تحلیل و طراحی سیستم ها
شبکه های عصبی مصنوعی

دکتر مهدی روانشادنیا

سرفصلهای بخش هشتم: شبکه های عصبی

الگوهای طبیعی شبکه عصبی زیستی

معرفی شبکه های عصبی مصنوعی(ANNها)

مبانی شبکه های عصبی مصنوعی

توپولوژی شبکه

فرآیند یادگیری شبکه

تجزیه و تحلیل داده ها توسط شبکه های عصبی مصنوعی

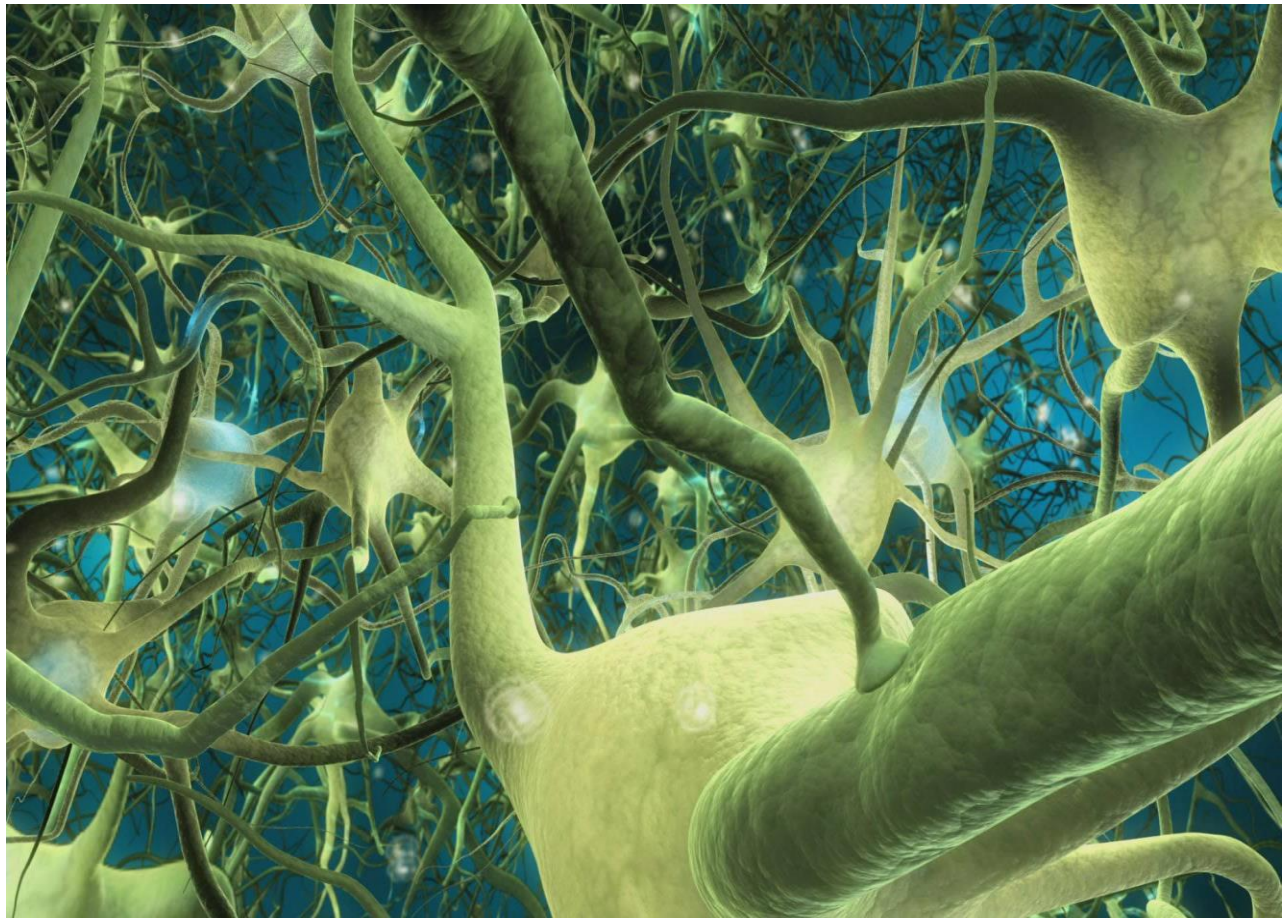
ایده ی اصلی شبکه های عصبی مصنوعی

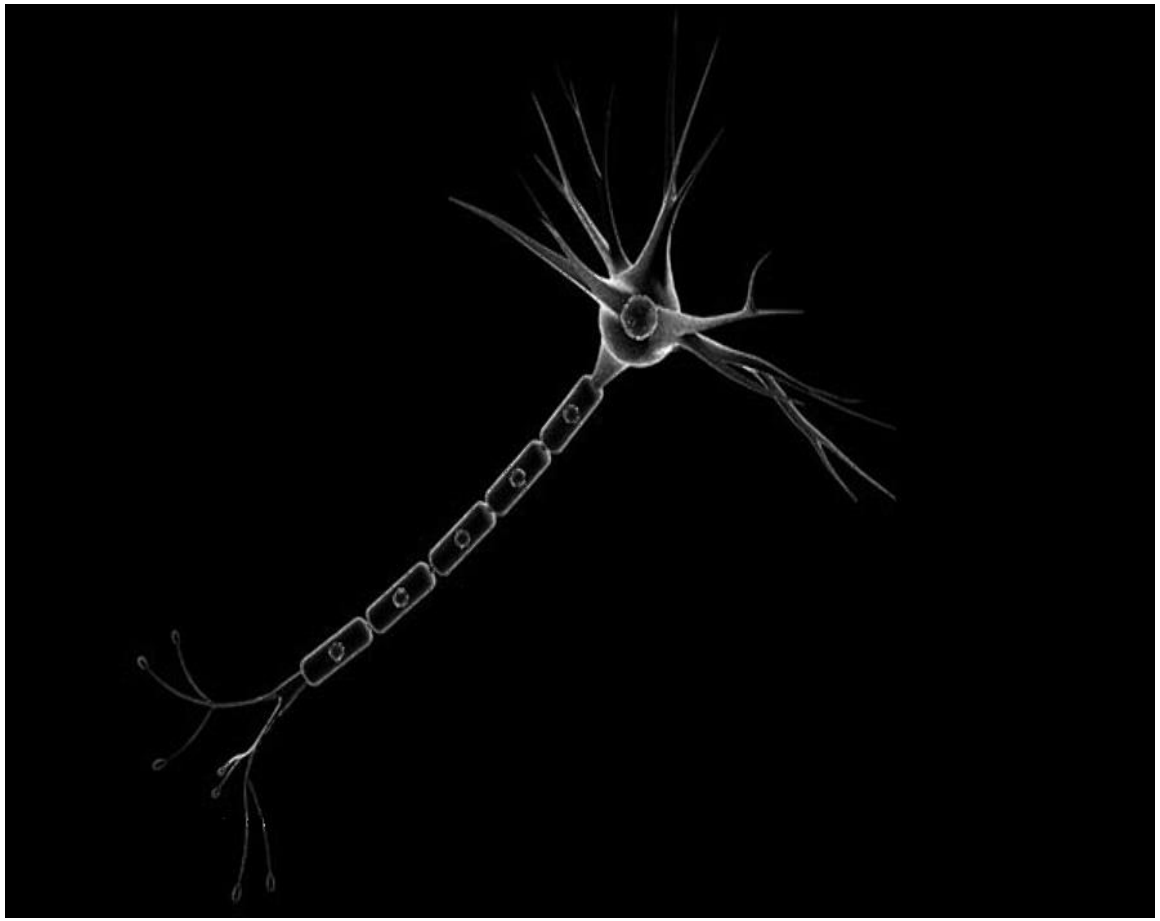
معایب شبکه های عصبی مصنوعی

کاربردهای شبکه های عصبی مصنوعی

شبکه عصبی فازی

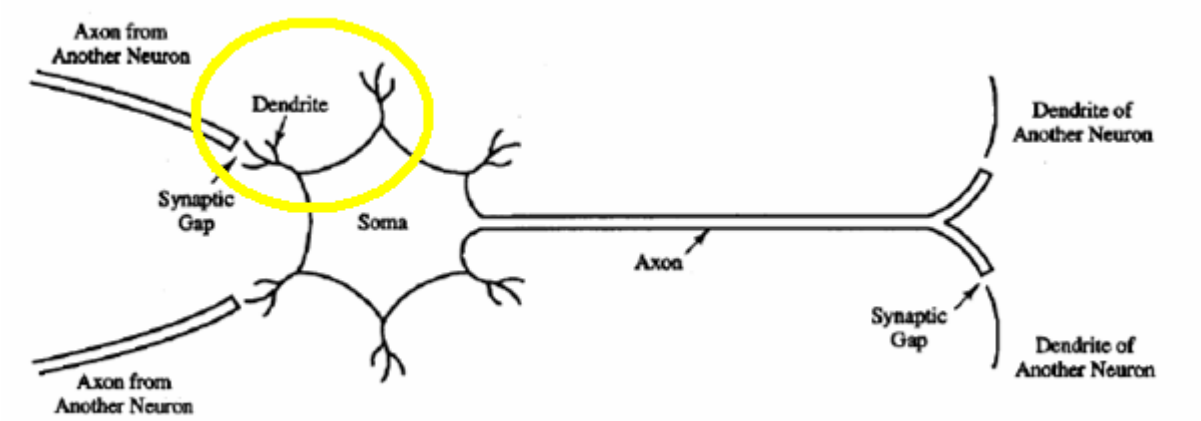
الهام از طبیعت:





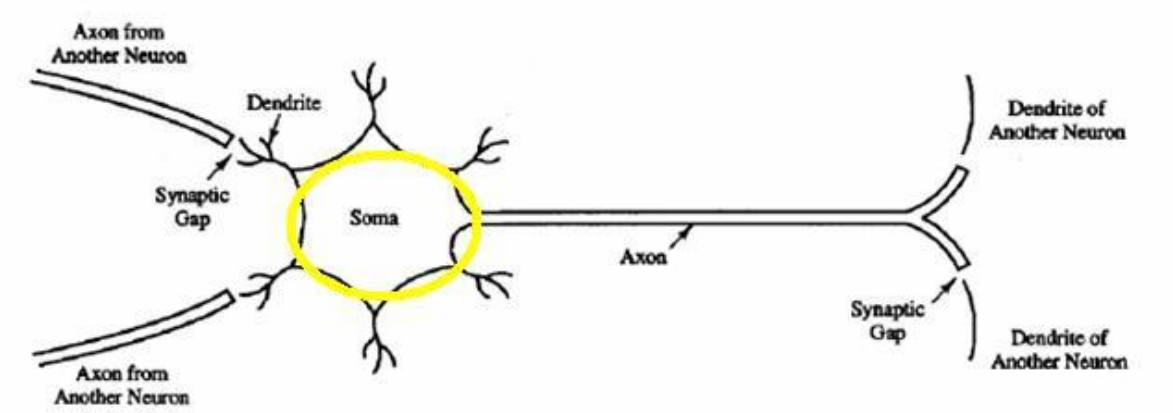
دندریت ها :

دندریت ها به عنوان مناطق دریافت سیگنال های الکتریکی هستند که دارای سطح نامنظم و شاخه های انشعابی بی شمار می باشند.



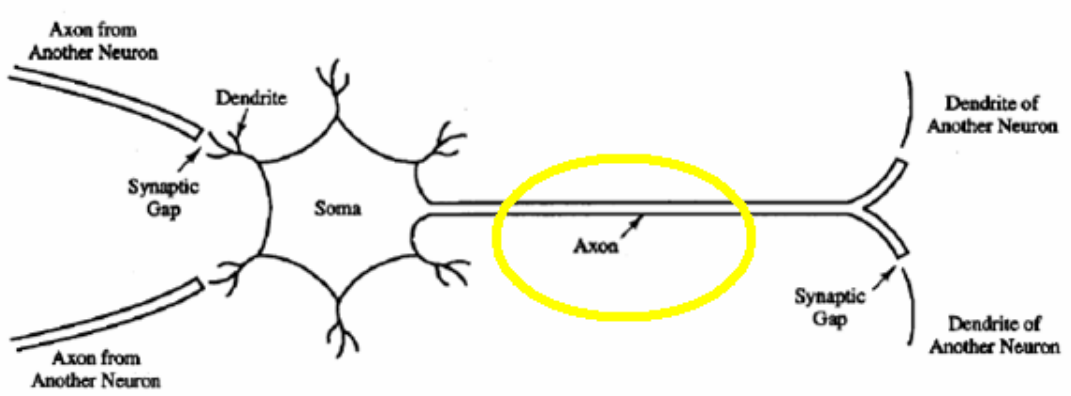
بدنه سلول:(some)

بدنه سلول ، وظیفه تامین انرژی مورد نیاز جهت فعالیت های نرون را به عهده دارد.



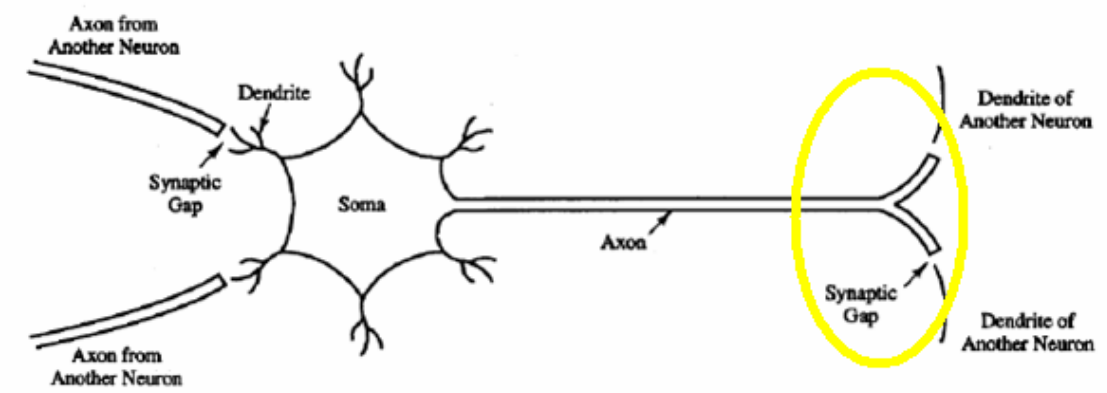
اکسون :

اکسون بر خلاف دندریت ها از سطحی هموارتر و تعداد شاخه های کمتری برخوردار می باشد. اکسون سیگنال های الکتروشیمیایی دریافتی از هسته سلول را به نرون های دیگر منتقل می کند.

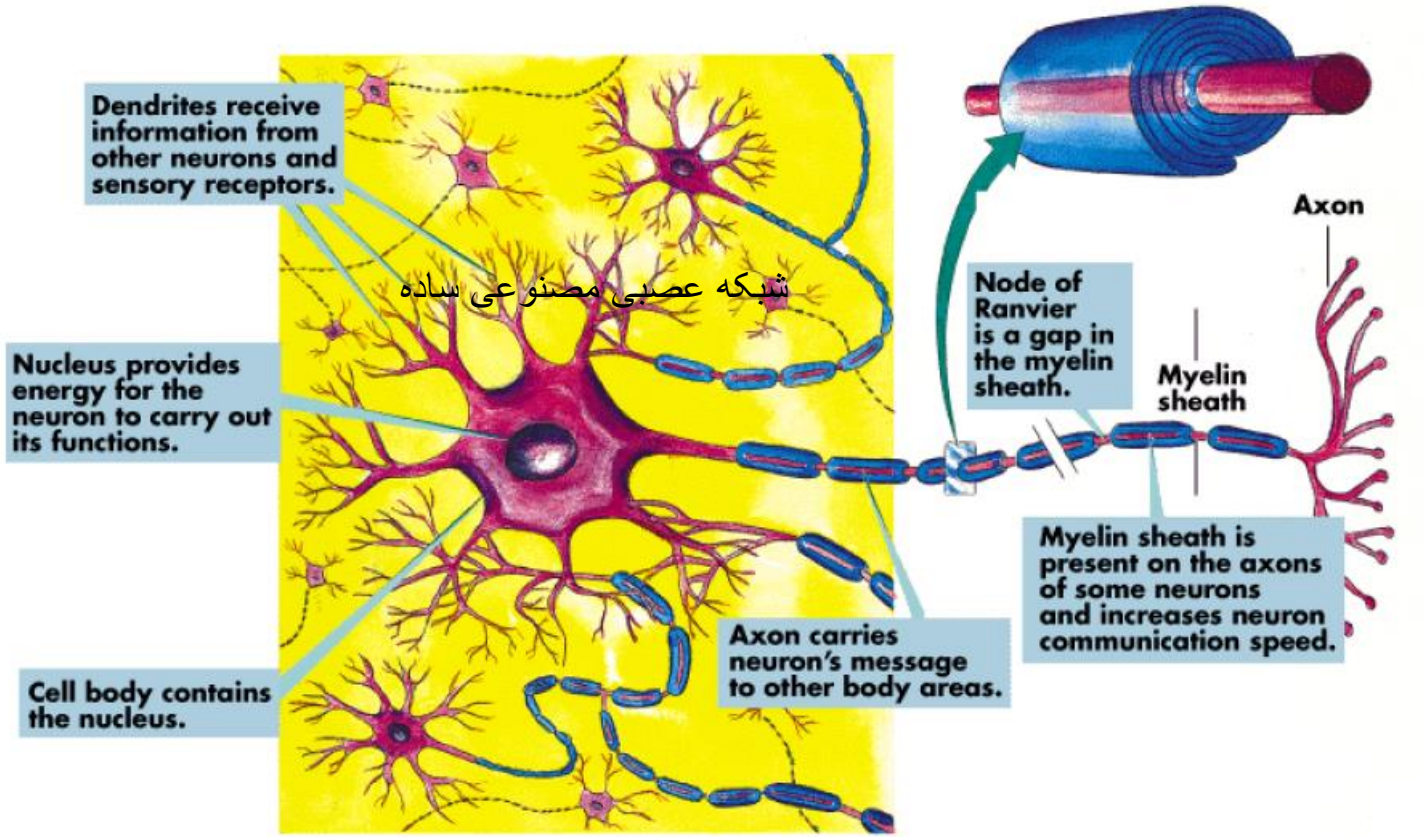


سیناپس :

محل تلاقی یک اکسون از یک سلول به دندریت های سلول های دیگر را سیناپس می گویند.
توسط سیناپس ها ارتباطات ما بین نرون ها برقرار می شود. به فضای مابین اکسون و دندریت ها فضای سیناپسی گویند.



بخشهای جزئی تر از یک شبکه عصبی:



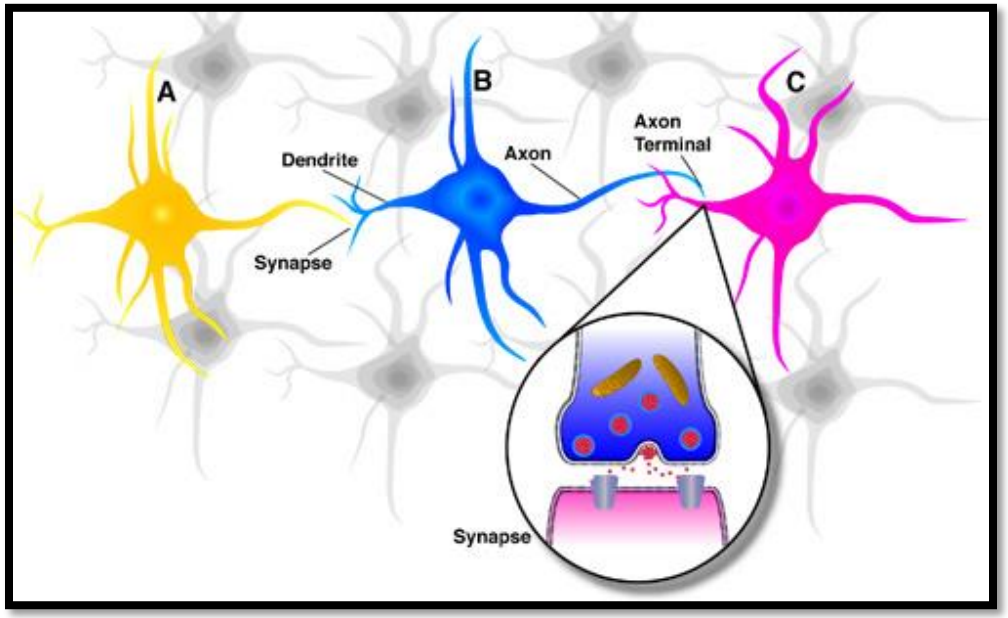
مغز انسان چگونه کار می کند؟

- مطالعه شبکه های عصبی مصنوعی تا حد زیادی ملهم از سیستم های یادگیر طبیعی است که در آنها یک مجموعه پیچیده از نرونهای به هم متصل در کار یادگیری دخیل هستند.
- گمان میرود که مغز انسان از تعداد 10^{11} نرون تشکیل شده باشد که هر نرون با تقریبا 10^4 نرون دیگر در ارتباط است.
- سرعت سوئیچنگ نرونها در حدود 10^{-3} ثانیه است که در مقایسه با کامپیوترها (10^{-10} ثانیه) بسیار ناچیز مینماید. با این وجود آدمی قادر است در 0.1 ثانیه تصویر یک انسان را بازشناسائی نماید. این قدرت فوق العاده باید از پردازش موازی توزیع شده در تعدادی زیادی از نرونها حاصل شده باشد.

آشنایی با شبکه های عصبی زیستی

- این شبکه ها مجموعه ای بسیار عظیم از پردازشگرهایی موازی به نام نورون اند که به صورت هماهنگ برای حل مسئله عمل می کنند و توسط سیناپس ها(ارتباط های الکترومغناطیسی)اطلاعات را منتقل می کنند.در این شبکه ها اگر یک سلول آسیب ببیند بقیه ی سلولها می توانند نبود آنرا جبران کرده و نیز در بازسازی آن سهیم باشند.
- این شبکه ها قادر به یادگیری اند.مثلا با اعمال سوزش به سلولهای عصبی لامسه، سلولها یاد می گیرند که به طرف جسم داغ نروند و با این الگوریتم سیستم می آموزد که خطای خود را اصلاح کند. back (propagation of error)
- یادگیری در این سیستم ها به صورت تطبیقی صورت می گیرد، یعنی با استفاده ازمثال ها وزن سیناپس ها به گونه ای تغییر می کند که در صورت دادن ورودی های جدید سیستم پاسخ درستی تولید کند.

مبانی کاربرد و عملکرد روش شبکه های عصبی مصنوعی



ساختار یک سلول عصبی:

۱- گیرنده های اطلاعات

۲- پردازشگر جزئی سلول

۳- انتقال دهنده ی داده ها

شبکه عصبی چیست؟

- **روشی برای محاسبه ارزیابی و پیش بینی است که بر پایه اتصال هدفمند و به هم پیوسته چندین واحد پردازشی کوچکتر ساخته می شود.**

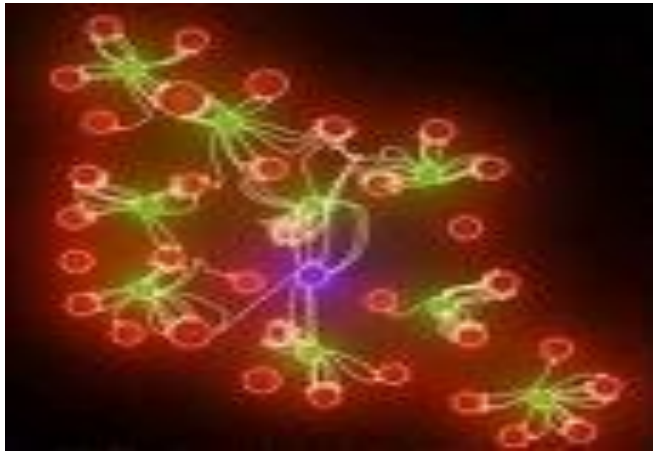
خلاصه نکات عمومی شبکه عصبی

- شبکه عصبی مصنوعی = توصیف ریاضی مشاهدات از شبکه های عصبی طبیعی مغز
- تک تک سلول های مغز به تنهایی ساده بوده و شامل دریافت کننده، پردازشگر و انتقال دهنده
- زمان پاسخگویی نورون های طبیعی ۱ میلی ثانیه، در حالیکه ترانزیستور ۱ نانوثانیه
- چه چیزی مغز را متفاوت می کند؟ تعداد سلول ها و تعداد ارتباطات(شبکه ارتباطی)
- مغز انسان ۱۰۰ میلیارد سلول که هر سلول به ۱۰ هزار سلول دیگر مرتبط است.
- چرا الهام از طبیعت؟ طبیعت به دلیل زمان زیادی که داشته به مکانیزمهای بهینه و انطباق پذیر دست یافته است.
- تفکیک، طبقه بندی

کاربردها و ویژگیهای شبکه عصبی مصنوعی

مزایای ANNها

- یادگیری انطباق پذیر
- سازماندهی توسط خود
- انجام محاسبات بصورت موازی
- تحمل اشتباه بدون ایجاد وقفه



شبکه عصبی چه قابلیتهائی دارد؟

- محاسبه یک تابع معلوم
- تقریب یک تابع ناشناخته
- شناسائی الگو
- پردازش سیگنال
- یادگیری

مسائل مناسب برای یادگیری شبکه های عصبی

- خطا در داده های آموزشی وجود داشته باشد. مثل مسائلی که داده های آموزشی دارای نویز حاصل از دادهای سنسورها نظیر دوربین و میکروفن ها هستند.
- مواردی که نمونه ها توسط مقادیر زیادی زوج ویژگی-مقدار نشان داده شده باشند. نظیر داده های حاصل از یک دوربین ویدئوئی.
- تابع هدف دارای مقادیر پیوسته باشد.
- زمان کافی برای یادگیری وجود داشته باشد. این روش در مقایسه با روشهای دیگر نظیر درخت تصمیم نیاز به زمان بیشتری برای یادگیری دارد.
- نیازی به تعبیر تابع هدف نباشد .زیرا به سختی میتوان اوزان یادگرفته شده توسط شبکه را تعبیر نمود.

