

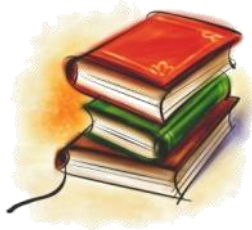
مبانی رایانش نرم

شبکه‌های عصبی: پرسپترون چند لایه (پس انتشار خطا)

هادی ویسی

h.veisi@ut.ac.ir

دانشگاه تهران - دانشکده علوم و فنون نوین



○ شبکه‌های عصبی پس‌انتشار (پرسپترون چند لایه)

- ساختار و الگوریتم آموزش

- نحوه استخراج قوانین یادگیری

- کاربردها و مثال‌ها

- نکات کاربردی

- مقداردهی اولیه به وزن‌ها و بایاس‌ها

- تعداد لایه‌های مخفی

- مدت زمان آموزش

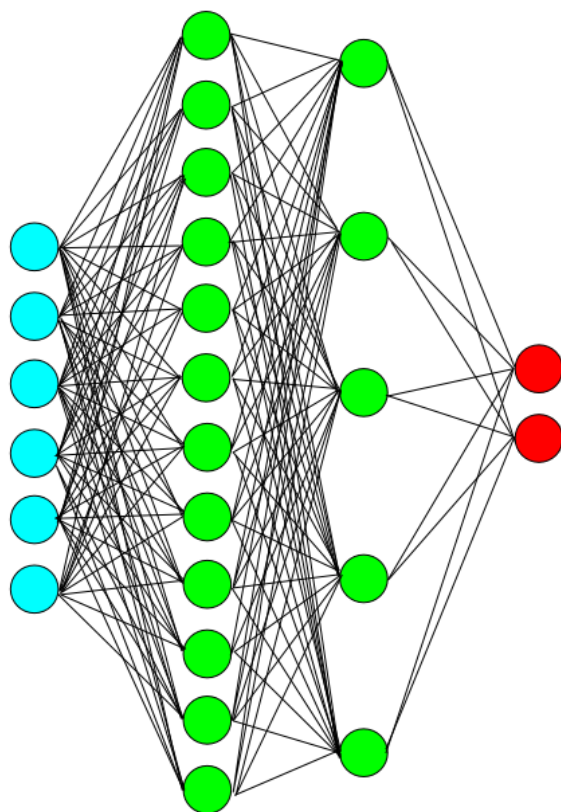
- نمایش داده‌ها

- مقدار داده آموزش

- روش‌های به‌روز کردن وزن‌ها (آموزش)

- توابع فعال‌سازی

- تقریب‌زننده جهانی (قضیه)





شبکه‌های پس‌انتشار ...

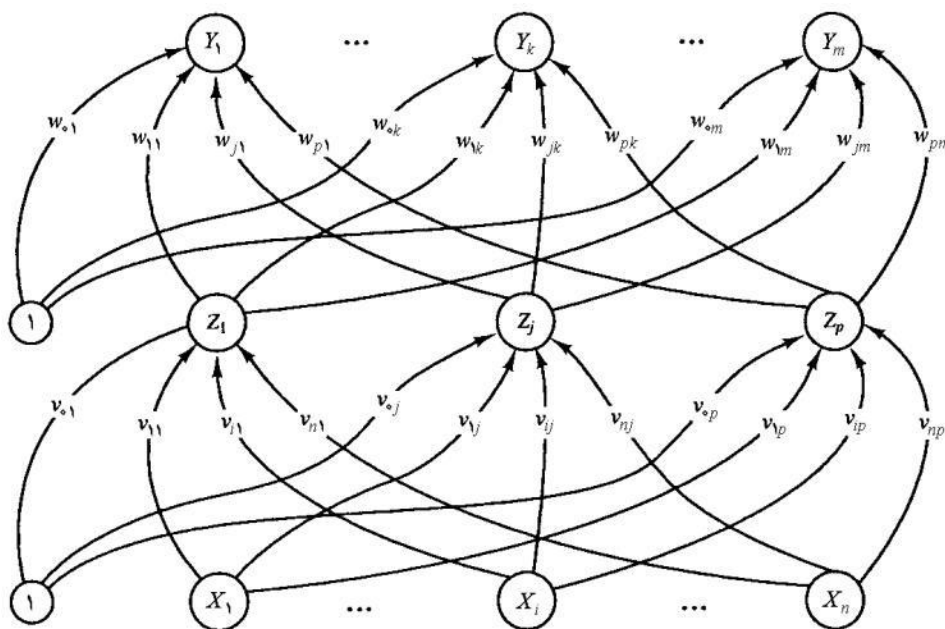
○ آموزش با پس‌انتشار خطا (Error Back-propagation)

- قانون دلتای تعمیم‌یافته (Generalized Delta Rule)
- روش کاهش گرادیان برای به حداقل رساندن کل مربعات خطای خروجی
- هدف آموزش شبکه با پس‌انتشار، رسیدن به تعادل بین قابلیت یادگیری و تعمیم است.
 - قابلیت یادگیری = پاسخگویی صحیح به الگوهای ورودی به کار رفته برای آموزش
 - تعمیم = پاسخدهی منطقی (خوب) به ورودی‌های شبیه اما نه دقیقاً یکسان با ورودی‌های آموزش
- آموزش شامل سه مرحله:
 - پیش‌خور کردن الگوی آموزش ورودی (Feed-forward)
 - محاسبه و پس‌انتشار کردن خطای مربوط
 - تنظیم وزن‌ها
- شبکه عصبی پرسپترون چندلایه (MLP: Multi-Layer Perceptron)

شبکه‌های پس‌انتشار: ساختار

○ شبکه سه لایه

- یک لایه ورودی (واحدهای X).
- یک لایه واحدهای مخفی (واحدهای Z).
- یک لایه خروجی (واحدهای Y).





شبکه‌های پس‌انتشار: الگوریتم آموزش ...

○ مراحل

- پیش‌خور کردن الگوی آموزش ورودی
- پس‌انتشار خطای مربوط
- تنظیم وزن‌ها

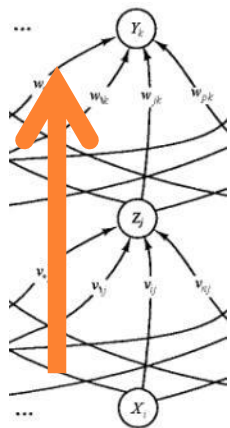
- مبانی ریاضی الگوریتم پس‌انتشار = بهینه‌سازی کاهش گرادیان (Gradient Descent)
 - گرادیان (شیب) یک تابع = نمایانگر جهتی که تابع در آن سریع‌تر افزایش می‌یابد
 - شیب با علامت منفی = جهتی نشان دهنده کاهش سریع‌تر آن تابع
 - در اینجا تابع مورد نظر = تابع خطای شبکه
 - متغیرهای مورد نظر = وزن‌های شبکه

شبکه‌های پس‌انتشار: الگوریتم آموزش ...

- مرحله ۰ - به وزن‌ها مقدار اولیه بدهید (مقادیر تصادفی کوچک را انتخاب کنید).
- مرحله ۱ - تا زمانی که شرایط توقف برقرار نیست، مراحل ۲ تا ۹ را انجام دهید.
- مرحله ۲ - برای هر جفت آموزش (مقادیر ورودی و هدف)، مراحل ۳ تا ۸ را انجام دهید.

پیش‌خور

- مرحله ۳ - ارسال سیگنال ورودی x_i به تمام واحدها در لایه بعدی (واحدهای مخفی)
- مرحله ۴ - محاسبه ورودی واحدهای مخفی و اعمال تابع فعال‌سازی

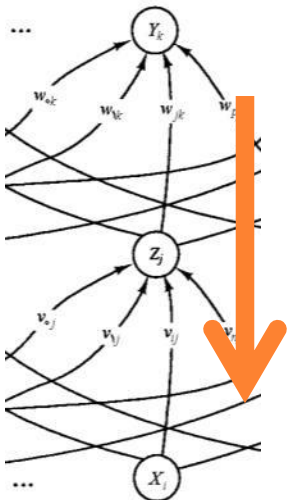


$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad z_j = f(z_in_j)$$

- مرحله ۵ - محاسبه ورودی واحدهای خروجی و اعمال تابع فعال‌سازی

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad y_k = f(y_in_k)$$

شبکه‌های پس انتشار: الگوریتم آموزش ...



○ پس انتشار خطا

- مرحله ۶- محاسبه خطا برای واحدهای خروجی (استفاده از الگوی هدف)

$$\delta_k = (t_k - y_k) f'(y_{in_k})$$

محاسبه پارامتر تصحیح وزن (بعداً در به روز کردن به کار می رود) $\Delta w_{jk} = \alpha \delta_k z_j$

محاسبه پارامتر تصحیح بایاس (بعداً در به روز کردن به کار می رود) $\Delta w_{0k} = \alpha \delta_k$
ارسال δ_k (مقادیر دلتا) به واحدهای لایه قبلی (لایه مخفی)

- مرحله ۷- دریافت ورودی‌های دلتا توسط واحدهای مخفی از واحدهای خروجی

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

ضرب در مشتق تابع فعال سازی جهت محاسبه پارامتر مربوط به اطلاعات خطا

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

محاسبه مقدار تصحیح وزن و بایاس (استفاده در به روز کردن)

