = 1 - \cos\theta$ $\theta = \angle u,v$ ° در

واحد برنده یا نزدیک‌ترین: X را دریافت می‌کند. کدام تعیین نزدیکترین W_i به X چند صورت ممکن است:

۱) هر واحد از لحاظ بردار I_i مربوط به خود فاصله آن تا ورودی محاسبه شده است و بر اساس آن تصمیم گیری می‌شود

۲) هیچ واحد (اتصالات جانبی) به طوری ایجاد شود بطوریکه تنها واحد برنده فعال باشد

۳- یک واحد برنده نیز I_i (فاصله) که برای آن محاسبه شده اعلام نماید

۴) واحد برنده مرتبط با بردار آستانه θ ارسال کند، اولین واحدی که فاصله I_i کمتر از آستانه باشد برنده محسوب شده. آستانه θ انتخاب می‌شود پس از هر بار افزایش می‌یابد

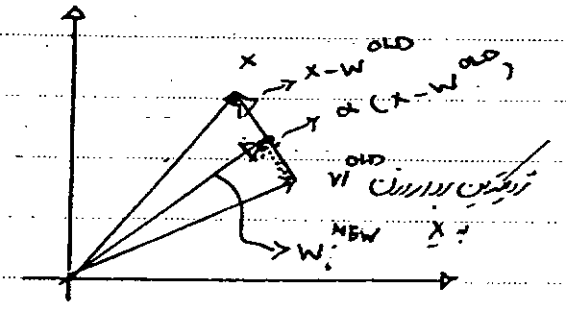
بهترین روش برای تعیین بردار وزنی W_i آن بردار است که فاصله آن تا بردار ورودی X کمترین باشد

$$W_i^{NEW} = W_i^{OLD} + \alpha (X - W_i^{OLD}) \quad 0 < \alpha < 1$$

در این حالت فاصله کوچک‌تر می‌شود

به صورت دیگر:

$$\begin{cases} W_i^{NEW} = (1-\alpha) W_i^{OLD} + \alpha X & \text{برای واحد برنده} \\ W_i^{NEW} = W_i^{OLD} & \text{برای سایر واحدها} \end{cases}$$



PAPCO

Subject:

Year. Month. Date. ( )

در اوایل فاز رشد، α نزدیک به صفر است و سپس به سرعت به حدود 0.8 افزایش می‌یابد. پیشرفت یادگیری α کاهش داده می‌شود تا از نوسانات در پایان یادگیری جلوگیری شود. این مقدار معمولاً در بازه 0.1 است.

- اعمال بردارهای ورودی مختلف به ورودی شبکه، بردار وزن‌ها را به گونه‌ای تغییر می‌دهد که بردار وزن‌ها به بردار ورودی نزدیک‌تر شود. نورون‌هایی که بردار وزن‌هایشان به بردار ورودی نزدیک‌تر است، مقدار خروجی بزرگتری تولید می‌کنند. در نهایت، پس از یادگیری، نورون‌های مختلف برای دسته‌های مختلف بردارهای ورودی پاسخ بزرگی تولید می‌کنند.

قبلاً اشاره شد که در خوشه‌بندی بردارهای W_i طوری تنظیم می‌شدند که بردارهای ورودی X با توجه به احتمال P احتمال اینکه X به W_i نزدیک باشد برابر $\frac{1}{m}$ باشد. با توجه به یادگیری کوهنن $P(w_i)$ نیز دارای خاصیت خاصی برای $\frac{1}{m}$ ای شده است. حقیقتاً که P در روی یک دامنه یک خاصیت مستقل دارای توزیع یکنواخت یا نزدیک به یکنواخت باشد.

اگر $P(w_i) \neq \frac{1}{m}$ شده چه اتفاقی می‌افتد؟

- در این حالت ممکن است به شکل گیری اتفاق بردارهای وزن در نواحی از ورودی و عدم حرکت وزن‌ها به نواحی مطلوب بر سر راه شویم. همچنین ممکن است یک بردار وزن خاص در یک خوشه بزرگ قرار گیرد به گونه‌ای که X های ورودی اندکی از آن خوشه نزدیک‌تر باشد و لذا هیچ‌گاه آن بردار وزن به ورودی‌های این خوشه نزدیک نشود و این حالت وزن‌ها را به یک راه حل چند واحدی (واحدهای مختلف) اصلاح نخواهد شد. واحدی که وزنش اصلاً اصلاح نشده را یک واحد مرده می‌گوییم.

چندین راه برای حل این مشکل پیشنهاد شده است:

1) اشعه‌ای شدن (Radial Sprouting)

هر بردار وزن را ابتدا برابر θ در نظر می‌گیریم. بردارهای ورودی را نیز ابتدا در یک ضریب (مثلاً یک دهم) ضرب می‌کنیم. با گذشت زمان، ضریب بردارها را به 1 نزدیک می‌شود.

- شکل این روش مطلوب بسیار کند آن است.

Subject:
Year. Month. Date. ( )

۲) اضافه کردن نویز:

یک نویز دارای توزیع یکنواخت به بردارهای X اضافه می‌کنیم بطوریکه در ابتدا نسبتاً قوی باشد و بر بردارهای X غالب. با گذشت زمان از قوت نویز کاسته می‌شود تا تدریجاً از بین برود. این روش نیز بسیار کند است.

۳- می‌توان در یک همسایگی هر واحد برنده، همه بردارهای وزن موجود را اصلاح نمود. به عبارت دیگر، به جای جذب بردار ورودی به وزنهای یک دسته‌ای از واحدها D در یک همسایگی واحد برنده اثر می‌کند و آنها را جذب می‌کند. در ابتدای آموزش، شعاع همسایگی نزدیک را بزرگ انتخاب می‌کنیم بطوریکه همه واحدها در همسایگی واحد برنده واقع شوند. با پیشرفت آموزش، شعاع همسایگی را کاهش می‌دهیم تا در پایان آموزش، همسایگی تنها شامل واحد برنده باشد.

۴) روش وجدان

برای شبکه یک وجدان در نظر می‌گیریم. در این روش، سابقه برنده شدن واحدها را نگه داشته، اگر واحدی بیش از $1/m$ دفعه برنده شود آنگاه آن را موقتاً از صحنه رقابت خارج می‌کنیم. این روش اکثر اوقات بسیار خوب عمل می‌کند.

پیاده‌سازی مکانیزم وجدان: پس از پایان رقابت و تعیین z_i ها، نسبت زمانی D واحدها را برنده می‌شوند به صورت زیر محاسبه می‌شود

$$f_i^{NEW} = f_i^{OLD} + \beta (z_i - f_i^{OLD})$$

(β: ضریب کوچک)

و همچنین تعصب را با این رابطه محاسبه می‌کنیم $b_i = \gamma (1/m - f_i^{NEW})$

(γ: مقدار مثبت بزرگ)

حال وزنهای واحدی اصلاح می‌شود D دارای کوچکترین مقدار $D(w_i, X) - b_i$ باشد.

Subject:

Year. Month. Date. ( )

الگوریتم یادگیری کوهونن

0 : مقادیر اولیه ای برای W_{ij} اختیار می‌کنیم.

پارامتری یک همسایگی اطراف هر گره تعیین می‌کنیم.

نرخ یادگیری

1 : تا زمانی که شرط خاتمه غلط است قدم‌های 2 الی 8 را انجام می‌دهیم.

2 : برای هر بردار ورودی X قدم‌های 3 الی 5 را انجام می‌دهیم:

3 : برای هر j فاصله D را محاسبه می‌کنیم.

$$D(j) = \sum (W_{ij} - X_i)^2$$

4 : اندیس J را که برای آن D(J) مینیم است پیدا می‌کنیم.

5 : برای همه گره‌های j در همسایگی J وزن‌ها را به روز می‌کنیم:

$$W_{ij}^{NEW} = (1-\alpha) W_{ij}^{OLD} + \alpha X_i$$

6 : نرخ یادگیری را به‌روز می‌آوریم. (α)

7 : شعاع همسایگی را در زمان‌های مشخص کم می‌کنیم.

8 : شرط خاتمه را آزمایش می‌کنیم.

نرخ یادگیری α را یک تابع کاهشی با زمان در نظر می‌گیریم.

کوهونن نشان داد که یک تابع خطی کاهشی با زمان اگر آهسته کم شود پس از مرحله اول، α با 1

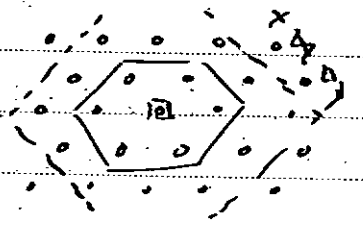
Subject:
Year. Month. Date. ( )

تعدیر کردیم را انتخاب کنیم

برای جلوگیری شدن از لازم است تجربه، نورشهی به بعضات متعدد به شبکه اعمال شود.

در آن به یک بعدی تب بالای بطور ستونی از اطراف نمونه برزده و ثابت می شود.

در 2 بعدی یک را، اهمیت که مستطیلی عمل کنیم. یعنی از همه ثابت در نظر بگیریم.



در حالت 3 هم جایگاه شش ضلعی خواهیم داشت.

انتخاب محل بهینه

مهم اهمیت: تأثیر یک سلول بر خانه سلول دیگر بیشتر است. برای لحاظ کردن این مسأله یک تابع تب بالای در نظر می گیریم.

$$h_{j,i}(x) = e^{-\frac{d^2_{j,i}}{2\sigma^2}}$$

$d_{j,i}$: فاصله نورون j از i

$h_{j,i}$: تب بالای حول نورون i با محل نورون j

برای حالت یک بعدی $d_{j,i} = |j-i|$ برای حالت 2 بعدی $d^2_{j,i} = \|r_j - r_i\|^2$

r_j و r_i بردار مکان نورون j و i هستند.

Subject:
Year. Month. Date. ( )

کاهش تدریجی اندازه همسایگی $\sigma(n) = \sigma_0 e^{-n/\tau_1}$ n= iteration شماره

σ_0 و τ_1 به انتخاب خود ما است

نرخ یادگیری $\eta(n) = \eta_0 e^{-n/\tau_2}$

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

الگوریتم کوهونن خودسازمانده به ۳ فاز تقسیم شده است:

۱) فاز خودسازماندهی یا مرتب‌سازی (Ordering Phase)

در این فاز، نورون‌ها مرتب می‌شوند. شعاع همسایه از 1000 شروع می‌شود تا الگوریتم به یک همگرایی نسبی برسد.

انتخاب $\eta(n)$ در ابتدا در این فاز اهمیت دارد. $\eta(n)$ را می‌توان با 0.1 شروع کرد و به تدریج کم کرد تا به 0.01 برسد. $\eta_0 = 0.1$ و $\tau_2 = 1000$ انتخاب شده است.

اندازه همسایگی در ابتدا بزرگ است تا همه نورون‌ها در یک همسایگی قرار گیرند و سپس این همسایگی کاهش می‌یابد تا در پایان فاز مرتب‌سازی تنها یک نورون برنده در همسایگی باشد. لذا σ_0 طوری انتخاب شده که شعاع همسایگی در ابتدا برابر $\frac{1000}{\log \sigma_0}$ باشد.

۲) فاز همگرایی

این فاز برای تنظیم دقیق نقشه است. تعداد الگوریتم در این فاز معمولاً حدود 500 برابر تعداد نورون‌های نقشه است.

در تمام طول فاز همگرایی $\eta(n)$ باید یک عدد ثابت کوچک مثلاً 0.01 باشد.

همچنین در این فاز، اندازه همسایگی باید به اندازه‌ای کوچک باشد که فقط شامل نورون برنده و همسایه‌های بسیار نزدیک باشد و عملاً نقشه دیگر تغییر محسوسی نکند.

PAPCO

Subject:
Year. Month. Date. ( )

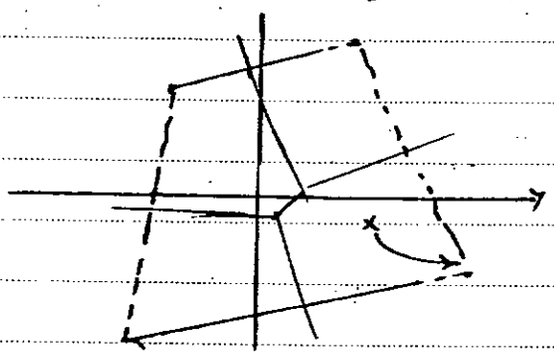
(LVQ)

Learning Vector Quantization Network شبکه یادگیری کوانتیزاسیون برداری

در این روش فضای برداری به تعدادی ناحیه تقسیم می شود و برای هر ناحیه یک بردار بازسازی تعیین می شود. بنحوی که اگر یک بردار جدید ورودی به سیستم وارد شود ابتدا ناحیه ای که بردار در آن واقع می شود مشخص می گردد و سپس بردار متناظر با آن ناحیه یعنی بردار بازسازی مربوطه ارائه می شود. با این روش تقاضا است در شبکه کاهش می یابد زیرا تنها شماره مربوط به بردار بازسازی به کار گرفته می شود. مجموعه ترم برداری بازسازی را کتاب کد بازسازی می نامند.