ویژگیهای شبکه عصبی

به طور خلاصه یک شبکه عصبی باید خصوصیات زیر را داشته باشد:

- ❖ بتواند الگوها را طبقه بندی کند.
- ❖ به اندازه کافی کوچک باشد تا از نظر فیزیکی واقع گرایانه باشد.
- ❖ با به کار گیری آموزش، قابل برنامه ریزی باشد و قدرت یادگیری داشته باشد. یعنی توانایی تنظیم پارامترهای شبکه (اوزان سیناپتیکی)، در مسیر زمان که محیط شبکه تغییر می کند و شبکه وارد شرایط جدیدی می شود. هدف از این کار این است که اگر شبکه برای یک وضعیت خاص آموزش دید و تغییر کوچکی در شرایط محیطی شبکه رخ داد، شبکه بتواند با آموزش مختصر، برای شرایط جدید نیز کارآمد باشد. دیگر این که اطلاعات در شبکه های عصبی در سیناپس ها ذخیره و هر نرون در شبکه به صورت بالقوه از کل فعالیت سایر نرون ها تأثیر می پذیرد. در نتیجه اطلاعات از نوع مجزا از هم نبوده و متأثر از کل شبکه می باشد.
- ❖ توانایی تعمیم را با استفاده از مثال های ارائه شده در فرآیند آموزش، داشته باشد.

ویژگیهای شبکه های عصبی

۱) قابلیت یادگیری
۲) قابلیت تعمیم
۳) پردازش موازی
۴) مقاوم بودن
۵) از شبکه های عصبی در جاهایی استفاده می کنیم که یکسری داده ورودی-خروجی داریم که پدیده اش را نمی شناسیم

کاربردهای شبکه های عصبی

- استخراج ویژگی ها
- پردازش سیگنال
- یادگیری
- طبقه و خوشه بندی
- فشرده سازی
- بهینه سازی
- کنترل و شناسایی سیستم
- حافظه های انجمنی
- پیش بینی
- اقتصاد
- مدیریت
- پزشکی

سایر کاربردها

- تشخیص بیماری
- تشخیص چهره
- انواع جدید سنسورها
- پیگیری هدف
- هدایت جنگ افزارها
- شناسایی تصویر /سیگنال
- بینایی ماشین
- مدل کردن غیر خطی
- ترکیب صدا
- کنترل فرآیند ساخت
- آنالیز مالی

- اختصار سخن
- بازبینی امضا
- ارزیابی سرمایه
- پیش بینی فروشهای آینده و نیازهای محصول
- کلاسه بندی نمودارهای مشتری/بازار
- پیش بینی هوا
- پیش بینی محصول
- مدل کردن کنترل فرآیند
- تشخیص هدف
- شبیه سازی مسیر
- پیش بینی ورشکستگی

نرم افزارهای شبکه های عصبی

نرم افزارهایی برای شبیه سازی، مطالعه و تحقیق سیستمهای عصبی زیستی و گسترش شبکه های عصبی مصنوعی وAdaptive system ها .

شبیه سازها: نرم افزارهایی برای شبیه سازی رفتار شبکه های عصبی زیستی و مصنوعی که به صورت مستقل عمل می کنند و قادرند فرآیند آموزش شبکه ی عصبی را به شکل تصویری نمایش دهند.

شبیه سازهای تحقیقاتی :برای مطالعه ی الگوریتم ها و ساختارهای شبکه ی عصبی که به فهم بهتر رفتارها و خصوصیات شبکه ی عصبی کمک می کنند.(مطالعه ی ویژگی های شیمیایی و زیستی بافتهای عصبی و پالس های الکترومغناطیسی بین نورونها).

رایجترین شبیه سازهای ANNها :

SNNS(stuttgart neural network simulator),PDP++(parallel distribution processing),JavaNNS

رایجترین شبیه سازهای شبکه های زیستی:

XNBC,BNN ToolBox

نرم افزارهای شبکه های عصبی

شبیه سازهای آنالیز داده: علی رغم دسته ی اول، کاربردهای عملی شبکه های عصبی را مطالعه می کنند. استفاده از آنها نسبتا ساده است در عوض تواناییهاشان محدود است. بر روی Data miningوپیش بینی ها کار می کنند.

بعضی از آنها عبارتند از:

Microsoft Excel,Matlab

Development Environment ها:برای گسترش و آرایش شبکه های عصبی به کار می روند.

رایج ترین نرم افزارهای این دسته عبارتند از:

MathWorks NN ToolBox,GBlearn2

مقایسه ی مدل سازی کلاسیک و مدل سازی شبکه ی عصبی

مدل سازی کلاسیک:

این مدل از نخستین قدم خطای بزرگی مرتکب می شود که فقط در سیستمهای ساده (خطی یا نزدیک به خطی)قابل صرفنظر است و آن محاسبه ی شاخصهای تمایل به مرکز و پراکندگی است که به این ترتیب راهمیت فردی تک تک داده ها از بین می رود و در نتیجه سیستم قادر به کشف پیچیدگی ها نخواهد بود.

مدل سازی شبکه ی عصبی:

در این مدل هر یک از کانالهای ورودی دارای یک ضریب عددی هستند که وزن سیناپسی نامیده می شود.شدت تحریک الکتریکی در این ضریب ضرب می شود و به جسم سلولی می رسد.

اگر مجموع تحریکات وارد به جسم سلولی به حد آستانه ی خاصی رسیده باشد،نورون شلیک می کند و در مسیرهای خروجی جریان الکتریکی ثابتی را ایجاد می کند.تحریکات لایه ی ورودی به یک یا چند لایه ی واسط می رود .ادامه ی جریان تحریکات در این لایه ها طوری هدایت میشود که پیچیدگیهای تاثیرات جریان ورودی را شبیه سازی می کند .سپس تحریکات به لایه ی خروجی می روند که هدف نهایی ماست.

مدل سازی شبکه ی عصبی(ادامه...)

اگر هدف پیشگویی کمی باشد ،مجموع تحریکات آخرین عصب خروجی ،آن عدد خواهد بود.

اگر هدف طبقه بندی باشد، فعالیت یا عدم فعالیت (on یا off بودن)نورونهای لایه ی آخر نمایانگر این امر خواهد بود. مثلا شلیک نورون خروجی(فعال بودن آن)نشانگر حضور بیماری و خاموش بودن آن نشانه ی سلامتی است.

سیستم شبکه ی عصبی در فرآیند یادگیری طوری وزنهای سیناپسی را تغییر می دهد که بتواند با هر سری تحریکات ورودی (یعنی داده های هر نمونه)جریان خروجی مناسب(پاسخ R)را تولید کند.

چگونگی ریاضی این تغییر وزنها ظریفترین بخش مکانیسم عملکرد شبکه است.

مهم ترین تفاوت حافظه ی انسان و حافظه ی کامپیوتر

- یکی از مهم‌ترین تفاوت‌های حافظه انسان با حافظه کامپیوتر در نوع آدرس دهی این دو نوع حافظه می‌باشد. در حافظه کامپیوتر اساس کار بر پایه آدرس خانه‌های حافظه یا آدرس اطلاعات بر روی حافظه دائم می‌باشد. به عنوان مثال برای دستیابی به یک تصویر یا متن خاص، باید آدرس حافظه یا فایل مربوط به آن تصویر یا متن را داشته باشید. اما با داشتن خود تصویر یا متن نمی‌توانید به سادگی آدرس حافظه مربوطه را بیابید (البته به این معنی که این کار با یک قدم قابل انجام نیست، وگرنه می‌توانید تصویر یا متن مورد نظر را با تمام موارد موجود در حافظه مقایسه کرده و در صورت تطبیق آدرس را بیابید. ناگفته پیداست که انجام چنین کاری بسیار زمان بر و پر هزینه می‌باشد).

- اما به سازوکار همین عمل در ذهن انسان دقت کنید. با دیدن یک تصویر ناقص اغلب بلافاصله کامل آنرا به خاطر می‌آورید یا با دیدن تصویر یک شخص سریعا نام او را می‌گویید، یا با خواندن یک متن سریعا تمامی مطالب مربوط به آن را به ذهن می‌آورید. در واقع ذهن انسان یک نوع حافظه آدرس‌دهی شده بر اساس محتواست (Content Addressable Memory). همانگونه که از این نام مشخص است در این نوع حافظه، با دادن محتوای یک خانه حافظه، بلافاصله آدرس آن به عنوان خروجی داده می‌شود.

معایب ANNها

با وجود برتری هایی که شبکه های عصبی نسبت به سیستم های مرسوم دارند، معایبی نیز دارند که پژوهشگران این رشته تلاش دارند که آن ها را به حداقل برسانند، از جمله:

- قواعد یا دستورات مشخصی برای طراحی شبکه جهت یک کاربرد اختیاری وجود ندارد.
- در مورد مسایل مدل‌سازی، نمی‌توان صرفاً با استفاده از شبکه عصبی به فیزیک مسأله پی برد. به عبارت دیگر مرتبط ساختن پارامترها یا ساختار شبکه به پارامترهای فرآیند معمولاً غیرممکن است.
- دقت نتایج بستگی زیادی به اندازه مجموعه آموزش دارد.
- آموزش شبکه ممکن است مشکل یا حتی غیرممکن باشد.
- پیش‌بینی عملکرد آینده شبکه (عمومیت یافتن) آن به سادگی امکان‌پذیر نیست.

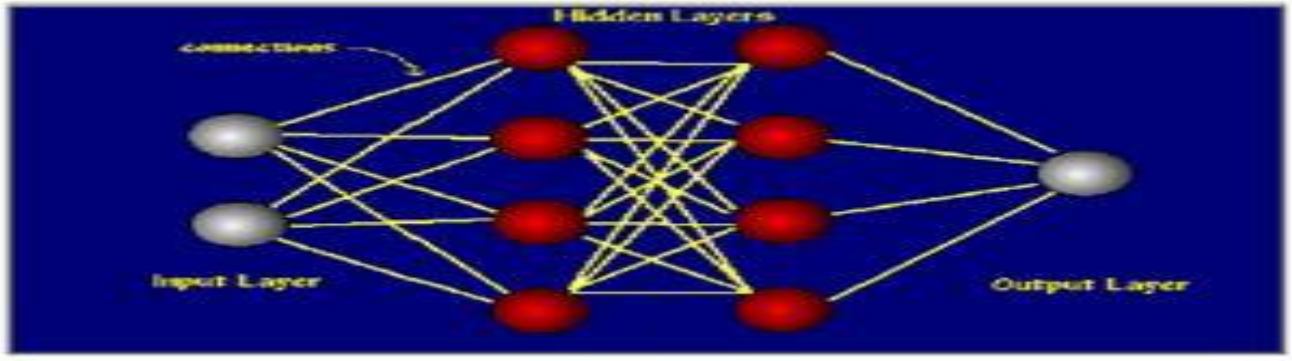
مدلسازی ریاضی شبکه عصبی

معرفی ANN ها

❖یک سیستم پردازشی داده ها که از مغز انسان ایده گرفته و پردازش داده ها را به عهده ی پردازنده های کوچک و بسیار زیادی سپرده که به صورت شبکه ای به هم پیوسته و موازی با یکدیگر رفتار می کنند تا یک مسئله را حل کنند.

❖در این شبکه ها به کمک د انش برنامه نویسی ، ساختا ر داده ای طراحی می شود که می تواند هما نند نورون عمل کند.که به این ساختارداده nodeیا گره نیزگفته می شود.بعد باایجاد شبکه ای بین این node ها و اعمال یک الگوریتم آموزشی به آ ن، شبکه را آموزش می دهند .

❖در این حافظه یا شبکه ی عصبی node ها دارای دو حالت **فعال**(on یا ۱) **وغیرفعال**(off یا ۰) اند و هر یال (سیناپس یا ارتباط بین node ها)دارای یک وزن می باشد.یالهای با وزن مثبت ،موجب تحریک یا فعال کردن nodeغیر فعال بعدی می شوند و یالهای با وزن منفی nodeمتصل بعدی را غیر فعال یا مهار(در صورتی که فعال بوده باشد) می کنند.



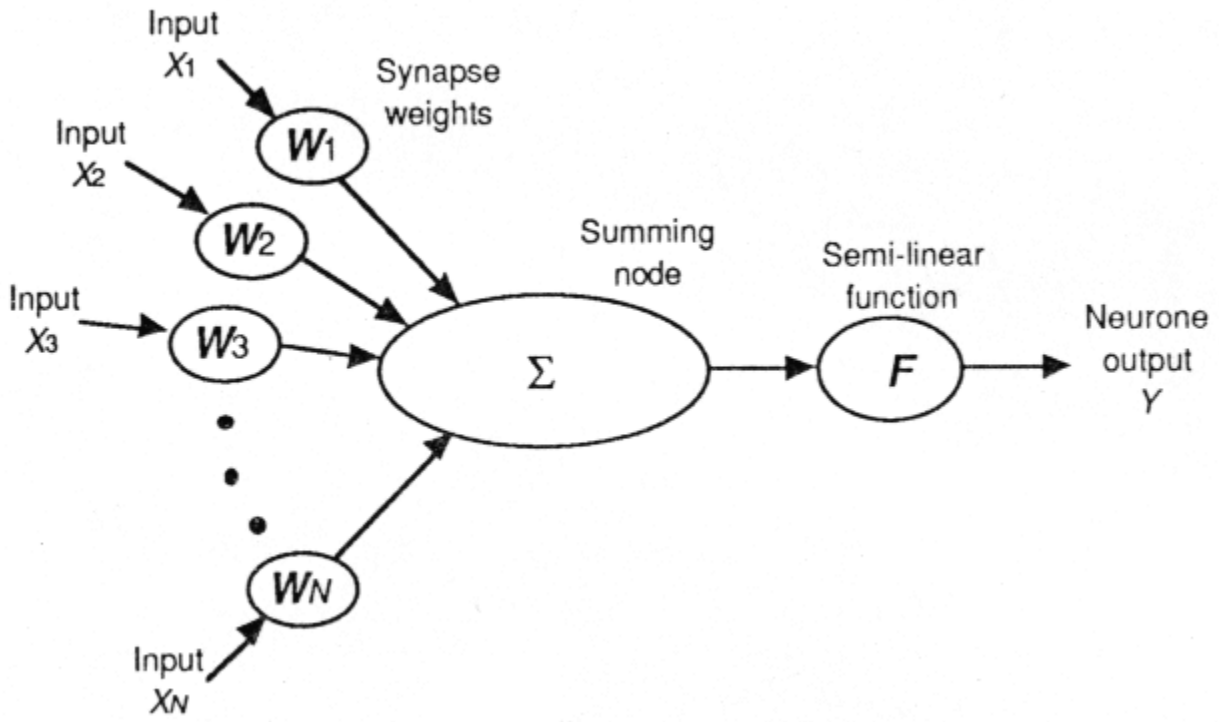
معرفی ANN ها(ادامه...)

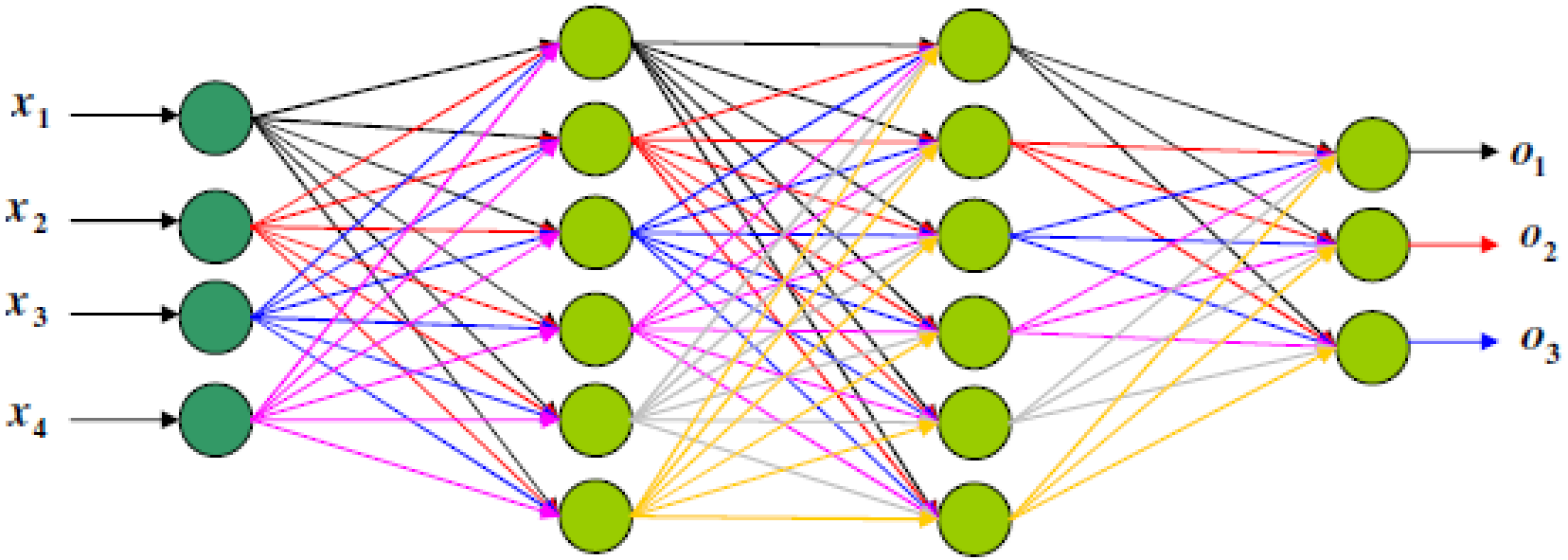
- ANN ها در واقع مثلثی هستند با سه ضلع مفهومی :
  I. سیستم تجزیه و تحلیل داده ها
  II. نورون یا سلول عصبی
  III. قانون کار گروهی نورونها (شبکه)

- ANN ها دست کم از دو جهت شبیه مغز انسا ن اند:
  I. مرحله ای موسوم به یاد گیری دارند.
  II. وزن های سیناپسی جهت ذخیره ی دانش به کار می روند.

- **هوش مصنوعی** و **مدل سا زی شناختی** سعی بر این دارند که بعضی خصوصیا ت شبکه های عصبی را شبیه سازی کنند. گرچه این دو روش ها یشان شبیه هم است، اما هدف هوش مصنوعی از این کار حل مسائل شخصی و هدف مدل سا زی شناختی ،ساخت مدلهای ریا ضی سیستم های نورونی زیستی می باشد .

توصیف ریاضی شبکه عصبی




x_1
x_2
x_3
x_4
o_1
o_2
o_3

مبانی ANN ها

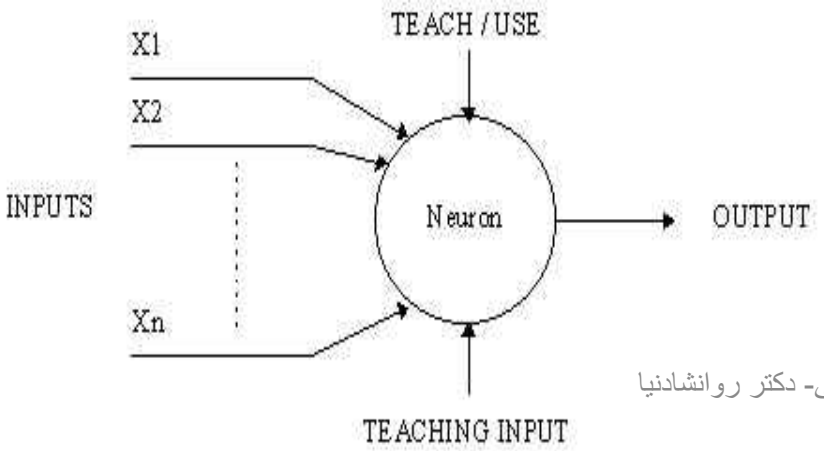
❖ شبکه های عصبی به طور کلی سیستمهای ریاضی یادگیر غیر خطی هستند. طرز کار این شبکه ها از روش کار مغز انسان الگو برداری شده است. در واقع شبکه های عصبی طبق تعریف ماشینی است برای ساخت یک مدل که می توان آن را بوسیله سخت افزار یا نرم افزار شبیه سازی کرد و عملکردی شبیه مغز انسان دارند.

❖ یک شبکه عصبی بر خلاف کامپیوترهای رقومی که نیازمند دستورات کاملا صریح و مشخص است, به مدل های ریاضی محض نیاز ندارد بلکه مانند انسان قابلیت یادگیری به وسیله تعدادی مثال مشخص را دارد.
هر شبکه عصبی سه مرحله آموزش, اعتبار سنجی و اجرا را پشت سر می گذارد. در واقع شبکه های عصبی را می توان در حل مسایلی که روابط دقیق ریاضی بین ورودی ها و خروجی های آن برقرار نیست بکار برد.

❖ آموزش دیدن شبکه های عصبی در واقع چیزی جز تنظیم وزن های ارتباطی این نرون ها به ازائ دریافت مثال های مختلف نیست تا خروجی شبکه به سمت خروجی مطلوب همگرا شود.

مدل ریاضی یک نورون

- همان گونه که ذکر شد نرون کوچکترین واحد یک شبکه عصبی مصنوعی است که عملکرد شبکه های عصبی را تشکیل می دهد.
- بدنه هر سلول عصبی از دو بخش تشکیل می شود, بخش اول را تابع ترکیب می گویند. وظیفه تابع ترکیب این است که تمام ورودی ها را ترکیب و یک عدد تولید می کند. در بخش دوم سلول تابع انتقال قرار دارد که به آن تابع تحریک نیز می گویند. درواقع همان گونه که یک سلول بیولوژیک باید به سطح آستانه تحریک خاصی برسد تا یک سیگنال تولید کند, توابع تحریک نیز تا زمانی که ورودی های ترکیب شده و وزن دار شده به یک حد آستانه ای خاص نرسند مقدار خروجی نظیر بسیار کوچکی تولید میکنند.
- وقتی ورودی های ترکیب شده به حد آستانه ای خاصی برسند, سلول عصبی تحریک شده و سیگنال خروجی تولید می کند. با مقایسه جواب خروجی شبکه با مقدار مطلوب مورد نظر بردار خطا محاسبه شده و این بردار با استفاده از الگوریتم های مختلف از آخر به سمت ابتدای شبکه پخش شده, به طوری که درسیکل بعد خطا کاهش یابد.



مراحل ساخت شبکه

- آموزش
- اعتبار سنجی
- اجرا

توپولوژی شبکه

وضعیت نسبی سلولها در شبکه(تعداد و گروه بندی و نوع اتصالات آنها)را توپولوژی شبکه گویند.توپولوژی در واقع سیستم اتصال سخت افزار نورونها به یکدیگر است که توام با نرم افزار مربوطه (یعنی روش ریاضی جریان اطلاعات و محاسبه ی وزنها)نوع عملکرد شبکه ی عصبی را تعیین می کند.

در این توپولوژی یک لایه ی ورودی وجود دارد که اطلاعات را دریافت می کند،تعدادی لایه ی مخفی وجود دارد که اطلاعات را از لایه های قبلی می گیرند و در نهایت یک لایه ی خروجی وجود دارد که نتیجه ی محاسبات به آنجا میرود و جوابها در آن قرار میگیرند.

FeedForward topology

Recurrent topology

انواع روشهای شبکه ی عصبی مصنوعی از رویکرد شیوه ی حل مسئله:

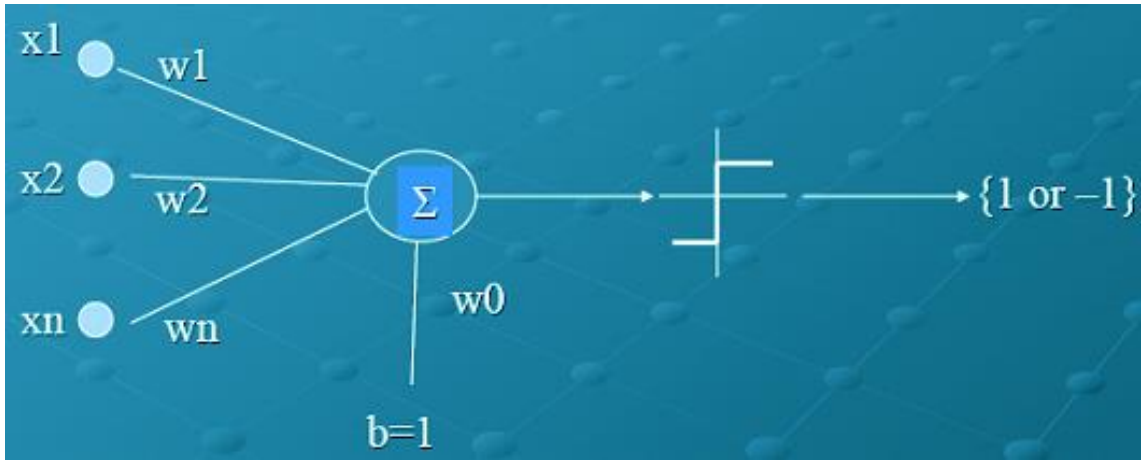
- شبکه های عصبی پرسپترون چند لایه (MLP)
- شبکه های عصبی مبتنی بر توابع پایه ای شعاعی (RBF)
- ماشین های بردار پشتیبان (SGM)

شبکه های عصبی ساده

شبکه عصبی مصنوعی ساده :

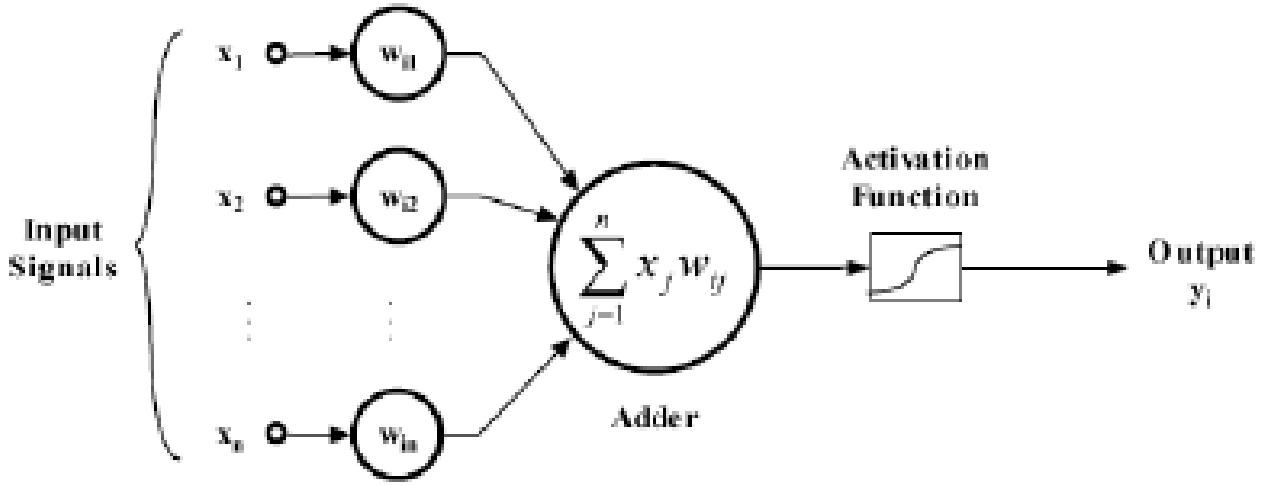
نوعی از شبکه عصبی برمبنای یک واحد محاسباتی به نام پرسپترون ساخته می شود:

- ✓پرسپترون برداری از ورودیهای را در یافت می کند .
- ✓ترکیب خطی از این ورودیها را محاسبه میکند.
- ✓اگر حاصل از یک مقدار آستانه بیشتر بود آتش می کند.