$$\Delta v_{ij} = \alpha \delta_j x_i$$

$$\Delta v_{0j} = \alpha \delta_j$$



شبکه‌های پس‌انتشار: الگوریتم آموزش

○ به‌روز کردن وزن‌ها و بایاس‌ها

• مرحله ۸ - به‌روز کردن وزن‌ها و بایاس‌های واحدهای خروجی

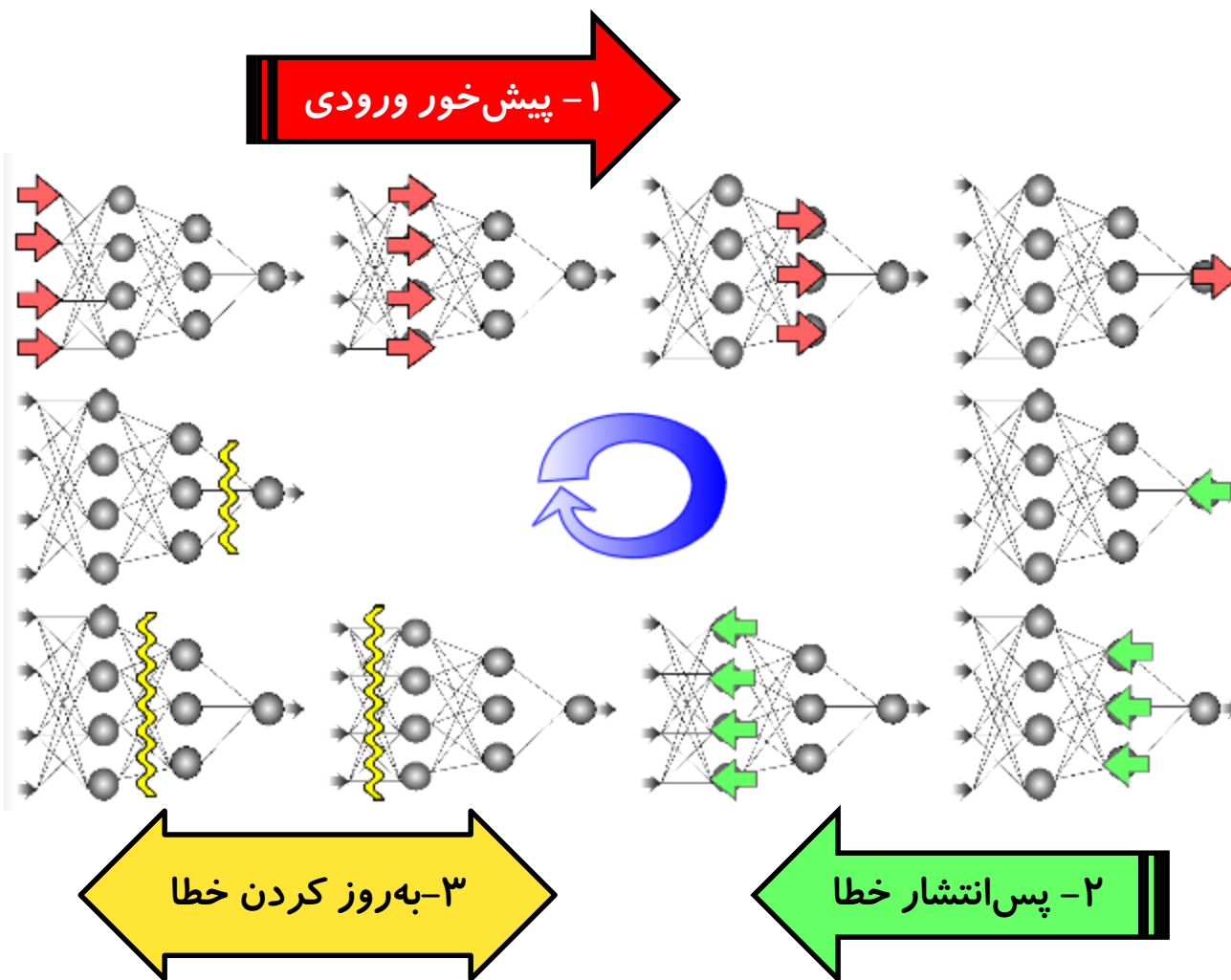
$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

به‌روز کردن وزن‌ها و بایاس‌های واحدهای مخفی

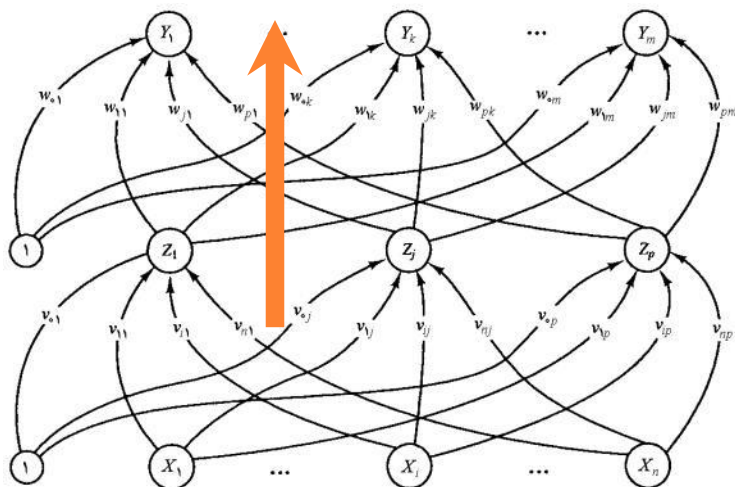
$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

• مرحله ۹ - شرایط توقف را بررسی کنید.

شبکه‌های پس انتشار: الگوریتم آموزش (مرور)



شبکه‌های پس‌انتشار: کاربرد



○ بعد از آموزش

- فقط مرحله پیش‌خور مورد نیاز است

- مرحله ۰: مقادیر وزن‌های شبکه را با استفاده از الگوریتم آموزش تعیین کنید.
- مرحله ۱: برای هر بردار ورودی، مراحل ۲ تا ۴ را انجام دهید.
- مرحله ۲: برای تمام نرون‌های ورودی، فعال‌سازی واحد ورودی را تعیین کنید،
- مرحله ۳: برای واحدهای مخفی:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \Rightarrow z_j = f(z_in_j)$$

- مرحله ۴: برای واحدهای خروجی:

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \Rightarrow y_k = f(y_in_k)$$

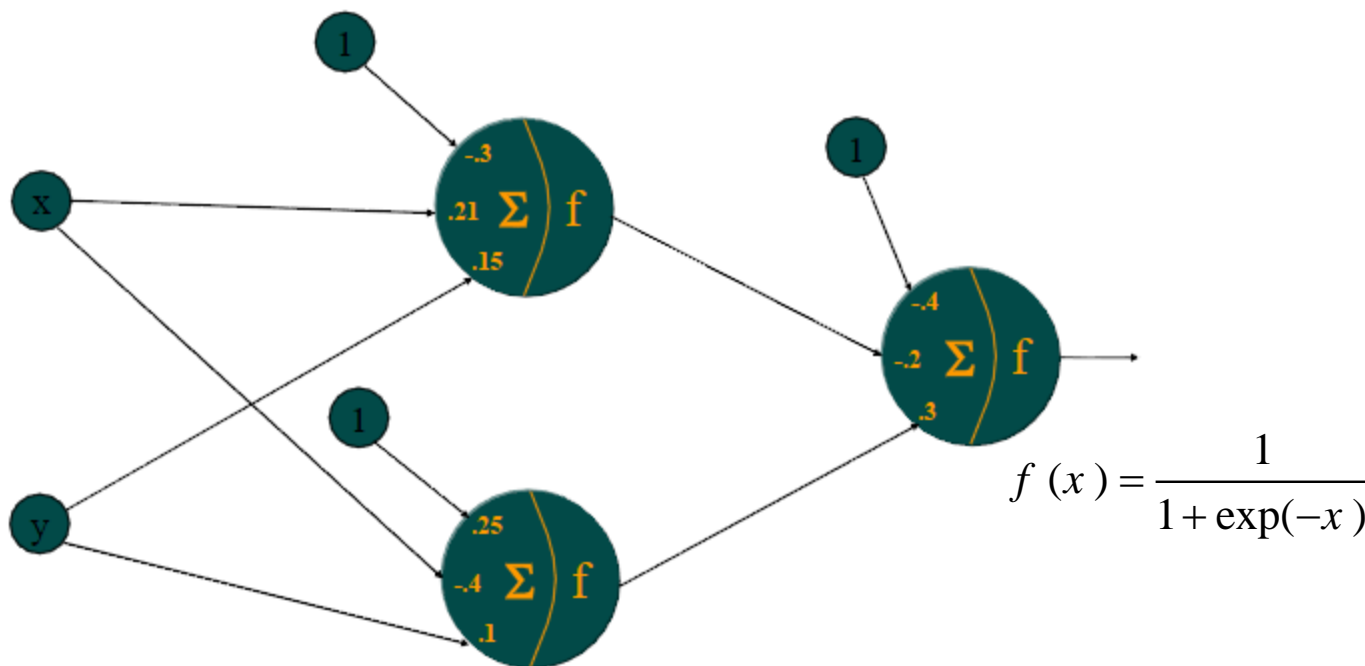


شبکه‌های پس‌انتشار: مثال ...

x_1	x_2	\rightarrow	y
1	1		0
1	0		1
0	1		1
0	0		0

○ تابع XOR: نمایش دودویی (۱ از ۶) ...

• مقدار دهی اولیه (تصادفی)

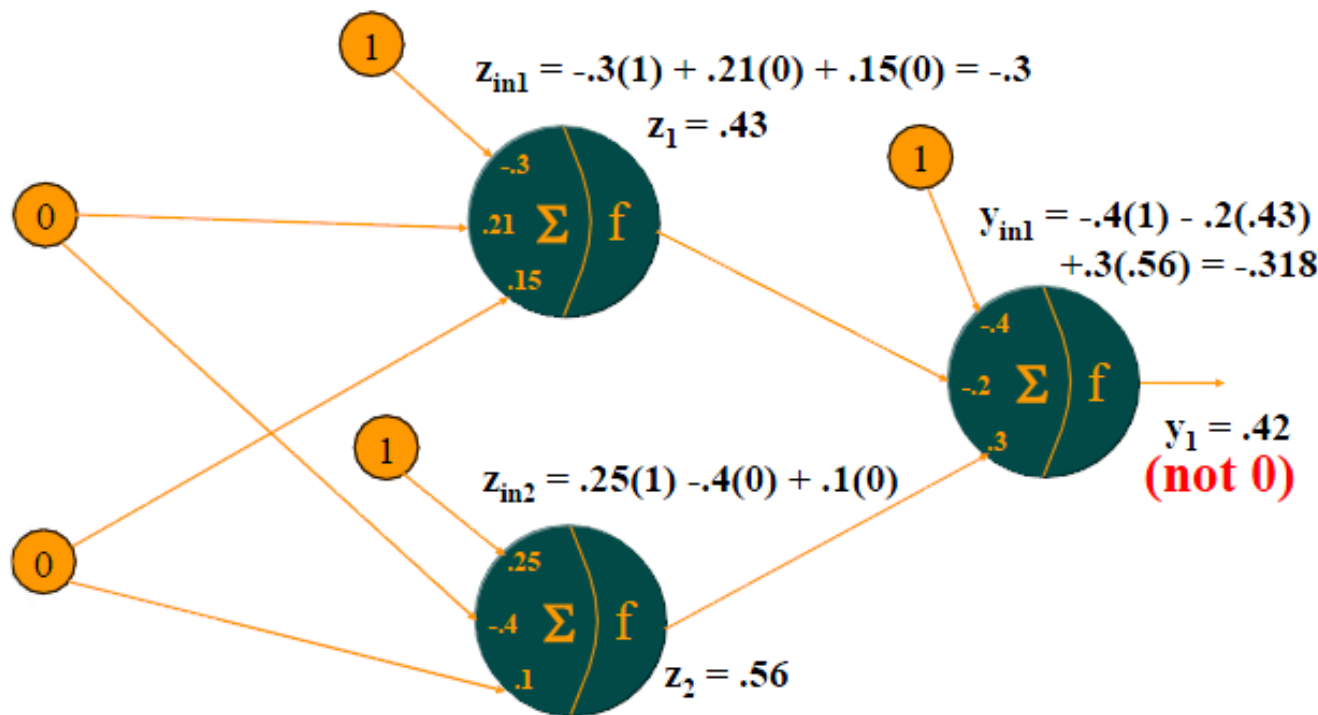


شبکه‌های پس‌انتشار: مثال ...