(تقسیم بندی فضا بر اساس فاصله انجام می شود.)

یک کوانتیزه کننده ورونوئی (Voronoi) یک کوانتیزه کننده برداری است که حداقل امید ریاضی است. در آن سلول های ورونوئی ایجاد شده حول نقاط در فضای ورودی توسط تقسیم بندی فضا با استفاده از نزدیکترین همسایه مبتنی بر فاصله اقلیدسی ایجاد می شود.



تعیین بردار ورونوئی
تصویر تقسیم بندی فضا
استفاده از الگوریتم رقابتی
بهبود بردار شخص شده.
روال دسته بندی الگو در یک شبکه برداری
تفاوت نیز در استفاده از شبکه ای یادگیری کوانتیزاسیون برداری نهفته شده است.
در این روش اطلاعات مربوط به کلاس دسته ای که برگشته می شود، بردار ورونوئی به طور جزئی تغییر داده شده، یک تقریب بهتر ایجاد می شود.

یک بردار ورودی S به صورت آنی از فضای ورودی انتخاب می شود. اگر دسته مربوط به S و دسته مربوط به نزدیکترین بردار ورونوئی W با هم تطبیق کند، W به سمت S جذب می شود و اگر دسته ای آنها با هم تطبیق

Subject:

Year. Month. Date. ( )

نکته: W را مقداردهی اولیه می‌شود.

پرسترون : 1	پرسترون ۹۸٪	
آدالاین : 2	〃	
شبکه‌ی عصبی چند : 3	۹۳٪	
پرسپترون : 4	〃	
شبکه‌ی لایه‌ی تابعی : 5	〃	
هاپفیلد : 6	۱۰٪	
ماشین بولتزمن : 7	〃	
هیمنگ : 8	〃	
کوهونن : 9	〃	
LVQ : 10	۱۰٪	

الگوریتم LVQ

ورودی شبکه $S = [s_1 \dots s_n]^T$ یک بردار

C_s : دسته صحیح برای بردار S

W_j : بردار وزن واحد خروجی ام

C_j : دسته ارائه شده توسط واحد خروجی ام

$\|S-W_j\|$: فاصله اقلیدسی بین بردار S و بردار وزن واحد خروجی ام

α : برداری وزن واحد را تعیین می‌کنیم. مقدار اولیه α را مشخص می‌کنیم.

1 - تا زمانی که شرط خاتمه خطا است قدم‌های 2 الی 6 را انجام می‌دهیم:

2 - برای هر بردار ورودی، مشابه‌ترین واحد قدم‌های 3 الی 4 را انجام می‌دهیم.

3 - مقدار j را که برای آن $\|S-W_j\|$ حداقل است مشخص می‌کنیم.

4 - مقدار W_j را به‌هنگام می‌کنیم:

اگر $C_s = C_j$

$$W_j^{NEW} = W_j^{OLD} + \alpha[S - W_j^{OLD}]$$

اگر $C_s \neq C_j$

$$W_j^{NEW} = W_j^{OLD} - \alpha[S - W_j^{OLD}]$$

5 - نرخ یادگیری α را کاهش می‌دهیم.

6 - شرط خاتمه را تست می‌کنیم.

Subject:

Year Month Date ( )

تعیین اولیه و ذهنی من توسط یادگیری رقابتی (شبکه SOM) و روش های خوشه بندی مانند روش (K-Means) انجام شده و با انتخاب m بردار از بردارهای آموزش به صورت تصادفی

نسخه های دیگر الگوریتم LVQ

(T بردار نزدیکترین بردار ورودی)

نسخه LVQ2 : در صورت برقراری شرایط زیر 2 بردار برنده و بردار برنده اصلاح می گردد

1: بردار برنده و بردار بردار برنده به 2 دسته مختلف متعلق باشند.

2- بردار ورودی متعلق به دسته بردار برنده باشد.

3. فاصله بردار ورودی با بردار برنده و بردار بردار برنده تقریباً مساوی باشد.

این شرایط را می توان بصورت زیر نیز بیان نمود:

اگر S بردار ورودی فعلی

W_c نزدیکترین بردار وزن به S

W_r " " " بعدی = S

d_c فاصله S تا W_c

d_r " " W_r

درنظر بگیریم $d_c < d_r$

$$\frac{d_c}{d_r} > 1-\varepsilon \quad , \quad \frac{d_r}{d_c} < 1+\varepsilon$$

بردار W_c و W_r اصلاح می گردند اگر شرط فوق برقرار باشد W_c و S به دو دسته مختلف متعلق باشند و S متعلق به دسته W_r باشد. اصلاح وزن ها بصورت زیر خلاصه میشود.

$$\left.\begin{aligned} W_c(t+1) &= W_c(t) - \alpha(t)\,[S(t) - W_c(t)] \\ W_r(t+1) &= W_r(t) + \alpha(t)\,[S(t) - W_r(t)] \end{aligned}\right\}$$ حال نایس

PaPCO

Subject :

Year ... Month ... Date ... ( )

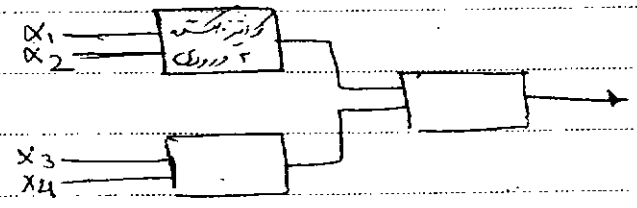
نسخه LVQ 2.1 : تغییر یافته (

" " " : LVQ 3 " "

کوانتیزاسیون برداری سلسله مراتبی

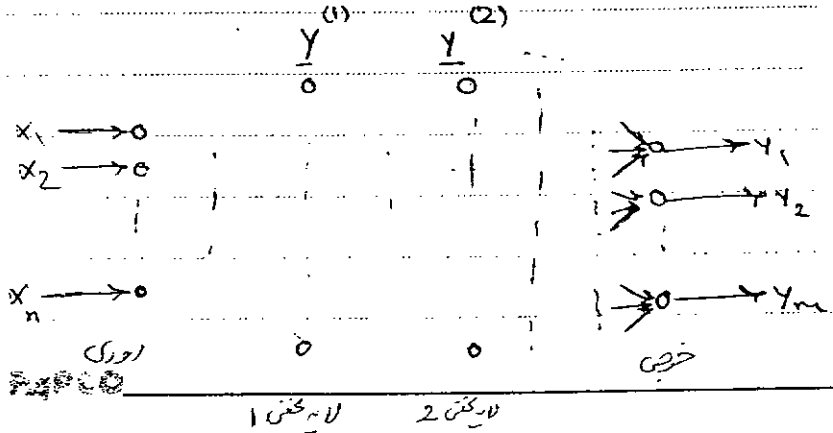
علت به کوانتیزاسیون برداری بواسطه نتیجه ای است که از تقریب (توابع) توزیع شانن حاصل شده است و آن این است که همیشه می‌توان با یک کد کردن بردار به جای استفاده از شیوه‌سازی اطلاعات آن را

با کوانتیزاسیون برداری، شکل حجم داده‌ای که باید مدیریت شود خصوصاً در حالتی که کد کردن داده‌ای برای هر بردار عضوی از یک مجموعه انتخاب شده باشد، بسیار کاهش پیدا می‌کند. کوانتیزاسیون برداری چند مرحله‌ای پیشنهاد شده است که در آن با فدا کردن کمی صحت بازیابی بطور قابل ملاحظه‌ای افزایش می‌یابد. از جمله روش‌های کوانتیزاسیون برداری چند مرحله‌ای، کوانتیزاسیون برداری سلسله مراتبی می‌باشد. در این حالت مثلاً برای کوانتیزه کردن بردار $X = [x_1 x_2 x_3 x_4]^T$ می‌توان از سه کوانتیزه کننده ۲ ورودی استفاده کرد.



کلاسیفایر پرسپترون چند لایه

Mulei layer Perceptron (MLP)



Subject :

Year . Month . Date ( )

اتصالات کامل است.

قانون یادگیری: انتشار خطا به عقب یا پس انتشار خطا

Back propagation Rule (BP) یا قانون دلتای تعمیم یافته

قانون BP ۲ مرحله دارد.

۱) اعمال ورودی به شبکه و محاسبه خروجی واحدها تا لایه آخر (انتشار رو به جلو)

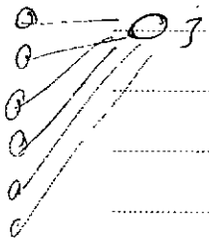
۲) محاسبه خطا در خروجی و خروجی لایه‌های مخفی (انتشار خطا به عقب)

۳) بهنگام آوردن وزنها با استفاده از نتایج مرحله ۲ (قانون واحد بطور هزمان / بهنگام آوری می‌شوند)

استخراج قانون BP

فرض کنید $\underline{s}_p = [s_{p_1} \dots s_{p_n}]^T$ یک ورودی آموزشی باشد و E_p تابع خطا برای این نمونه و $\underline{T}_p = [T_{p_1} \dots T_{p_m}]^T$ خروجی مطلوب برای ورودی $\underline{s}_p$ باشد و $\underline{Y}_p^{(\ell)} = [Y_{p_1}^{(\ell)} \dots Y_{p_{M_\ell}}^{(\ell)}]^T$ خروجی مشاهده شده برای ورودی $\underline{s}_p$ در لایه ℓ باشد $W_{ij}^{(\ell)}$ وزن اتصال از واحد i در لایه $\ell-1$ به واحد j در لایه ℓ باشد

خطای شبکه برای ورودی $\underline{s}_p$ $\quad E_p = \frac{1}{2}\sum_j (t_{pj} - y_{pj})^2$



فعالیت واحد j در لایه ℓ $\quad I_{pj}^{(\ell)} = \sum_k W_{kj}^{(\ell)} y_{pk}^{(\ell-1)}$

خروجی واحد j در لایه ℓ $\quad y_{pj}^{(\ell)} = f_j(I_{pj}^{(\ell)})$

$$\frac{\partial E_p}{\partial W_{ij}^{(\ell)}} = \frac{\partial E_p}{\partial I_{pj}^{(\ell)}} \cdot \frac{\partial I_{pj}^{(\ell)}}{\partial W_{ij}^{(\ell)}}$$

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

PAPCO

Subject:

Year ____ Month ____ Date ____ ( )

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

$$= y_{pi}^{(e-1)}$$

$$\Rightarrow -\frac{\partial E_p}{\partial w_{ij}^{(e)}} = -\nabla_w E_p = \delta_{pj}^{(e)} y_{pi}^{(e-1)}$$

تغییر وزن‌ها در BP تنها به کاهش خطا کمک کند.

$$w_{ij}^{New(e)} = w_{ij}^{OLD(e)} + \eta \left(-\frac{\partial E_p}{\partial w_{ij}^{(e)}} \right)$$

$$= w_{ij}^{OLD(e)} + \eta \left(\delta_{pj}^{(e)} y_{pi}^{(e-1)} \right) \quad : \text{BP قانون}$$

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

$\frac{\partial y_{pj}^{(e)}}{\partial I_{pj}^{(e)}} \rightarrow f'(I_{pj}^{(e)})$

برای لایه خروجی:

$$\frac{\partial E_p}{\partial y_{pj}^{(e)}} = -(t_{pj} - y_{pj})$$

$$\Rightarrow \delta_{pj} = (t_{pj} - y_{pj}) f_j'(I_{pj})$$

⇐ به شرطی است که تابع فعالیت ما مشتق‌پذیر باشد.

برای نقاط درونی مخفی:

$$\frac{\partial E_p}{\partial y_{pj}^{(e)}} = \sum_k \frac{\partial E_p}{\partial I_{pk}^{(e+1)}} \cdot \frac{\partial I_{pk}^{(e+1)}}{\partial y_{pj}^{(e)}}$$

$-\delta_{pk}^{(e+1)}$

Subject :

Year . Month . Date . ( )

$$= \sum \left(-\delta_{P_k}^{(e+1)}\right) \frac{\partial}{\partial y_{P_j}^{(e)}} \sum_i w_{ik}^{(e+1)} y_{P_i}^{(e)}$$

$$= -\sum_k \delta_{P_k}^{(e+1)} w_{jk}^{(e+1)}$$

$$\Rightarrow \delta_{P_j}^{(e)} = f'\left(I_{P_j}^{(e)}\right) \sum_k \delta_{P_k}^{(e+1)} w_{jk}^{(e+1)}$$ برای لایه‌های مخفی

اگر f تابع سیگموئید باینری باشد:

$$f(z) = \frac{1}{1+e^{-kz}}$$

$$f'(z) = k f(z)\left(1-f(z)\right)$$

سیگموئید متقارن:

$$f(z) = \frac{2}{1+e^{-kz}} - 1$$

$$f'(z) = \frac{k}{2}\left[1+f(z)\right]\left[1-f(z)\right]$$

الگوریتم آموزش شبکه HPL

(بدنی)

۱. شرط خاتمه: خطا بیست قدم‌های 2 الی 7 را انجام می‌دهیم.