شبکه عصبی مصنوعی ساده (ادامه)

سیگنال های ورودی X1 تا Xn معادل سیگنال ها ی عصبی ورودی و وزن های W1 تا Wn معادل مقادیر اتصالات سیناپسی ورودی های نرون می باشند که جمعا ورودی های نرون را تشکیل داده است.

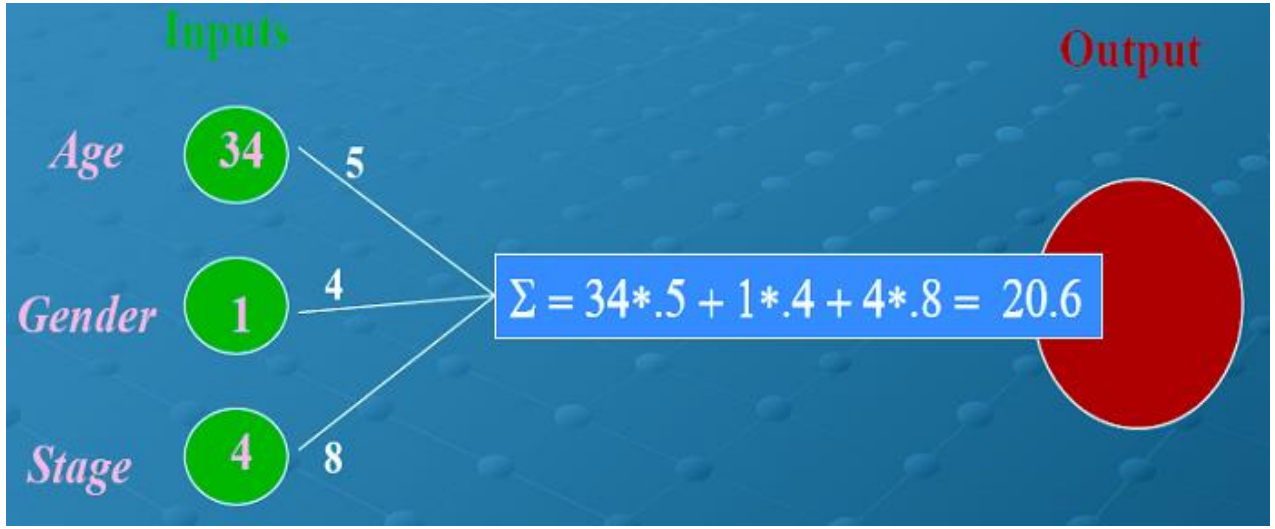


خروجی پرسپترون توسط رابطه زیر مشخص میشود :

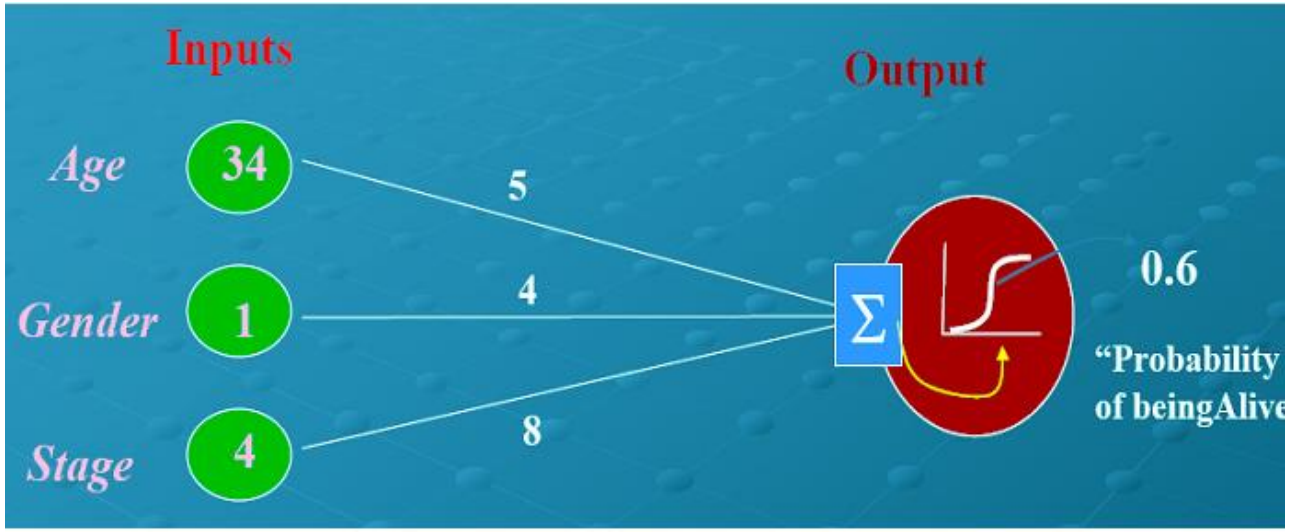
$$y_i = ActivationFunction(\sum_{j=1}^{n} x_i w_{ij})$$

$$\begin{cases} 1 \text{ if } w0 + w1x1 + w2x2 + \ldots + wnxn > 0 \\ -1 \text{ otherwise} \end{cases}$$

مثال:



نقش تابع در خروجی شبکه :



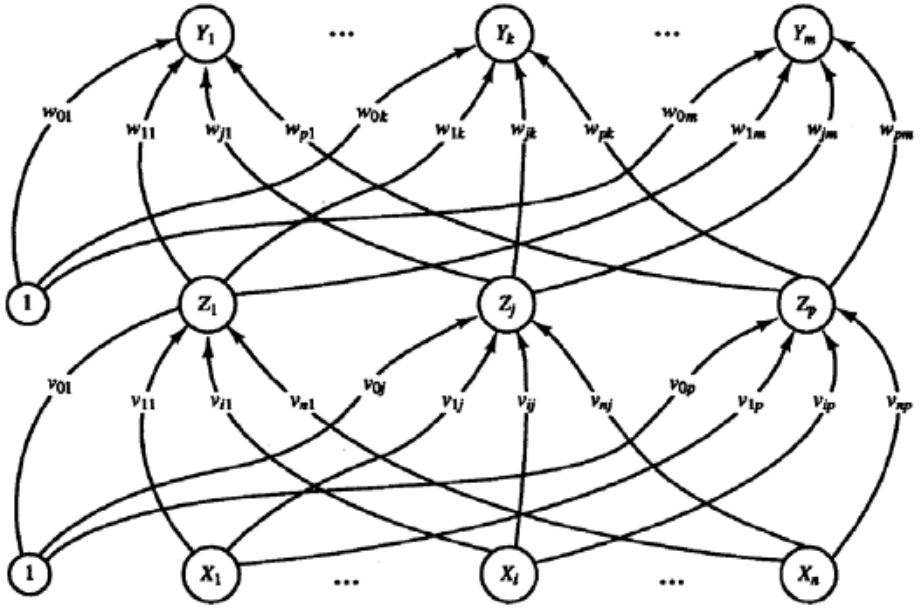
یادگیری یک پرسپترون ؟

یادگیری پرسپترون عبارت است از:

پیدا کردن مقادیردرستی برای W

شبکه های پرسپترون چند لایه :

شبکه های پرسپترون از یک لایه ورودی ، تعدادی لایه پنهان و یک لایه خروجی تشکیل شده است. در شکل زیر یک شبکه پرسپترون با یک لایه پنهان نشان داده شده است.



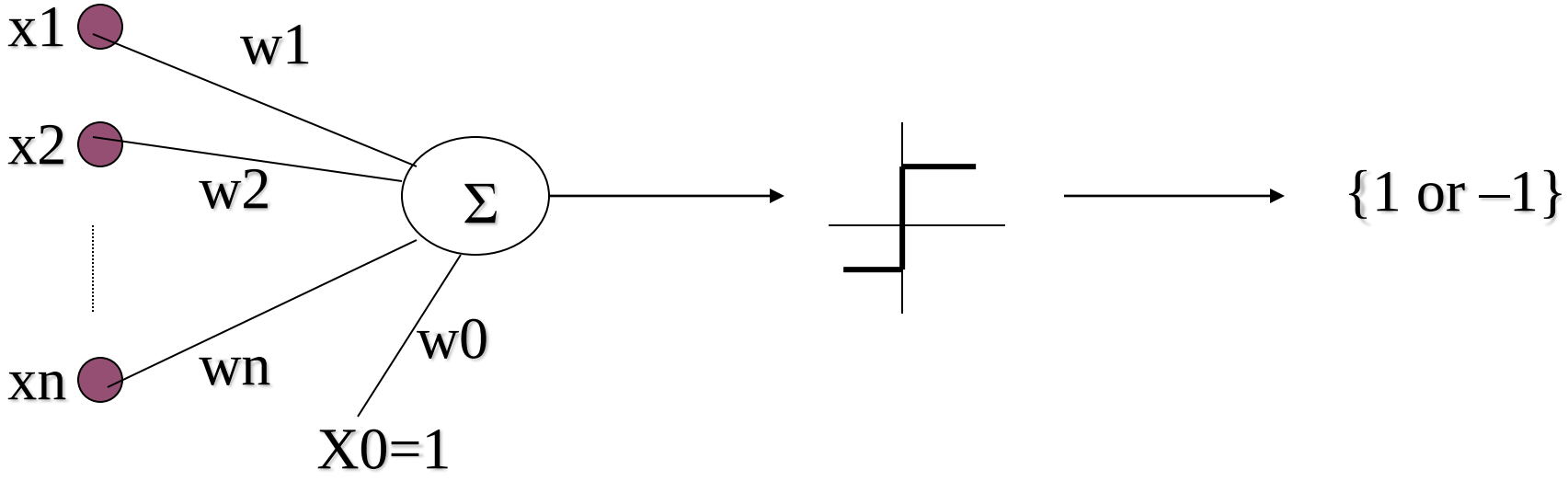
یک نکته :

در شبکه های پرسپترون چند لایه ، تعداد لایه های پنهان می تواند هر تعداد باشد. البته در بیشتر کاربردها یک لایه پنهان کفایت می کند . در بعضی مواقع نیز دو لایه پنهان یادگیری شبکه را ساده تر می کند.

شبکه های چند لایه

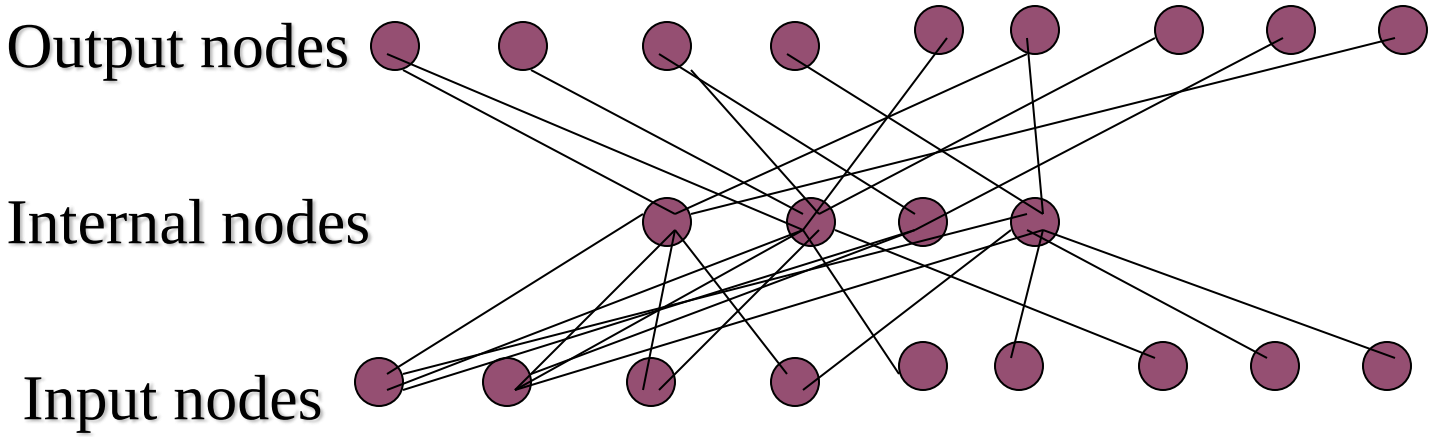
Perceptron

- نوعی از شبکه عصبی برمبنای یک واحد محاسباتی به نام پرسپترون ساخته میشود .یک پرسپترون برداری از ورودیهای با مقادیر حقیقی را گرفته و یک ترکیب خطی از این ورودیها را محاسبه میکند .اگر حاصل از یک مقدار آستانه بیشتر بود خروجی پرسپترون برابر با 1 و در غیر اینصورت معادل-1خواهد بود.

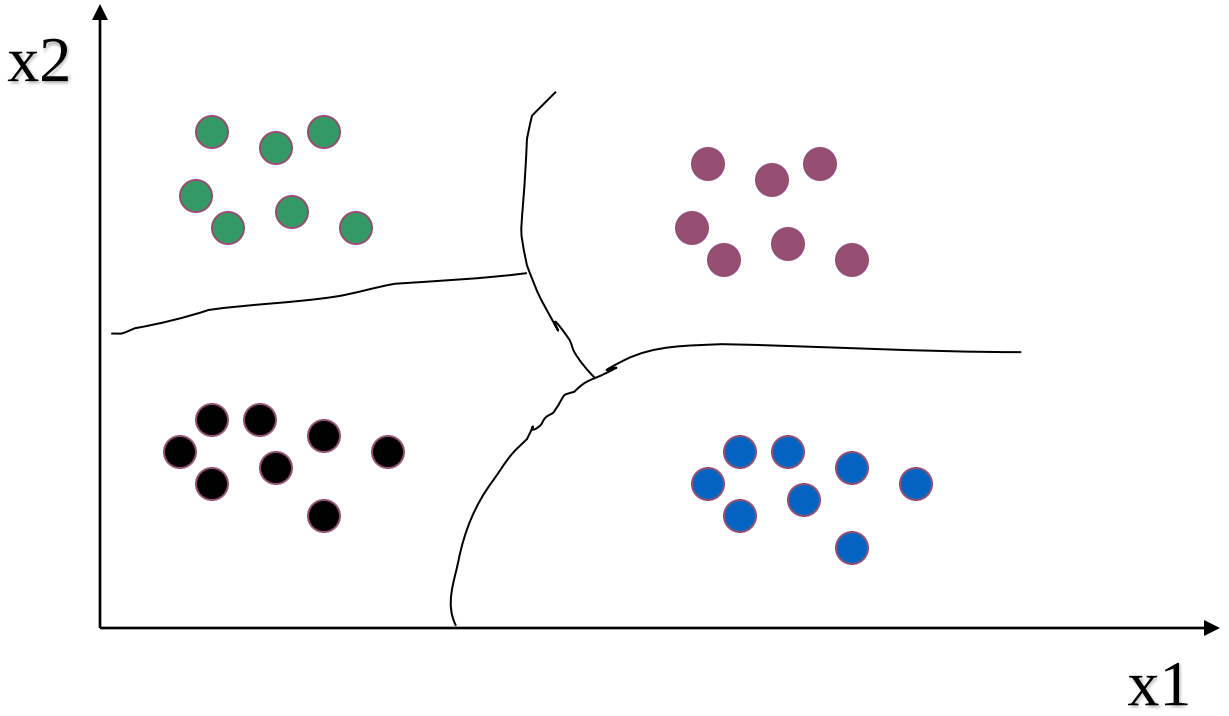


شبکه های چند لایه

بر خلاف پرسپترونها شبکه های چند لایه میتوانند برای یادگیری مسائل غیر خطی و همچنین مسائلی با تصمیم گیری های متعدد بکار روند.

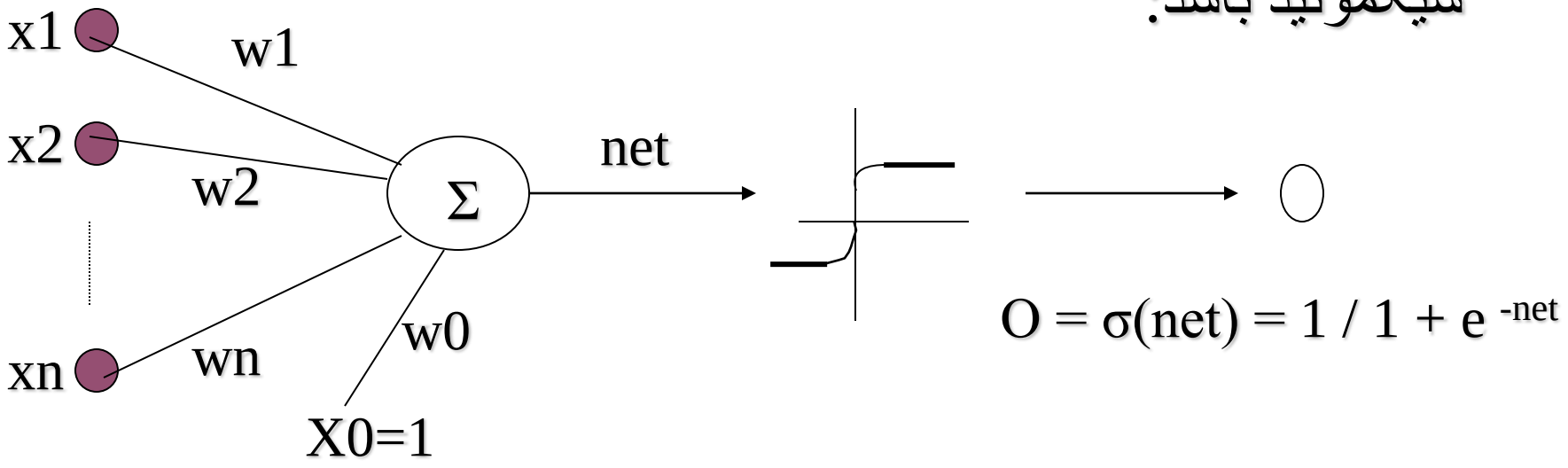


مثال



یک سلول واحد

برای اینکه بتوانیم فضای تصمیم گیری را بصورت غیر خطی از هم جدا بکنیم، لازم است تا هر سلول واحد را بصورت یک تابع غیر خطی تعریف نمائیم .مثالی از چنین سلولی میتواند یک واحد سیگموئید باشد:



یادگیری یک پرسپترون

- خروجی پرسپترون توسط رابطه زیر مشخص میشود:

$$O(x1,x2,\ldots,xn) = \begin{cases} 1 \text{ if } w0 + w1x1 + w2x2 + \ldots + wnxn > 0 \\ -1 \text{ otherwise} \end{cases}$$

- که برای سادگی آنرا میتوان بصورت زیر نشان داد:

$$O(\mathbf{X}) = sgn(\mathbf{WX}) \text{ where}$$

$$Sgn(y) = \begin{cases} 1 \text{ if } y > 0 \\ -1 \text{ otherwise} \end{cases}$$

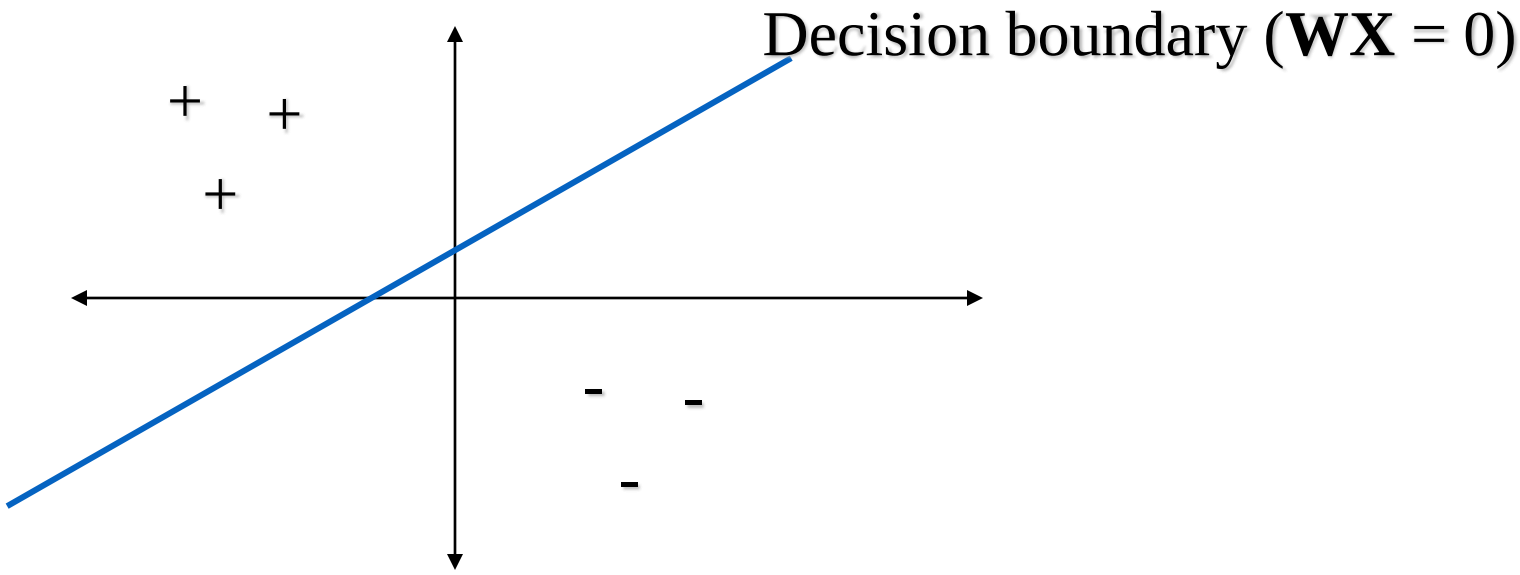
عبارت یادگیری پرسپترون از است:

W برای مقادیردرستی کردن پیدا

ممکن حقیقی مقادیر تمام ازمجموعه است عبارت یادگیری پرسپترون در H فرضیه فضای بنابراین وزن بردارهای برای.

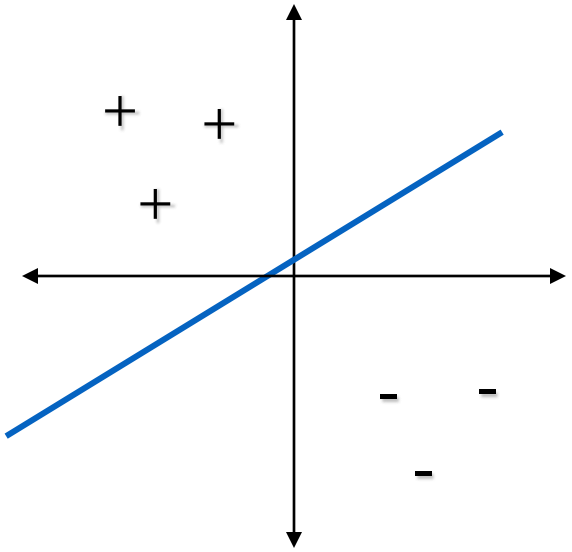
پرسپترون توانائی

- پریسپترون را میتوان بصورت یک سطح تصمیم hyperplane در فضای n بعدی نمونه ها در نظر گرفت .پرسپترون برای نمونه های یک طرف صفحه مقدار 1 و برای مقادیر طرف دیگر مقدار 1- بوجود میاورد.

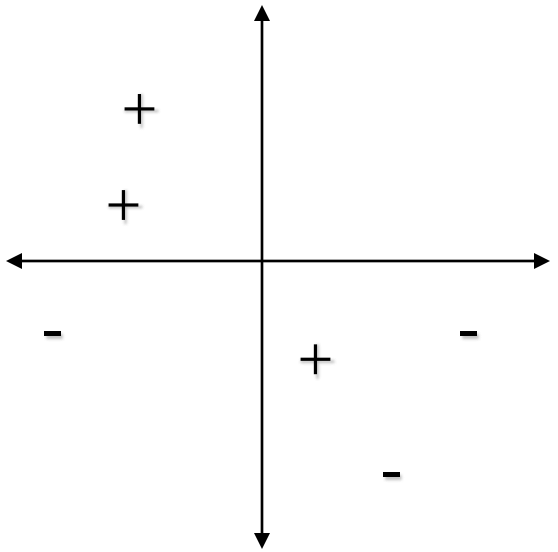


آنها یادگیری به قادر که پرسپترون توابعی میباشد

- یک پرسپترون فقط قادر است مثالهائی را یاد بگیرد که بصورت خطی جداپذیر باشند. اینگونه مثالها مواردی هستند که بطور کامل توسط یک hyperplaneقابل جدا سازی میباشند.



Linearly separable

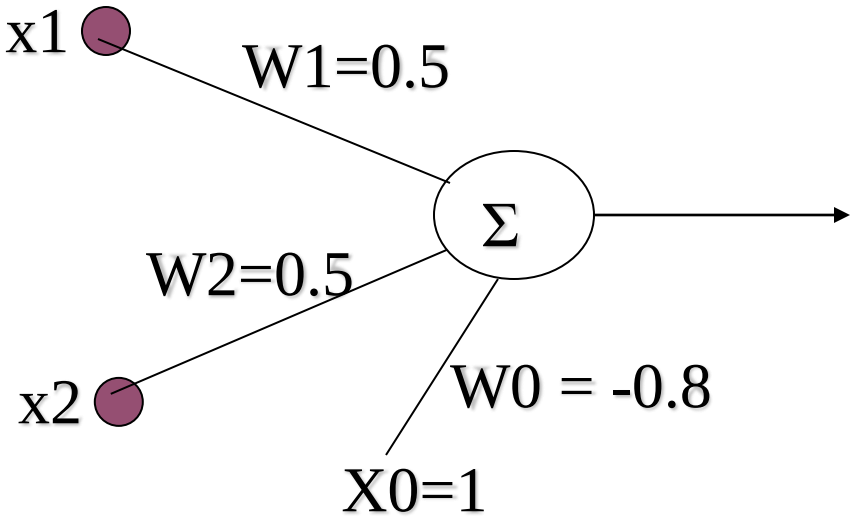


Non-linearly separable

توابع بولی و پرسپترون

- یک پرسپترون میتواند بسیاری از توابع بولی را نمایش دهد نظیر AND, OR, NAND, NOR
- اما نمیتواندXOR را نمایش دهد.