○ تابع XOR: نمایش دودویی (۲ از ۶) ...

• پیش‌خور کردن ورودی

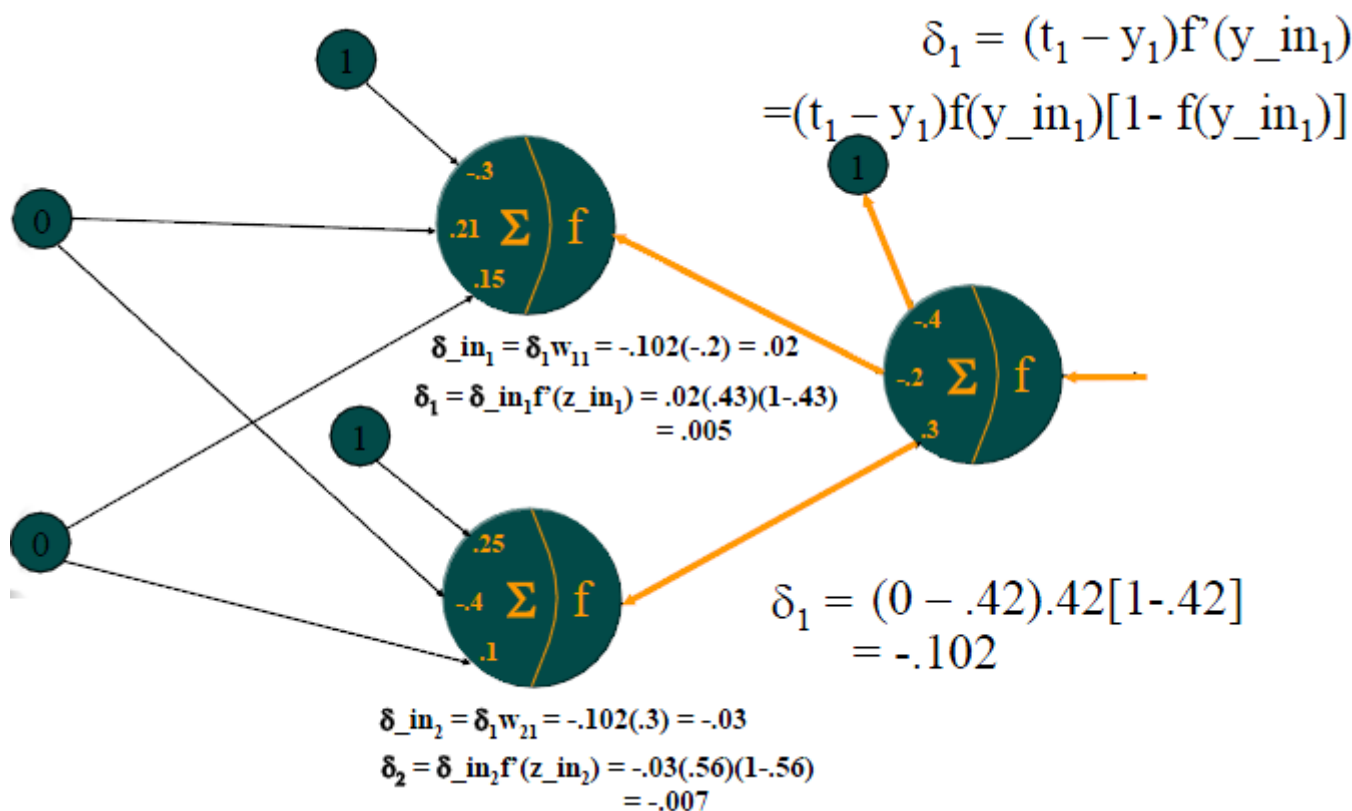
x_1	x_2	y
0	0	0



شبکه‌های پس‌انتشار: مثال ...

○ تابع XOR: نمایش دودویی (۳ از ۶) ...

• پس‌انتشار خطا



شبکه‌های پس انتشار: مثال ...

○ تابع XOR: نمایش دودویی (۴ از ۶) ...

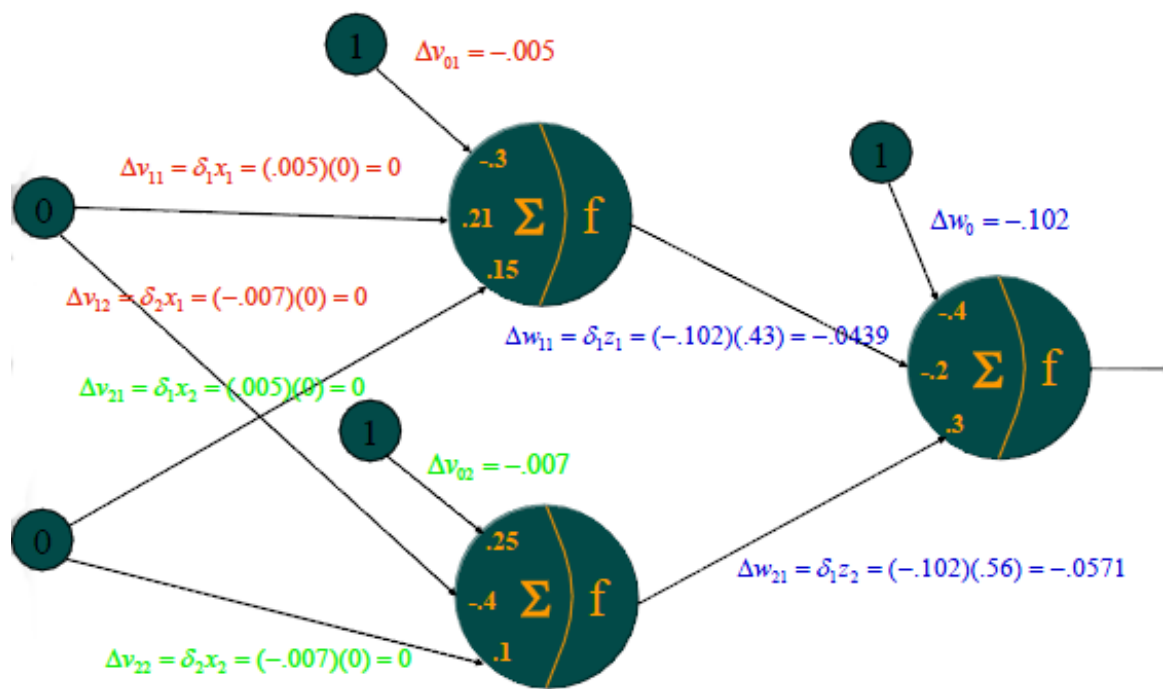
• محاسبه وزن‌ها

$$\Delta v_{ij} = \alpha \delta_j x_i \quad j = 1, 2$$

$$\Delta v_{0j} = \alpha \delta_j$$

$$\Delta w_{j1} = \alpha \delta_1 z_j \quad j = 1, 2$$

$$\Delta w_0 = \alpha \delta_1$$

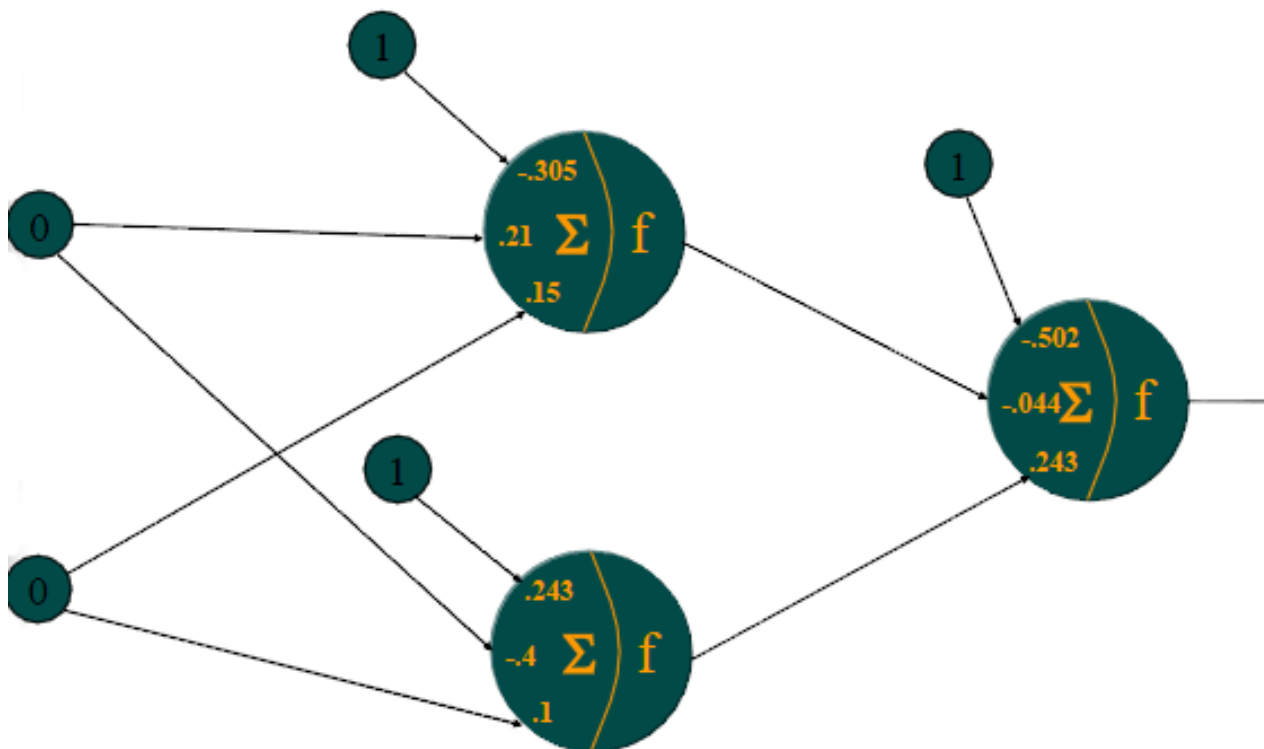




شبکه‌های پس‌انتشار: مثال ...

○ تابع XOR: نمایش دودویی (۵ از ۶) ...

