

عنوان دوره:

هوش مصنوعی با استفاده از نرم افزار متلب

مدرس دوره : معین سلیمی

صورت استاندارد مسائل بهینه‌سازی چند هدفی

○ " یافتن بردار طراحی X به نحوی که بردار توابع هدف $f(X)$ را با توجه به قیود زیر بهینه کند."

$$X = [x_1^*, x_2^*, \dots, x_n^*]^T$$

$$f(X) = [f_1(X), f_2(X), \dots, f_m(X)]^T$$

$$h_i(X) = 0$$

$$i = 1, 2, \dots, k$$

$$g_j(X) \leq 0$$

$$j = 1, 2, \dots, l$$

بهینه سازی تک هدفی

$$\text{Minimize } Z = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Subject to:

$$x_1 \geq 0$$

$$x_2 \geq 0$$



بهینه سازی تک هدفی

$$\min_{\mathbf{x}} (-8 + x_1)^2 + (-2 + x_2)^2$$

subject to:

$$-\frac{1}{3}x_1 - x_2 \geq -4.5$$

$$-0.1x_1^2 + x_2 \geq 0$$

$$0 \leq x_1 \leq 200$$

$$0 \leq x_2 \leq 200$$



بهینه سازی تک هدفی

$$\min_{\mathbf{x}} (-8 + x_1)^2 + (-2 + x_2)^2$$

subject to:

$$-\frac{1}{3}x_1 - x_2 \geq -4.5$$

$$-0.1x_1^2 + x_2 \geq 0$$

$$0 \leq x_1 \leq 200$$

$$0 \leq x_2 \leq 200$$

$$\forall i = 1, 2 : x_i \in \mathbb{Z}$$



بهینه سازی تک هدفی

An example : **single** objective function

$$\text{Min } f(z_1, z_2, z_3) = (-100 - (z_1 - 5)^2 - (z_2 - 5)^2 + (z_3 - 5)^2) / 100$$

Subject to;

$$h(z_1, z_2, z_3) = (z_1 - 3)^2 + (z_2 - 2)^2 + (z_3 - 5)^2 - 0.0625 \leq 0$$

where;

$$0 \leq z_i \leq 10;$$

Constrained Problem



بهینه سازی تک هدفی

An example : **Multi** objective function

$$\begin{aligned}f_1(x_1, x_2) &= 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \\f_2(x_1, x_2) &= \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \\f_3(x_1, x_2) &= \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2)\end{aligned}$$

objective function **min(f1) & min(f2) & min(f3)**

Unconstrained Problem



بهینه سازی چند هدفی

$$\text{Minimize } F = (f_1(x, y), f_2(x, y))$$

$$f_1(x, y) = 4x^2 + 4y^2$$

$$f_2(x, y) = (x - 5)^2 + (y - 5)^2$$

$$0 \preceq (x - 5)^2 + y^2 - 25$$

$$0 \preceq -(x - 8)^2 - (y + 3)^2 + 7.7$$

$$0 \leq x \leq 5$$

$$0 \leq y \leq 3$$

بهینه سازی چند هدفی

Definition of Optimization Problem (cont.)



8

An example : **Multi objective function**

$$\begin{aligned} \min = & \begin{cases} f_1(\vec{x}) = 1.5 - x_1(1 - x_2) \\ f_2(\vec{x}) = 2.25 - x_1(1 - x_2^2) \\ f_3(\vec{x}) = 2.625 - x_1(1 - x_2^3) \end{cases} \\ \text{subject to;} & \begin{cases} g_1(\vec{x}) = x_1^2 + x_2^2 - 225 \leq 0 \\ g_2(\vec{x}) = x_1 - 3x_2 + 10 \leq 0 \end{cases} \end{aligned}$$

Constrained Problem

Workshop on Intelligent System and Applications (ISA'17), Faculty of Computers and Informatics, Benha University.

شاخصه ی خروج از فرایند بهینه سازی

۱	میانگین تغییرات تابع هدف کمتر از تلورانس تعیین شده میباشد
۳	مقادیر تابع هدف در تعداد مشخصی افزایش نسل تغییری نکرده است
۴	گام حرکت کوچکتر از حداقل مقدار تعریف شده برای ماشین میباشد
۵	تابع هدف به حداقل تعیین شده برای تابع هدف رسیده است
۰	افزایش نسل به مقدار تعیین شده رسیده است
۱-	بهینه سازی به وسیله ی یک تابع خارجی یا ترسیمی متوقف شده است
۲-	محدوده ی مجاز پیدا نشد
۴-	عدم تغییر تابع هدف در مدت زمان تعیین شده
۵-	زمان بهینه سازی به مقدار تعیین شده رسیده است

الگوریتم ژنتیک

الگوریتم های تکاملی، روشهای جستجو و بهینه سازی هستند که از روند تکامل تدریجی طبیعت و نظریه تکامل تدریجی داروین الگوبرداری کرده اند. مفاهیم اصلی تمامی این الگوریتم ها شبیه به هم می باشد. بطور ساده، یک الگوریتم تکاملی به منظور یافتن و حفظ یک جمعیت برتر از بین جمعیت هایی که طی یک فرآیند تکراری و تصادفی، تولید و انتخاب شده است، بکار می رود. در تمامی این الگوریتم ها یک مجموعه از جمعیت ها به عنوان کاندیدای جواب در نظر گرفته می شوند و با استفاده از ترکیبات تصادفی جمعیت های جدید تولید می شوند، با استفاده از یک الگوریتم انتخاب سعی می شود تا برازندگی جمعیت ها افزایش پیدا کند.