۲. برای هر نوع بردار آموزش (S_p, T_p) قدم‌های 3 الی 6 را انجام می‌دهیم.

۳. محاسبه خروجی واحد از لایه مخفی اول تا لایه خروجی

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

$$y_{P_j}^{(e)} = f_j\left(\sum_i w_{ij}^{(e)} y_{P_i}^{(e-1)}\right)$$

۴. محاسبه δ_{P_j} برای واحدهای خروجی

$$\delta_{P_j} = \left(t_{P_j} - y_{P_j}\right) f'\left(I_{P_j}\right)$$

EXPCO

Subject:

Year Month Date

5. برای لایه مخفی با استفاده از $\delta_{pj}^{(l+1)}$ ، $\delta_{pj}^{(l)}$ را محاسبه می‌کنیم

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

6. اصلاح وزن‌ها

$$w_{ij}^{NEW} = w_{ij}^{OLD} + \eta \, \delta_{pj}^{(l)} y_{pi}^{(l-1)}$$

7. آزمایش شرط خاتمه

برای آموزش شبکه با این الگوریتم، مجموعه آموزشی باید یک بار به شبکه ارسال کرده و اصلاح وزن‌ها صورت گیرد. هر بار ارسال مجموعه آموزشی به شبکه را یک دوره یا یک epoch می‌نامند.

آموزش این شبکه زمان بر است.

پس از آموزش شبکه برای استفاده از آن از روابط زیر استفاده می‌کنیم.

0 : وزن‌ها با مقادیر یادگرفته شده در آموزش قرار داده می‌شوند.

1 : برای هر یک از الگوهای ورودی 2 و 3 را انجام می‌دهیم.

2. محاسبه خروجی لایه مخفی 1

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

3. محاسبه خروجی برای سایر لایه‌ها

$$y_i^{(l)} = f\left(\sum_j w_{ij}^{(l)} y_j^{(l-1)} \right)$$

Subject:

Year Month Date ( )

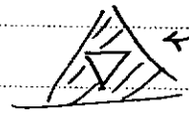
بینی از عملکرد شبکه

شبکه یک تابع خطا (انرژی) را تعریف می‌کند و با بهره‌گیری از روش نزول در راستای گرادیان به دنبال حداقل کردن آن است. گراف تابع خطا در فضای وزن‌ها، سطحی پر از پستی و بلندی است. (سطح خطا یا سطح انرژی) اگر ما یک گوی را در حداقل (ی) این سطح بگذاریم می‌شود این گوی از این حداقل (ها) حرکت کند و یا نکند (شبکه به دنبال آن است. شبکه با استفاده از یکی از این گام‌ها (روش‌ها) سعی می‌کند از یک طرف (نقطه) هر خطا بزرگ می‌شود

تعریف یک دسته‌بندی کننده:

* ۱) تعدادی (از بردارهای) یک خط را می‌کشند. بعضی از این خط‌ها نزدیک دسته‌بندی در صفحه‌ها انجام می‌دهد *
* ۲) این بردارها نواحی‌بندی انجام می‌دهد و دو خط را جدا می‌کند *

برای حل یک مسئله در فضای الگو (رسم می‌کنیم) با ترکیب خطی (ی) این نواحی (برای لایه بعدی) می‌توانیم فضای الگو را به طور مرتب تقسیم کنیم. نواحی ایجاد شده در این حالت، نواحی محدب خواهند بود. (نقطه‌ها یعنی) (تعداد اضلاع آن (یا) (برابر صفحه)) افزایش تعداد نواحی (در لایه اول برابر) افزایش تعداد نواحی به (ی) مرتب شکل شده می‌باشد. اگر یک لایه دیگر نیز اضافه کنیم نواحی این لایه، نواحی لایه قبل را با هم ترکیب کرده، نواحی محدب‌تری ایجاد می‌کند.



بنابراین برای داشتن حداقل ۴ ناحیه (۲ لایه پنهان) می‌توان صفحه‌ها به هر شکل (دلخواه) نواحی که (یعنی برای) دسته‌بندی را می‌شود با آن حل کرد.

قدرت حفظ کردن و قدرت تعمیم دادن

قدرت حفظ کردن: توانایی شبکه در پاسخ صحیح دادن به نمونه (ی) آموزشی

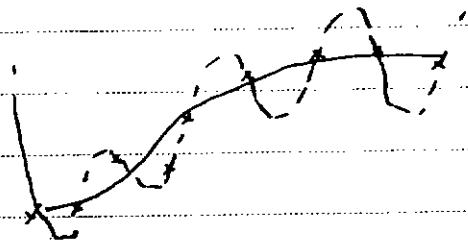
قدرت تعمیم دادن: " " " به نمونه (ی) که در حین آموزش دیده نشده (اما نمونه (ی) جدید)

قدرت حفظ کردن در تضاد با قدرت تعمیم دادن می‌باشد. معلوم است (این) که شبکه (با) باشد ای (تعادل) بین (دو) این دو نیاز کند.

PAGE

Subject:
Year ___ Month ___ Date ___ ( )

برازش منحنی Curve fitting



منحنی پیوسته مطلوب ما چیزی ساده تر است
تغییرات آن کمتر است و بر روی خطای آن
نسبت به خطاها ... بیشتر است. آموزش شبکه
نیز همین طور است.

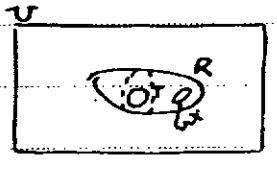
قدرت حفظ کردن مثال قبل، قدرت تعمیم خوب ——
" " خوب، " " ضعیف ----

آموزش شبکه متناظر است با حل برازش منحنی به داده های آموزشی
قدرت تعمیم به قدرت درون یابی

آموزش زیادی (own training) آموزش خیلی زیاد، نمونه های آموزشی بیش از حد، درجه های بخش بیشتر ——
آموزش مناسب، نمونه ای مناسب، مقدار کمتری از درجه آزادی ——
تعداد

در مسائل برازش منحنی و رگرسیون همیشه بین ساده ترین و پیچیده ترین پاسخ ها، جهت قابل قبول، کار را انجام دهد.

نکته دیگر اینکه با توجه به تعمیق های صورت گرفته در شبکه و نقصان اطلاعات صدمه دیده، اگر شبکه بخوبی تعلیم یافته باشد، به مطلق نظر مانند شده بعید از این مسئله جدا نمی شود.



می خواهیم R را راضی کنیم ... تا آموزش دهیم در اثر
این آموزش تجربه ای مثلاً ! یاد گرفته می شود.
مثلاً اگر بخواهیم با X تست کنیم خب مثلاً جواب خوبش اینه
دلیلش (؟) چون نمونه X را نیز از بین کرات جواب خوب انتخاب کردیم

Subject :

Year Month Date ( )

تقریب توابع با نگاشت شبکه‌های چندلایه

تقریب یک تابع ریاضی با یک نگاشت، استفاده از نمونه‌های (ورودی-خروجی) آن تابع در نقاط مختلف فضای ورودی است. شبکه‌های MLP بخوبی قادر به حل این مسأله می‌باشند. در شبکه‌های عصبی نگاشتی هدف پیدا کردن یک تقریب ساده برای یک نگاشت است.

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

با استفاده از نمونه‌های نگاشت $(\delta_1, T_1), (\delta_2, T_2), \ldots, (\delta_L, T_L)$ است که در آن $T_k = f(\delta_k)$ می‌باشد.

صحت تقریب را با یک مجموعه نمونه‌های جدید آموزش می‌سنجیم.

وقتی یک بردار آزمایشی X_k به شبکه اعمال شده بردار خطای خروجی برابر است با:

$$F_k(X_k, W) = \left| f(X_k) - Y_{ik}(X_k, W) \right|^2$$

برای N بردار نمونه بردار خطا برحسب متوسط برابر است با $F(W) \geq 0$،

$$F(W) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} F_k(X_k, W)$$

این حد برای یک شبکه خاص، افزایش تعداد نمونه‌ها تعداد مشخصی همگرا می‌شود.

یک راه برای مشخص کردن اینکه آیا نمونه مجموعه آزمایشی خیلی بزرگ شده آن است که به تقریب خوبی از مجموعه نمونه‌ها نزدیک شده‌ایم یا نه، جایی که اختلاف بین خطا به کمتر از یک مقدار ثابت همگرا شده.

در این نوع آموزش شبکه‌ها تابع $F(W)$ یک سطح پیچیده با دره‌های عمیق و نقاط کمینه (سطح خطا) در آموزش شبکه‌ها سعی می‌کنیم از $F(W)$ یک حداقل محلی بگیریم.

Subject:

Year Month Date ( )

قضیه کالموگروف: فرض کنید $f:[0,1]^n \to \mathbb{R}^m$ یک تابع پیوسته دلخواه و $\underline{y} = f(\underline{x})$ باشد. تابع f را می‌توان با یک شبکه سه لایه و n واحد ورودی در لایه اول $(2n+1)$ واحد در لایه وسط و m واحد خروجی در لایه سوم پیاده‌سازی کرد که در آن واحدهای مخفی دارای تابع تبدیل

$$z_k = \sum_{j=1}^{n} \lambda^k \psi(x_j + k\epsilon) + k$$

و واحدهای خروجی دارای تابع تبدیل

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

می‌باشند و در آنها

λ: مقدار حقیقی ثابت وابسته به n مستقل از f

ψ: تابع حقیقی پیوسته و افزاینده یکنوا، وابسته به n، مستقل از f

ϵ: عدد ثابت گویا $0 < \epsilon \leq \frac{1}{2n}$ δ: مقدار ثابت

g_i: یک تابع حقیقی پیوسته وابسته به f و ϵ

این قضیه تنها وجود یک شبکه برای پیاده‌سازی برای تابع f را مطرح می‌کند اما بدون نمی‌کند که چگونه چنین شبکه‌ای را پیدا کنیم. هیچ دستور خاصی از ψ و ϵ مشخص نیست و هیچ نمونه‌ای از g_i در دست نیست.

قلمرو تابع می‌تواند از $[0,1]^n$ به هر مجموعه مشترک گسترش یابد؛ لذا از این جهت قضیه کلیت خود را دارد.

(مجموعه مشترک: بسته و کراندار)

تعریف: اگر تابع f $f:[0,1]^n \to \mathbb{R}^m$ دوسویی باشد می‌گوییم f در L_2 است، f متعلق به L_2 است اگر هر یک از توابع مختصات f یکی از شکل‌های صورت مربع قابل انتگرال‌گیری باشد. یعنی

$$\int |f(x)|^2 \, dx$$

وجود داشته باشد

تکلیف

Subject :

Year . Month . Date . ( )

قضیه : فرض کنید $f:[0,1]^n \to \mathbb{R}^m$ یک تابع دلخواه و پیوسته باشد. یک شبکه سه لایه با اندازه مخفی کافی وجود دارد که تابع f را با حداکثر خطای ε تقریب زند. تعداد n_2 در حقیقت شامل یک تابع خطی است که در عمل مطرح می‌شود.

گرچه می‌توان یک تابع را با یک شبکه سه لایه پیاده‌سازی نمود اما اکثراً بهتر است ۴ یا ۵ لایه انتخاب شود زیرا با یک شبکه سه لایه ممکن است تعداد واحدهای لایه مخفی خیلی بیش از حد بزرگ شود و پیاده‌سازی شبکه را غیرعملی سازد. با افزایش تعداد لایه‌ها تعداد واحدهای مخفی لازم کاهش می‌یابد.

معادل جدید در رابطه با شبکه :

۱) انتخاب اولیه وزن‌ها : تأثیر زیادی در سرعت همگرایی و نزدیک شدن به مینیمم سراسری تابع خطا یا همگرایی در سرعت نزدیک شدن دارد.

خیلی مقادیر بزرگ باعث می‌شود که در ابتدا تابع فعالیت لایه آخر در ناحیه اشباع باشد که باعث می‌شود گرادیان کوچک شده و اصلاح وزن‌ها کند می‌شود.

خیلی کوچک هم نباید باشند چون در روش‌های گرادیان لایه‌ها کوچک می‌شوند و در شبکه‌های عمیق لایه‌های اولیه بسیار کوچک می‌شوند و اصلاح وزن‌ها کند می‌شود.