• به‌روز کردن وزن‌ها

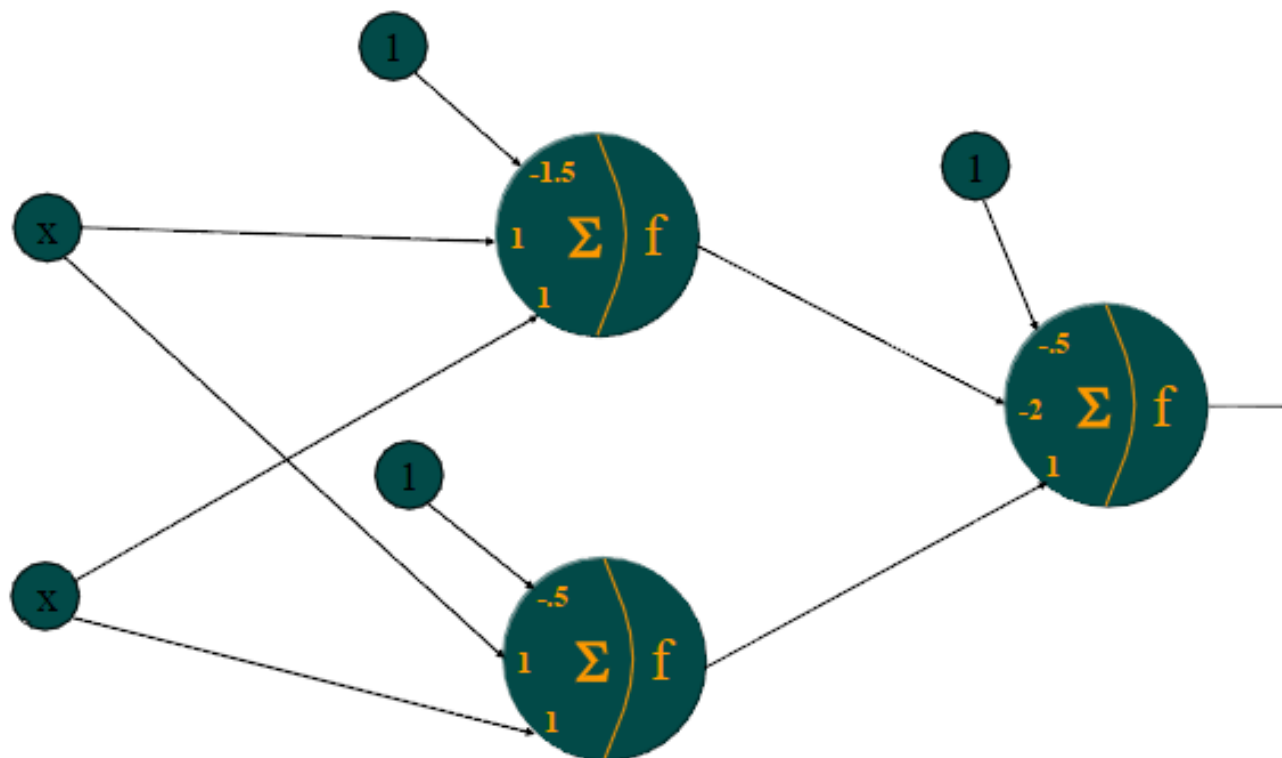


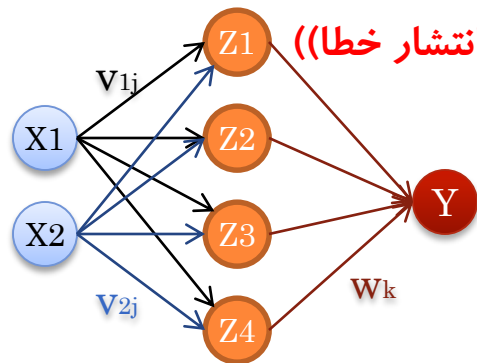


شبکه‌های پس‌انتشار: مثال ...

○ تابع XOR: نمایش دودویی (۶ از ۶)

• وزن‌های نهایی (بعد از ۵۰۰ تکرار)





شبکه‌های پس انتشار: مثال ...

تابع XOR: نمایش دودویی

- ساختار ۱-۴-۲ (۲ واحد ورودی، ۴ واحد مخفی در یک لایه مخفی، ۱ واحد خروجی)

- وزن‌های اولیه تصادفی

-0,3378 0,2771 0,2859 -0,3329

- بایاس‌های چهار واحد مخفی

0,1970 0,3191 -0,1448 0,3594

- وزن‌های اولین واحد ورودی به لایه مخفی

0,3099 0,1904 -0,0347 -0,4861

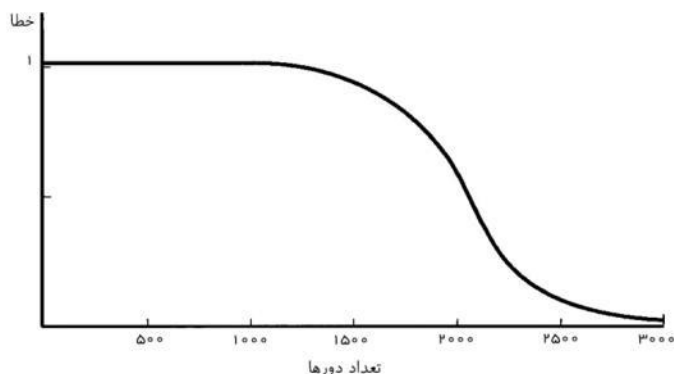
- وزن‌های دومین واحد ورودی به لایه مخفی

-0,1401 0,4919 -0,2913 -0,3979 0,3581

- وزن‌های (و بایاس) واحدهای مخفی به واحد خروجی

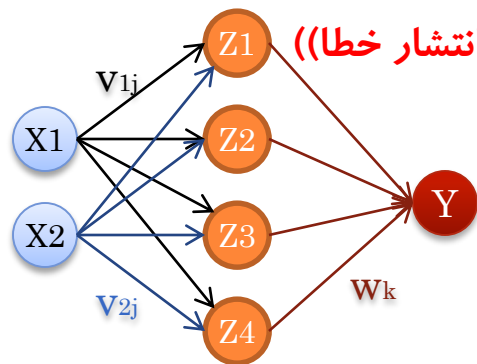
- نرخ یادگیری = ۰.۰۲

- ادامه آموزش تا زمانی که کل مربعات خطا برای چهار الگوی آموزش کمتر از ۰.۰۵ باشد



- آموزش نسبتاً آهسته

- تقریباً در ۳۰۰۰ دور



شبکه‌های پس انتشار: مثال ...

تابع XOR: نمایش دوقطبی

- ساختار ۱-۴-۲ (۲ واحد ورودی، ۴ واحد مخفی در یک لایه مخفی، ۱ واحد خروجی)
- وزن‌های اولیه تصادفی

-0,3378 0,2771 0,2859 -0,3329

○ بایاس‌های چهار واحد مخفی

0,1970 0,3191 -0,1448 0,3594

○ وزن‌های اولین واحد ورودی به لایه مخفی

0,3099 0,1904 -0,0347 -0,4861

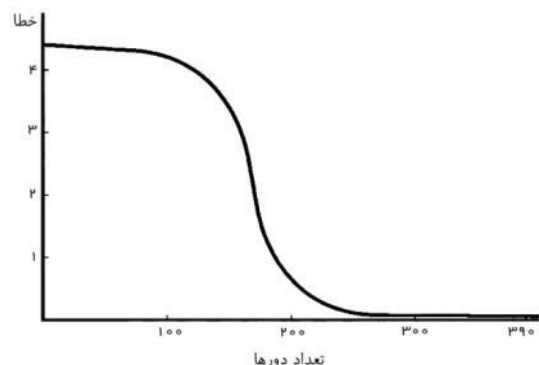
○ وزن‌های دومین واحد ورودی به لایه مخفی

-0,1401 0,4919 -0,2913 -0,3979 0,3581

○ وزن‌های (و بایاس) واحدهای مخفی به واحد خروجی

- نرخ یادگیری = ۰.۰۲

- ادامه آموزش تا زمانی که کل مربعات خطا برای چهار الگوی آموزش کمتر از ۰.۰۵ باشد

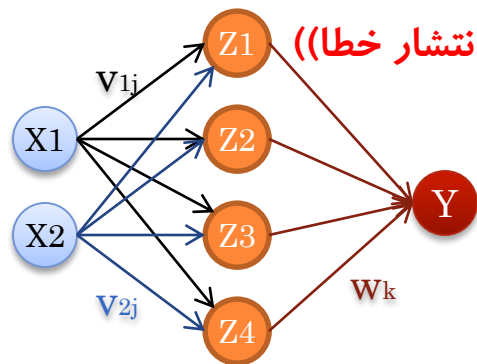


- آموزش سریع‌تر به نسبت حالت دودویی

- آموزش در ۳۸۷ دور



درس: مبانی رایانش نرم - شبکه‌های عصبی (پرسپترون چند لایه (پس‌انتشار خطا))



شبکه‌های پس‌انتشار: مثال ...

تابع XOR: نمایش دوقطبی تغییر داده شده

- ساختار ۱-۴-۲ (۲ واحد ورودی، ۴ واحد مخفی در یک لایه مخفی، ۱ واحد خروجی)
- وزن‌های اولیه تصادفی

-0,3378 0,2771 0,2859 -0,3329

○ بایاس‌های چهار واحد مخفی

0,1970 0,3191 -0,1448 0,3594

○ وزن‌های اولین واحد ورودی به لایه مخفی

0,3099 0,1904 -0,0347 -0,4861

○ وزن‌های دومین واحد ورودی به لایه مخفی

-0,1401 0,4919 -0,2913 -0,3979 0,3581

○ وزن‌های (و بایاس) واحدهای مخفی به واحد خروجی

- نرخ یادگیری = ۰.۰۲
- ادامه آموزش تا زمانی که کل مربعات خطا برای چهار الگوی آموزش کمتر از ۰.۰۵ باشد

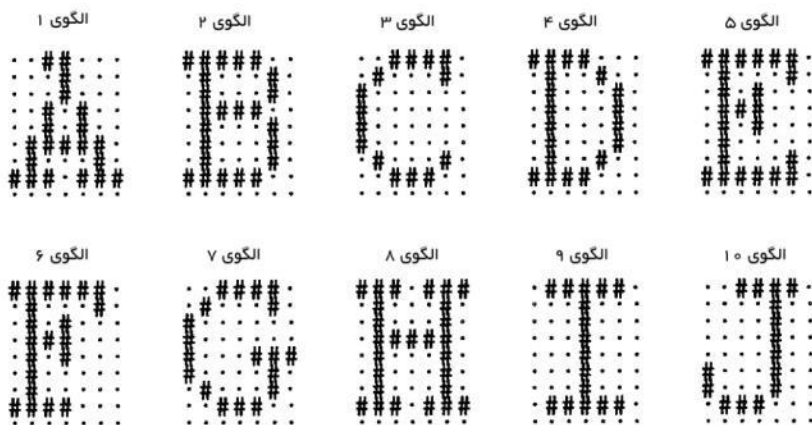
- ایده: اگر مقادیر هدف در مجانب قرار نداشته باشند، همگرایی بهبود می‌یابد
- مقادیر هدف‌های بین ۰.۸ و -۰.۸
- آموزش در ۲۶۴ دور



شبکه‌های پس‌انتشار: مثال ...