AND:

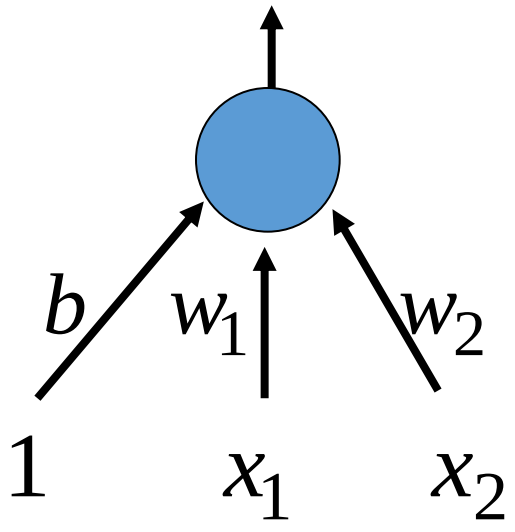


- در واقع هر تابع بولی را میتوان با شبکه ای دوسطحی از پرسپترونها نشان داد.

اضافه کردن بایاس

$$\hat{y} = b + \sum_i x_i w_i$$

- افزودن بایاس موجب میشود تا استفاده از شبکه پرسپترون با سهولت بیشتری انجام شود.
- برای اینکه برای یادگیری بایاس نیازی به استفاده از قانون دیگری نداشته باشیم بایاس را بصورت یک ورودی با مقدار ثابت ۱ در نظر گرفته و وزن W0 را به آن اختصاص میدهیم.



$$\hat{y} = w_0 + \sum_{i=1} x_i w_i$$

آموزش پرسپترون

- چگونه وزنهای یک پرسپترون واحد را یاد بگیریم به نحوی که پرسپترون برای مثالهای آموزشی مقادیر صحیح را ایجاد نماید؟
- دو راه مختلف:
  - قانون پرسپترون
  - قانون دلتا

آموزش پرسپترون

الگوریتم یادگیری پرسپترون

1. مقادیری تصادفی به وزنها نسبت میدهیم
2. پریسپترون را به تک تک مثالهای آموزشی اعمال میکنیم .اگر مثال غلط ارزیابی شود مقادیر وزنهای پرسپترون را تصحیح میکنیم.
3. آیا تمامی مثالهای آموزشی درست ارزیابی میشوند:
   - بله ← پایان الگوریتم
   - خیر←به مرحله 2 برمیگردیم

قانون پرسپترون

- برای یک مثال آموزشی $X = (x1, x2, ..., xn)$ در هر مرحله وزنها بر اساس قانون پرسپترون بصورت زیر تغییر میکند:

$$wi = wi + \Delta wi$$

که در آن

$$\Delta wi = \eta (t - o) xi$$

t: target output

o: output generated by the perceptron

η: constant called the learning rate (e.g., 0.1)

اثبات شده است که برای یک مجموعه مثال جداپذیرخطی این روش همگرا شده و پرسپترون قادر به جدا سازی صحیح مثالها خواهد شد.

قانون دلتا Delta Rule

- وقتی که مثالها بصورت خطی جداپذیر نباشند قانون پرسپترون همگرا نخواهد شد .برای غلبه بر این مشکل از قانون دلتا استفاده میشود .
- ایده اصلی این قانون استفاده از gradient descent برای جستجو در فضای فرضیه وزنهای ممکن میباشد .این قانون پایه روش Backpropagation است که برای آموزش شبکه با چندین نرون به هم متصل بکار میرود.
- همچنین این روش پایه ای برای انواع الگوریتمهای یادگیری است که باید فضای فرضیه ای شامل فرضیه های مختلف پیوسته را جستجو کنند.

قانون دلتا Delta Rule

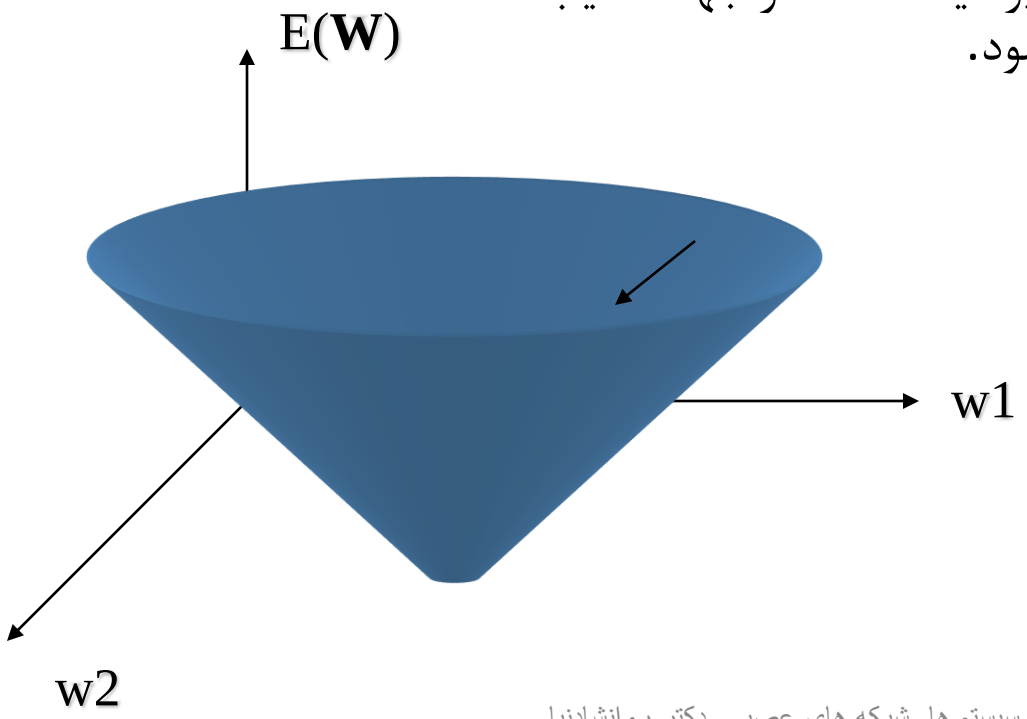
- برای درک بهتر این روش آنرا به یک پرسپترون فاقد حد آستانه اعمال میکنیم .در انجا لازم است ابتدا تعریفی برای خطا ی آموزش ارائه شود .یک تعریف متداول این چنین است:

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

- که این مجموع برای تمام مثالهای آموزشی انجام میشود.

الگوریتم gradient descent

- با توجه به نحوه تعریف E سطح خطا بصورت یک سهمی خواهد بود .ما بدنبال وزنهائی هستیم که حداقل خطا را داشته باشند . الگوریتم gradient descentدر فضای وزنها بدنبال برداری میگردد که خطا را حداقل کند .این الگوریتم از یک مقدار دلبخواه برای بردار وزن شروع کرده و در هر مرحله وزنها را طوری تغییر میدهد که در جهت شیب کاهشی منحنی فوق خطا کاهش داده شود.



بدست آوردن قانون gradient descent

- ایده اصلی :گرادیان همواره در جهت افزایش شیب E عمل میکند.
- گرادیان E نسبت به بردار وزن w بصورت زیر تعریف میشود:

$$E(W) = [E'/w0, E'/w1, \ldots, E'/wn]$$

- که در آن E (W) یک بردارو E’ مشتق جزئی نسبت به هر وزن میباشد.

$$\nabla$$

$$\nabla$$

Delta Rule **قانون دلتا**

- برای یک مثال آموزشیx) = Xش (x1, x2, ..., xnدر هر مرحله وزنها بر اساس قانون دلتا بصورت زیر تغییر میکند:

$$wi = wi + \Delta wi$$

$$\text{Where } \Delta wi = -\eta \; E'(\mathbf{W})/wi$$

η: learning rate (e.g., 0.1)

علامت منفی نشان دهنده حرکت در جهت کاهش شیب است.

گرادیان محاسبه

- با مشتق گیری جزئی از رابطه خطا میتوان بسادگی گرادیان را محاسبه نمود:

$$E'(W)/ w_i = \Sigma_i (t_i - O_i) (-x_i)$$

- لذا وزنها طبق رابطه زیر تغییر خواهند نمود.

$$\Delta wi = \eta \Sigma i (ti - oi) xi$$

خلاصه یادگیری قانون دلتا

الگوریتم یادگیری با استفاده از قانون دلتا بصورت زیر میباشد.

1. به وزنها مقدار تصادفی نسبت دهید
2. تا رسیدن به شرایط توقف مراحل زیر را ادامه دهید
   - هر وزن w_i را با مقدار صفر عدد دهی اولیه کنید.
   - برای هر مثال :وزن w_i را بصورت زیر تغییر دهید:

$$w_i = w_i + \eta (t - o) x_i$$

مقدار w_i را بصورت زیر تغییر دهید:

$$w_i = w_i + w_i$$

تا خطا بسیار کوچک شود

مشکلات روش gradient descent

1. ممکن است همگرا شدن به یک مقدار مینیمم زمان زیادی لازم داشته باشد.
2. اگر در سطح خطا چندین مینیمم محلی وجود داشته باشد تضمینی وجود ندارد که الگوریتم مینیمم مطلق را پیدا بکند.

در ضمن این روش وقتی قابل استفاده است که:

- فضای فرضیه دارای فرضیه های پارامتریک پیوسته باشد.
- رابطه خطا قابل مشتق گیری باشد

تقریب افزایشی gradient descent

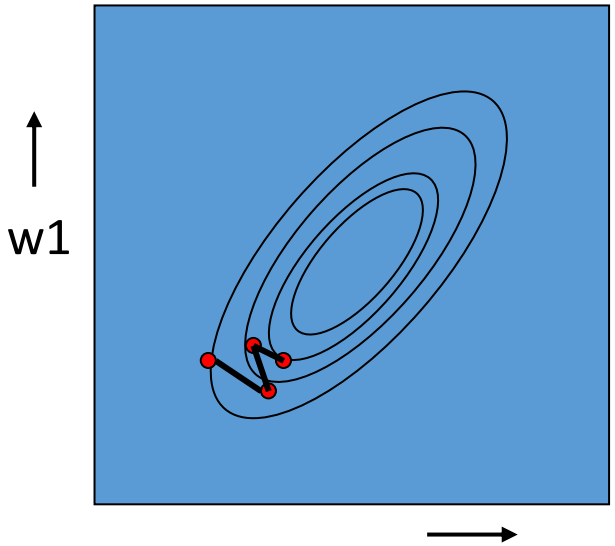
- میتوان بجای تغییر وزنها پس از مشاهده همه مثالها، آنها را بازا هر مثال مشاهده شده تغییر داد .در این حالت وزنها بصورت افزایشی incremental تغییر میکنند .این روش را stochastic gradient descentنیزمینامند.

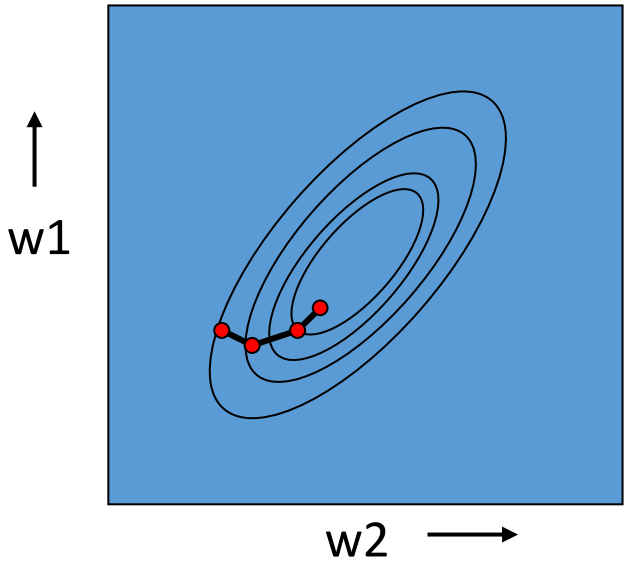
$$w_i = \eta (t-o) x_i$$

در بعضی موارد تغییر افزایشی وزنها میتواند از بروز مینیمم محلی جلوگیری کند .روش استاندارد ∇ نیاز به محاسبات بیشتری دارد درعوض میتواند طول step بزرگتری هم داشته باشد.

مقایسه آموزش یکجا و افزایشی

- آموزش افزایشی (Online learning)
- آموزش یکجا (Batch learning)

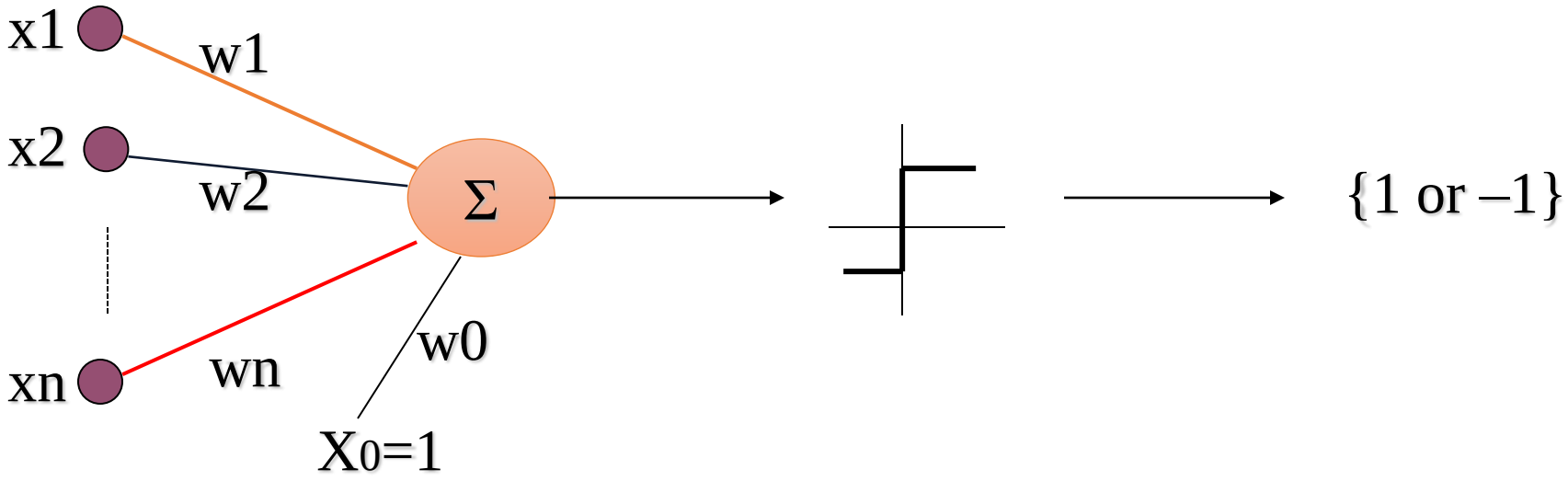




شبکه های چند لایه

شبکه های عصبی پرسپترون چند لایه

- نوعی از شبکه عصبی برمبنای یک واحد محاسباتی به نام پرسپترون ساخته می شود. یک پرسپترون برداری از ورودیهای با مقادیر حقیقی را گرفته و یک ترکیب خطی از این ورودیها را محاسبه میکند. اگر حاصل از یک مقدار آستانه بیشتر بود خروجی پرسپترون برابر با 1 و در غیر اینصورت معادل 1- خواهد بود.



یادگیری یک پرسپترون

خروجی پرسپترون توسط رابطه زیر مشخص می شود:

$$O(x1,x2,\ldots,xn) = \begin{cases} 1 \text{ if } w0 + w1x1 + w2x2 + \ldots + wnxn > 0 \\ -1 \text{ otherwise} \end{cases}$$

که برای سادگی آنرا میتوان بصورت زیر نشان داد:

$$O(\mathbf{X}) = sgn(\mathbf{WX}) \text{ where}$$

$$Sgn(y) = \begin{cases} 1 \text{ if } y > 0 \\ -1 \text{ otherwise} \end{cases}$$

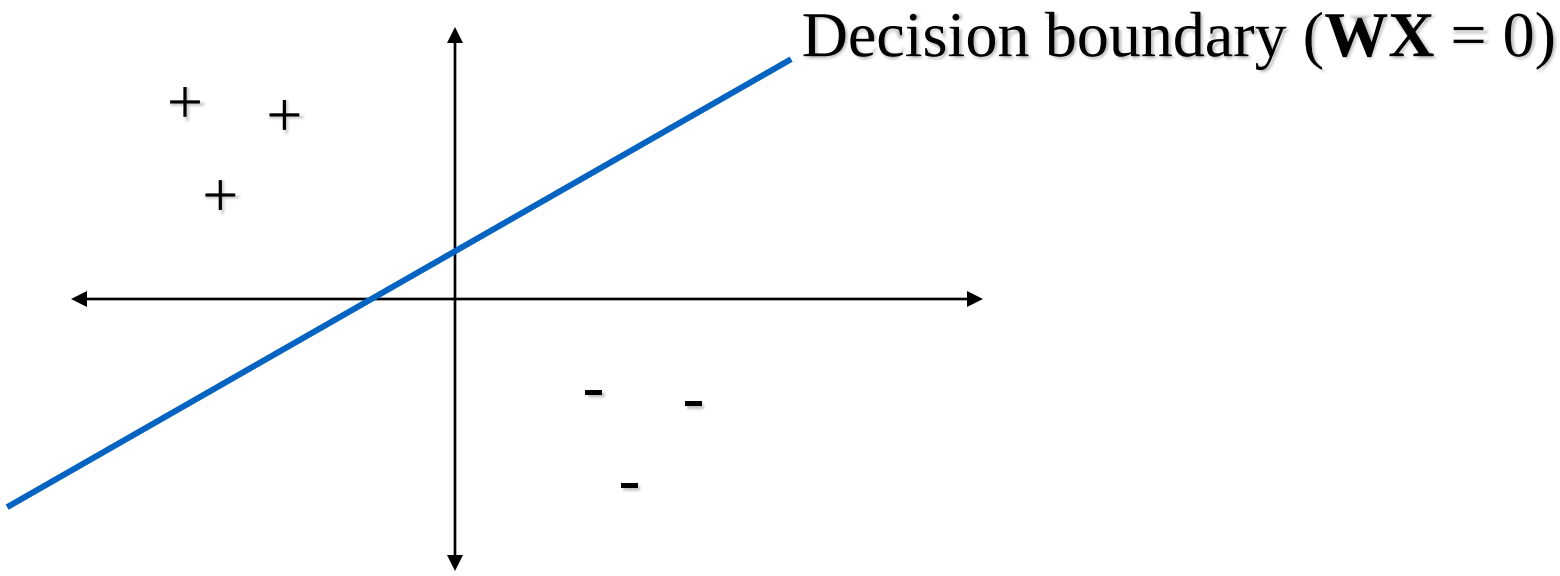
یادگیری پرسپترون عبارت است از:

پیدا کردن مقادیردرستی برای W

بنابراین فضای فرضیه H در یادگیری پرسپترون عبارت است ازمجموعه تمام مقادیر حقیقی ممکن برای بردارهای وزن.

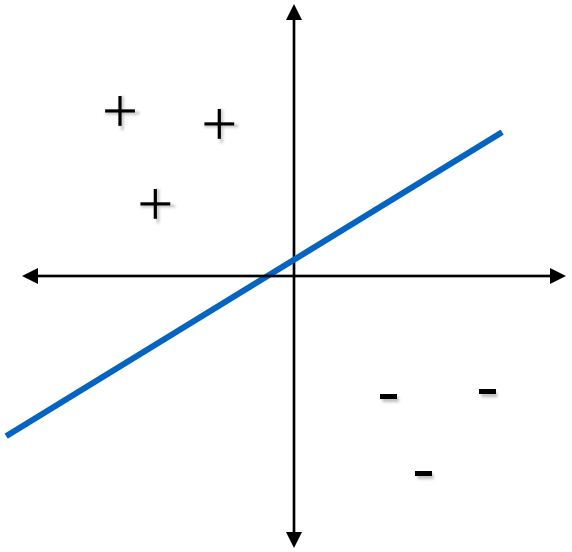
توانائی پرسپترون

- پریسپترون را میتوان بصورت یک سطح تصمیم Hyperplane در فضای nبعدی نمونه ها در نظر گرفت. پرسپترون برای نمونه های یک طرف صفحه مقدار 1 و برای مقادیر طرف دیگر مقدار 1- بوجود میاورد.

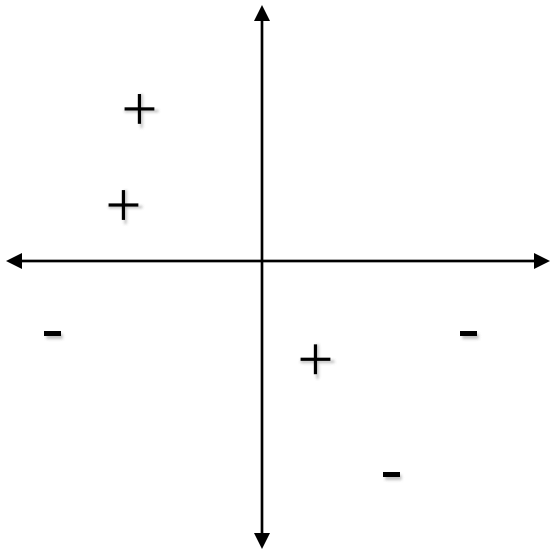


توابعی که پرسپترون قادر به یادگیری آنها میباشد

- یک پرسپترون فقط قادر است مثالهائی را یاد بگیرد که بصورت خطی جداپذیر باشند. اینگونه مثالها مواردی هستند که بطور کامل توسط یکhyperplane قابل جدا سازی میباشند.



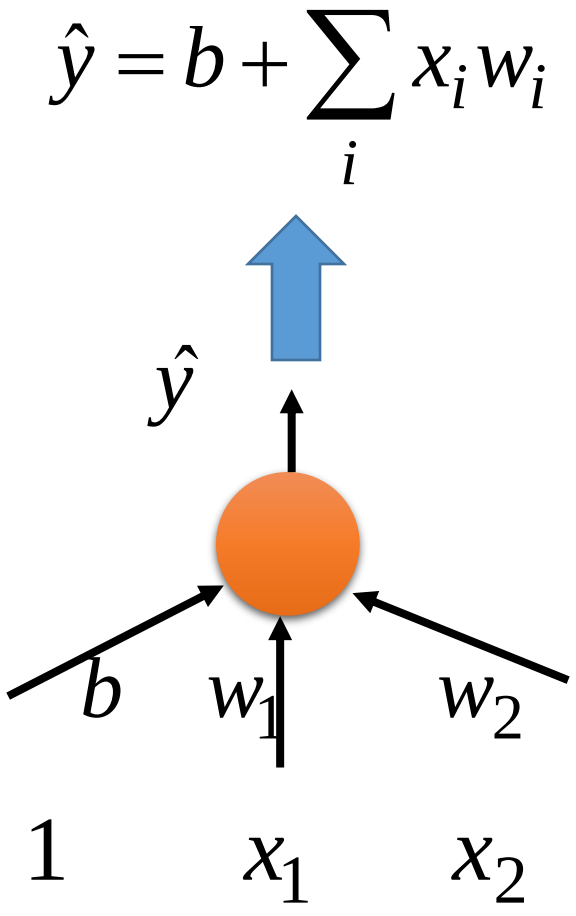
Linearly separable



Non-linearly separable

اضافه کردن بایاس

- افزودن بایاس موجب می شود تا از ایجاد لوپ در فرآیند تربیت جلوگیری شده و استفاده از شبکه پرسپترون با سهولت بیشتری انجام شود.
- برای اینکه برای یادگیری بایاس نیازی به استفاده از قانون دیگری نداشته باشیم بایاس را بصورت یک ورودی با مقدار ثابت ۱ در نظر گرفته و وزن W_0 را به آن اختصاص میدهیم.



آموزش پرسپترون:

- چگونه وزنهای یک پرسپترون واحد را یاد به شبکه بیاموزیم به نحوی که پرسپترون برای مثالهای آموزشی مقادیر صحیح را ایجاد نماید؟
- دو راه مختلف :
  - ✓ قانون پرسپترون (Perceptron Rule)
  - ✓ قانون دلتا (Delta Rule)

راهکار آموزش پرسپترون:

الگوریتم یادگیری پرسپترون

1. مقادیری تصادفی به وزن ها (Wi) نسبت می دهیم.
2. پرسپترون را به تک تک مثالهای آموزشی اعمال می کنیم. اگر مثال غلط ارزیابی شود آنگاه مقادیر وزنهای پرسپترون را تصحیح می کنیم.
3. آیا تمامی مثالهای آموزشی درست ارزیابی می شوند:
   - بله ← پایان الگوریتم
   - خیر←به مرحله ۲ برمی گردیم

قانون پرسپترون (Perceptron Rule)

- برای یک مثال آموزشی $X = (x1, x2, \ldots, xn)$ در هر مرحله وزنها بر اساس قانون پرسپتون بصورت زیر تغییر می کند:

$$wi = wi + \Delta wi$$

که در آن:

$$\Delta wi = \eta (t - o) xi$$

t: target output

o: output generated by the perceptron

η: constant called the learning rate (e.g., 0.1)

✓ اثبات شده است که برای یک مجموعه مثال جداپذیرخطی این روش همگرا شده و پرسپترون قادر به جدا سازی صحیح مثالها خواهد شد .

قانون دلتا (Delta Rule)

- وقتی که مثالها بصورت خطی جداپذیر نباشند قانون پرسپترون همگرا نخواهد شد. برای غلبه بر این مشکل از قانون دلتا استفاده می شود.
- ایده اصلی این قانون استفاده از Gradient descent برای جستجو در فضای فرضیه وزنهای ممکن می باشد. این قانون پایه روش Back Propagation (BP) است که برای آموزش شبکه با چندین نرون به هم متصل بکار می رود.
- همچنین این روش پایه ای برای انواع الگوریتمهای یادگیری است که باید فضای فرضیه ای شامل فرضیه های مختلف پیوسته را جستجو کنند.