تکامل نسل ها

تکامل در گونه های طبیعی با استفاده از سه عملگر اصلی پیوند، جهش و انتخاب گونه های برتر ژنتیکی صورت می گیرد. هدف اصلی از جهش، تولید کروموزومهای جدید و حفظ تنوع در جمعیت های طبیعی می باشد. استفاده از پیوند باعث انتقال خواص خوب دو کروموزوم به عنوان والدین به کروموزومهای جدید تحت عنوان فرزندان می شود، بنابراین باعث تکامل و بهبود در جمعیت ها می گردد. در طبیعت شاهد آن هستیم که جمعیت های برتر همواره احتمال انتخاب بیشتر داشته و اثرات بیشتری از خود در محیط پیرامون بر جای می گذارند، برای مدلسازی این فرایند در الگوریتم های تکاملی از ابزاری به نام برازندگی استفاده می شود. با استفاده از این ابزار، جمعیت ها با برازندگی بیشتر، قدرت بیشتری داشته و تاثیر بیشتری در تولید جمعیت های نسل آینده خواهند داشت.

برای حل یک مساله بهینه سازی با استفاده از الگوریتم تکاملی پنج مورد زیر از اهمیت بالایی برخوردارند

- ۱. تعریف کروموزوم مناسب برای مساله
- ۲. تولید جمعیت اولیه
- ۳. تعریف تابع هدفی که بتواند توصیف کننده برازندگی کروموزومها باشد
- ۴. عملگرهای ژنتیکی که بتواند کروموزومهای جدید را تولید کند
- ۵. تعیین مقادیر اولیه الگوریتم ژنتیک نظیر، تعداد جمعیت اولیه، احتمال پیوند، احتمال جهش، تعداد تکرار.

الگوریتم های انتخاب

انتخاب یکی از فرآیندهای اصلی در الگوریتم های تکاملی می باشد. الگوریتم های انتخاب هم برای انتخاب کروموزومهای برای عملگرهای پیوند و جهش و هم برای انتخاب کروموزومهای نسل بعد کاربرد دارد. در انتخاب کروموزومها سه الگوریتم اصلی وجود دارد:

- انتخاب قطعی
- انتخاب تورنمنت
- انتخاب براساس مکانیزم چرخ گردان

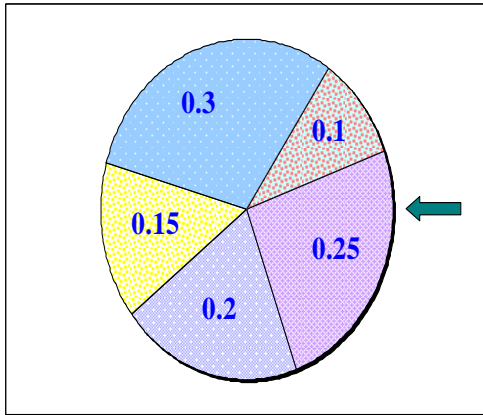
انتخاب صورت قطعی

در این روش تنها بخشی از جمعیت ها در هر تکرار تعویض می شوند. عملگرهای تکاملی در این روش کروموزوم جدید از کروموزوم والد تولید می کنند. سپس کروموزوم بر اساس برازندگی مرتب می شوند در نهایت جمعیت با کمترین برازندگی حذف خواهد شد.

انتخاب بر اساس تورنمنت

انتخاب بر اساس تورنمنت به صورتهای گوناگونی قابل اجرا است. ساده‌ترین راه جداکردن دو به دو است. هر عضو جمعیت در تعداد مشخصی مسابقه شرکت داده می‌شوند و در نهایت اعضای که بیشترین برد یا امتیاز را بدست آورده‌اند انتخاب می‌گردند.

انتخاب براساس مکانیزم چرخ گردان



در مکانیزم انتخاب با استفاده از چرخ گردان، متناظر با هر رشته یک مقدار برازندگی تعیین می‌شود. عدد برازندگی برای هر کروموزوم برابر مقدار تابع هدف آن کروموزوم، تقسیم بر مجموع توابع هدف کروموزومها، تعریف می‌شود، که به آن احتمال بقاء هر جمعیت نیز گفته می‌شود