○ فشرده‌سازی داده‌ها ...

- یک شبکه خودانجمنی (بردار ورودی آموزش و بردار خروجی هدف یکسان)
- تعداد واحدهای مخفی کمتر از تعداد واحدهای ورودی



• هر حرف $9 \times 7 = 63$ پیکسل

○ یک بردار با ۶۳ مؤلفه

• شبکه عصبی با ۶۳ واحد ورودی

• تعداد واحدهای خروجی $= 63$

• واحدهای مخفی کمتر از واحدهای ورودی

• مجموعه‌ای از N الگوی ورودی متعامد را می‌توان به $\log_2 N$ واحد مخفی نگاشت کرد

○ استفاده از این اصل = بازیابی کامل پس از فشرده‌سازی (فشرده‌سازی بدون ضرر Lossless Compression)

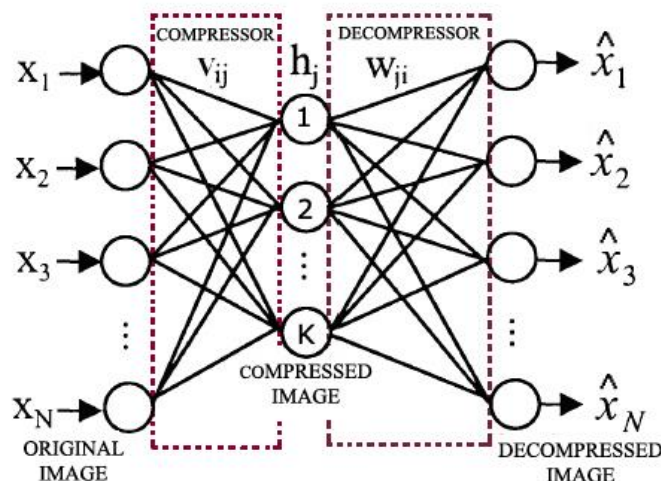
• حروف موجود در مجموعه الگوهای ورودی متعامد نیستند

○ کران پایین نظری برای تعداد واحدهای مخفی $= \log_2 N$

شبکه‌های پس‌انتشار: مثال

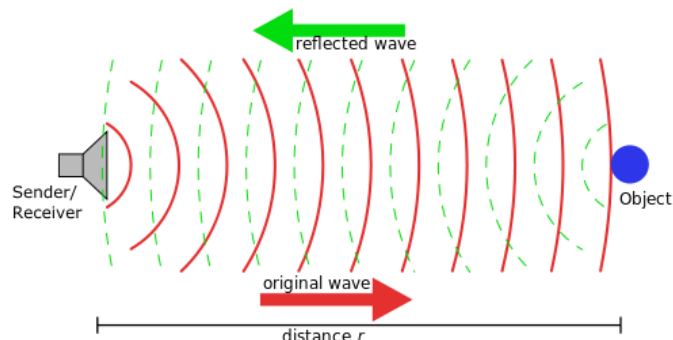
○ فشرده‌سازی داده‌ها

- فشرده‌سازی با ضرر تصویر
- شکستن تصویر به بلوک‌های کوچک (مثلا $8 \times 8 = 64$ نرون ورودی و خروجی)
- مقدار ورودی و خروجی (خروجی تابع فعال‌سازی) پیوسته هستند



شبکه‌های پس‌انتشار: کاربردها ...

○ طبقه‌بند سونار ...



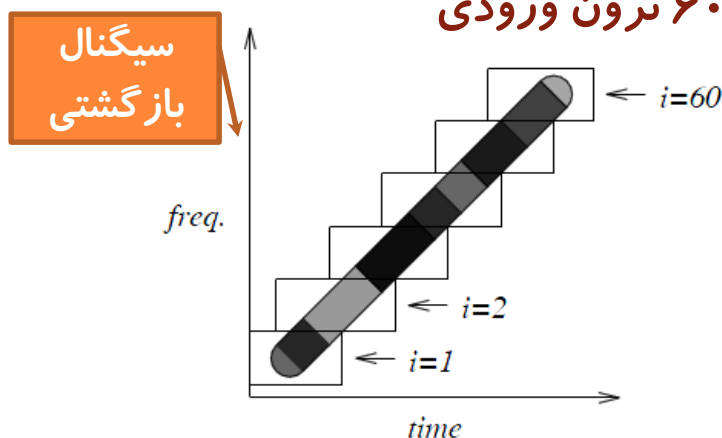
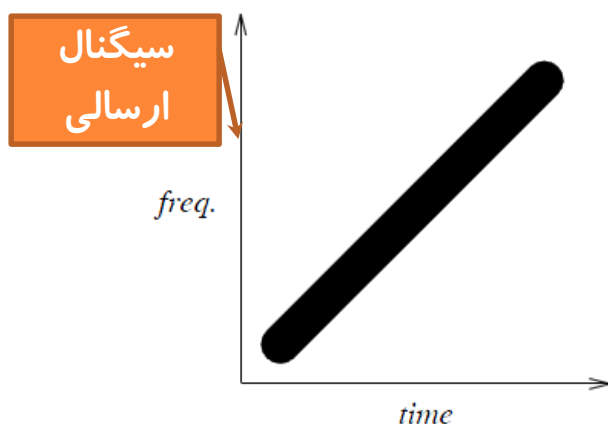
• SONAR = Sound Navigation And Ranging = سونار

• سیستم کاشف زیردریایی با امواج صوتی

• طبقه‌بندی بین سیگنال‌های دریافتی از صخره یا فلزات (مین) = ۲ کلاس خروجی

• بردن سیگنال‌ها به حوزه فرکانس (با تبدیل فوریه) و تقسیم فضای زمان-فرکانس حاصل

به ۶۰ بخش = ۶۰ نرون ورودی

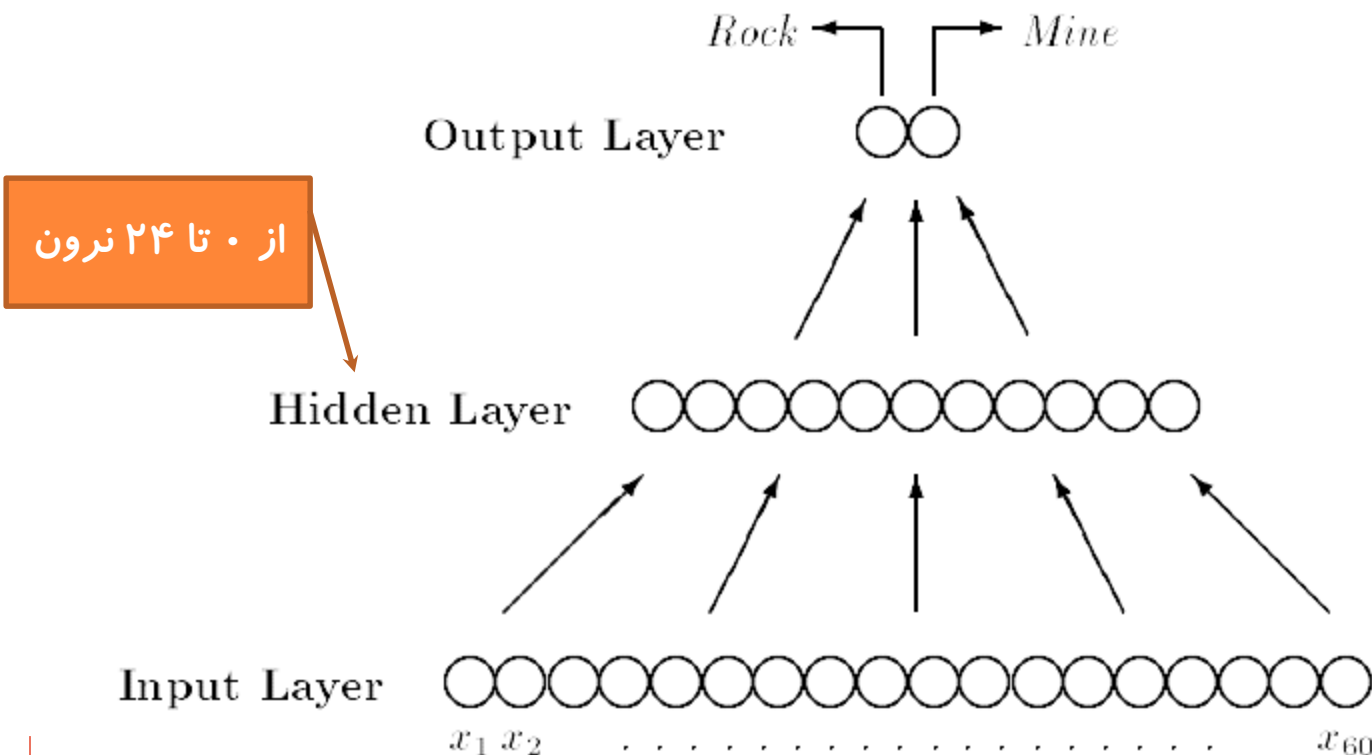


R. Paul Gorman and Terrence J. Sejnowski, "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Objects, Neural Networks, 1:75-89, 1988



شبکه‌های پس‌انتشار: کاربردها ...

○ طبقه‌بند سونار ...





شبکه‌های پس‌انتشار: کاربردها ...

○ طبقه‌بند سونار ...

- داده‌ها = ۲۰۸ نمونه: تقسیم به ۱۳ دسته ۱۶ نمونه‌ای
- آموزش و آزمون به روش 13-fold validation

• نتایج

Hidden Units	% Correct on Training Data	% Correct on Test Data
0	89.0	77.1
2	96.0	81.9
3	98.8	82.0
6	99.7	83.0
12	99.8	84.7
24	99.8	84.5

شبکه‌های پس انتشار: کاربردها ...

○ هدایت خودکار خودرو (ALVINN) ...