معمولاً بین $[-0.5, 0.5]$ یا $[-1, 1]$ انتخاب می‌شوند.

روش Nguyen & Widrow

$w_{ij}^{(0)}$ همه وزن‌ها ابتدا بطور اتفاقی بین $[-0.5, 0.5]$ انتخاب شوند

n تعداد واحدهای ورودی

p تعداد واحدهای لایه مخفی

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

وزن‌های لایه و واحد مخفی را به صورت زیر تغییر می‌دهیم :

PAPCO

Subject:

Year Month Date ( )

$$y_{ij}^{(s)} = \beta \frac{y_{ij}^{(s)}}{\| y_j^{(s)} \|}$$

متغیر y_{ij} بین $[-\beta, \beta]$ نرمال می‌کنیم.

با انتقال مرکز جرم به مبدأ، انتقال پیدا می‌کند.

۲- افزایش داده‌ها:

زمانی که تعداد زیادی از جفت‌ها را بسته به آن ممکن است مسئله حل شده یا نشده (ساده‌تر حل شود یا مشکل‌تر)

- گسسته‌سازی: ساده‌تر، محدوده‌های مختلف
- پیوسته‌سازی: سخت‌تر، یادگیری مشکل‌تر
- بهینه‌ی: در صورتی که بهینه‌ی شدیدتر دیگری را به تعادل شود.
- دوقطبی: دقت یادگیری بیشتر

۳- تعمیم واحدی محض:

تشکیل: عدم تشکیل سیستم‌های محلی به تشکیل دسته‌بندی صحیح وابسته است. اگر دو دسته متمایز تحلیل شوند دسته تحلیل‌کننده شده سیستم‌های محلی ایجاد خواهد شد.

تعمیم واحدی محض در ایجاد یا عدم ایجاد سیستم‌های محلی موثر است. افزایش تعمیم این واحد متناظر است با کاهش احتمال تشکیل سیستم‌های محلی.

یک رابطه معمولی برای تعیین تعداد واحدی محض موجود است. در بعضی از تعمیم‌های محض به طور خاص با بعد مدل مرتبط می‌شود.

۴- تعمیم نمونه‌های آموزشی لازم:

Subject:

Year. Month. Date. ( )

فرض کنید تعداد نمونه‌های موجود در شبکه N_w باشد

ϵ : خطای تابع تبدیل در دسته‌بندی الگوی آزمایشی

اگر شبکه بطوری آموزش داده شده (با تعداد نمونه‌ی آموزشی) را با خطای $\frac{1}{2}\epsilon$ دسته‌بندی کند و تعداد نمونه‌ی آموزشی برابر $L = \frac{N_w}{\epsilon}$ باشد، خطای دسته‌بندی الگوی آموزشی از ϵ تجاوز نخواهد کرد.

مثال: $\epsilon = 10\%$ $N_w = 50$ $L = 500$

شبکه‌ای را 500 نمونه با تعداد آموزش دهید (خطا برای نمونه‌ی آموزشی زیر 5٪ برسد!)

آموزش دسته‌بندی کننده ← مجموعه آموزش

آزمایش دسته‌بندی کننده ← با داده‌ی آموزش ← درصد خطا

← با داده‌ی جدید ← درصد خطا

۵- مجموعه‌ی آموزش - آزمایشی و میزان آموزش

مجموعه‌ی لازم برای تقریب تقریباً اندازه‌ای بزرگ بستگی (به) راه‌حل مسئله می‌کند.

چه‌قدر اندازه‌ی؟ ! وقتی مجموعه بزرگ شد تغییر محسوس ایجاد نکند. (idubt way!)

اگر دقت کاربردی، داده (خیلی کم) است، حل مسأله با شبکه عصبی موفق نخواهد بود!

اگر داده‌ی موجود زیاد باشد ← مجموعه آموزشی با اندازه کافی داده ($\frac{1}{2}$ داده)

مجموعه آزمایشی " " " ($\frac{1}{2}$ یا $\frac{1}{3}$ داده)

مجموعه آزمایشی 2 " " " ($\frac{1}{3}$ " $\frac{1}{4}$ ")

که برای تست شبکه دیگری: در اغلب طراحی‌ها از این روش!

PAGE

Subject:
Year. Month. Date.

اگر داده (جمعیت) زیاد نباشد:

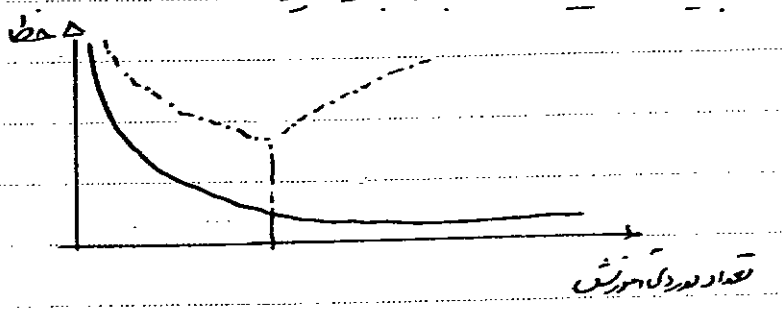
I- مجموعه آموزش را بار تمام داده در نظر می‌گیریم (با نمونه)

چقدر شبکه را آموزش دهیم؟

شبکه را تا زمانی یادگیری سنتی در آن به خوبی فرم همبط یاد گرفته: با ادامه آموزش فرم همبط تغییر می‌کند و سنتی در شبکه بیشتر از داده (ی نامه، در نتیجه سنتی دانایی (در این شرط خطا می‌شود. با دقت مکرر مجموعه آموزش، فرم مورد بحث بیشتر در هم می‌پیچد تا اینکه از داده (عبور کند در نقطه ای به تعادل می‌رسد عبور از نقاط و درونیابی می‌رسد. آموزش باید در این نقطه متوقف شود!

یادگیری ۷۰ تعمیم

در بسیاری از مسائل وضعیت شبکه به صورت زیر است:



نتیجه اخلاقی: با آموزش می‌دهیم خطا دیگر با افزایش تعداد دوره‌ای آموزش، کاهش چشمگیری نداشته باشد.

II. با بار شبکه را با نمونه آموزش دهیم. توقف آموزش در جایی موقعی است که یعنی خطای مجموعه آموزش تقریباً مسطح می‌شود.

خطا را برای نمونه با اندازه n در آموزش به کار رفته، برآورد می‌کنیم.

متوسط مربع خطا برای n نمونه فوق را محاسبه می‌کنیم.

خطای آزمایش شبکه

PRESS (Leave One Out)

Subject :

Year . Month . Date . ( )

III . جمل داده از بردی داده (ی موجه

A . بازی در تغییر تعداد داده (ی موجه

B . بازی در تغییر نویز به داده (ی موجه

(خوبیدت خوبیات هم باشه بتونه کاری کنه از آن جمل که دیم باشد هم آنها نیز تغییر دهیم)

نسخه (ی مختلف الگوریتم

نقاط ضعف الگوریتم اولیه : ۱. ثابت بودن در سیستم (ی محلی

۲. یک بعدی بودن

روش (ی برای رفع این مشکلات پیشنهاد شده است که :

۱. روش (ی مبتنی بر تغییر نحوه اصلاح وزنها

الف) استفاده از مومنتم

اگر η بزرگ باشد نوسانات شدیدی پیش می آید

اگر η کوچک باشد همگرایی کند خواهد بود

برای رفع مشکل به وزنها اینطور می دهیم

$$w_{ij}^{(e)}(t+1) = w_{ij}^{(e)}(t) + \eta \delta_{pj}^{(e)} y_{pi}^{(e)} + \mu [w_{ij}^{(e)}(t) - w_{ij}^{(e)}(t-1)]$$

Subject:

Year Month Date ( )

با ایکار :

- حرکت ، کارهای بزرگی انجام می‌دهیم لذا احتمال بروز از روی سیستم ایمنی زیاد می‌شود
- برای جلوگیری از آن باید
- نگاه به نقطه ثابت و نقطه فعلی جهت حرکت را مشخص می‌کند - حرکت در جهت گوشه
در نقطه فعلی است
- اطلاع دادن آموزشی خیلی مؤثر است با تغییر شیوه در یادگیری این کار می‌کنیم

الگوریتم : بستن ———

الگوریتم پیاده‌ی ———

بی‌ی پنج (۵) بنج گلستانم و زدوک
ملک‌ی لب (۱) تشنه مرا طلبم و زدوک
ای آفتاب جلنی بهن از ابر دی
کاین چرخ بشیخ تا بنم اندوک
گفتی ز نازبین بری جان سپار رو
آن گشته دلبش مرجانم اندوک
وعده دیدن گشته (۱) بروشه بر خانه‌ت
ماه نازه و نازه دیدنی در بنم اندوک
تقویت وعده وصل (۱) سفارش بی‌ی زنم
وعده‌ی خوب دیدنی تنگ بنم و زدوک
ماه سه (۳) شهر یا تو سرا حبس می‌شوه
آمدنی دیدنی و دیدنی بنم و زدوک
زینی همراه سمت منصور دلم گرفت
شیر خطه و بستم دستانم اندوک
زینی خلق پرشکات گریان شدم دلل
آن دی دیدنی و نغره مستانم و زدوک
گریام ز بلبل ای ز رشک مقام
حرکت مردم و افغانم اندوک
دل شیخ با چراغ همی گشت گرد شهر
گرد در و دیوار ملولم و انسانم و زدوک
گفتند بیت می‌نشوه حسته‌ام با
گفتم آنک بیت می‌نشوه بنم اندوک
پنج ز دوره (۵) و بیه دوره (۱۰) و زدوک
آن آشکار صفت نهانم و زدوک
گفتم شنیده قصه‌ای من و مت شد
گفتم چشم صورت ای بنم اندوک
من هم دیدنی بشتم و بشتم دیدنی با ک
طاس لطف‌ای زخمه جانم اندوک
با این غزل طرای مطرب ظریف
زینی زبان همی شکار طرز سلم اندوک
نبی‌ی نفس عنبر بیز او ز مشرق
من صدهم محمود سلیمی و زدوک

Subject:

Year Month Date ( )

نسخه های مختلف الگوریتم BP

۱- روش های مبتنی بر تغییر نحوه اصلاح وزن ها

الف- استفاده از مشتق

ب) هنگام ارائه نمونه ها دسته ای وزن ها

آموزش { الگو به الگو Pattern Mode
دسته ای Batch Mode

موجب تابع خطای اولیه شده، برای الگوهای مورد نظر
مقادیر وزن ها کاهش می یابد

سرعت بالاتر می رود.

$E_{av.} = \frac{1}{2L} \sum_{p=1}^{L} \sum_{j \in C} e_{pj}^2 \qquad (e_{pj} = t_{pj} - y_{pj})$

ج) تطبیق کردن نرخ یادگیری

ج-۱) در ابتدای یادگیری از نرخ انتخاب شده و در حین ۱ کاهش یابد به طوری که در
نزدیکی همگرایی ۱ نسبتاً کوچک باشد. سرعت یادگیری زیاد و نوسانات کم می شود.

ج-۲) روش دلتا بار دلتا: هر وزن نرخ یادگیری مخصوص خود را دارد و نرخ
با زمان نیز تغییر کند.

قاعده تجربی برای اصلاح ۲: اگر تغییرات وزن در چند قدم متوالی در یک جهت باشد
نرخ یادگیری افزایش یابد. ولی اگر نوسان شدید
نرخ کاهش یابد.

PAGE

Subject:

Year Month Date

اگر میزان خطا برای یک نمونه در زمان t-1 برابر E باشد، تغییر وزن از رابطه زیر خواهد بود:

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

α ضریب یادگیری است و بین ۰ و ۱ است.

در این روش نسبت مشابهی از مشتق خطا تغییر می‌یابد بدین‌گونه تغییر وزن در راستای کاهش گرادیان است.

$$\Delta_{jk} = \frac{\partial E}{\partial w_{jk}} = -\delta_k y_j$$ برای یک نمونه ورودی

$$\Delta_{ij}^{(e)} = \frac{\partial E}{\partial w_{ij}^{(e)}} = -\delta_j^{(e)} y_i^{(e-1)}$$ برای یک نمونه ورودی لایه e

$$\bar{\Delta}_{jk}^{(t)} = (1-\beta)\,\Delta_{jk}^{(t)} + \beta\,\bar{\Delta}_{jk}^{(t-1)}$$ برای یک نمونه ورودی

$$0 < \beta < 1$$

$$\bar{\Delta}_{ij}^{(e)(t)} = (1-\beta)\,\Delta_{ij}^{(e)(t)} + \beta\,\bar{\Delta}_{ij}^{(e)(t-1)}$$ برای $t = 0$

$$\alpha_{jk}^{(t)} = \begin{cases} \alpha_{jk}^{(t-1)} + \lambda & \bar{\Delta}_{jk}^{(t-1)}\,\Delta_{jk}^{(t-1)} > 0 \\ (1-\gamma)\,\alpha_{jk}^{(t-1)} & \text{در غیر این صورت} \\ \alpha_{jk}^{(t-1)} & \text{در غیر این صورت} \end{cases}$$

λ و γ و β توسط کاربر مشخص شوند.