قانون دلتا Delta Rule:

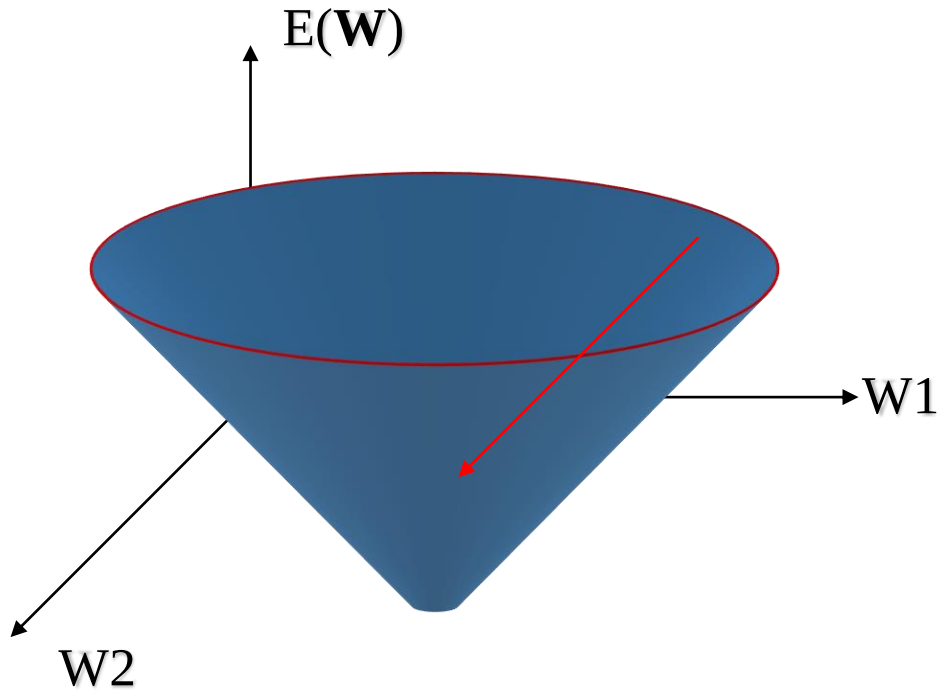
- برای درک بهتر این روش آنرا به یک پرسپترون را فاقد حد آستانه اعمال می کنیم (همگرایی تعریف نمی شود). در انجا لازم است ابتدا تعریفی برای خطای آموزش ارائه شود. یک تعریف متداول این چنین است:

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

t_i داده های رده ی تست و o_i داده های رده جوابها هستند.

- که این مجموع برای تمام مثالهای آموزشی انجام می شود.

Gradient descent الگوریتم



$$E(W) = \frac{1}{2} \sum_i (t_i - o_i)^2$$

- با توجه به نحوه تعریف E، سطح خطا بصورت یک سهمی خواهد بود. در این سهمی فضایی ما بدنبال وزنهائی هستیم که حداقل خطا را داشته باشند . الگوریتم Gradient descent در فضای وزنها بدنبال برداری میگردد که خطا را حداقل کند. این الگوریتم از یک مقدار دلخواه برای بردار وزن شروع کرده و در هر مرحله وزنها را طوری تغییر میدهد که در جهت شیب کاهشی منحنی فوق خطا کاهش داده شود.

خلاصه یادگیری قانون دلتا

الگوریتم یادگیری با استفاده از قانون دلتا بصورت زیر میباشد.

1. در ابتدا به وزن ها مقادیر تصادفی نسبت دهید
2. تا رسیدن به شرایط توقف مراحل زیر را ادامه دهید.
   - هر وزن w_i را با مقدار صفر عدد دهی اولیه کنید.
   - برای هر مثال وزن w_i را بصورت زیر تغییر دهید:

$$\Delta w_i = \Delta w_i + \eta (t - o) x_i$$

مقدار w_i را بصورت زیر آنقدر تغییر دهید:

$$w_i = w_i + \Delta w_i$$

تا خطا بسیار کوچک شود.

مشکلات روش Gradient descent

1. ممکن است همگرا شدن به یک مقدار مینیمم زمان زیادی لازم داشته باشد.
2. اگر در سطح خطا چندین مینیمم محلی وجود داشته باشد تضمینی وجود ندارد که الگوریتم مینیمم مطلق را پیدا بکند.

در ضمن این روش وقتی قابل استفاده است که:

- فضای فرضیه دارای فرضیه های پارامتریک پیوسته باشد.
- رابطه خطا قابل مشتق گیری باشد.

تقریب افزایشی Gradient descent

- میتوان بجای تغییر وزنها پس از مشاهده همه مثالها، آنها را در هر مثال مشاهده شده تغییر داد .
- در این حالت وزن ها به صورت افزایشی Incremental تغییر می کنند.این روش را Stochastic Gradient Descent نیزمی نامند.

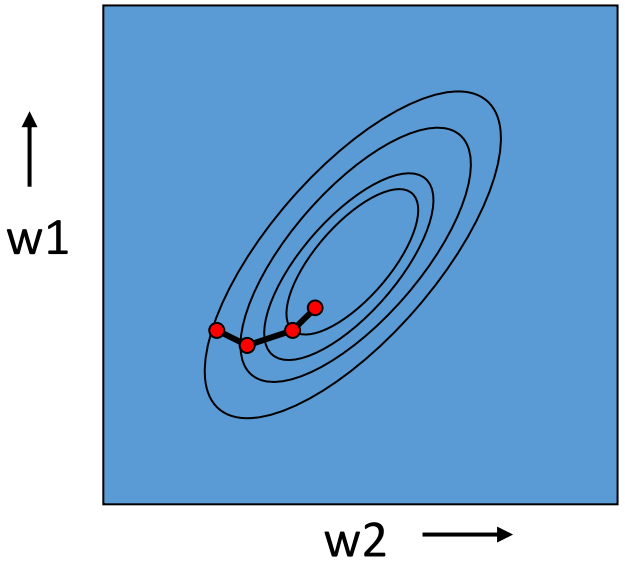
$$\nabla w_i = \eta (t-o) x_i$$

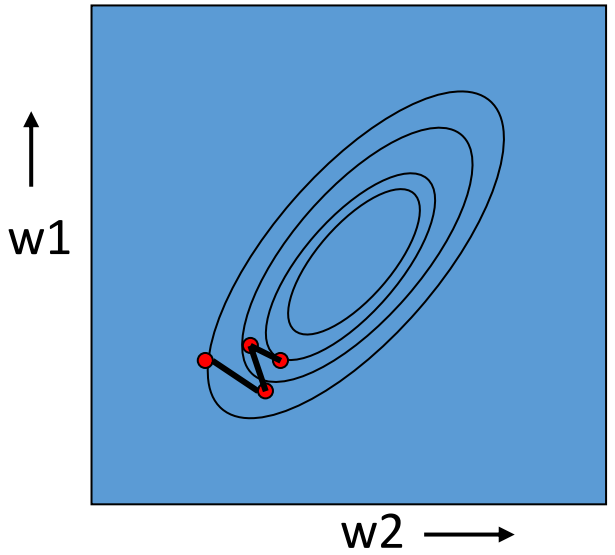
در بعضی موارد تغییر افزایشی وزن ها می تواند از بروز مینیمم محلی جلوگیری کند. روش استاندارد نیاز به محاسبات بیشتری دارد درعوض میتواند طول Step بزرگتری هم داشته باشد.

مقایسه آموزش یکجا و افزایشی

- آموزش یکجا
(Batch learning)

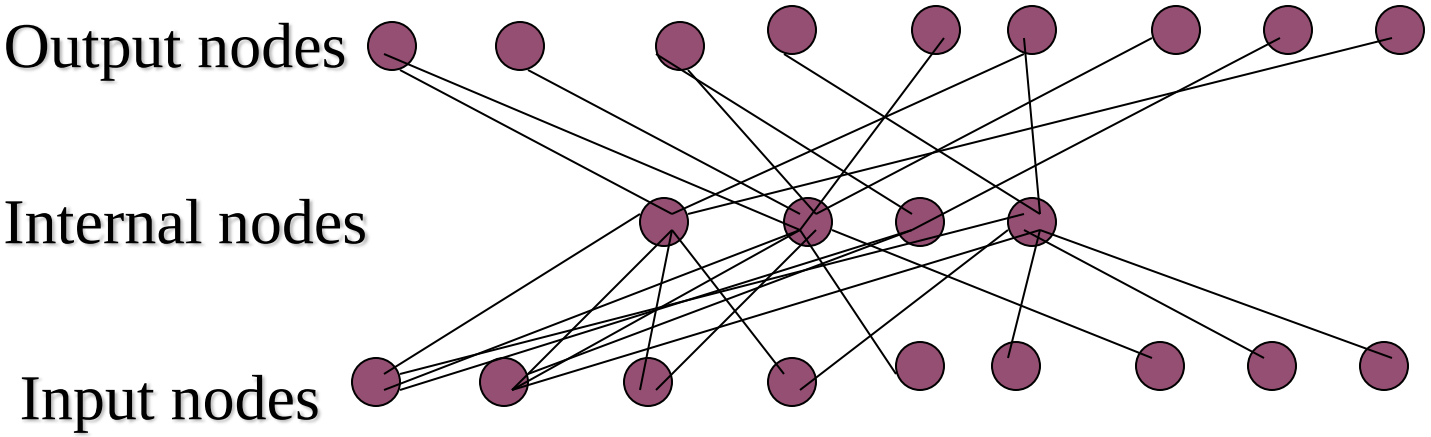
- آموزش افزایشی
(Online learning)





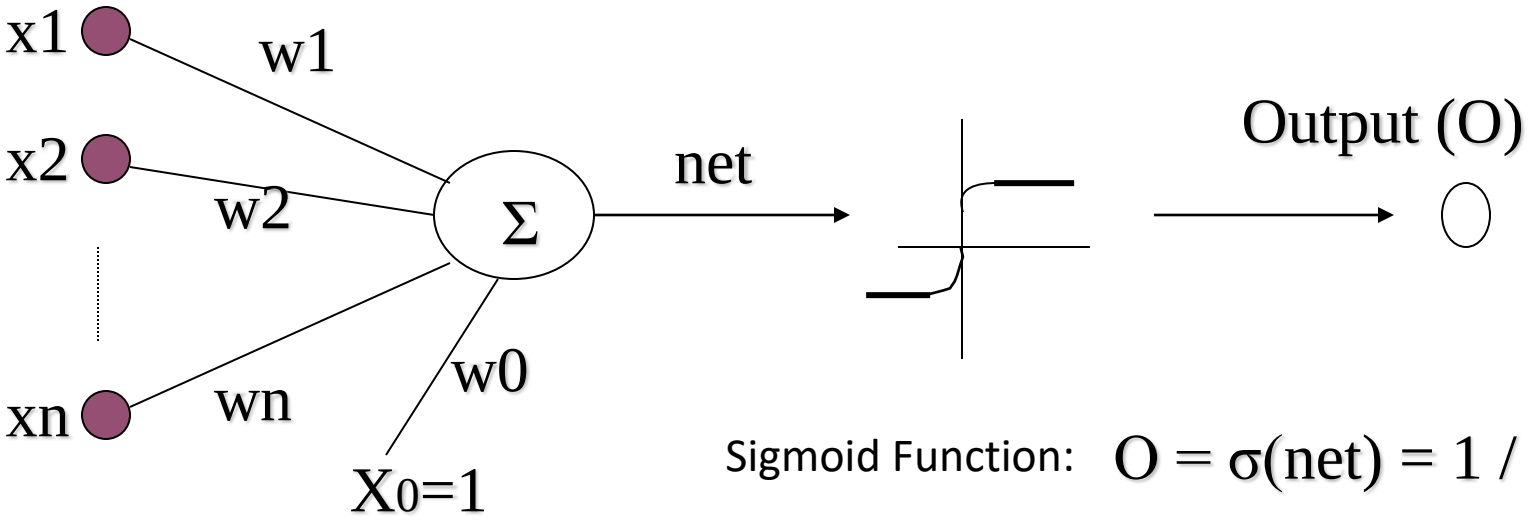
شبکه های چند لایه (Multi-Layer Perceptron)

بر خلاف پرسپترونها، شبکه های چند لایه (MLP) می توانند برای یادگیری مسائل غیر خطی و همچنین مسائلی با تصمیم گیری های متعدد بکار روند.



شبکه های چند لایه (Multi-Layer Perceptron)

برای اینکه بتوانیم فضای تصمیم گیری را بصورت غیر خطی از هم جدا کنیم، لازم است تا هر سلول واحد را به صورت یک تابع غیر خطی تعریف نمائیم.
مثالی از چنین سلولی میتواند یک واحد سیگموئید باشد:



Sigmoid Function: $O = \sigma(net) = 1 / 1 + e^{-net}$

تابع سیگموئید:

خروجی این سلول واحد را بصورت زیر میتوان بیان نمود:

$$O(x1,x2,\ldots,xn) = \sigma (WX)$$

where: $\sigma (WX) = 1 / 1 + e^{-WX}$

تابع σ تابع سیگموئید یا لجستیک نامیده می شود. این تابع دارای خاصیت زیر است:

$$d\,\sigma(y) / dy = \sigma(y)\ (1 - \sigma(y))$$

الگوریتم Back Propagation:

- برای یادگیری وزن های یک شبکه چند لایه از روش Back Propagation استفاده می شود. در این روش با استفاده از Gradient descentسعی می شود تا مربع خطای بین خروجی های شبکه و تابع هدف مینیمم شود.
- خطا به صورت زیر تعریف می شود:

$$E\left(\vec{W}\right) \equiv \frac{1}{2}\sum_{d \in D}\sum_{k \in outputs}\left(t_{kd} - O_{kd}\right)^2$$

✓ منظور از outputs خروجیهای مجموعه واحد های لایه خروجی و t_{kd} و o_{kd} مقدار هدف و خروجی متناظر با k امین واحد خروجی و مثال آموزشی d است.

الگوریتم Back propagation

- فضای فرضیه مورد جستجو در این روش عبارت است از :

فضای بزرگی که توسط همه مقادیر ممکن برای وزنها تعریف میشود. روش Gradient descentسعی می کند تا با مینیمم کردن خطا به فرضیه مناسبی دست پیدا کند.

اما تضمینی برای اینکه این الگوریتم به مینیمم مطلق برسد وجود ندارد.

الگوریتم پیاده سازی روش BP:

1. شبکه ای با n_{in} گره ورودی، n_{hidden} گره مخفی، و n_{out} گره خروجی ایجاد کنید.
2. همه وزنها را با یک مقدار تصادفی کوچک عدد دهی کنید.
3. تا رسیدن به شرط پایانی (کوچک شدن خطا) مراحل زیر را انجام دهید:

   برای هر x متعلق به مثالهای آموزشی:

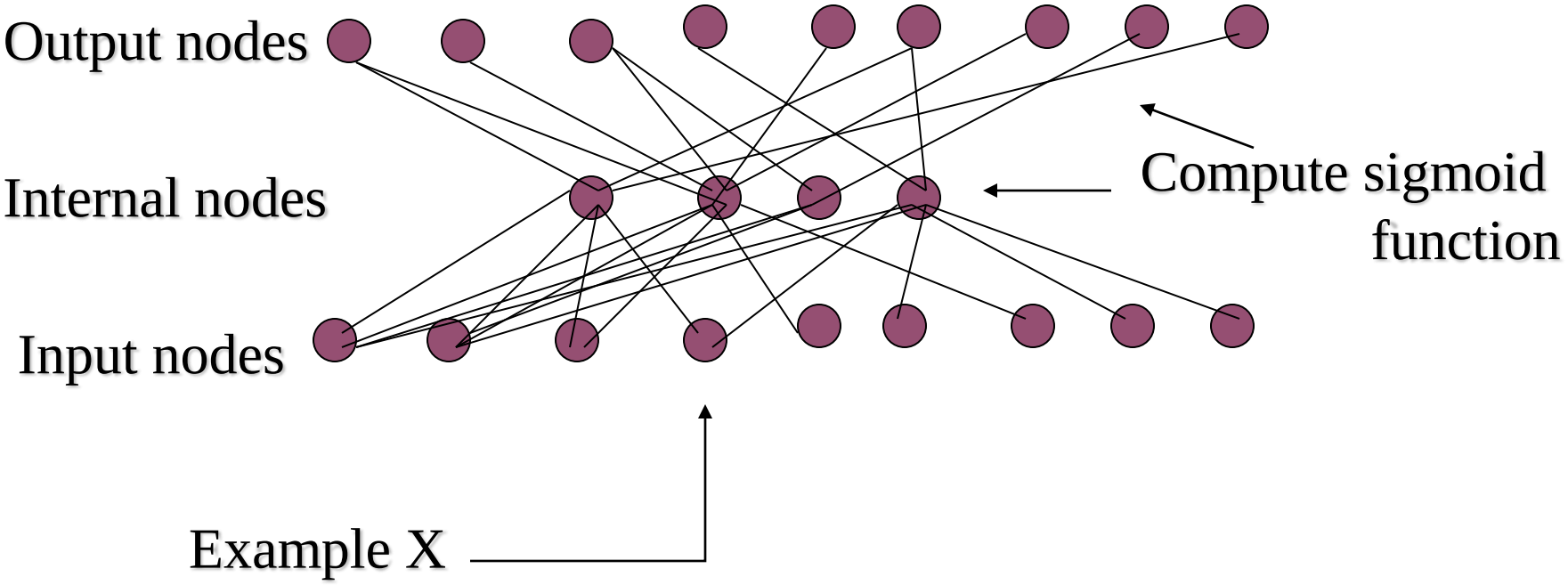
   مثال X را به سمت جلو در شبکه انتشار دهید

   خطای E را به سمت عقب در شبکه انتشار دهید.

هر مثال آموزشی بصورت یک زوج (x,t) ارائه می شود که بردار x مقادیر ورودی و بردار t مقادیر هدف برای خروجی شبکه را تعیین می کنند.

انتشار به سمت جلو

- برای هر مثال X مقدار خروجی هر واحد را محاسبه کنید تا به گره های خروجی برسید.



الگوریتم روش انتشار به سمت عقب

1. برای هر واحد خروجی جمله خطا را بصورت زیر محاسبه کنید:

$$\delta_k = O_k (1-O_k)(t_k - O_k)$$

2. برای هر واحد مخفی جمله خطا را بصورت زیر محاسبه کنید:

$$\delta_h = O_h (1-O_h) \Sigma_k W_{kh} \delta_k$$

3. مقدارهر وزن را بصورت زیر تغییر دهید:

$$W_{ji} = W_{ji} + \Delta W_{ji}$$

که در آن:

$$\Delta W_{ji} = \eta \delta_j X_{ji}$$

η عبارت است از نرخ یادگیری

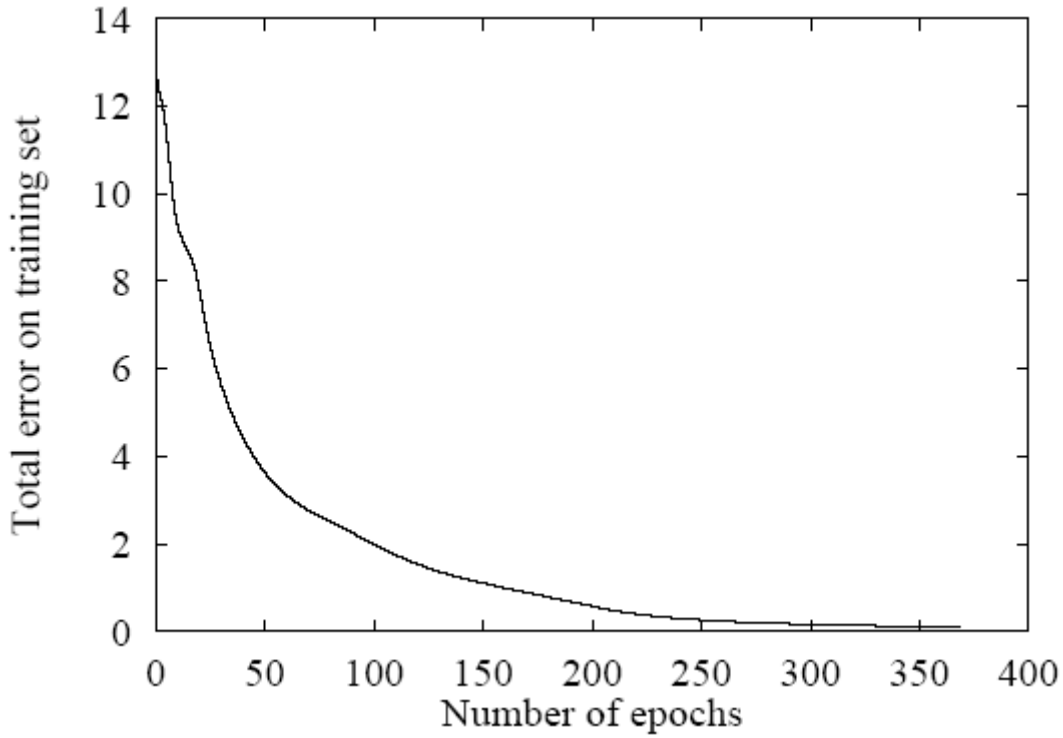
شرط خاتمه

معمولا الگوریتم BP پیش از خاتمه هزاران بار با استفاده همان داده های آموزشی تکرار میگردد شروط مختلفی را میتوان برای خاتمه الگوریتم بکار برد:

- توقف بعد از تکرار به دفعات معین
- توقف وقتی که خطا از یک مقدار تعیین شده کمتر شود
- توقف وقتی که خطا در مثالهای مجموعه تائید از قاعده خاصی پیروی نماید

اگر دفعات تکرار کم باشد خطا خواهیم داشت و اگر زیاد باشد مسئله Overfitting رخ خواهد داد.

منحنی یادگیری



مرور الگوریتم BP

- این الگوریتم یک جستجوی Gradient descent در فضای وزنها انجام میدهد.
- ممکن است در یک مینیمم محلی گیر بیافتد.
- در عمل بسیار موثر بوده است.

برای پرهیز از مینیمم محلی روشهای مختلفی وجود دارد:

- افزودن ممنتم
- استفاده ازStochastic Gradient Descent
- استفاده ازشبکه های مختلف با مقادیر متفاوتی برای وزنهای اولیه

افزودن ممنتم

- می توان قانون تغییر وزنها را طوری در نظر گرفت که تغییر وزن در تکرار n ام تا حدی به اندازه تغییروزن در تکرار قبلی بستگی داشته باشد.

$$\Delta W_{ji}(n) = \eta\, \delta_j\, X_{ji} + \alpha \Delta W_{ji}(n-1)$$

که در آن مقدارممنتم α بصورت $0 < \alpha < 1$ می باشد

عبارت ممنتم **قانون تغییر وزن**

افزودن ممنتم باعث می شود تا با حرکت در مسیر قبلی در سطح خطا:

- از گیر افتادن در مینیم محلی پرهیز شود
- از قرارگرفتن در سطوح صاف پرهیز شود
- با افزایش تدریجی مقدار پله تغییرات، سرعت جستجو افزایش یابد.

قدرت نمایش توابع

- گرچه قدرت نمایش توابع به توسط یک شبکه forward feed بسته به عمق و گستردگی شبکه دارد، با این وجود موارد زیر را میتوان به صورت قوانین کلی بیان نمود:
- توابع بولی :هر تابع بولی را میتوان توسط یک شبکه دو لایه پیاده سازی نمود.
- توابع پیوسته :هر تابع پیوسته محدود را میتوان توسط یک شبکه دو لایه تقریب زد. تئوری مربوطه در مورد شبکه هائی که از تابع سیگموئید در لایه پنهان و لایه خطی در شبکه خروجی استفاده میکنند صادق است.
- توابع دلخواه :هر تابع دلخواه را میتوان با یک شبکه سه لایه تا حد قابل قبولی تقریب زد.

با این وجود باید درنظر داشت که فضای فرضیه جستجو شده توسط روش gradient deescent ممکن است در برگیرنده تمام مقادیر ممکن وزنها نباشد.

فضای فرضیه و بایاس استقرا

- فضای فرضیه مورد جستجو را می توان بصورت یک فضای فرضیه اقلیدسی n بعدی از وزنهای شبکه در نظر گرفت. (که n تعداد وزنهاست)
- این فضای فرضیه بر خلاف فضای فرضیه درخت تصمیم یک فضای پیوسته است.
- بایاس استقرا این روش را می توان بصورت زیر بیان کرد:

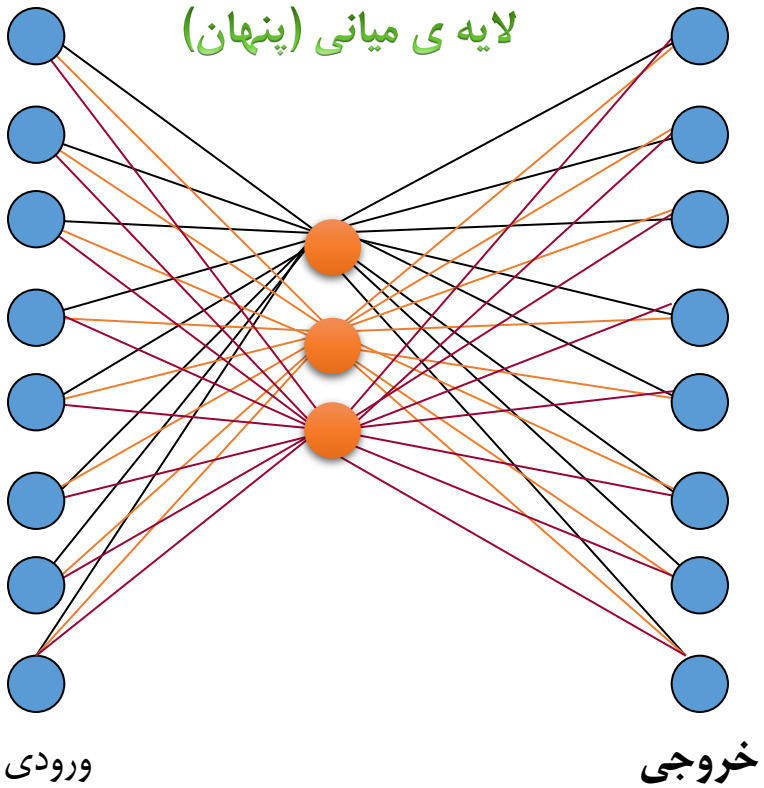
“smooth interpolation between data points”

به این معنا که الگوریتم BP سعی میکند تا نقاطی را که به هم نزدیکتر هستند در یک دسته بندی قرار دهد.

قدرت نمایش لایه پنهان

- یکی از خواص BP این است که میتواند در لایه های پنهان شبکه ویژگیهای نا آشکاری از داده ورودی نشان دهد.

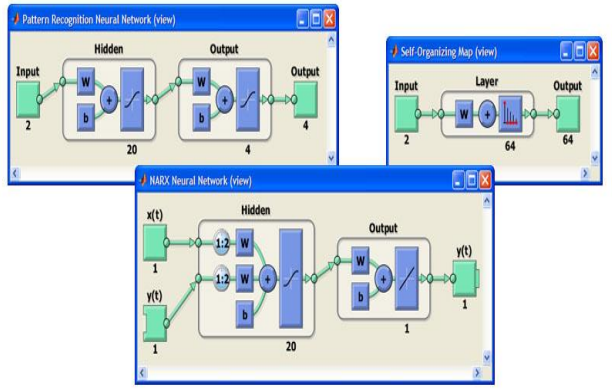
برای مثال شبکه 8x3x8 زیر طوری آموزش داده میشود که مقدار هر مثال ورودی را عینا در خروجی بوجود آورد (تابع $f(x)=x$ را یاد بگیرد) ساختار خاص این شبکه باعث میشود تا واحد های لایه وسط ویژگی های مقادیر ورودی را به نحوی کدبندی کنند که لایه خروجی بتواند از آنان برای نمایش مجدد داده ها استفاده نماید .