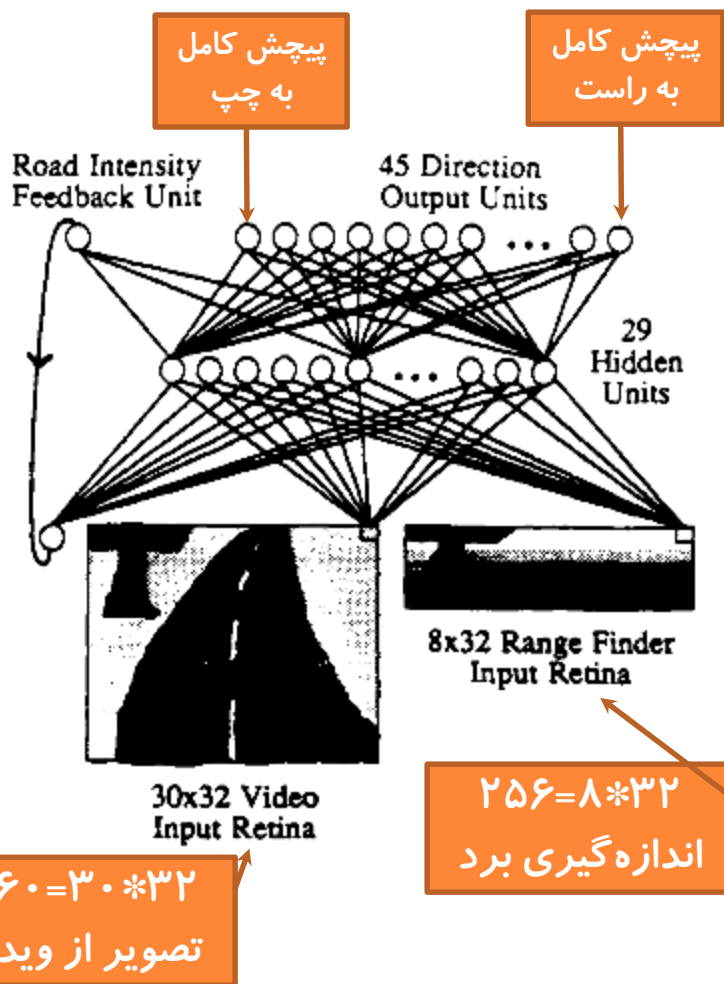
ALVINN (Autonomous Land Vehicle in a Neural Network)

رانندگی خودکار خودرو در جاده مارپیچ

• ورودی: $1217 = 1 + 960 + 256$

• نرون‌های مخفی: ۲۹

• نرون‌های خروجی: ۴۵ (جهت فرمان)



Dean A. Pomerleau, "ALVINN: AN AUTONOMOUS LAND VEHICLE IN A NEURAL NETWORK", Technical Report, ALP-77 (15213-3890), Department of Psychology Carnegie Mellon University, January 1989

شبکه‌های پس‌انتشار: کاربردها ...

○ هدایت خودکار خودرو (ALVINN) ...

- مجموعه آموزش: ۱۲۰۰ تصویر از جاده‌های مختلف
- آموزش در ۴۰ تکرار
- دقت ۹۰٪
- رانندگی با سرعت ۵ کیلومتر بر ساعت! (دو برابر سرعت روش‌های دیگر!!)





شبکه‌های پس‌انتشار: وزن‌ها و بایاس‌های اولیه ...

○ نکات مهم در انتخاب مقادیر اولیه

- تأثیر مقادیر وزن‌های اولیه بر همگرایی شبکه به حداقل خطای سراسری (Global) یا فقط همگرایی شبکه به حداقل خطای محلی (Local)
- انتخاب وزن‌های اولیه‌ای که فعال‌سازی‌ها یا مشتق‌های فعال‌سازی‌ها را صفر نمی‌کنند
 - وابستگی به‌روز کردن وزن بین دو واحد به مشتق تابع فعال‌سازی واحد بعدی و فعال‌سازی واحد قبلی
- تأثیر بر سرعت همگرایی شبکه
- مقادیر وزن‌های اولیه نباید خیلی بزرگ و یا خیلی کوچک باشند
 - باعث می‌شوند سیگنال‌های ورودی به واحدهای مخفی یا واحدهای خروجی در ناحیه اشباع قرار بگیرند که در آن مشتق تابع سیگموئید مقدار بسیار کوچکی دارد.
 - اگر وزن‌های اولیه خیلی کوچک باشند، ورودی شبکه به واحد مخفی یا به واحد خروجی به صفر نزدیک خواهد بود که موجب کند شدن یادگیری می‌شود.



شبکه‌های پس‌انتشار: وزن‌ها و بایاس‌های اولیه ...

○ مقادیر اولیه تصادفی

- مقادیر می‌توانند مثبت یا منفی باشند
- زیرا وزن‌های نهایی بعد از آموزش ممکن است هر علامتی داشته باشند
- بازه متداول برای مقادیر تصادفی وزن‌ها و بایاس‌ها بین ۰.۵ و -۰.۵ (یا بین ۱- و ۱)

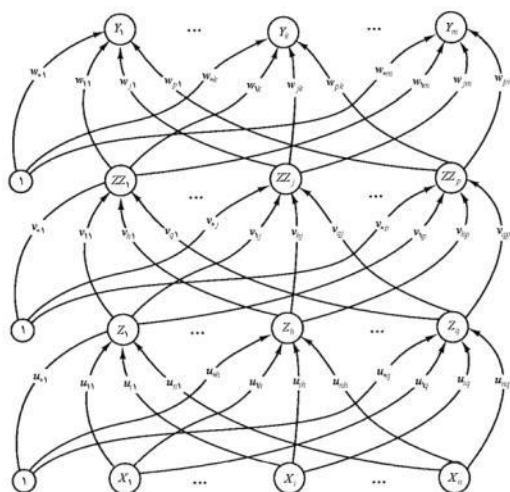
○ تعیین مقدار اولیه با روش نوگن-ویدرو

- ایجاد یادگیری سریع‌تر
- بهبود یادگیری واحدهای مخفی = مقدار اولیه وزن‌های واحدهای ورودی به واحدهای مخفی
- توزیع وزن‌ها و بایاس‌های اولیه به گونه‌ای که برای هر الگوی ورودی، مقدار ورودی شبکه به هر کدام از واحدهای مخفی در دامنه‌ای قرار داشته باشد که در آن دامنه یادگیری آن نرون مخفی به راحتی صورت گیرد.

شبکه‌های پس‌انتشار: تعداد لایه‌های مخفی

○ آموزش شبکه عصبی با بیش از یک لایه مخفی

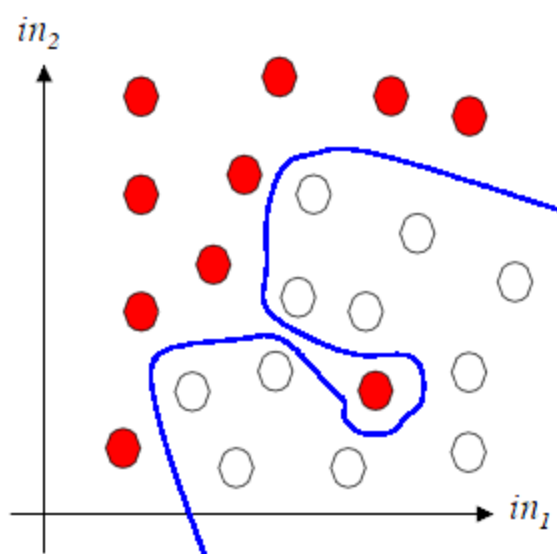
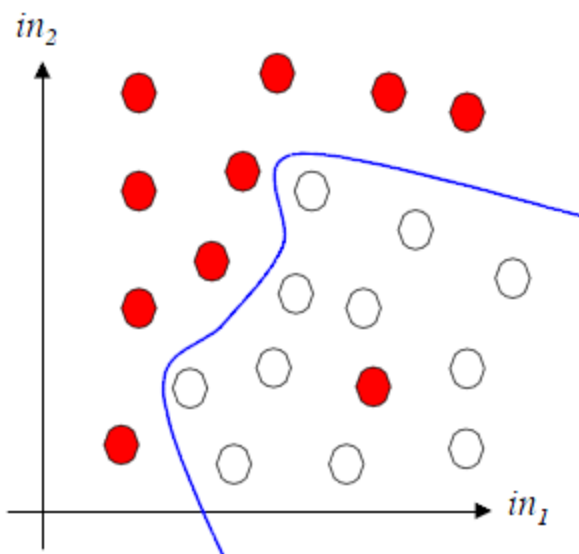
- مشابه الگوریتم آموزش با یک لایه مخفی
- محاسبه‌ها برای هر لایه مخفی اضافی مشابه لایه مخفی بیان شده در الگوریتم است
- برای هر لایه مخفی، گام ۴ در مرحله پیش‌خور و گام ۷ در مرحله پس‌انتشار تکرار می‌شود.
- یک لایه مخفی در شبکه پس‌انتشار برای تقریب زدن هر نگاشت پیوسته‌ای از الگوهای ورودی به الگوهای خروجی با میزان دلخواهی از دقت کافی است.
- در برخی شرایط استفاده از دو لایه مخفی، آموزش شبکه را آسان‌تر می‌کند.



شبکه‌های پس‌انتشار: تعمیم ...

○ هدف آموزش شبکه

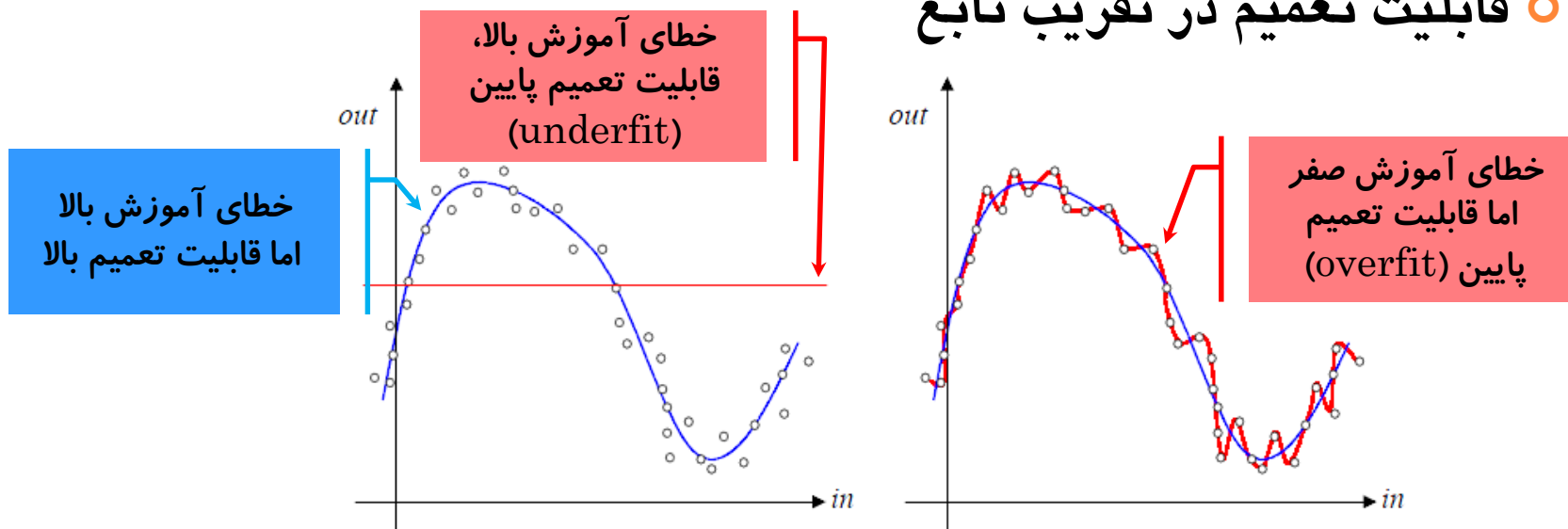
- تعادل بین یادگیری الگوها و تعمیم
 - پاسخ صحیح به الگوهای آموزش داده شده به شبکه و تولید پاسخ مناسب به الگوهای جدید
 - شبکه قوانین حاکم بر داده‌ها را یاد بگیرد نه فقط نمونه‌های آموزش
- ادامه آموزش شبکه زمانی که مقدار مربعات خطا واقعاً حداقل شده، الزاماً مفید نمی‌باشد



خطای آموزش صفر
اما قابلیت تعمیم
پایین

شبکه‌های پس انتشار: تعمیم ...