Subject :

Year. Month. Date. ( )

۴- روش دی مبتنی بر تغییر تابع فعالیت ملاحظه

اگر تابع فعالیت متقارن باشد ($f(-x) = -f(x)$) احتمالاً سرعت یادگیری افزایش می‌یابد.

توابع مختلف:

- تابع سیگموئید در فضای محدود به بازه $[a, b]$

$$f_1(x) = \frac{1}{1+e^{-\alpha x}} \qquad f_2(x) = (b-a) f_1(x) + a$$

$$f_2'(x) = \frac{\alpha}{b-a} (f_2(x)-a)(b-f_2(x))$$

- تابع سیگموئید با شیب تطبیقی

تغییر قبلی α شکل است. می‌توان α را نیز در طی فرایند یادگیری مانند وزن تغییر کند.

معادلات شبکه با این تعدیل تغییر می‌یابد.

- تابع $\cot g^{-1}$

$$f(x) = \frac{2}{\pi} \cot g^{-1} x$$

سرعت یادگیری بیشتر از تابع سیگموئید است با $\tanh$

- توابع ایستا و نشده

- بقیه توابع لگاریتمی، توابع نمایی، توابع کاوشی

PAPCO

Subject :

Year ... Month ... Date ...

۲- روش های دیگر

الف) استفاده از الگوریتم های ژنتیک

مثلاً اینکه آموزش، الگوریتم BP و ۶ بزرگ را دنبال پس در مرحله ای از روش های ژنتیک برای ایجاد وزن های جدید استفاده شود.

ب) استفاده از مدل های زیستی

استرد آموزش، BP و پس، انتخاب یک مدل فزیکی مناسب برای وزن ها و استفاده از تئوری تکامل، وزن های جدید ایجاد می شود.

در کار دوم این شیوه:

- ابتدا هر کلاس بندی
- تخمین زدن هر تابع، هر چقدر دقت مورد نیاز

جدایپذیری الگوها

قبل نقش های مدرن به فضای بعد بالاتر می تواند قابلیت جدایپذیری الگوها را افزایش دهد.

فرض کنید مجموعه X شامل L بردار x_i باشد که یک زیرمجموعه X^+ و X^- است.

برای هر الگوی $x \in X$ بردار جدید تبدیل یافته به صورت زیر تعریف می کنیم. بردار

$$\varphi(x) = \begin{bmatrix} \varphi_1(x) \\ \vdots \\ \varphi_m(x) \end{bmatrix}$$

$\varphi(x)$ نقطه فضای n بعدی را به فضای m بعدی نگاشت می کند. $\varphi_i(x)$ را تابع تبدیل می نامند. دسته بندی $\{X^+, X^-\}$ از X، φ جدایی پذیر گفته می شود اگر یک بردار m بعدی W وجود داشته باشد بطوریکه

$$W^T \varphi(x) > 0 \quad x \in X^+$$

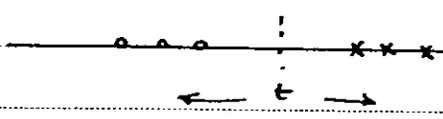
$$< 0 \quad x \in X^-$$

Subject :

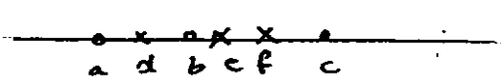
Year Month Date ( )

در این حال، از صفحه $\underset{\text{zero}}{\Phi} = w^T \varphi(x)$ سطح جدایی (جدایکننده) رفتن به خواهد بود.

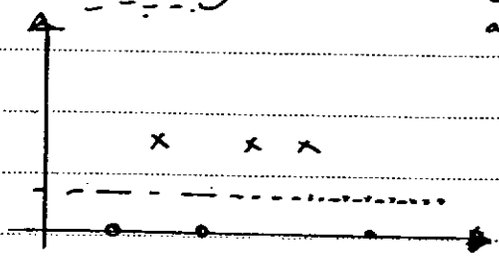
مثال ۱ :



مثال دیگر :



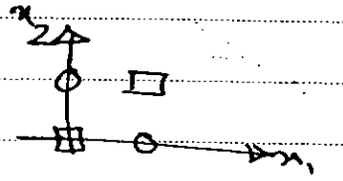
خط جدا کننده



$a \to \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ $b \to \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ $c \to \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$

$d \to \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ $e \to \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$ $f \to \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$

مثال بعدی : XOR



○ خروجی $\to 1$

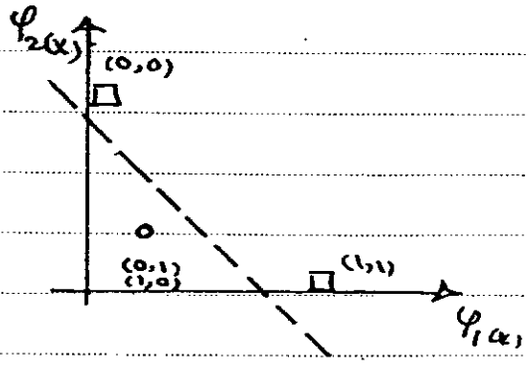
□ " $\to 0$

تابع گوس :

$$\varphi_1(x) = e^{-\|x-c_1\|^2} \qquad c_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\varphi_2(x) = e^{-\|x-c_2\|^2} \qquad c_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

x	$\varphi_1(x)$	$\varphi_2(x)$
$(1,1)$	1	0.135
$(0,0)$	0.135	1
$(0,1)$	0.368	0.368
$(1,0)$	0.368	0.368



در این بعد داده ها خطی جدا پذیر شدند و توانستیم برای آنها یک تفکیک خطی پیدا کنیم.

حل کنید

Page 60

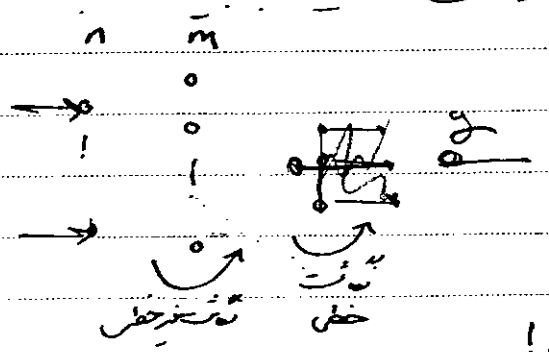
Subject:
Year Month Date ( )

شبکه‌های تابع پایه شعاعی (RBF) (Radial Basis Function)

محتوای:

۳ لایه: ورودی، مخفی، خروجی

تبدیل فضای ورودی به فضای لایه مخفی، یک تبدیل غیرخطی است و در لایه خروجی تبدیل خطی می‌باشد.



دو گام متوجه از هم مستقل می‌باشند.

از آنجا لایه‌های مخفی و خروجی کارهای مختلفی انجام می‌دهند، نظریات آموزش‌سازی لایه‌های اول و سوم به طور جداگانه انجام شود!

آموزش شبکه: تعیین پارامترهای توابع پایه شعاعی در لایه خروجی

توابع پایه شعاعی به روش‌های مختلف انتخاب می‌شوند. (وابسته به فاصله از مرکز نزدیک‌ترین مرکز است) تابع گاوسی

$$G(\|x-c_i\|^2) = e^{-\frac{H}{D^2}\|x-c_i\|^2}$$

انتخاب خوشه‌ها به روش سادگی

با کمک روش خوشه‌بندی، مراکز توابع تشخیص داده می‌شود سپس با استفاده از روش K نزدیک‌ترین همسایه

K میانگین

تحویل پذیره

چهارشنبه ۱۰/۶ ساعت ۱۴:۰۰ الی ۱۵
شنبه ۱۰/۹ " ۸:۰۰ " ۱۱

Subject: R.418
Year. Month. Date. ( )

تعداد مجهولات خطی: ابتدا بردار تعداد نمونه (بردار ورودی) را در یک ماتریس ستونی آنها کاشی دسیم.
با استفاده از یک الگوریتم خوشه بندی مراکز را پیدا می کنیم و سپس انحراف معیار آنها را محاسبه می کنیم.

با داشتن شکل توابع پایه مرتبط با لایه خطی انتخاب کرده و وزن های لایه خروجی را با یک روش حداقل کمتر مربعات خطا مشخص می کنیم.

تقسیم از فرم تابع گاوسی که در آن تقارن شعاعی به تقارن بیضوی (Elliptically Symm) تغییر می یابد.

جایگزین $(x_i - c_i)$ با $\frac{x_i - c_i}{\beta_i}$ توابع پایه شعاعی را به توابع پایه بیضوی تبدیل می کند.

در این کار، فاصله اقلیدسی به فاصله ماهالانوبیس تبدیل می شود که شبکه RBF تعمیم یافته تغییر یافته است!

روش یادگیری

در این روش مراکز توابع پایه شعاعی و پارامترهای دیگر شبکه تحت یک فرآیند یادگیری با نظارت انتخاب می شوند. برای این کار از روش نزول در امتداد گرادیان استفاده می شود.

$$e_j = d_j - F(x_j) = d_j - \sum_{i=1}^{n} w_i G(\|x_j - c_i\|_{D_i})$$

$$E = \frac{1}{2} \sum_{j=1}^{k} e_j^2$$

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

$$= e^{-\frac{1}{2}(x_j - c_i)^T S_i^{-1} (x_j - c_i)}$$

$$\left(D_i^T D_i = \frac{1}{2} S_i^{-1} \right)$$

از حد جدیدترین :D

PARCO

Subject:

Year Month Date ( )

مرحله‌ی c_iها، w_iها و S_i^{-1}ها را طوری تعیین کنیم که ع مینیمم شود!

۱) وزن‌های لایه‌ی خروجی

$$w_i(n+1) = w_i(n) - \eta_1 \frac{\partial \mathcal{E}(n)}{\partial w_i(n)} \quad i = 1 \cdots M$$

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

۲) مراکز RBF

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

$$\frac{\partial \mathcal{E}(n)}{\partial c_i(n)} = 2 w_i(n) \sum_{j=1}^{L} e_j(n) G'(\|x_j - c_i(n)\|_{D_i}) S_i^{-1}(x_j - c_i(n))$$

۳) توزیع‌ها

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

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

where $Q_{ji}(n) = (x_j - c_i(n))(x_j - c_i(n))^T$

مسئله‌ی درون‌یابی به تقریب تابع

فرض کنید مجموعه‌ای از نقاط مختلف x_i و متناظر با آن‌ها اعداد حقیقی d_i داده شده‌اند.

$x_i \in \mathbb{R}^n \quad i = 1, \ldots, L$

$d_i \in \mathbb{R} \quad i = 1, \ldots, L$

می‌خواهیم تابعی مثل F پیدا کنیم

$$F : \mathbb{R}^L \longrightarrow \mathbb{R}^1$$

که شرط زیر را ارضا کند.

$$F(x_i) = d_i$$

PAPCO

Subject:

Year. Month. Date. ( )

روش RBF شامل انتخاب یک تابع F به صورت زیر است:

$$F(x_i) = \sum_{j=1}^{L} \omega_j \varphi(\|x_i - c_j\|) = \sum_{j=1}^{L} \omega_j \varphi_{ij}$$

به ازای $\{\varphi(\|x_i - c_j\|) \mid i=1,2,\dots,L \;\; j=1,\dots,L\}$ مجموعه‌ای از تعاریف متمایز خواهد بود که تعاریف به این شکل نامیده می‌شوند. رابطه فوق را به صورت ماتریسی هم می‌توان نوشت:

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

$$\underline{\bar{\Phi}} \qquad \underline{W} \qquad \underline{d}$$

در این صورت: $\underline{\bar{\Phi}} \times \underline{W} = \underline{d}$

شرط بدست آمدن W این است که در صورت تغییر بعضی نقاط x_i، ماتریس $\bar{\Phi}$ غیر منفرد و برای دسته‌ای از توابع φ ماتریس به دست آمده تضمین خطی بودن نمونه‌ای از این توابع:

$$\varphi(r) = \frac{1}{\sqrt{r^2 + c^2}} \qquad c > 0 \quad r \geq 0$$

$$\varphi(r) = e^{-\frac{r^2}{2\sigma^2}} \qquad \sigma > 0 \quad r \geq 0$$

در این حال ضرایب وزنی از رابطه $\underline{W} = \bar{\Phi}^{-1} \underline{d}$ بدست می‌آید.