قدرت نمایش لایه پنهان

در این آزمایش که به تعداد ۵۰۰۰ بار تکرار شده از ۸ داده مختلف به عنوان ورودی استفاده شده و شبکه با استفاده از الگوریتم BP موفق شده تا تابع هدف را بیاموزد.

10000000
01000000
00100000
00010000
00001000
00000100
00000010
00000001

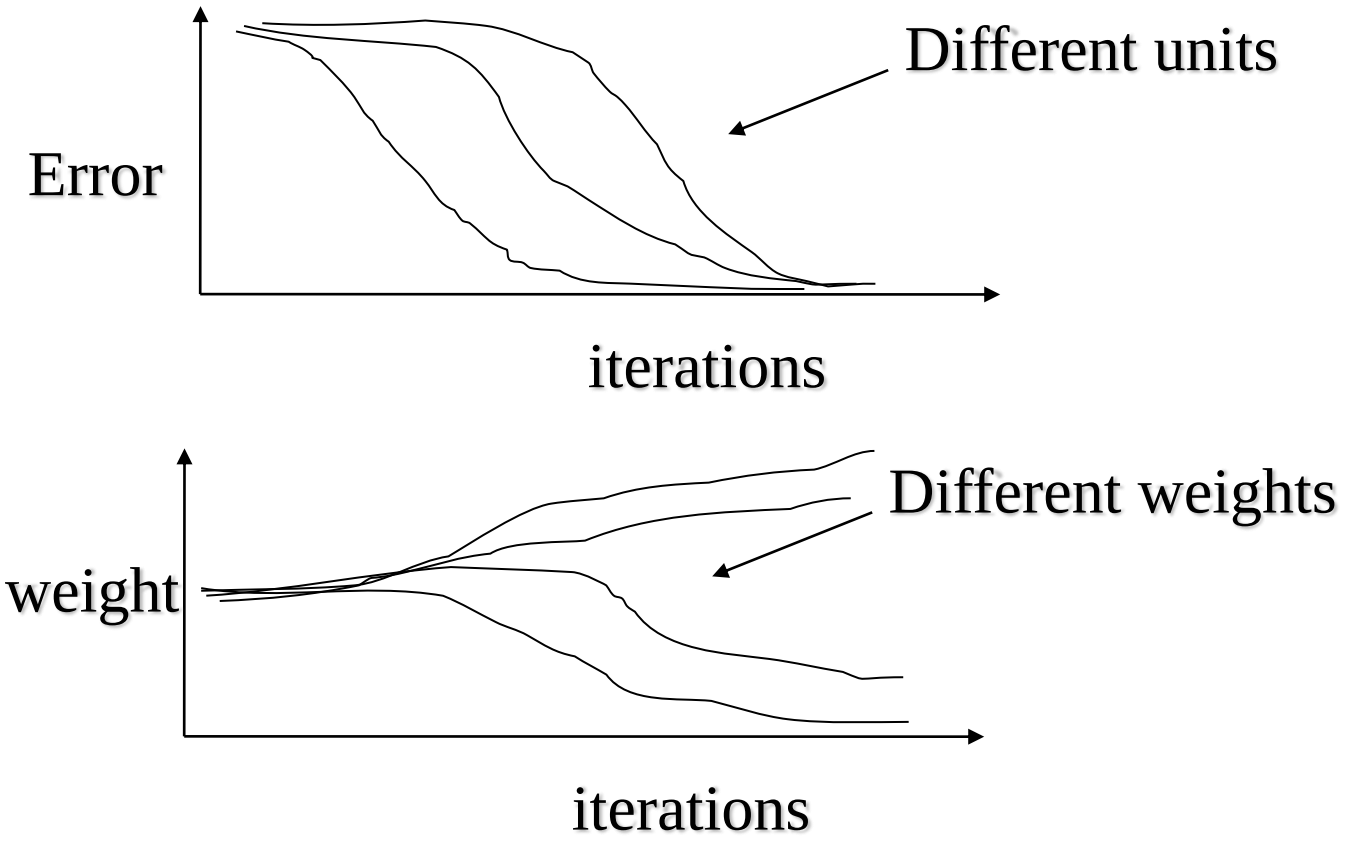


Hidden nodes

10000000
01000000
00100000
00010000
00001000
00000100
00000010
00000001

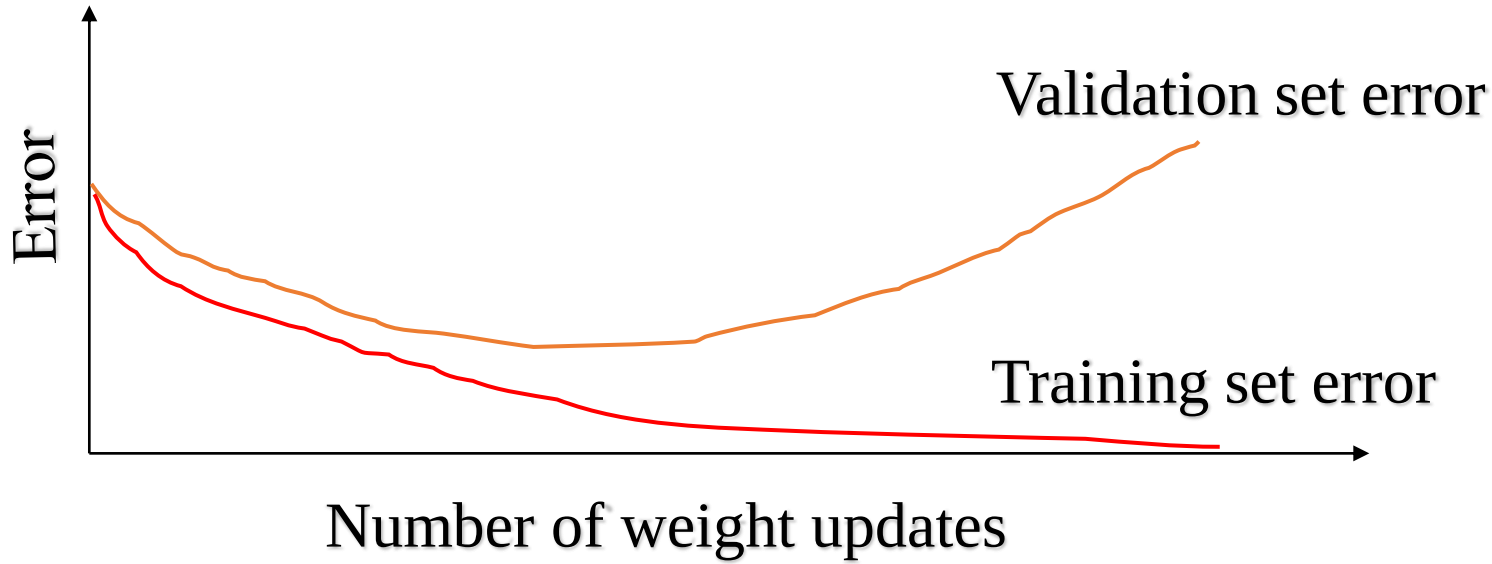
با مشاهده خروجی واحد های لایه میانی مشخص میشود که بردار حاصل معادل انکدینگ استاندارد داده ههای ورودی بوده است. (000,001,...,111)

نمودارخطا


Different units
Error
iterations
Different weights
weight
iterations

قدرت تعمیم وoverfitting

- شرط پایین الگوریتم BP چیست؟
- یک انتخاب این است که الگوریتم را آنقدر ادامه دهیم تا خطا از مقدار معینی کمتر شود. این امر میتواند منجر به overfitting شود.



دلایل رخ دادن Overfitting

- Overfitting ناشی از تنظیم وزنها برای در نظر گرفتن مثالهای نادری است که ممکن است با توزیع کلی داده ها مطابقت نداشته باشند. تعداد زیاد وزنهای یک شبکه عصبی باعث میشود تا شبکه درجه آزادی زیادی برای انطباق با این مثالها داشته باشد.

- با افزایش تعداد تکرار، پیچیدگی فضای فرضیه یادگرفته شده توسط الگوریتم بیشتر و بیشتر میشود تا شبکه بتواند نویز و مثالهای نادر موجود در مجموعه آموزش را بدرستی ارزیابی نماید.

راه حل

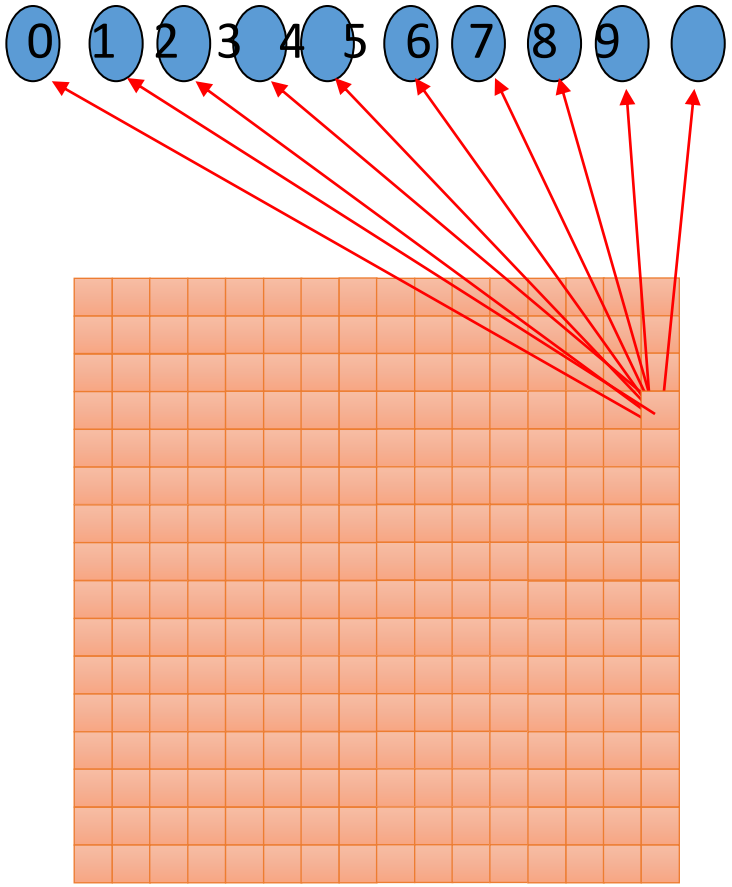
- استفاده از یک مجموعه تائید Validation و توقف یادگیری هنگامی که خطا در این مجموعه به اندازه کافی کوچک می شود.
- بایاس کردن شبکه برای فضاهای فرضیه ساده تر: یک راه میتواند استفاده از weight decay باشد که در آن مقدار وزنها در هر بارتکرار باندازه خیلی کمی کاهش داده می شود.
- k-fold cross validation وقتی که تعداد مثالهای آموزشی کم باشد میتوان m داده آموزشی را به K دسته تقسیم بندی نموده و آزمایش را به تعداد k دفعه تکرار نمود. در هر دفعه یکی از دسته ها بعنوان مجموعه تست و بقیه بعنوان مجموعه آموزشی استفاده خواهند شد. تصمیم گیری بر اساس میانگین نتایج انجام میشود.

روشهای دیگر

راه های بسیار متنوعی برای ایجاد شبکه های جدید وجود دارد از جمله:

- استفاده از تعاریف دیگری برای تابع خطا
- استفاده از روشهای دیگری برای کاهش خطا در حین یادگیری
  - Hybrid Global Learning
  - Simulated Annealing
  - Genetic Algorithms
- استفاده از توابع دیگری در واحدها
  - Radial Basis Functions
- استفاده از ساختار های دیگری برای شبکه
  - Recurrent Network

مثال: تشخیص ارقام



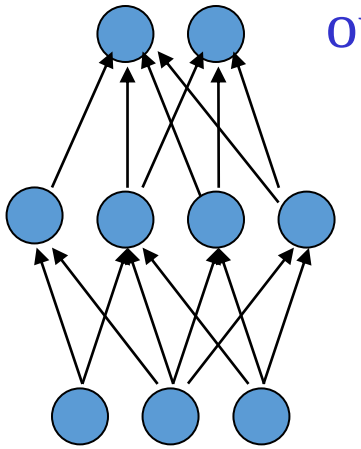
- فرض کنید بخواهیم با استفاده از یک شبکه دو لایه ارقام دستنویس را تشخیص دهیم.
- نرونهای لایه اول شدت روشنائی پیکسلها را تقریب میزنند ونرونهای لایه آخر شکل ارقام را تعیین میکنند.

شبکه چه چیزی را یاد میگیرد؟

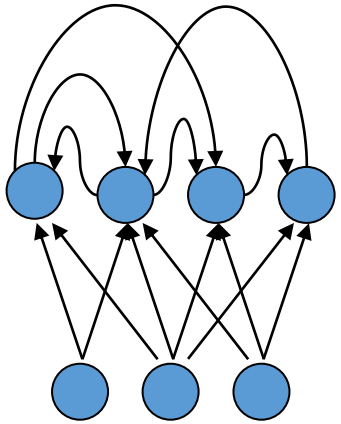
- در این مثال یک شبکه با دو لایه معادل با استفاده از یک سری template یا قالب است که شبکه قالبی را که بهترین تطبیق با ورودی را داشته باشد بر میگزیند!
- اما برای مسئله ارقام دستنویس شکلهای ورودی بسیار متنوع هستند لذا یک قالب ساده که با همه ورودیها سازگار باشد وجود ندارد. در نتیجه چنین شبکه ای هم نمیتواند راه حل مسئله در حالت کلی باشد!
- برای اینکه بتوان مسئله را در حالت کلی حل نمود بایدشکل های ورودی به مجموعه ای از ویژگی ها تبدیل شده و شبکه را بر اساس ویژگی ها آموزش داد.

انواع اتصالات شبکه

- Feed forward networks



- Recurrent networks
  - این شبکه ها بیشتر به سیستم های بیولوژیکی شبیه تر هستند.
  - بعلت داشتن فید بک دارای دینامیک پیچیده تری هستند.



انواع مختلف یادگیری

- Supervised learning
  - سیستم یاد میگیرد که با داشتن بردار ورودی مقدار خروجی را پیش بینی کند
  - ناظری لازم است تا خروجی صحیح را تهیه نماید.
- Reinforcement learning
  - سیستم یاد میگیرد که پاداش دریافتی را حدااکثر کند
  - سیگنال پاداش اطلاعات زیادی ندارد
  - اغلب با تاخیر است

Unsupervised learning

- یک مدل داخلی از ورودی درست میشود مثلا از طریق کلاسترینگ
- چگونه میتوان فهمید که این مدل صحیح است؟

اعمال Back Propagation به تشخیص اشیا

- انسانها براحتی میتوانند اسکال را تشخیص دهند
- در صورتیکه اینکار برای کامپیوتر ها بسیار سخت است.
- دلایل سختی این کار عبارت است از:

- Segmentation: Real scenes are cluttered.
- In variances: We are very good at ignoring all sorts of variations that do not affect the shape.
- Deformations: Natural shape classes allow variations (faces, letters, chairs).
- A huge amount of computation is required.

تابع سیگموئید

خروجی این سلول واحد را بصورت زیر میتوان بیان نمود:

$$O(x1,x2,\ldots,xn) = \sigma (WX)$$

where: $\sigma (WX) = 1 / 1 + e^{-WX}$

تابع σ تابع سیگموئید یا لجستیک نامیده میشود .این تابع دارای خاصیت زیر است:

$$d\,\sigma(y) / dy = \sigma(y)\ (1 - \sigma(y))$$

الگوریتم Back propagation

- برای یادگیری وزن های یک شبکه چند لایه از روش Back Propagation استفاده میشود . در این روش با استفاده از gradient descent سعی میشود تا مربع خطای بین خروجی های شبکه و تابع هدف مینیمم شود.
- خطا بصورت زیر تعریف میشود:

$$E\left(\vec{W}\right) \equiv \frac{1}{2} \sum_{d \in D} \sum_{k \in outputs} \left(t_{kd} - o_{kd} \right)^2$$

مراد از outputsخروجیهای مجموعه واحد های لایه خروجی و t_{kd} و o_{kd}مقدار هدف و خروجی متناظر با k امین واحد خروجی و مثال آموزشی d است.

Back propagation الگوریتم

- فضای فرضیه مورد جستجو در این روش عبارت است از فضای بزرگی که توسط همه مقادیر ممکن برای وزنها تعریف میشود .روش gradient descent سعی میکند تا با مینیمم کردن خطا به فرضیه مناسبی دست پیدا کند .اما تضمینی برای اینکه این الگوریتم به مینیمم مطلق برسد وجود ندارد.

الگوریتم BP

1. شبکه ای با n_{in} گره ورودی، n_{hidden} گره مخفی، و n_{out} گره خروجی ایجاد کنید .
2. همه وزنها را با یک مقدار تصادفی کوچک عدد دهی کنید.
3. تا رسیدن به شرط پایانی (کوچک شدن خطا)مراحل زیر را انجام دهید:

برای هر x متعلق به مثالهای آموزشی:

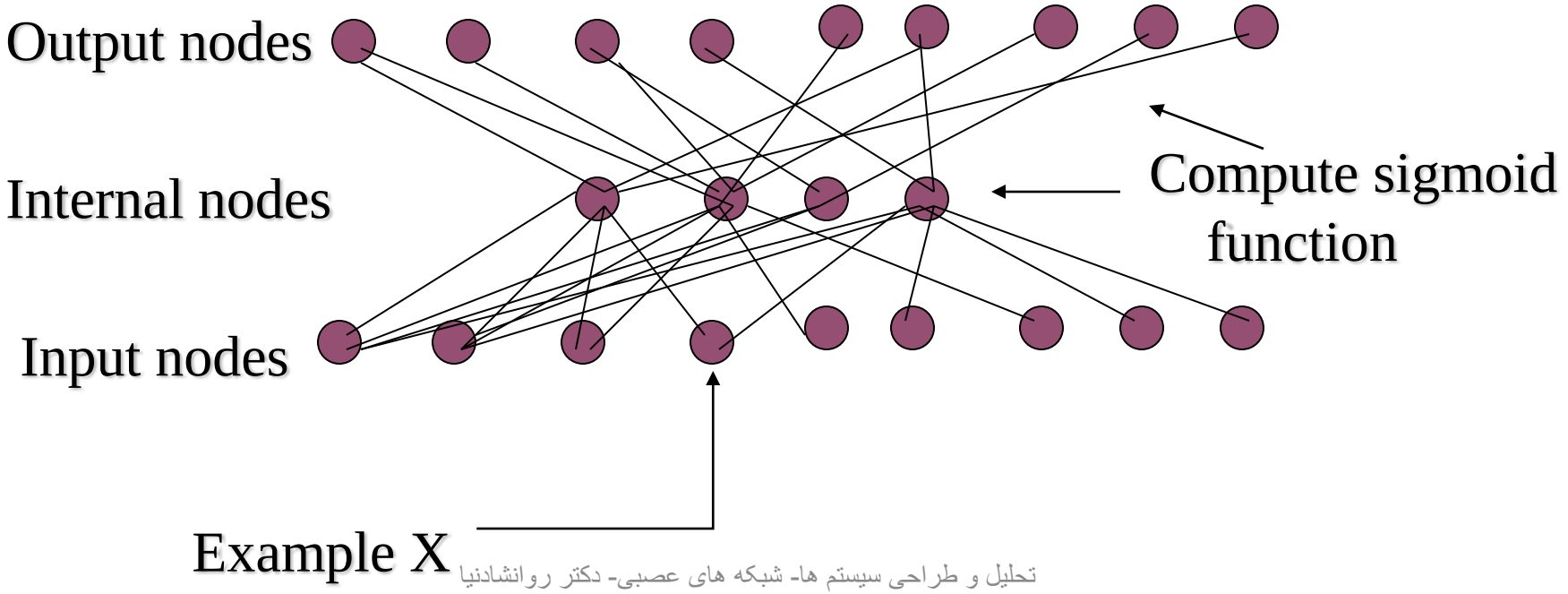
مثال X را به سمت جلو در شبکه انتشار دهید

خطای E را به سمت عقب در شبکه انتشار دهید.

هر مثال آموزشی بصورت یک زوج (x,t) ارائه میشود که بردار x مقادیر ورودی و بردار t مقادیر هدف برای خروجی شبکه را تعیین میکنند.

انتشار به سمت جلو

- برای هر مثال X مقدار خروجی هر واحد را محاسبه کنید تا به گره های خروجی برسید.



انتشار به سمت عقب

1. برای هر واحد خروجی جمله خطا را بصورت زیر محاسبه کنید:

$$\delta_k = O_k (1-O_k)(t_k - O_k)$$

1. برای هر واحد مخفی جمله خطا را بصورت زیر محاسبه کنید:

$$\delta_h = O_h (1-O_h) \Sigma_k W_{kh} \delta_k$$

1. مقدارهر وزن را بصورت زیر تغییر دهید:

$$W_{ji} = W_{ji} + \Delta W_{ji}$$

که در آن:

$$\Delta W_{ji} = \eta \delta_j X_{ji}$$

یادگیری نرخ از است عبارتη

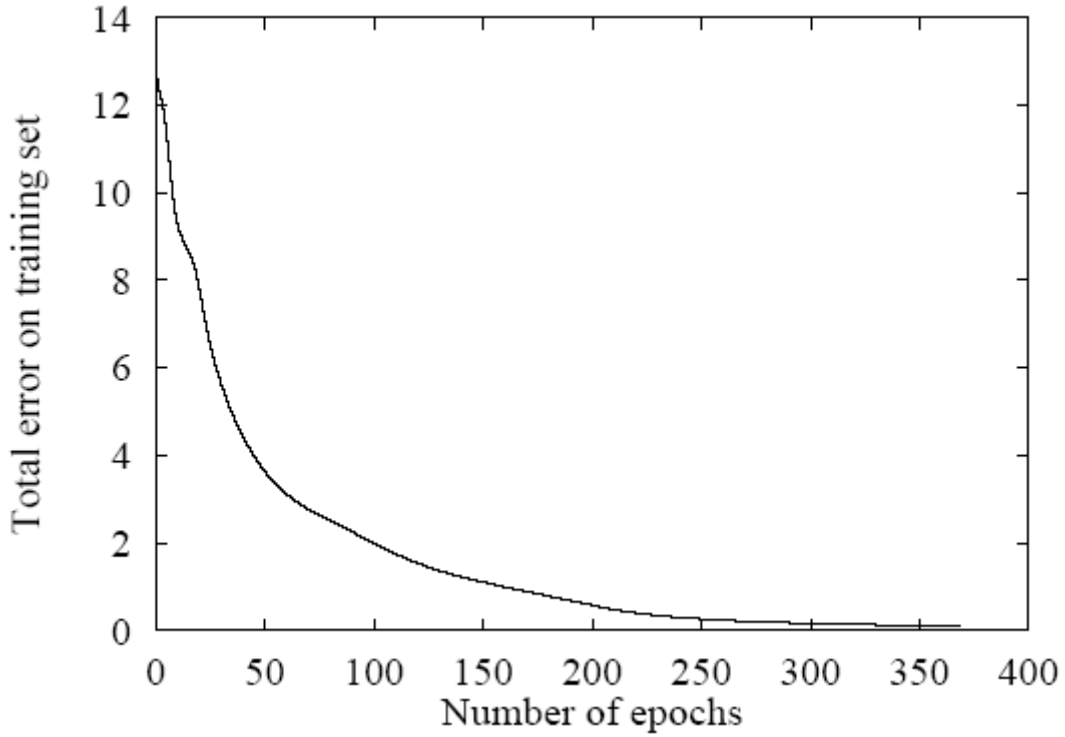
شرط خاتمه

معمولا الگوریتم BP پیش از خاتمه هزاران بار با استفاده همان داده های آموزشی تکرار میگردد شروط مختلفی را میتوان برای خاتمه الگوریتم بکار برد:

- توقف بعد از تکرار به دفعات معین
- توقف وقتی که خطا از یک مقدار تعیین شده کمتر شود.
- توقف وقتی که خطا در مثالهای مجموعه تائید از قاعده خاصی پیروی نماید.

اگر دفعات تکرار کم باشد خطا خواهیم داشت و اگر زیاد باشد مسئله Overfitting رخ خواهد داد.

منحنی یادگیری



مرور الگوریتم BP

- این الگوریتم یک جستجوی gradient descent در فضای وزنها انجام میدهد.
- ممکن است در یک مینیمم محلی گیر بیافتد
- در عمل بسیار موثر بوده است

برای پرهیز از مینیمم محلی روشهای مختلفی وجود دارد:

- افزودن ممنتم
- استفاده از stochastic gradient descent
- استفاده ازشبکه های مختلف با مقادیر متفاوتی برای وزنهای اولیه

افزودن ممنتم

- میتوان قانون تغییر وزنها را طوری در نظر گرفت که تغییر وزن در تکرار n ام تا حدی به اندازه تغییروزن در تکرار قبلی بستگی داشته باشد.

$$\Delta W_{ji}(n) = \eta\, \delta_j\, X_{ji} + \alpha \Delta W_{ji}(n-1)$$

که در آن مقدارممنتم α بصورت $0 <= \alpha <= 1$ میباشد

ممنتم عبارت — وزن تغییر قانون

افزودن ممنتم باعث میشود تا با حرکت در مسیر قبلی در سطح خطا:

- از گیر افتادن در مینیم محلی پرهیز شود
- از قرارگرفتن در سطوح صاف پرهیز شود
- با افزایش تدریجی مقدار پله تغییرات، سرعت جستجو افزایش یابد.

قدرت نمایش توابع

- گرچه قدرت نمایش توابع به توسط یک شبکه feedforward بسته به عمق و گستردگی شبکه دارد، با این وجود موارد زیر را میتوان به صورت قوانین کلی بیان نمود:
- توابع بولی :هر تابع بولی را میتوان توسط یک شبکه دو لایه پیاده سازی نمود.
- توابع پیوسته :هر تابع پیوسته محدود را میتوان توسط یک شبکه دو لایه تقریب زد .تئوری مربوطه در مورد شبکه هائی که از تابع سیگموئید در لایه پنهان و لایه خطی در شبکه خروجی استفاده میکنند صادق است.
- توابع دلخواه :هر تابع دلخواه را میتوان با یک شبکه سه لایه تا حد قابل قبولی تفریب زد.