○ قابلیت تعمیم در تقریب تابع



○ استفاده از دو مجموعه داده مجزا در زمان آموزش شبکه

- یک مجموعه برای آموزش الگوها و یک مجموعه برای آموزش-آزمون الگوها

○ مجموعه تایید اعتبار: validation

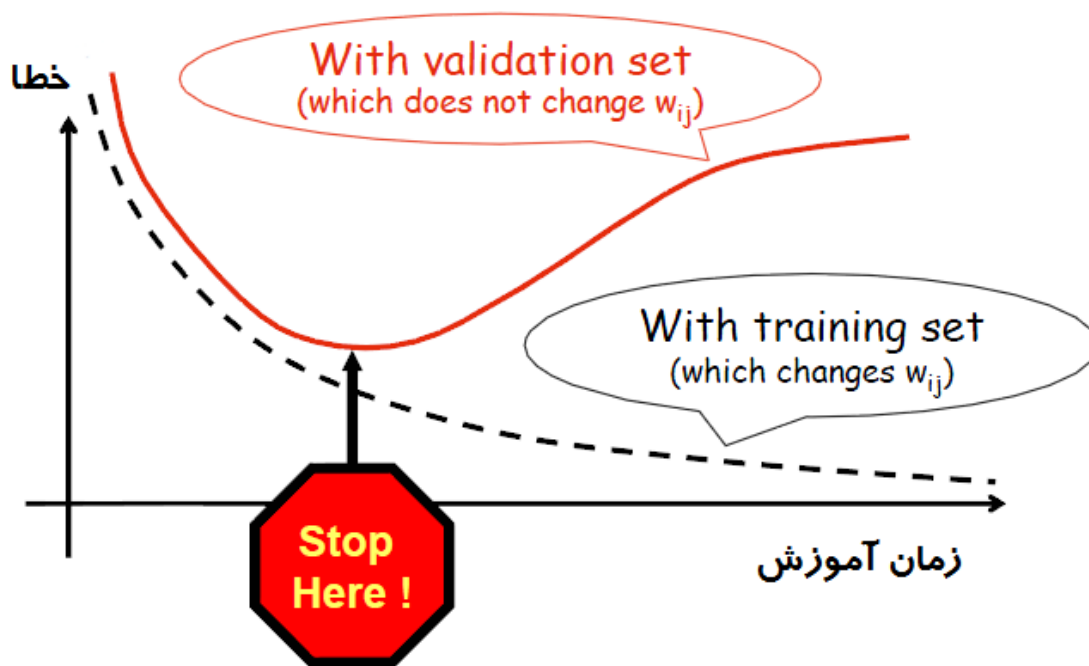
- روش cross validation: تقسیم داده آموزش به K زیرمجموعه

○ هر بار یکی از زیر مجموعه‌ها برای تایید اعتبار استفاده می‌شود

شبکه‌های پس‌انتشار: تعمیم

○ نکاتی که قابلیت تعمیم را افزایش می‌دهد

- تعداد نرون‌های کمتر در لایه مخفی
- overfit نکردن: توقف شبکه با افزایش خطای مجموعه تست
- داده‌های آموزش پوششی از انواع و تنوع نمونه‌ها باشد





شبکه‌های پس‌انتشار: نمایش داده‌ها

- واحدهایی که فعال‌سازی‌های واحد قبلی آنها صفر است، یادگیری نخواهند داشت، چون

$$\Delta w_{jk} = \alpha \delta_k z_j; \quad \delta_k = (t_k - y_k) f'(y_{in_k})$$

○ یادگیری بهتر با نمایش دوقطبی

- نمایش دوقطبی برای ورودی و تابع سیگموید دوقطبی برای تابع فعال‌سازی

○ یادگیری آسان‌تر پاسخ‌های مجزا در مقایسه با مقدار پیوسته

- تبدیل یک متغیر با مقدار پیوسته به گسسته معادل با «یک مجموعه یا محدوده»
- برای تمامی داده‌ها (ورودی یا الگوهای هدف)
- مثال: دمای غذا

○ دمای واقعی = متغیری با مقدار پیوسته (یک نرون)

○ حالت گسسته = یکی از چهار حالت: یخ زده، سرد، دمای نرمال یا داغ (چهار نرون هر یک با مقادیر دوقطبی)

- اشکال تبدیل مقدار پیوسته به گسسته: پیچیده کردن یادگیری نمونه‌های روی (یا نزدیک) مرز گروه‌ها



شبکه‌های پس‌انتشار: تعداد داده‌های آموزش

○ قاعده تجربی

- P = تعداد الگوهای آموزش موجود،
- W = تعداد وزن‌های مورد آموزش در شبکه
- e = صحت دسته‌بندی مورد نظر
- آموزش شبکه برای دسته‌بندی صحیح کسری معادل $1 - (e/2)$ از الگوهای آموزشی،
- می‌توان مطمئن بود که شبکه $1 - e$ الگوی آزمایش را نیز به درستی دسته‌بندی کند؟
- کافی بودن الگوهای آموزشی: $\frac{W}{P} = e$ یا $P = \frac{W}{e}$
- مثال: با $e=0.1$ ، شبکه‌ای با ۸۰ وزن، ۸۰۰ الگوی آموزش لازم خواهد داشت تا از دسته‌بندی صحیح ۹۰٪ الگوهای آزمایش اطمینان حاصل شود، با این فرض که شبکه برای دسته‌بندی صحیح ۹۵٪ الگوهای آموزشی، آموزش دیده باشد.



شبکه‌های پس‌انتشار: روش‌های به‌روز کردن وزن ...

○ پس‌انتشار با گشتاور (Momentum)

- تغییر روش کاهش گرادیان: مقدار تغییر وزن ترکیبی از گرادیان (شیب) فعلی و گرادیان قبلی
- به‌روز شدن وزن‌های زمان $t+1$ وابسته به وزن‌های زمان‌های قبل‌تر (مانند t و $t-1$)

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \delta_k z_j + \mu [w_{jk}(t) - w_{jk}(t-1)]$$

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \delta_j x_i + \mu [v_{ij}(t) - v_{ij}(t-1)]$$

$$\Delta w_{jk}(t+1) = \alpha \delta_k z_j + \mu \Delta w_{jk}(t)$$

$$\Delta v_{jk}(t+1) = \alpha \delta_j x_i + \mu \Delta v_{ij}(t)$$

گرادیان فعلی

گرادیان قبلی

پارامتر ممان (بین ۰ تا ۱)



شبکه‌های پس‌انتشار: روش‌های به‌روز کردن وزن ...

○ پس‌انتشار با گشتاور (Momentum): مزایا

- استفاده از جمع وزن‌دار تغییر وزن‌های قبلی و فعلی

- حرکت شبکه در جهت ترکیبی از گرادیان فعلی و گرادیان وزن قبلی

- عدم حرکت فقط در جهت گرادیان

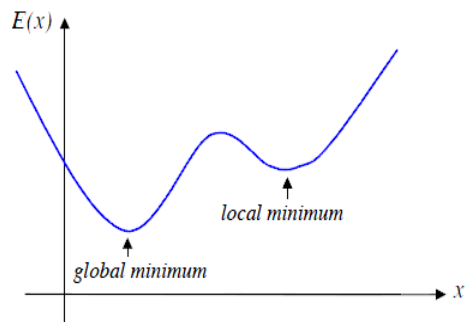
- مقابله با اثرات منفی داده‌های آموزشی غلط یا دارای تفاوت زیاد با سایر داده‌های آموزش

- استفاده از نرخ یادگیری کوچک‌تر = عدم پاسخ بزرگ به خطاها در الگوهای آموزشی

- کمک به همگرایی سریع‌تر: زمانی که داده‌های آموزش نسبتاً شبیه به هم هستند

- تغییر وزن‌ها با گام بزرگ‌تر برای چند الگوی آموزشی که در یک جهت قرار دارند

- کاهش احتمال گیر کردن در نقطه کمینه محلی



○ پس‌انتشار با گشتاور (Momentum): معایب

- نرخ یادگیری کران بالایی برای مقدار تغییر وزن‌ها ایجاد می‌کند

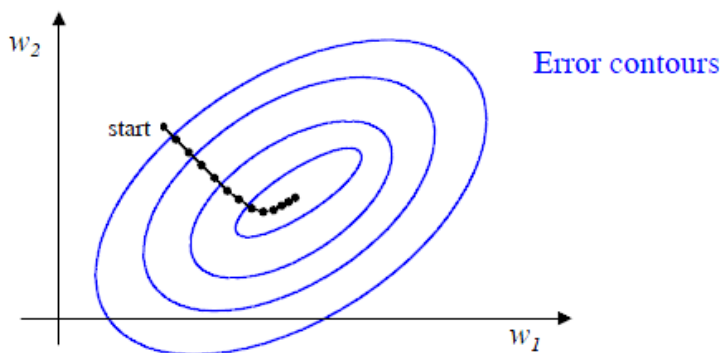
- ممکن است موجب تغییر وزن در جهتی شود که خطا را افزایش دهد



شبکه‌های پس‌انتشار: نرخ یادگیری و فنی ...