Page:

Subject:

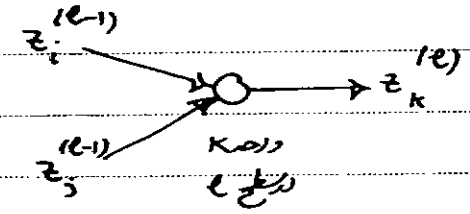
Year Month Date ( )

Group Method of Data Handling GMDH شبکه

در سال 1968 برای پیشگویی جمعیت بلغارستان به کار رفت. توسط ایواخنکو (who the hell ?) معرفی شد. برای کاربردهای مدل‌سازی متعدد دیگر هم بسیار خوب کار کرده است.

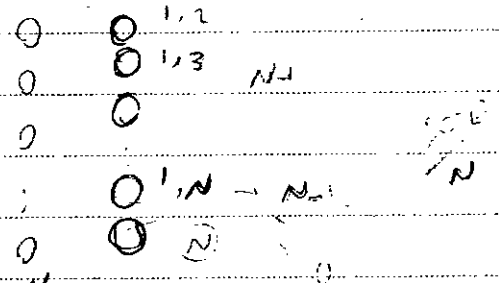
این شبکه عموماً برای تقریب تابع $f : A \subset \mathbb{R}^n \to \mathbb{R}$ به کار رفته است. برای تقریب توابع چندمتغیره برای هر مولفه بردار خروجی یک شبکه‌ی ای دیگر.

واحدهای شبکه



(تقاطع ورودی)

$$z_k^{(l)} = a_k^{(l)} \left(z_i^{(l-1)} \right)^2 + b_k^{(l)} \left(z_i^{(l-1)} z_j^{(l-1)} \right) + c_k^{(l)} \left(z_j^{(l-1)} \right)^2 + d_k^{(l)} z_i^{(l-1)} + e_k^{(l)} z_j^{(l-1)} + f_k^{(l)}$$



در ابتدا شبکه در سطح اول شامل چند جمله‌ای ایجاد شده در 2 متغیر شروع می‌شود. لذا اگر در شبکه به جز لایه ورودی، l سطح داشته باشد چند جمله‌ای خروجی آن درجه 2^l خواهد بود. این چند جمله‌ای چند جمله‌ای ایواخنکو نامیده می‌شود.

Subject :

Year . Month . Date . ( )

ساخت این شبکه بدین صورت است که از ورودی، لایه به لایه شبکه ساخته می‌شود. در هر لایه اولین قدم آن است که برای هر نرون خروجی لایه قبل یک واحد قائل شویم. اگر تعداد واحدهای در لایه l ام را با M_l نشان دهیم تعداد واحدهای در لایه $l+1$ ام برابر $\binom{M_l}{s}$ خواهد بود. هر لایه یک واحد اضافه می‌شود که مقدار ثابت تعلق داده شده و ثابت می‌ماند.

در ادامه هر واحد در هر لایه با پس می‌کند خروجی خطی را تولید می‌کند. تعلیم T تایی بر واحد مستقر از تعداد زیادی نمونه‌ی ورودی - خطی انجام می‌شود. با اعمال بر ورودی S_p خروجی‌های $z_j^{(l-1)}$ در لایه $(l-1)$ ام ایجاد شده و در نهایت مقادیر خطا t_p را ایجاد می‌کند. $z_i^{(l)}$ را با t_p در نظر می‌گیریم؛ لذا برای هر واحد در هر لایه برای هر نمونه آموزشی (S_p, t_p) یک معادله با T پارامتر ایجاد می‌شود!

هر لایه با استفاده از یک مجموعه آموزشی جدید آموزش داده می‌شود. معادلات حاصل از آموزش یک واحد K در سطح l را می‌توان به صورت زیر نوشت:

$$\begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_L \end{matrix} \begin{bmatrix} (z_i^{(l-1)})^2 & (z_i^{(l-1)})(z_j^{(l-1)}) & (z_j^{(l-1)})^2 & z_i^{(l-1)} & z_j^{(l-1)} & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix}$$

ماتریس $Z_k^{(l-1)}$

$$z_k^{(l-1)} \times \begin{bmatrix} a_k^{(l)} \\ b_k^{(l)} \\ c_k^{(l)} \\ d_k^{(l)} \\ e_k^{(l)} \\ f_k^{(l)} \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_L \end{bmatrix} \Rightarrow Z_k^{(l-1)} \; P_k^{(l)} = T$$

PAPCO

Subject:
Year ... Month ... Date ... ( )

معادله فوق را برای پیش تریبی خواهیم داد برای حداقل کردن MSE عبارت است از:

$$P_{ik}^{(e)} = \left(Z_{ik}^{(e-1)}\right)^{+} T$$

برای آنکه بدانیم بزرگی اندازه مجموعه آموزش کافی است یا نه می‌توانیم به این (با یک) مجموعه، پارامتر را با اندازه مجموعه را مثلاً ۲ برابر کنیم و مجدداً پارامتر را محاسبه کنیم. اگر تغییرات پارامتر زیاد نباشد، مجموعه اول کافی است.

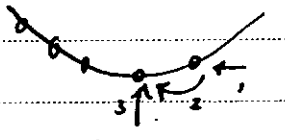
به تعیین پارامتر برای هر واحد، عملکرد هر واحد را در رسیدن به هدف حداقل کردن MSE یک مجموعه آموزش جریمه ارزیابی می‌شود. اگر یک نمونه از این مجموعه را باشد مقدار

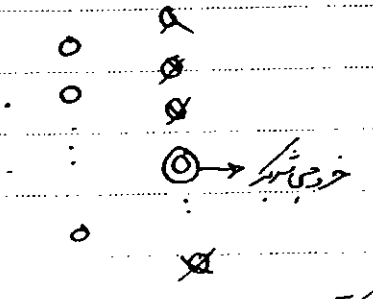
$$\frac{1}{2} \left\| Z_{ik}^{(e-1)} P_{k}^{(e)} - T \right\|^2$$

را می‌سنجیم. اگر این خطا برای یک واحد از یک مقدار آستانه بزرگتر باشد واحد مزبور را حذف می‌کنیم. با این روش، واحد یکی پس از دیگری در لایه (ای) می‌شوند تا جایی که هیچ کدام توقف رخ دهد.

اگر در لایه کمترین MSE را در نظر بگیریم، این مقدار با افزایش لایه (تا حدی) کاهش می‌یابد و پس از آن افزایش پیدا می‌کند. در آخرین مرحله قبل از افزایش، لایه جدید را متوقف می‌کنیم.

کمترین خطای بین لایه را با خط‌چین نشان می‌دهیم.





خروجی شبکه

لایه خروجی شبکه

در لایه خروجی، غیر از آن واحدی که کمترین خطا را داشته، البته واحد حذف می‌شود. حال، جدت (ن)

Subject:

Year. Month. Date. ( )

واحد (، بطوری است واحد دین در لایه قبل بوده ، شبکه به دیگر اتصالی به واحد جریمه نداشته ، بشبه
بدین ترتیب آن واحد نیز حذف می شود .

این شبکه ها را دید چه چیزی را دارد ؟ هان ؟ بگو دیگه ! چون فل بگو !

شبکه ای که ۲ ورودی می گیرد و معادله درجه ۲ ای برای ما تشکیل می دهد :

بهینه سازی

یک تابع هزینه تشکیل می دهیم و به حداقل کردن آن حل هزینه مسئله را پیدا می کنیم .

روش های سنتی بهینه سازی : روش نیوتن لاگرانژ ، برنامه ریزی خطی ، غیر خطی و ...

استفاده از شبکه های عصبی : در سال های که روش های سنتی جواب حل نمی کنند این روش ها جواب
حل می کنند : مثلاً در جایی که بهینه سازی با وجود محدودیت های قید فضا
انجام شود .

شبکه های عصبی قطعی (Deterministic)
مشکل گیر افتادن در نقاط بهینه محلی

شبکه های عصبی احتمالی (Stochastic)
مشکل فوق را ندارند .

PAPCO

Subject:

Year Month Date

گرم و سرد کردن فلزات (Annealing)

۱- حرارت دادن فلز تا حالت نزدیک ذوب ← ترک و شکستگی های شبکه فلز رفع می شود

۲- سرد کردن فلز بصورت آرام و تدریجی

مجموعه حالات ممکن یک فلز $\{\alpha_i\}$

انرژی کل سیستم در حالت α_i E_{α_i}

درجه حرارت T

اگر $T > 0^{\circ}K$ (صفر مطلق) ← سیستم دارای نوسانات حرارتی خواهد بود.

انتقال از حالتی به حالت دیگر که انرژی را به طور جزئی کم یا زیاد می کند.

تعادل حرارتی: نوسانات سیستم حول مقادیر متوسط ثابتی است و از این مقادیر تغییر نمی کند.

احتمال هنگ سیستم به حالت α_i از حالت تعادل که P_{α_i} نشان می دهند.

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

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

k_B را یک ثابت کلی برابر 1 می گیریم.

P4PCO

(۵۶)

Subject :

Year . Month . Date . ( )

نسبت احتمال β حالت α_i و α_j

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

اگر $T \gg 1$ $\Rightarrow$ $\frac{P_{\alpha_i}}{P_{\alpha_j}} \to 1$

در دمای بالا احتمال رفتن سیستم به یک حالت با انرژی بالاتر بسیار احتمال رفتن سیستم به یک حالت پایین‌تر است.

بهینه‌سازی با روش تابکاری شبیه‌سازی شده (Simulated Annealing)

تابع هزینه بهینه‌سازی $\longleftrightarrow$ تابع انرژی فلز

متغیرهای مسأله $\longleftrightarrow$ ذرات فلز

قبول‌ترین انتقال با Δ تابع هزینه را کاهش می‌دهد $\longleftrightarrow$ کاهش سریع دمای حرارت

الگوریتم برای محاسبه $\longleftrightarrow$ دمای بزرگ و کندی کاهش دما

اجازه به بالا رفتن در سیستم کلی $\longleftrightarrow$ اجازه انتقال به حالت با انرژی بالاتر

یک جواب اولیه به یکی از همسایه‌های خود با احتمالی مانند P انتقال می‌یابد و الگوریتم هم اجازه انتقال به حالت با انرژی بالاتر هم به حالت با انرژی پایین‌تر را می‌دهد.

در هر قدم در یک متغیر یک تغییر اندک ایجاد می‌شود تا از حالت α_i به حالت α_j برویم

$$\Delta E = E_{\alpha_j} - E_{\alpha_i}$$

PqPCO

Subject:
Year. Month. Date. ( )

اگر $\Delta E < 0$ ⟵ انتقال قبول می‌شود، احتمال 1

و $\Delta E \geq 0$ ⟵ انتقال، احتمال $e^{-\frac{\Delta E}{T}}$ پذیرفته می‌شود.

فرض کنید می‌خواهیم تابع $f(x)$ را نسبت به بردار X راحداقل کنیم. روال تابکاری شبیه‌سازی شده برای این کار به قرار زیر است:

۱- یک بردار X را به صورت اتفاقی انتخاب می‌کنیم.

یک مقدار اولیه بزرگ برای T انتخاب می‌کنیم. (به طوریکه برای همه ΔE ها $e^{-\frac{\Delta E}{T}} > 0.999$ باشد)

۲- یک مؤلفه x_i از X را به طور اتفاقی یکنواخت انتخاب می‌کنیم.

3- مقدار مؤلفه‌ای x'_i را برابر مؤلفه‌ای x قبلی در نظر گرفته و مؤلفه x_i را برای آن

$x'_i = \bar{x}_i$ (عکس بیت)

و $\Delta E = f(x') - f(x)$ را محاسبه می‌کنیم.

4- اگر $\Delta E < 0$، x را برابر x' قبلی در نظر گرفته و 6 می‌رویم.

5- اگر $\Delta E \geq 0$، x را با احتمال $e^{-\frac{\Delta E}{T}}$ برابر x' قبلی در نظر گرفته و T را کمی کوچک می‌کنیم. (هرچه T بزرگ‌تر است احتمال نزدیک ۱ است)

برای اینکار یک عدد اتفاقی ξ بین صفر و یک، یعنی $0 \leq \xi < 1$ انتخاب می‌کنیم اگر $e^{-\frac{\Delta E}{T}} > \xi$ آنگاه x را برابر x' قبلی در نظر گرفته و در غیر اینصورت x را تغییر نمی‌دهیم.