فضای فرضیه و بایاس استقرا

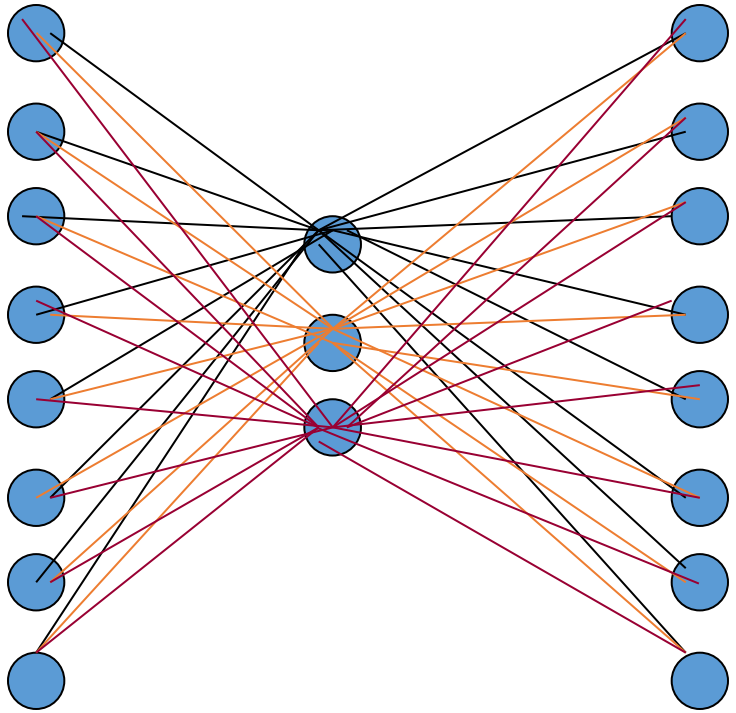
- فضای فرضیه مورد جستجو را میتوان بصورت یک فضای فرضیه اقلیدسی n بعدی از وزنهای شبکه در نظر گرفت(که nتعداد وزنهاست)
- این فضای فرضیه بر خلاف فضای فرضیه درخت تصمیم یک فضای پیوسته است.
- بایاس استقرا این روش را میتوان بصورت زیر بیان کرد:

“smooth interpolation between data points”

به این معنا که الگوریتم BP سعی میکند تا نقاطی را که به هم نزدیکتر هستند در یک دسته بندی قرار دهد.

نمایش لایه قدرت پنهان

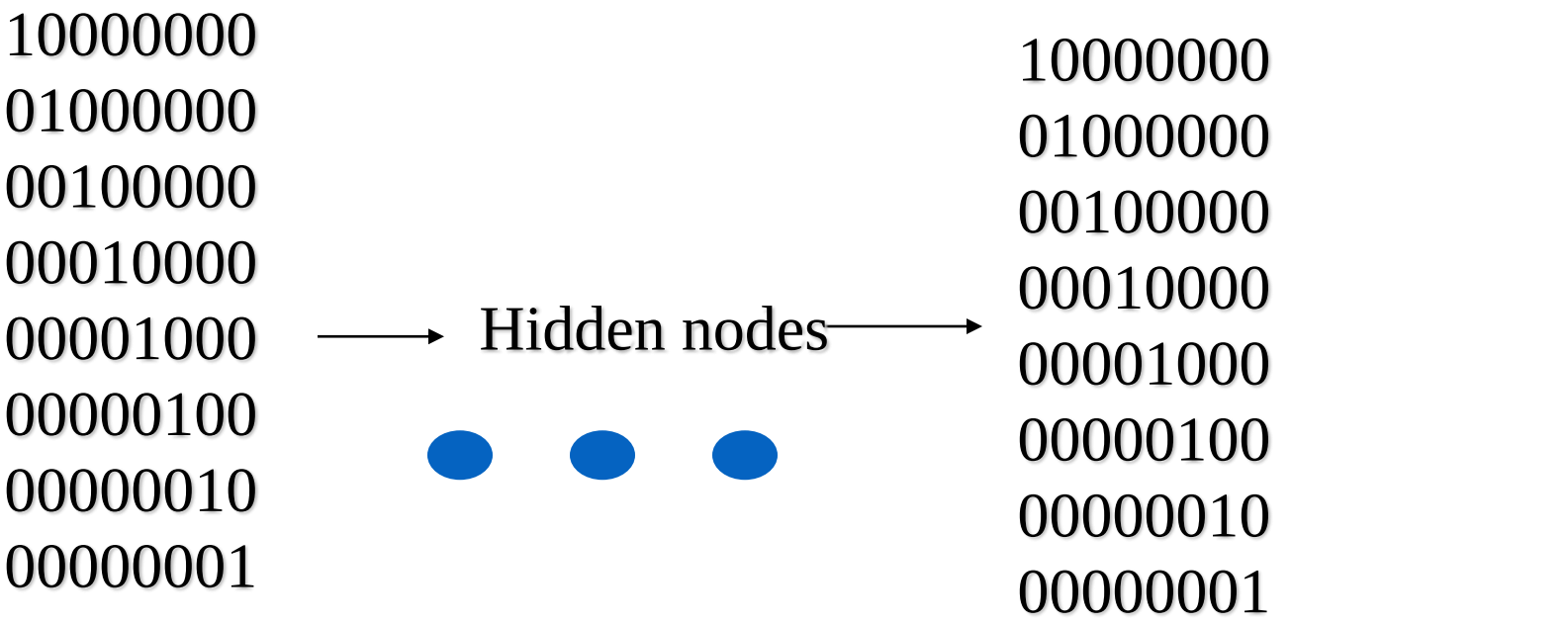
- یکی از خواص BP این است که میتواند در لایه های پنهان شبکه ویژگیهای نا آشکاری از داده ورودی نشان دهد.



برای مثال شبکه 8x3x8 زیر طوری آموزش داده میشود که مقدار هرمثال ورودی را عینا در خروجی بوجو د آورد (تابع f(x)=x را یاد بگیرد .)ساختار خاص این شبکه باعث میشود تا واحد های لایه وسط ویژگی های مقادیر ورودی را به نحوی کد بندی کنند که لایه خروحی بتواند از آنان برای نمایش مجدد داده ها استفاده نماید .

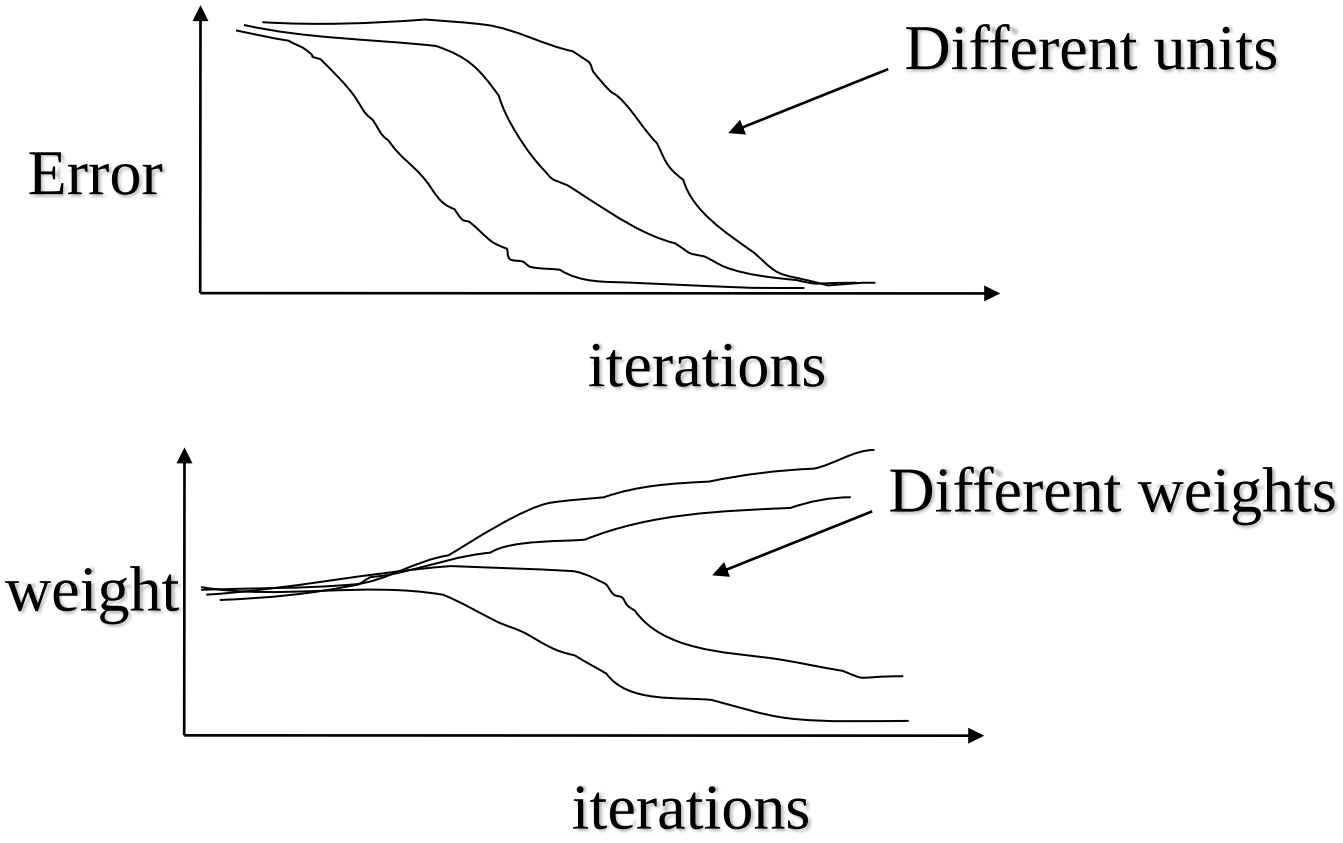
نمایش لایه قدرت پنهان

در این آزمایش که به تعداد 5000 بار تکرار شده از 8 داده مختلف به عنوان ورودی استفاده شده و شبکه با استفاده از الگوریتم BP موفق شده تا تابع هدف را بیاموزد.



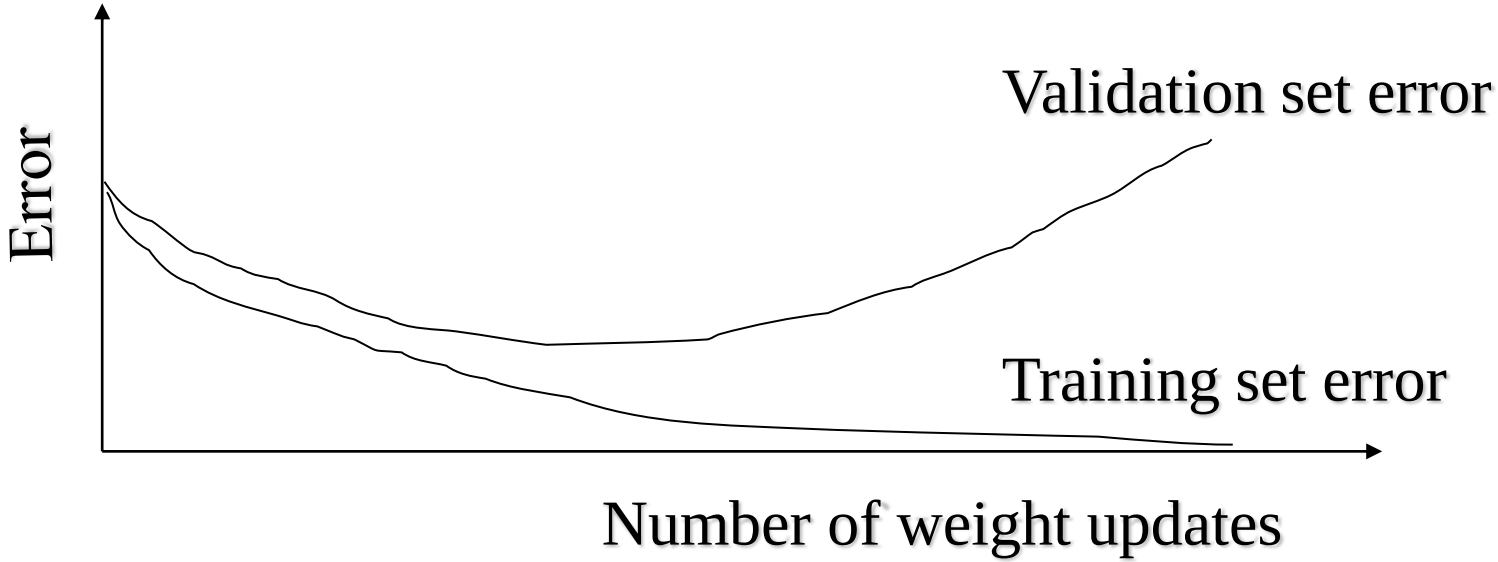
معادل حاصل بردار که میشود مشخص میانی لایه های واحد خروجی مشاهده با (000,001,,...,111) است بوده ورودی همهای داده استاندارد انکدینگ

نمودار خطا


Different units
Error
iterations
Different weights
weight
iterations

قدرت تعمیم وoverfitting

- شرط پاین الگوریتم BP چیست؟
- یک انتخاب این است که الگوریتم را آنقدر ادامه دهیم تا خطا از مقدار معینی کمتر شود .این امر میتواند منجر به overfittingشود.



دلایل رخ دادن overfitting

- overfittingناشی از تنظیم وزنها برای در نظر گرفتن مثالهای نادری است که ممکن است با توزیع کلی داده ها مطابقت نداشته باشند .تعداد زیاد وزنهای یک شبکه عصبی باعث میشود تا شبکه درجه آزادی زیادی برای انطباق با این مثالها داشته باشد.
- با افزایش تعداد تکرار، پیچیدگی فضای فرضیه یادگرفته شده توسط الگوریتم بیشتر و بیشتر میشود تا شبکه بتواند نویز و مثالهای نادر موجود در مجموعه آموزش را بدرستی ارزیابی نماید.

راه حل

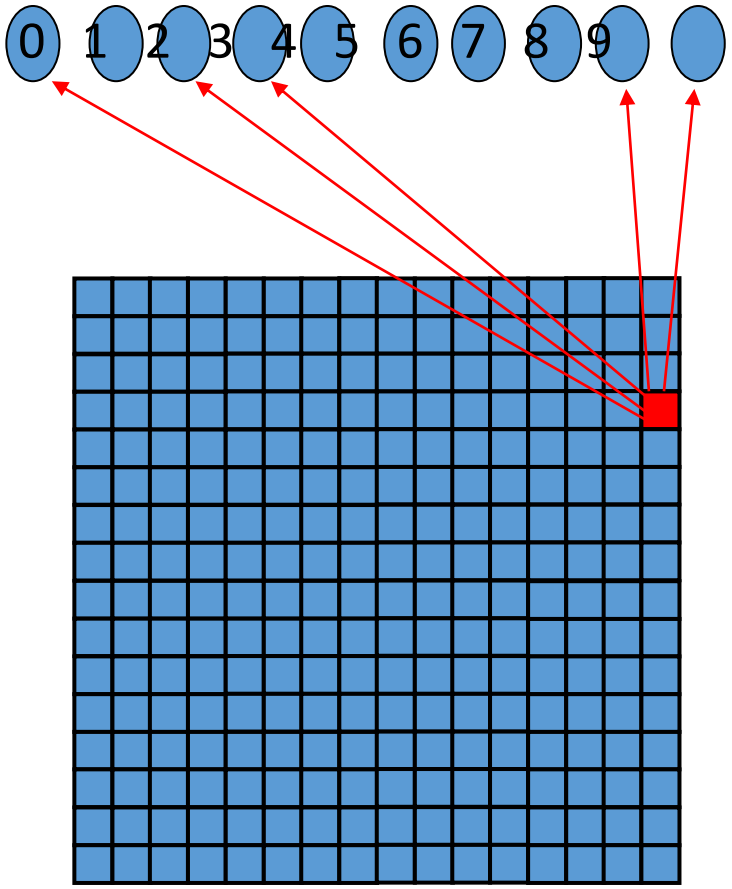
- استفاده از یک مجموعه تائید Vallidation و توقف یادگیری هنگامی که خطا در این مجموعه به اندازه کافی کوچک میشود.
- بایاس کردن شبکه برای فضاهای فرضیه ساده تر :یک راه میتواند استفاده ازweight decay باشد که در آن مقدار وزنها در هر بارتکرار باندازه خیلی کمی کاهش داده میشود.
- k-fold cross validationوقتی که تعداد مثالهای آموزشی کم باشد میتوان m داده آموزشی را به K دسته تقسیم بندی نموده و آزمایش را به تعداد k دفعه تکرار نمود .در هر دفعه یکی از دسته ها بعنوان مجموعه تست و بقیه بعنوان مجموعه آموزشی استفاده خواهند شد .تصمیم گیری براساس میانگین نتایج انجام میشود.

روشهای دیگر

راه های بسیار متنوعی برای ایجاد شبکه های جدید وجود دارد از جمله:

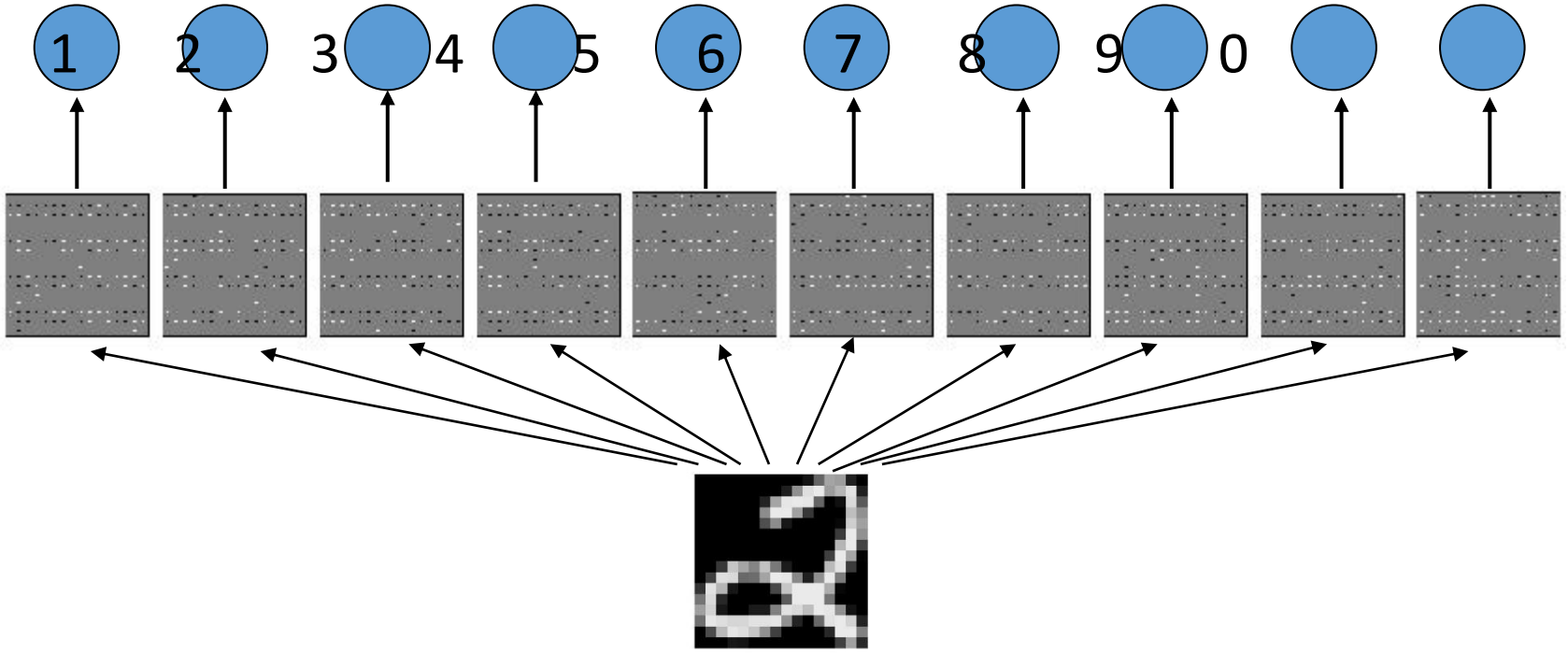
- استفاده از تعاریف دیگری برای تابع خطا
- استفاده از روشهای دیگری برای کاهش خطا در حین یادگیری
  - Hybrid Global Learning
  - Simulated Annealing
  - Genetic Algorithms
- استفاده از توابع دیگری در واحدها
  - Radial Basis Functions
- استفاده از ساختار های دیگری برای شبکه
  - Recurrent Network

مثال: تشخیص ارقام



- فرض کنید بخواهیم با استفاده از یک شبکه دو لایه ارقام دستنویس را تشخیص دهیم.
- نرونهای لایه اول شدت روشنائی پیکسلها را تقریب میزنندو
- نرونهای لایه آخر شکل ارقام را تعیین میکنند.

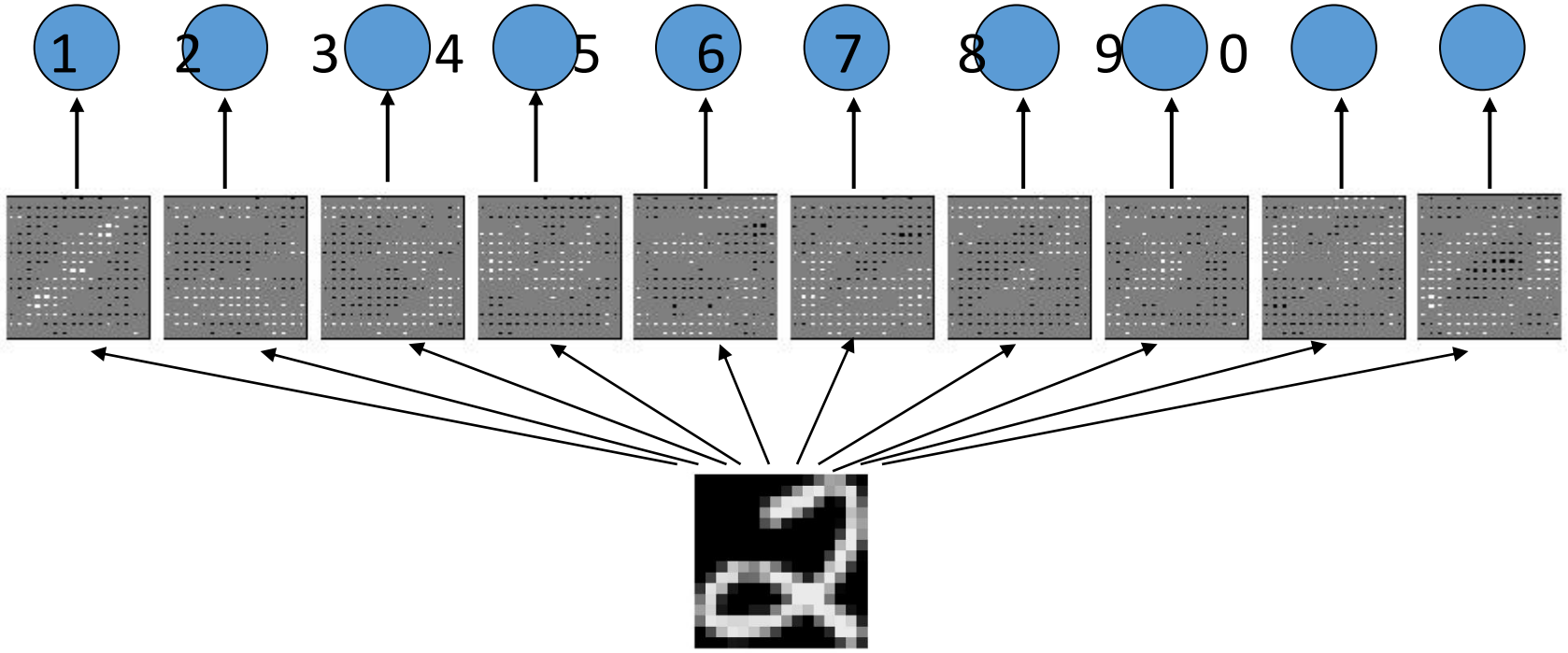
روشی که وزنها یاد گرفته میشوند:



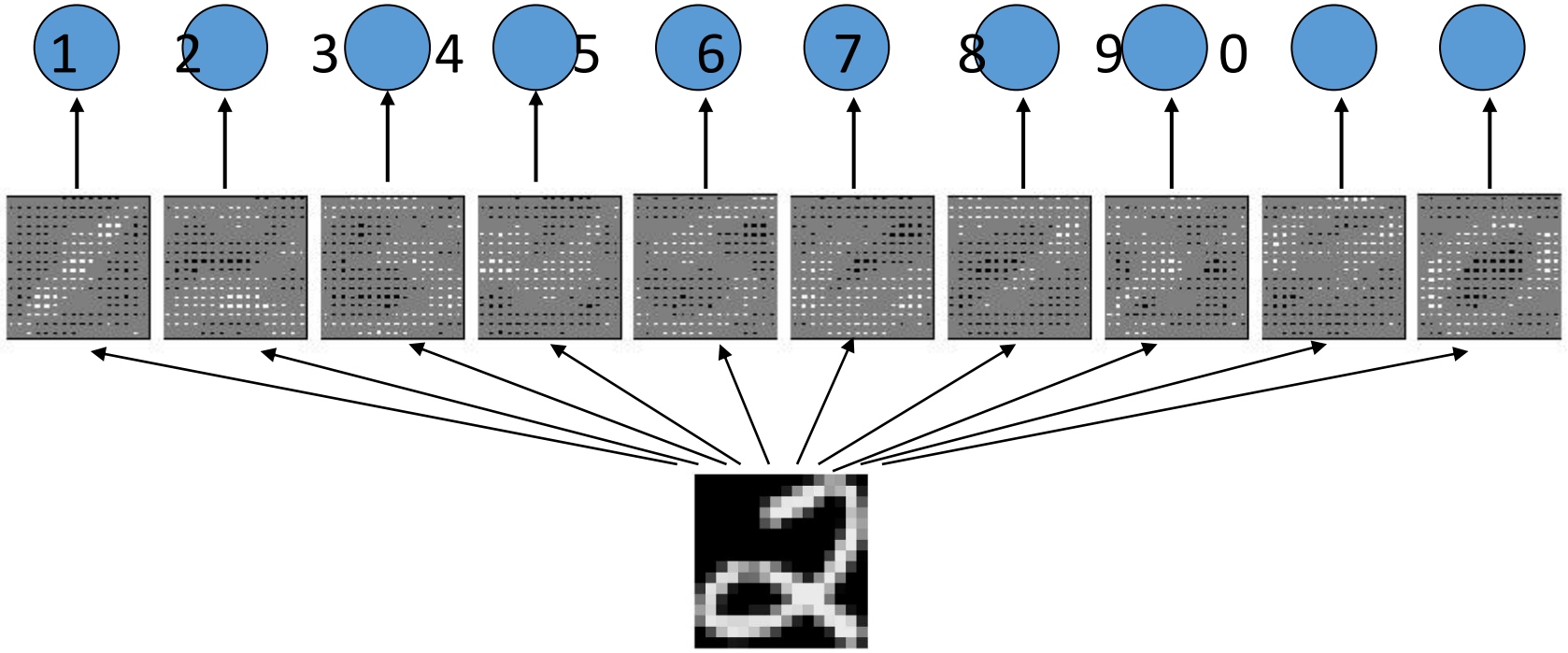
تصویر ورودی

تصویر به شبکه ارائه شده و وزنهای پیکسلهای فعال بتدریج اضافه میشوند. وزن پیکسلهای غیر موثر نیز بتدریج کاهش میابد.