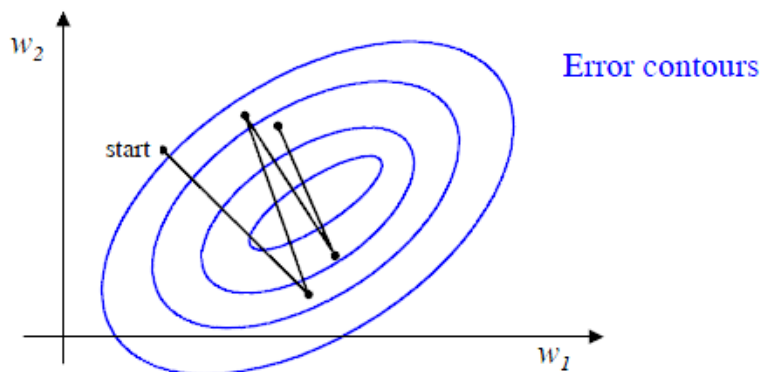
○ نرخ یادگیری کوچک

- سرعت همگرایی پایین
- همگرایی هموار



○ نرخ یادگیری بزرگ

- همگرایی ناهموار (احتمال بالای واگرایی)
- سرعت همگرایی بالا





شبکه‌های پس‌انتشار: نرخ یادگیری و فنی ...

○ تغییر نرخ یادگیری در حین آموزش = بهبود سرعت آموزش

○ حالتی خاص از دسته‌بندی الگو

• تعداد الگوهای آموزش برخی از دسته‌ها بسیار کمتر از داده آموزش سایر دسته‌هاست

○ روش‌های قدیمی برای حل این مشکل

○ دو برابر کردن الگوهای آموزش

○ ساخت کپی‌های نویزی شده از الگوهای آموزش

○ روش دیگر: نرخ یادگیری

○ افزایش نرخ یادگیری در هنگام ارائه الگوهای آموزش دسته‌های با داده آموزشی کم

○ روش دلتا-بار-دلتا (Delta-Bar-Delta)



شبکه‌های پس‌انتشار: نرخ یادگیری و فنی

○ دلتا-بار-دلتا (Delta-Bar-Delta)

- فراهم کردن نرخ یادگیری مخصوص برای هر وزن (نرخ یادگیری وابسته به وزن)
- تغییر نرخ‌های یادگیری با پیشروی آموزش
- استفاده از دو روش ابتکاری برای تعیین تغییرات نرخ یادگیری برای هر وزن
 - اگر تغییر وزن در چند مرحله زمانی در یک جهت باشد (افزایش یا کاهش)، نرخ یادگیری برای آن وزن باید افزایش یابد.
 - تغییر وزن برای چند مرحله زمانی در یک جهت = مشتق جزئی خطای مربوط به آن وزن در آن چند مرحله زمانی علامت یکسانی داشته باشد
 - اگر جهت تغییر وزن (علامت مشتق جزئی) عوض شود، نرخ یادگیری باید کاهش یابد
- شامل دو قانون برای به‌روز کردن
 - به‌روز کردن وزن
 - به‌روز کردن نرخ یادگیری



شبکه‌های پس‌انتشار: به‌روز کردن دسته‌ای وزن‌ها

○ به‌روز کردن دسته‌ای (Batch Updating)

- به جای به‌روز کردن وزن‌های شبکه بعد از ارائه هر الگوی آموزشی
- ادغام مقدار تصحیح (تغییر) وزن را برای چند الگو یا تمام الگوها در یک دور کامل
- تشکیل یک مقدار تنظیم وزن برای هر وزن، برابر با میانگین عبارات تصحیح وزن‌ها
- آسان‌تر کردن تصحیح وزن‌ها
- مقابله با داده‌های نویزی
- محاسبات موازی
- افزایش احتمال نزدیک شدن به کمینه محلی



شبکه‌های پس‌انتشار: تابع فعال‌سازی ...

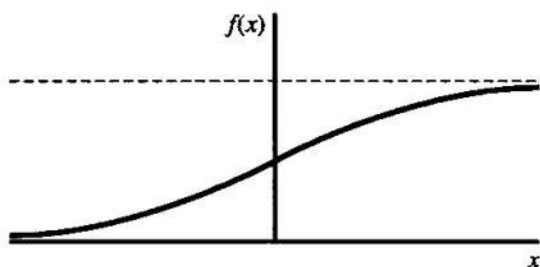
ویژگی‌های مورد نیاز

- پیوسته
- مشتق‌پذیر
 - دارا بودن کارایی محاسباتی (به راحتی قابل محاسبه باشد)
 - مشتق تابع را بتوان برحسب مقدار خود تابع نوشت
- به صورت یکنوا غیرنزولی
- قابلیت اشباع (Saturate)
 - به صورت مجانبی به مقادیر بیشینه و کمینه خود نزدیک شود



شبکه‌های پس‌انتشار: تابع فعال‌سازی

سیگموئید دودویی

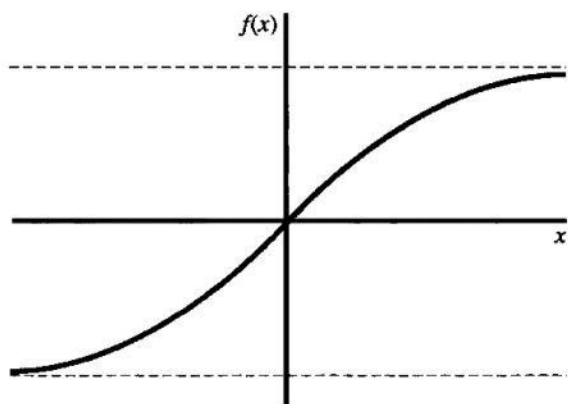


$$f_1(x) = \frac{1}{1 + \exp(-x)}$$

$$f'_1(x) = f_1(x) [1 - f_1(x)]$$

سیگموئید دوقطبی

• شباهت به تانژانت هیپربولیک



$$f_2(x) = \frac{2}{1 + \exp(-x)} - 1$$

$$f'_2(x) = \frac{1}{2} [1 + f_2(x)] [1 - f_2(x)]$$



شبکه‌های پس‌انتشار: توابع فعال‌سازی ...

○ تابع سیگموئید دودویی ...

$$f(x) = \frac{1}{1 + \exp(-x)} \Rightarrow f'(x) = f(x)[1 - f(x)]$$

- می‌توان به گونه‌ای تغییر داد که هر برد دلخواهی را دربرگیرد
- بردن به بازه $[a, b]$ - متغیرهای کمکی $\gamma = b - a, \quad \eta = -a$

$$g(x) = \gamma f(x) - \eta \Rightarrow g'(x) = \frac{1}{\gamma} [\eta + g(x)][\gamma - \eta - g(x)]$$

- تابع فعال‌سازی دوقطبی

$$[a, b] = [-1, 1] \Rightarrow \gamma = 2 \quad \eta = 1$$

$$\Rightarrow g(x) = 2f(x) - 1 \Rightarrow g'(x) = \frac{1}{2} [1 + g(x)][1 - g(x)]$$



شبکه‌های پس انتشار: توابع فعال‌سازی ...

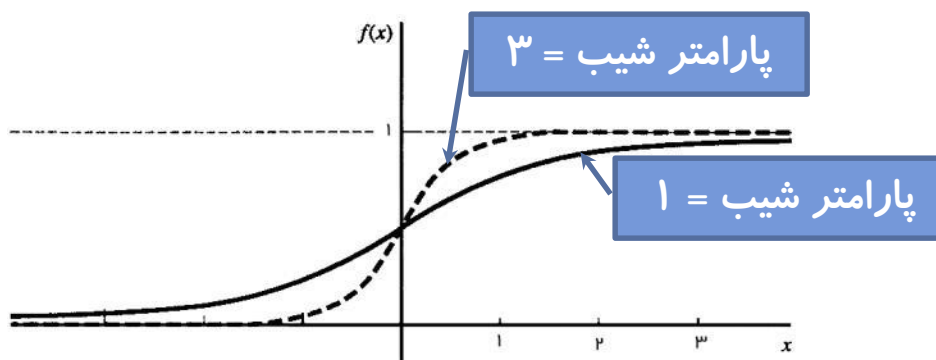
○ تابع سیگموید دودویی

$$f(x) = \frac{1}{1 + \exp(-x)} \Rightarrow f'(x) = f(x)[1 - f(x)]$$

• تغییر شیب تابع

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \Rightarrow f'(x) = \sigma f(x)[1 - f(x)]$$

پارامتر شیب



• حالت کلی

$$g(x) = \gamma f(x) - \eta = \frac{\gamma}{1 + \exp(-\sigma x)} - \eta \Rightarrow g'(x) = \frac{\sigma}{\gamma} [\eta + g(x)][\gamma - \eta - g(x)]$$



شبکه‌های پس‌انتشار: توابع فعال‌سازی ...

○ تابع آرکتانژانت

$$f(x) = \frac{2}{\pi} \arctan(x) \Rightarrow f'(x) = \frac{2}{\pi} \frac{1}{1+x^2}$$

○ توابع فعال‌سازی غیراشباع

$$f(x) = \begin{cases} \log(1+x) & \text{for } x > 0 \\ -\log(1-x) & \text{for } x < 0 \end{cases} \Rightarrow f'(x) = \begin{cases} \frac{1}{1+x} & \text{for } x > 0 \\ \frac{1}{1-x} & \text{for } x < 0 \end{cases}$$

- همانی
- لگاریتمی

○ توابع فعال‌سازی غیرسیگموئید

$$f(x) = \exp(-x^2)$$

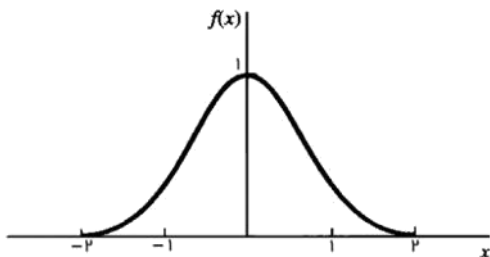
$$f'(x) = -2xf(x)$$

• توابع پایه شعاعی (Radial Basis Functions (RBF))

• برای تمام مقادیر، پاسخ غیرمنفی تولید می‌کند

• با دور شدن از مرکز تابع پاسخ به صفر کاهش می‌رسد

• تابع گاوسی



شبکه‌های عصبی چندلایه. تقریب‌زننده‌های جهانی

- تقریب زدن تابع پیوسته به عنوان یکی از کاربردهای شبکه‌های عصبی
- سوال: شبکه چندلایه تقریب تابع را با چه کیفیتی انجام می‌دهد؟

• پاسخ: قضیه شبکه عصبی کولموگوروف (Kolmogorov)

○ یک شبکه عصبی پیش‌خور با سه لایه نرون (واحدهای ورودی، واحدهای مخفی و واحدهای خروجی) می‌تواند هر تابع پیوسته‌ای را به صورت دقیق نمایش دهد.

- تقریب زدن تابع پیوسته توسط شبکه = تقریب‌زننده جهانی (Universal Approximator)