6- اگر N تغییر متوالی در x، یعنی N تغییر در x از زمان آخرین تغییر در درجه حرارت T (متوالی یا نامتوالی)

PqPCO

Subject:

Year. Month. Date. ( )

اتفاق افتاده است، مقدار T را برابر $T \propto T$ تغییر دهیم: $T \leftarrow \alpha T$

$0.8 < \alpha < 1$, $M \ll N$

7. اگر مقدار حداقل F در ۵ تکرار آخر بیشتر از ۴ کاهش نیافت، الگوریتم را خاتمه می‌دهیم در غیر اینصورت به قدم ۲ برمی‌گردیم.

ثابت شده است که این الگوریتم همگرا می‌شود (دلیل نمی‌خواهد). این در عمل میزبت است اما به تقاضای آهنگش محدود نیز همگرایی بسیار خوبی از آن نمی‌آید و تعداد سیکل صعودی از روی توابع خوش رفتار به طور تقریبی کم است.

اعمال روش تابکاری شبیه‌سازی شده بر شبکه گسسته هاپفیلد

برای استفاده از روش SA در شبکه (فیلد) توابع تبدیل وضعیت را به شکل زیر تغییر می‌دهیم:

$$y_i^{NEW} = \begin{cases} -y_i^{OLD} & (\Delta E < 0) \cup (\Delta E \geq 0 \ , \ \zeta < e^{\frac{\Delta E}{T}}) \\ y_i^{OLD} & \text{otherwise} \end{cases}$$

در اینجا نیز در هر زمان تنها یک وضعیت را انتخاب (و می‌آزمایم). این وضعیت به طور اتفاقی و یکنواخت انتخاب می‌شود. در صورت نیز پس از تعداد زیادی دفعه انتخاب سازی، کاهش قابل توجهی در انرژی حاصل آورده می‌شود. پیچ شبکه حتی اگر به همگرایی کامل نرسد، عموماً نزدیک به پیچ بهینه است. بعضی از این شبکه‌ها (Hopfield + SA) روشهای بهتری نیز هستند.

Subject:

Year Month Date ( )

ماشین بولتزمن (Boltzman Machine)

در سال 1983 توسط هینتون و کنوفسکی جزئی گوید در این ماشین واحد دارای حالت باینری (انتقال حالات، اتصالی بستند و وزنی دارای مقدیر ثابت می باشند.

"Vanity, is definitely My Favourite Sin"

Al pacino in Devil's Advocate

وزن w_{ij} نشانه‌ای از میزان مطلوبیت فعال بودن هر زوج واحد i و j می باشد. برای استفاده در پیش‌سازی توابع کمک می‌کند پیش‌سازی و محدودیت‌ها در وزن‌ها ذخیره می‌شوند.

ماشین بولتزمن از مجموعه‌ای از واحدها و اتصالات در هم تشکیل شده است. اگر واحدهای i و j هم وصل باشند $w_{ij} \neq 0$ و $w_{ij} = w_{ji}$. اتصال به خود نیز با وزن w_{ii} ممکن وجود داشته باشد. هدف حداکثر کردن یک تابع C است به صورت زیر:

$$C = \sum_i \sum_{j \leq i} w_{ij} x_i x_j$$

تابع هزینه شبکه

حداکثر کردن تابع هزینه با استفاده از روش SA انجام می‌شود. برای هر نرون ممکن است که به واحد i اجازه تغییر حالت داده شده و در این حال تغییر C را به شکل زیر خواهیم بود:

$$\Delta C(i) = [1 - 2x_i]\left[w_{ii} + \sum_{j \neq i} w_{ij} x_j\right]$$

x_i در حال ... عدد صفر

این شرط تغییر حالت نرون را به احتمال $A(i,T)$ می‌دهد که:

$$A(i,T) = \frac{1}{1 + e^{\frac{-\Delta C(i)}{T}}}$$

در هر حرارت T تدریجاً کاهش می‌یابد. عملیات شبکه دیگر تغییر نمی‌کند و نرون‌ها به توجیه کامل تغییر می‌کنند.

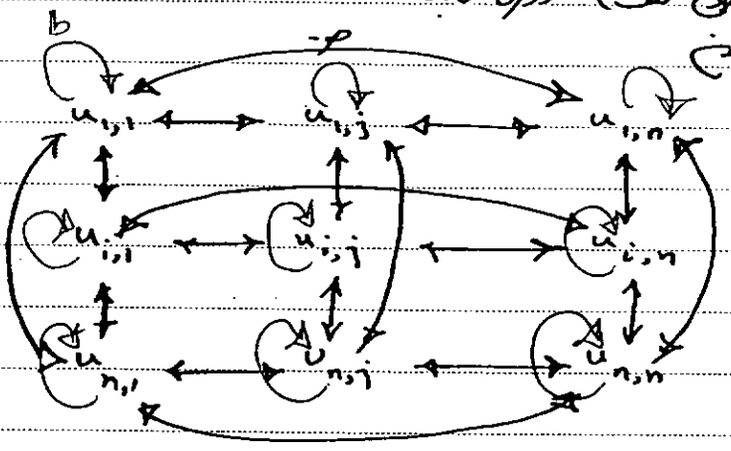
PAPCO

Subject:

Year. Month. Date. ( )

معماری:

هر واحد به صورت یک آرایه دو بعدی پیاده شده. اتصالات کامل بین همه واحدها برقرار است (وزن‌های مختلف) وزن ثابت



وزن تمام اتصالات: $-p$

وزن خود اتصالی: b

الگوریتم شبکه

وزن بین واحد (i,j) با واحد (I,J) را $w(i,j;I,J)$ نشان می‌دهیم.

$w(i,j;i,j) = b$ $\quad w(i,j;I,J) = -p$ برای $i=I$ و $j=J$ (این همپوشانی دارد)

شبکه به صورت آرایه $n \times n$

θ: مقادیر اولیه وزن‌ها مشخص می‌کنیم به طوری که حداکثر دامنه‌ای مساوی را داشته باشد.

پارامتر کنترل T را مقدار دهی اولیه می‌کنیم. (T_0)

مقادیر حالات اولیه را به صورت اتفاقی برابر 0 یا 1 انتخاب می‌کنیم.

۱: تا زمانی که شرط خاتمه غلط است قدم‌های 2 الی 5 را تکرار می‌کنیم:

Subject:
Year. Month. Date. ( )

۲- تعریف ی 3 تا ی 6 n^2 بار انجام می دهیم (epoch = بار n^2)

3- مقادیر صحیح I و J را به طور تصادفی بین ۱ و n انتخاب می کنیم.

4. تغییر ورود یا خروج را محاسبه می کنیم:

$$\Delta C = [1 - 2X_{IJ}][W(I,J;I,J) + \sum\sum_{i,j \neq I,J} W(i,j;I,J)\, X_{ij}]$$

5. محاسبه احتمال قبول تغییر حالت

$$A(T) = \frac{1}{1 + e^{\frac{-\Delta C}{T}}}$$

6. تصمیم گیری در مورد قبول یا رد تغییر حالت ی یک عدد تصادفی بین ۰ و ۱ انتخاب می کنیم اگر $A > \xi$ ی تغییر حالت را قبول می کنیم $X_{IJ} = 1 - X_{IJ}$
وگرنه $A \leq \xi$ ی تغییر حالت قبول نمی شود.

7. کاهش پارامتر T

$$T_e = 0.95\, T_{e-1}$$

8. شرط خاتمه را آزمایش می کنیم.

اگر برای تعدادی epoch پیش تغییر حالتی رخ نداده و T به یک مقدار حداقل رسیده باشد توقف الگوریتم و غیر این صورت ادامه می دهیم (مقدار اولیه T باید در حدی بزرگ باشد و احتمال قبول حدود 0.5 شود)

کاربرد: مسأله فروشنده دوره گرد (TSP)

n : تعداد شهرها (مثلاً 10)

i : اندیس نشان دهنده شهر $(1 \le i \le n)$

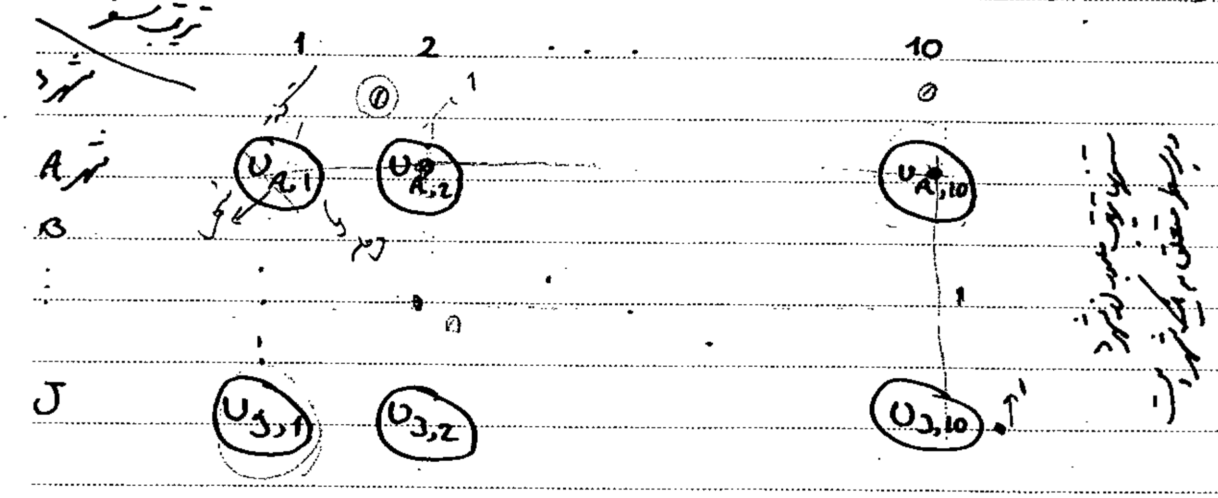
j : " " ترتیب سفر $(\mathrm{mod}\ n)$

$V_{i,j}$: واحدی که فرض گذر از شهر i ام در مرحله j ام از سفر را نشان می‌دهد.

$v_{i,j}$: فعالیت واحد $V_{i,j}$ یعنی
- $v_{i,j}=1$ فرض $v_{i,j}$ صحیح است.
- $v_{i,j}=0$ " " " نیست.

$d_{i,k}$: فاصله بین شهر i و شهر k $(i \neq k)$ ، d : حداکثر فاصله بین هر دو شهر

معماری شبکه



- هر واحد $V_{i,j}$ یک خود اتصال با وزن b خواهد داشت و یک اتصال با همه واحدهای سطر خود با وزن $-p$

خود اتصالی $v_{i,j}$ مطلوبیت عبور از شهر i در مرحله j را نشان می‌دهد.

- اتصال هر واحد با سایر واحدها در یک سطر این محدودیت که از هر شهر نباید بیش از یک بار عبور کرد را نشان می‌دهد.

- " " " سایر واحدهای ستون خود با وزن $-p$ ارائه دهنده‌ی این محدودیت است که از دو شهر نمی‌توان در یک مرحله عبور کرد.

- واحد $V_{i,j}$ به واحد $V_{k,j+1}$ برای $1 \le k \le n$ ، $k \neq i$ با وزن $-d_{i,k}$ وصل است. این اتصال فاصله‌ی طی شده از شهر i در مرحله j تا شهر k در مرحله‌ی $j+1$ را ارائه می‌دهد.

Subject:

Year ... Month ... Date ... ( )

- واحد $U_{i,j}$ به واحد $U_{k,j-1}$ برای $1 \le k \le n$ ، $k \neq i$ ، وزن $w_{i,k}$ له - وصل است.
این اتصال نشان می‌دهد که اطلاعات طی شده از نرون k در مرحله $j-1$ به نرون i در مرحله j را ارائه می‌دهد.

- مقدار تابع تشخیص نقطه b با شیب $p > b$ تابع آزاد، برای یک راه حل قابل قبول مقدار بزرگتری نسبت به b برای یک راه حل غیرقابل قبول.

- همچنین اگر $b > 2d$ مقدار تابع آزاد برای یک راه حل قابل قبول کوتاه‌تر مقدار بزرگتری نسبت به d برای یک راه حل قابل قبول طولانی‌تر

مثال حدی بالا:

A = (0.4 0.4439) B = (0.2439 0.1463)

C = (0.1707 0.2293) D = (0.2293 0.7610)

E = (0.5171 0.9414) F = (0.8732 0.6536)

G = (0.6879 0.5219) H = (0.8488 0.3609)

I = (0.6683 0.2536) J = (0.6195 0.2634)

	A	B	C	D	E	F	G	H	I	J
A	0									
B	0.3361									
C	0.3141	0.1107								
D	0.3601	0.6149	5049							
E	0.5111	8907	7919	3397						
F	0.5176	8683	8207	8207	4579					
G	2982	5815	5941	5941	4529	2274				
H	9564	6418	6908	6908	6632	2937	2277			
I	3289	4318	4932	4932	7042	4494	2690	2100		
J	2392	3934	4501	4501	6857	4654	2674	2492	0498	

PqPCO

Subject:

Year. Month. Date. ( )

انتخاب مقادیر پارامترها $b=60$ $p=70$ $T_0=20$ $T_j=0.9T_{j-1}$

۱۰۰ حالت مختلف تولید، حدود نصف مواقع در هر یک، آزمایش گمراه است.