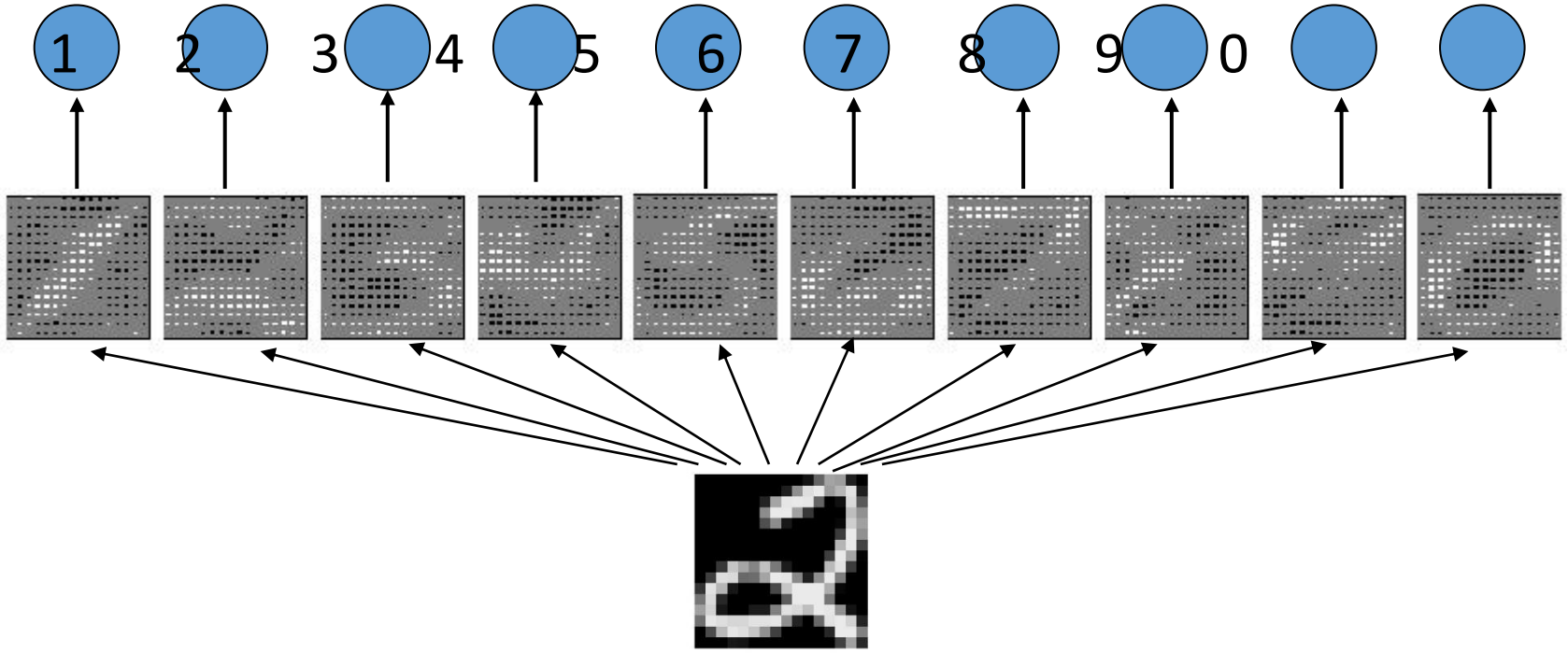
شکل گیری وزنها:



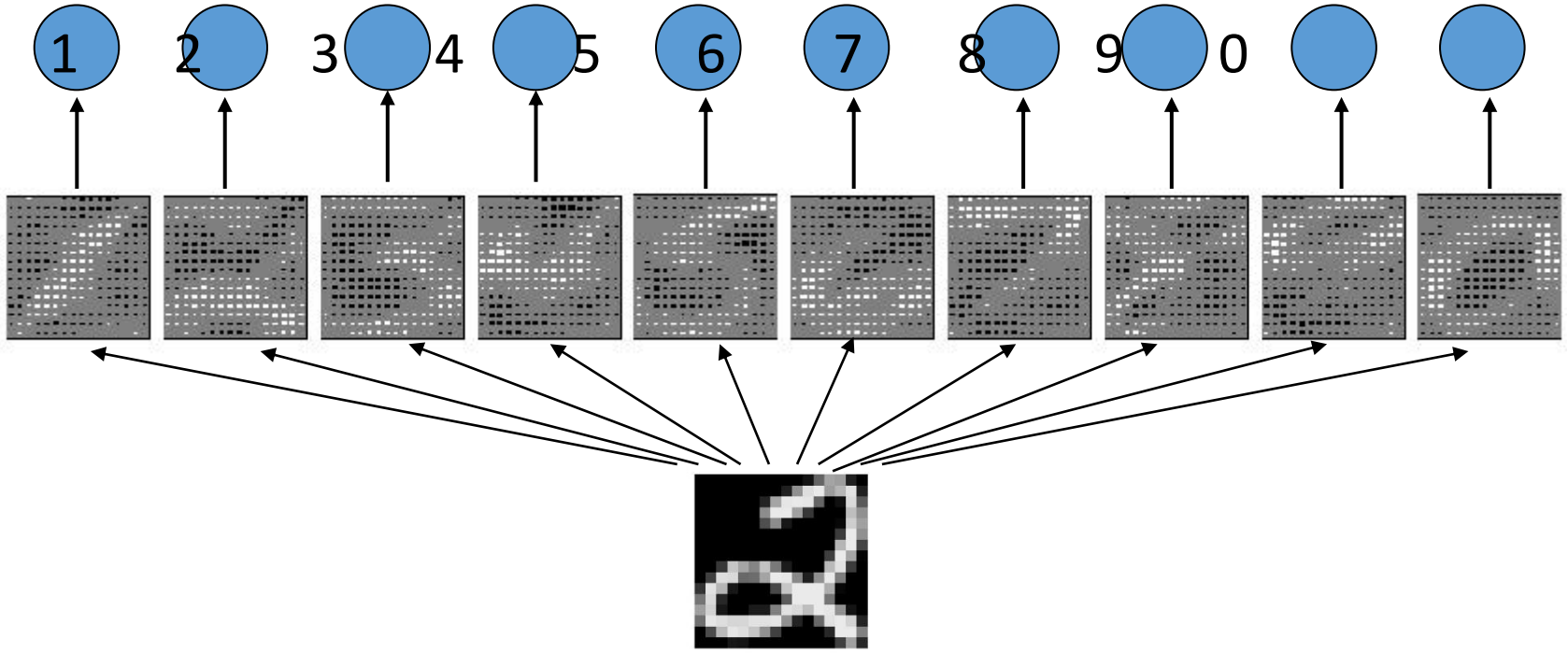
تصویر ورودی



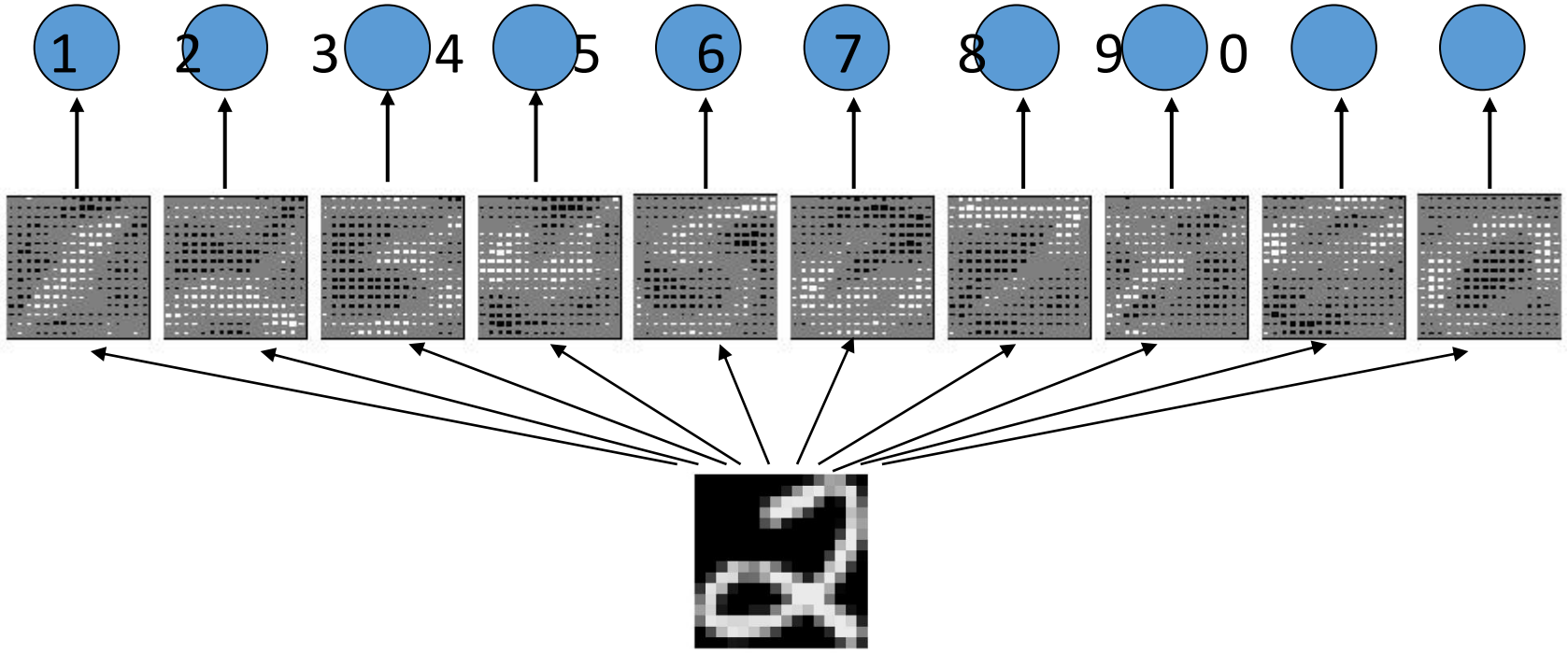
تصویر ورودی



تصویر ورودی

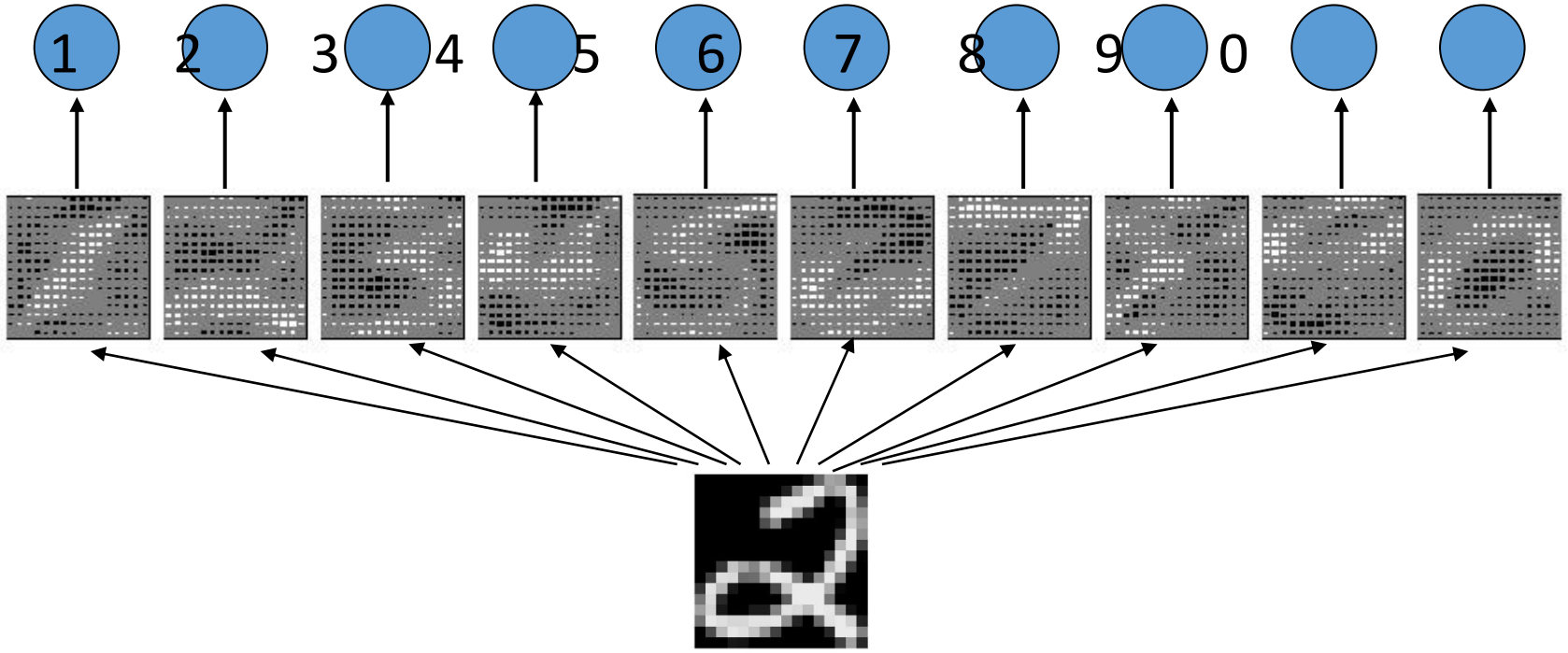


تصویر ورودی



تصویر ورودی

The learned weights

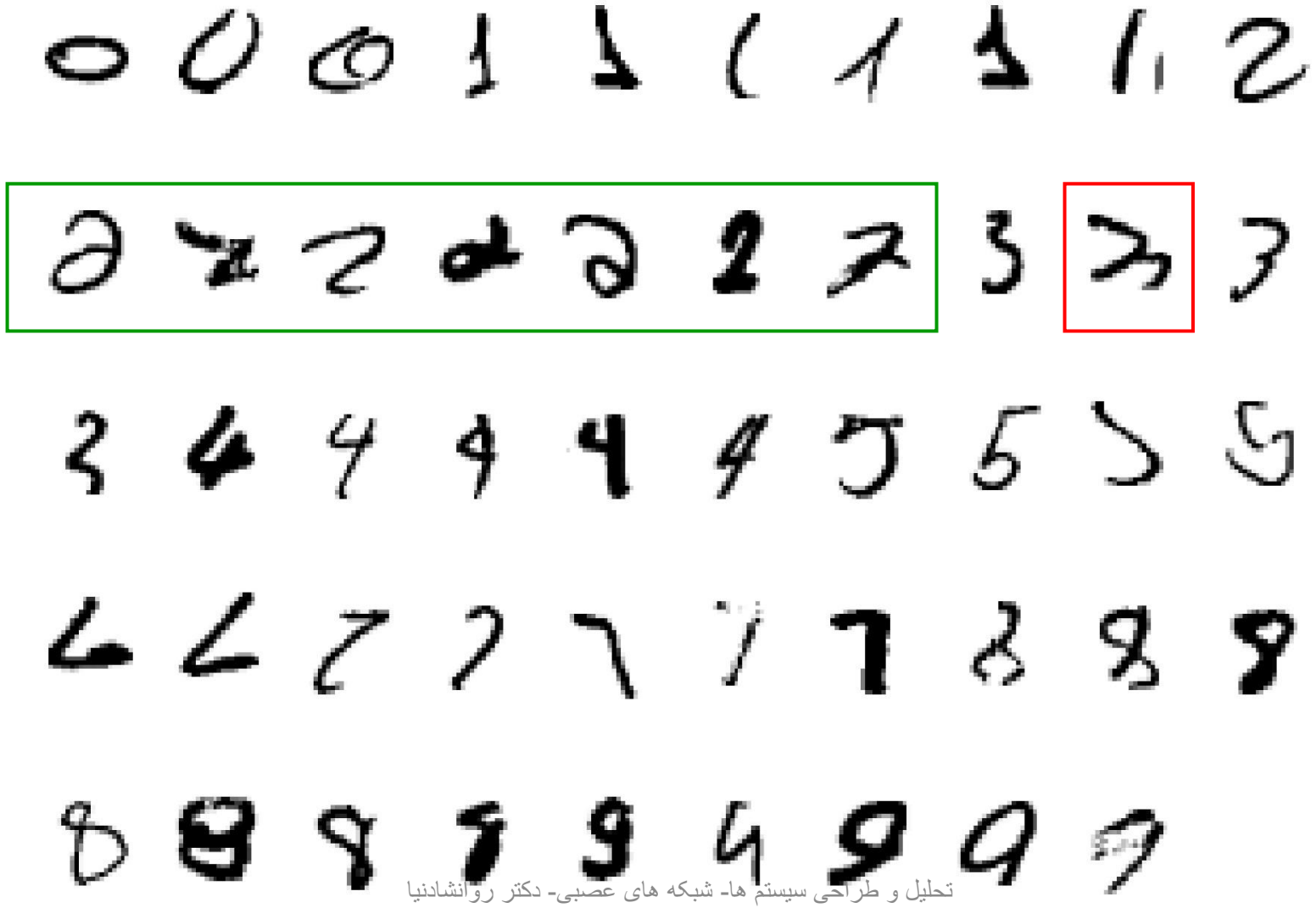


تصویر ورودی

شبکه چه چیزی را یاد میگیرد؟

- در این مثال یک شبکه با دو لایه معادل با استفاده از یک سری template یا قالب است که شبکه قالبی را که بهترین تطبیق با ورودی را داشته باشد بر میگزیند!
- اما برای مسئله ارقام دستنویس شکلهای ورودی بسیار متنوع هستند لذا یک قالب ساده که با همه ورودیها سازگار باشد وجود ندارد. در نتیجه چنین شبکه ای هم نمیتواند راه حل مسئله در حالت کلی باشد!
- برای اینکه بتوان مسئله را در حالت کلی حل نمود بایدشکل های ورودی به مجموعه ای از ویژگی ها تبدیل شده و شبکه را بر اساس ویژگی ها آموزش داد.

مثالی از تنوع ارقام دستنویس



شبکه عصبی فازی

سیستم استنتاج فازی عصبی تطبیق یافته یا ANFIS

- طراحی سیستماتیک یه مجموعه ی فازی با استفاده از داده های موجود:
- بهینه سازی رفتار عملی یک سیستم و نزدیک ساختن آن با یک الگوی واقعی: (مفهوم آموزش شبکه)

خطای آموزش: $E = \frac{1}{2} \sum_i (t_i - o_i)^2$

✓صفر کردن تابع خطا در حالت طبیعی ممکن نیست

- t_i(Target Value were gained from Real System)
- o_i (Output Value were gained from Trained Model)

روش بهینه سازی تابع خطا:

- بر مبنای روش گرادیان نزولی شرح داده شده:

$$E(\boldsymbol{\theta})=F(E|\boldsymbol{\theta})$$

$$\boldsymbol{\theta}= \boldsymbol{\theta}1, \boldsymbol{\theta}2,..., \boldsymbol{\theta}k$$

$\boldsymbol{\theta}$ پارامتر تنظیم تابع خطاست که براساس مینیم سازی مقدار تابع خطا تنظیم می شود. یعنی متغیر تصمیم فرآیند بهینه سازی زیر:

$$Min(E)$$

$$For\ \boldsymbol{\theta}$$

- ایرادات روش GD

مفهوم مدلسازی فازی:

- حال اگر سیستم تقریب خروجی ها (Target) برای تقریب متغیرهای فازی در نظر گرفته شود روال کمینه سازی مطرح شده به چه شکل خواهد بود؟
- اگر برای الگوی P مقدار تابع خطا برابرep و مقدار جوابهای Target برابر tp و برای رول r مقدار خروجی برابر yp باشد برای حالت یک متغیر ورودی داریم:

$$e_p{}^2 = \left(\frac{\sum_r (t_p - y_p) w_r)}{\sum_r w_r} \right)^2$$

خطای آموزش فازی:

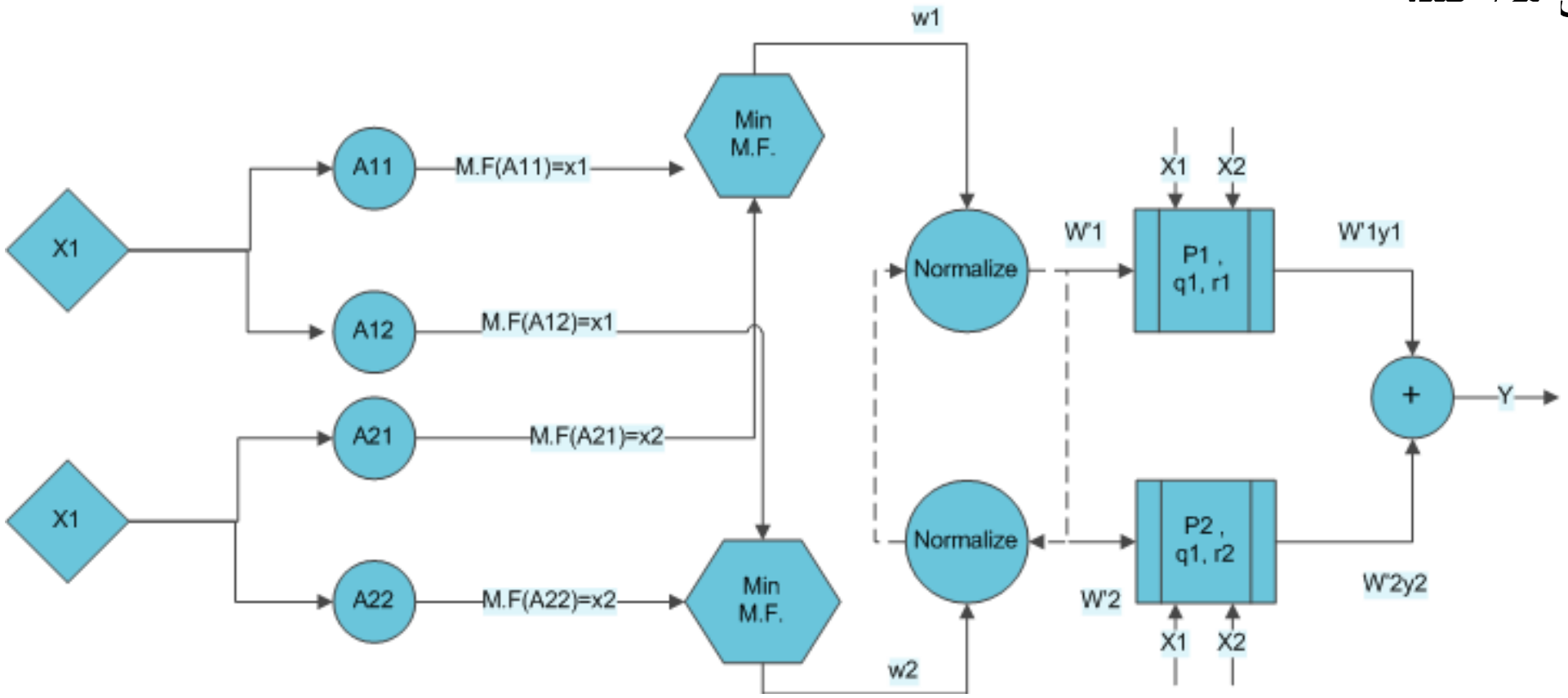
- تابع خطای تعریف شده همواره مشتق پذیر است و مقدار مشخصی دارد که با روال تکراری محاسبه ی آن پارامترهای تنظیم به سمت مشخصی تابع خطا (صفر) همگرا می کند. (تابع خطا را با مقدار مشخصی Adaptive می کنند)
- راهکارهای مختلفی برای پیاده سازی مفهوم خطای فازی جهت ساخت مدل یادگیری داده ها ارائه شده است که ساخت یافته ترین آنها مدل

Adaptive Network-based (Nueral-Fuzzy) Fuzzy Inference System (ANFIS)

است که در مرجع اصلی آن به این نام معرفی شده است.

سیستم استنتاج فازی عصبی تطبیق یافته

- ANFIS یک سیستم فازی درجه ی یک است که به طور مفهمومی منطبق بر مدل ساختاری زیر عمل مـ ـ کند.



توضیح مدل مفهومی ANFIS

در مدل مفهمومی ارائه شده جهت شناخت رفتار مدل ANFIS عملا لایه های زیر برای عملکرد سیستم وجود دارد:

- لایه ی Fuzzifier: به منظور انجام محاسبات درجات انطباق یا Membership Function.
- لایه ی Infliction: لایه اعمال قید جهت استنتاج مساله
- لایه نرمال سازی اوزان اولیه
- لایه محاسبه ی خروجی های وزن دار شده
- لایه خطی آخر جهت اخذ خروجی نهایی

نکات حائز اهمیت در مدل مفهومی

- پارامترهای محاسبه شده در مدل و محل قرارگیری آنها خصوصا در مورد مرکز M.F. ها بسیار در خروجی اثرگذار است.

- مشخصات رول ها و پارامترهای لایه محاسبه ی خروجی های وزن دار شده

- تمامی این پارامترهای تعریف شده با همان قاعده ی گرادیان نزولی و براساس مفهوم بهینه سازی آموزش بدست آمده و عملکرد شبکه ی ترسیم شده را تنظیم می کنند.

الگوریتم عملکرد ANFIS:

1. دریافت ورودی ها از بین مجموعه ی داده ها
2. تعریف توابع عضویت فازی برای مساله
3. اخذ درجه ی عضویت و اعمال اثر استلزام بر روی آنهار(فازی سازی)
4. نرمالیزه کردن اوزان اولیه
5. ایجاد خروجی های وزن دار (نافازی سازی)
6. جمع وزنهای بدست آمده
7. خروجی مساله

با آرزوی موفقیت شما
مهدی روانشادنیا