در بهمه این حالات راه حل قابل قبول در کمتر از 20 اپک حاصل شد.

بهترین پاسخ بدست آمده:

GFDEACBJIA 3036 مسافت:

DAIJGFHECB 3.713 ~ ← جواب بعدی

با مقادیر کوچکتر برای b و p ($b=30$, $p=35$) آزمایش تکرار شد. با صد مقدار اولیه مختلف هیچ راه حلی با مسافت کوچکتر از راه حل بالا پیدا نشد. (جواب همگرایم. اما مسافت (طولانی تر)

با مقادیر باز هم کوچکتر برای p و b ($b=6$, $p=7$) آزمایش باز هم تکرار شد. با همه مقادیر اولیه مختلف هیچ راه حل قابل قبول پیدا نشد.

نظریه تشدید تطبیقی (Adaptive Resonance Theory)

۱۹۸۷ توسط گروسبرگ و کارپنتر معرفی شد شبکه ART1

ART2 و ART3 گسترش یافت.

ART1 : برای ورودی های باینری

ART2 : ~ ~ ~ پیوسته

ART3 : شبکه ای اصلاح شده ی ART2

Subject:
Year Month Date

۱/ مسئله: پایداری ↔ پذیرش از نمونه‌های جدید

Stability
- plasticity
Dilemma

2/ مسئله دیگر: چند واحد در لایه خروجی داشته باشیم!

ثابت شده: اگر قدرت از داده‌های جدید بگیریم، از کل مجموعه داده‌ی جدید به یک خوشه جدید متعلق نباشد؟ بله!

راه حل: واحدهایی غیرفعال در واحد خروجی در نظر داشته باشیم که در صورت نیاز فعال کنیم

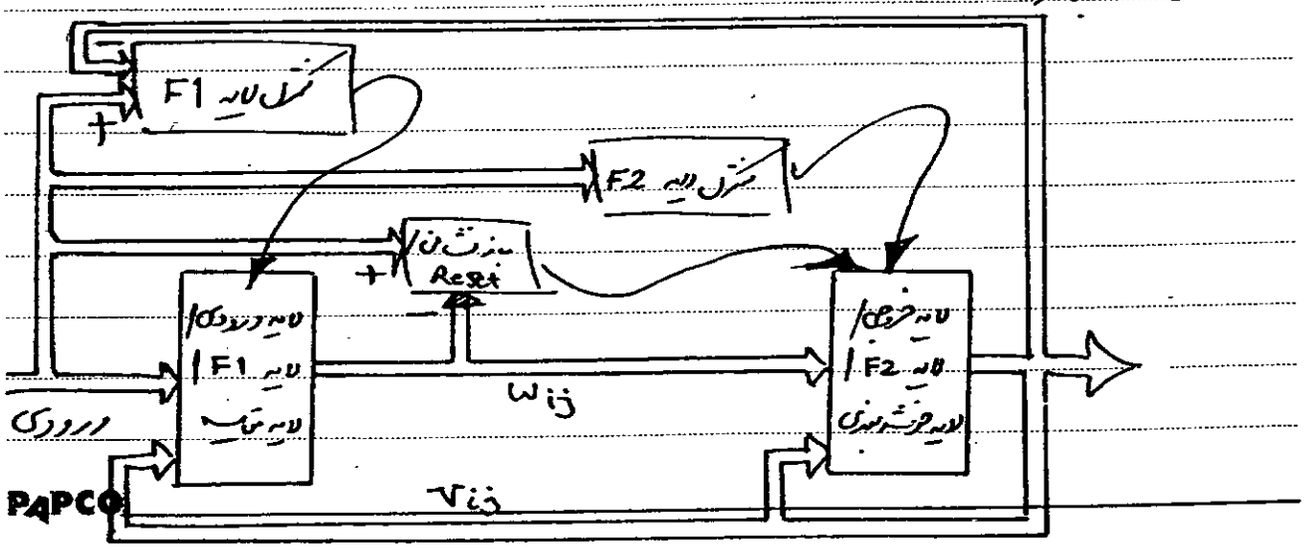
* شبکه ART1

ویژگی عمده این شبکه: تعادل در تغییر حالت بین دو حالت تأثیرپذیری و پایداری

تأثیرپذیری: انجام یادگیری و تغییر پارامترهای شبکه

پایداری: این حالت به یادگیری شبکه کمک و دسته‌بندی انجام می‌شود.

- معماری شبکه



PAPCO

(91)

Subject:

Year. Month. Date. ( )

خوشه‌بندی شبکه

چارچوب:

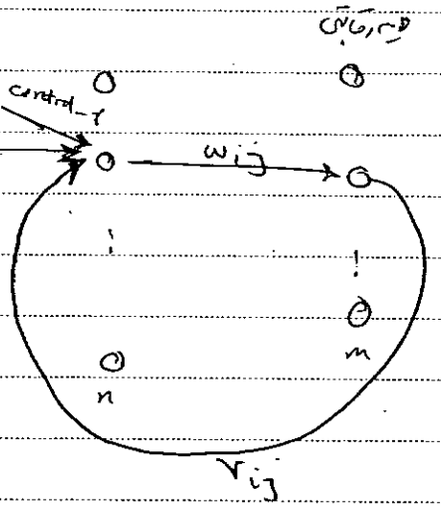
۱- تعیین معیار تشابه: فاصله اقلیدسی مشخص می‌شود. $Y_{ij} = 1$

$Y_{ij} = \frac{1}{1+n}$ " " بیشتر "

؟ آستانه مراقبت (Vigilance threshold) $0 < \rho < 1$

با فعال شدن خوشه جدید، سیگنال control_1 به ۱ می‌شود و به F1 می‌رود. در حالت عادی قادر می‌دهد یا اگر هیچ‌یک از واحدهای F1 به خوشه فعال باشند، اجباراً control_1 به صفر می‌شود.

با فعال شدن خوشه جدید control_2 = 1 می‌شود و واحدهای خوشه فعال می‌کند. برای ؟ یک آستانه مراقبت سفت، control_2 = 0 می‌شود و واحدهای خوشه را غیرفعال و خوشه‌های آن را صفر می‌کند.



۲- خوشه‌بندی

هر واحد در لایه ورودی n ورودی دارد

بردار داده‌ها از لایه ورودی با تعداد $2/3$ کنترل می‌شود یعنی اگر n ورودی از n ورودی غیر صفر باشند ← خروجی واحد = 1

در غیر اینصورت $= 0$

Subject:
Year. Month. Date.

لایه F2 یک لایه رقابتی است. برداری ورودی به این لایه با الگوی ذخیره شده در وزنهای واصدی لایه F2 مقایسه می‌شود و برترین تطبیق مشخص می‌شود.

برترین تطبیق، کاسه به ضرب داخلی اتصالات جایی بازگشتی بین واحدی لایه F2 هر واحد جز واحدی که دارای قویترین پاسخ است را خاموش می‌کند.

با فعال شدن خروجی واحد برنده وزنهای اتصالات بازگشتی به F1 وارد می‌شود.

C - بازه‌ها

یک خروجی لایه F2 فعال است به سیگنال control 1 صفر می‌شود.

دو بهره واحد قابل، 2/3 اعمال می‌شود که در حقیقت برابر ورودی خارجی و ورودی از خروجی لایه F2 را با هم AND می‌کند. نتیجه این عمل در خروجی F1 ظاهر می‌شود.

این برداره و برداره ورودی به بخش Reset وارد می‌شود. در این بخش آستانه مراقبت ρ سنجیده می‌شود

$$r = \frac{\sum V_{ji} X_i}{\sum X_i}$$

(تعداد 1 های مشترک در الگوی ذخیره شده و ورودی)

تعداد 1 های ورودی

بیانگر میزان تشابه دو بردار ورودی و وزن

ρ میزان است از تشابه دو بردار

اگر $r \geq \rho$ واحد فعال لایه F2 جزء ورودی را نشان می‌دهد. خوشه‌بندی عام‌تر و وزنها اصلاح می‌شوند.

اگر $r < \rho$ برترین الگوی بعدی برای این ورودی پیدا نشده. واحد فعال غیر فعال می‌شود (موج التل) و به جستجو می‌رویم.

PqPCO

Subject:

Year. Month. Date. ( )

۴- فاز جستجو

وارد برنده مثل ← غزال ← خروجی آن صفر می‌شود ← در تطبیق‌های بعدی حرکت نکند

سیگنال control-1 صفر می‌شود.

ورودی به لایه دوم اعمال می‌شود و برترین تطبیق دوباره مشخص می‌شود.

سپس بررسی می‌کند و فاز مقایسه می‌شود و تطبیق جدید برای (رضا) آزمایش آستانه برآورده نشده است می‌شود. این فرآیند مکرراً تکرار می‌شود تا واحدهای خروجی غزال می‌شوند تا اینکه واحدی پیدا شود که در محدوده قابل قبول برای مقدار آستانه برآورده باشد و ورودی تطبیق داشته باشد. اگر چنین واحدی پیدا نشود شبکه ورودی جدید را به یک واحد جدید استفاده نشده برای خوشه جدید نسبت می‌دهد.

الگوریتم:

۱- تعیین مقادیر اولیه برای

$V_{ij}(0) = 1$ $\quad 0 \le j \le m-1$

$W_{ij}(0) = \frac{1}{1+n}$ $\quad 0 \le i \le n-1$

و تعیین مقدار ρ (معمولاً 0.9)

۲- ورودی جدید به شبکه اعمال شود.

۳- محاسبه میزان تطبیق با الگوی نمونه

$$\mu_j = \sum_{i=0}^{n-1} W_{ij}(t) X_i \qquad j = 0 \dots m-1$$

۴- تعیین برترین تطبیق

$$\mu_J = \max_j (\mu_j)$$

Subject:

Year ___ Month ___ Date ___ ( )

۵- آزمون برابری

$$R = \frac{\sum_{i=0}^{n-1} V_{ij}(t) X_i}{\sum_{i=0}^{n-1} X_i}$$

اگر $r \geq \rho$ به قدم 7 می‌رویم

وگرنه " " 6 "

۶- واحد J را غیرفعال می‌کنیم. حوزه‌ی اطرافی را Φ می‌کنیم و به قدم 3 می‌رویم.

۷- وزن‌های برنده تطبیق یا اصلاح می‌کنیم:

$$V_{ij}(t+1) = V_{ij}(t) X_i$$

$$V1_{ij}(t+1) = \frac{V_{ij}(t) X_i}{0.5 + \sum_{i=0}^{n-1} V_{ij}(t) X_i}$$

۸- واحدهای غیرفعال را فعال شده به قدم 2 می‌رویم.

این شبکه در هر زمان می‌تواند در حالت آموزش دیده و داده‌های جدید را در آموزش داشته شبکه به کار گیرد.

شبکه ART 1 در طی آموزش نسبت به تغییرات بسیار حساس است. حساس‌ترین آن نسبت به مقادیر اولیه وزن‌ها (یا کوچک‌ترین) است.

اگر مقدار ρ کوچک باشد (مثلاً 0.4) یک کلاس بزرگ (خوشه بزرگ) از ورودی‌ها این خوشه‌ها به دست آن کلاس جای گذاری می‌شوند.

یک مقدار بزرگ ρ (مثلاً نزدیک به 1) منجر به یک دسته‌بندی ریز ورودی‌ها خواهد شد

Subject :

Year. Month. Date. ( )

که در آن تغییرات جزئی نیز منجر به ایجاد دسته‌ای جدید خواهد گردید.

یک تعداد نزدیک به اکثریت جامعه مثل از جمله شبکه‌ها تغییرات بین نمونه‌ای حرکتی را می‌کند
ارادی که به این شبکه گرفته شده عکس صحیح آن در سیاریت ، داده‌های نوری بوده است.
شبکه‌ی برای راه آن ، شده و طی آزمایش م را تغییر دهیم ، یک حلقه‌ی کوچک در ابتدای کار یک دسته‌ی
خنثی اولیه ایجاد می‌شود ، و آزمایش م ، دسته‌ی بعدی به صورت دقیق‌تر انجام می‌شود.

حروفی با هنر نقاش
تا نگاه می‌کنی دقت رنگ است

گرتی

با نیم نگاه حکایت بیشتر

پیش از آنکه بخوابم
کفه‌ی ترازو تند تا گیر می‌شود

یک

ای دریغ و حیرت بیشتر
آگاهی چقدر زود می‌میرد

نصر امیر

Subject :

Year . Month . Date . ( )

PAPCO