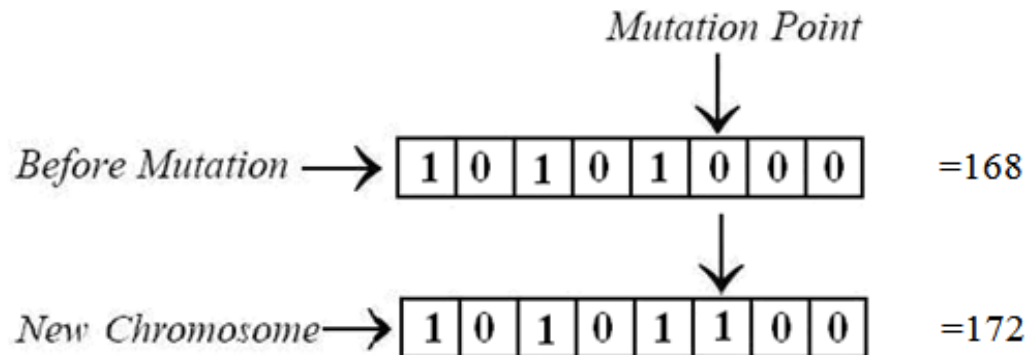
عملگر پیوند

عملگر پیوند، روی دو و یا چند رشته عمل کرده و یک زوج رشته جدید تولید می‌کند، در حقیقت در فرآیند پیوند با ترکیب کردن ژنهای دو یا چند والدین با یکدیگر، دو فرزند جدید ایجاد می‌شود.

Parent 1	11000	011 00111001
Parent 2	01110	110 11011100
Offspring 1	11000	110 11011100
Offspring 2	01110	011 00111001

عملگر جهش

جهش روی یک رشته به صورت تغییر تصادفی در یکی از بیت های اطلاعاتی آن انجام می شود. این عمل شامل یک فرآیند تکراری می باشد که بر روی تمامی ژنها صورت می گیرد.



ویژگی های مربوط به تنظیمات

Option	Description	Values
ConstraintTolerance	حد مجاز برای قیود خطی و غیرخطی در صورت استفاده از gaoptimset از TolCon استفاده میشود.	Positive scalar {1e-3}
CreationFcn	تابع برای ساخت جمعیت اولیه	{'gacreationuniform'} {'gacreationlinearfeasible'}* Custom creation function
CrossoverFcn	تابعی که برای تولید فرزندان از والدین بر اساس crossover استفاده میشود	{'crossovergathered'} for ga,{'crossoverintermediate'}* for gamultiobj 'crossoverheuristic' 'crossoversinglepoint' 'crossovertwoopoint' 'crossoverarithmetic' Custom crossover function
CrossoverFraction	کسری از جمعیت که با استفاده از crossover فرزندان را تولید میکنند	Positive scalar {0.8}
Display	مواردی که در خروجی نمایش داده میشود	'off' 'iter' 'diagnose' {'final'}
DistanceMeasureFcn	تابعی که مقدار فاصله ی بین نمونه ها را محاسبه میکند	{'distancecrowding'} means the same as{@distancecrowding,'phenotype'} {@distancecrowding,'genotype'} Custom distance function

ویژگی های مربوط به تنظیمات

EliteCount	مقدار مثبتی که تعیین میکند چه تعداد از افراد در جمعیت کنونی، در نسل بعد حضور داشته باشند.	Positive integer {ceil(0.05*PopulationSize)} {0.05*(defaultPopulationSize)} for mixed-integer problems
FitnessLimit	مقدار حد پایین برای تابع هدف، جهت توقف فرایند بهینه سازی.	Scalar {-Inf}
FitnessScalingFcn	برای مشاهده ی تابع هدف در یک مقیاس مشخص از این ویژگی استفاده میشود	{'fitscalingrank'} 'fitscalingshiftlinear' 'fitscalingprop' 'fitscalingtop' Custom fitness scaling function
FunctionTolerance	حد آستانه ای تغییرات میانگین تابع هدف جهت توقف فرایند بهینه سازی در صورت استفاده از gaoptimset از TolFun استفاده میشود.	Positive scalar {1e-6} for ga, {1e-4} for gamultiobj
HybridFcn	تابعی که پس از پایان فرایند بهینه سازی، محاسبات را ادامه میدهد.	Function name or handle 'fminsearch' 'patternsearch' 'fminunc' 'fmincon' {}

ویژگی های مربوط به تنظیمات

InitialPenalty	مقدار اولیه برای پارامتر جریمه	Positive scalar {10}
InitialPopulationMatrix	مقادیر اولیه جمعیت برای فرایند بهینه سازی در نسل اول	Matrix {} در صورت استفاده از gaoptimset از InitialPopulation استفاده میشود.
InitialPopulationRange	محدوده ی تعیین شده برای جمعیت اولیه	Matrix or vector {-10;10} for unbounded
	در صورت استفاده از gaoptimset از PopInitRange استفاده میشود.	components, {-1e4+1;1e4+1}
InitialScoresMatrix	مقدار اولیه برای score	Column vector for single objective
	در صورت استفاده از gaoptimset از InitialScore استفاده میشود.	matrix for multiobjective {}
MaxGenerations	بیشترین تعداد تکرار در فرایند بهینه سازی	Positive integer {100*numberOfVariables} for ga, {200*numberOfVariables} for ga
	در صورت استفاده از gaoptimset از Generation استفاده میشود.	multiobj

ویژگی های مربوط به تنظیمات

MaxStallGenerations	در صورتی که مقدار میانگین تابع هدف در تعداد نسل مشخص شده در این آرایه تغییر نکند ، فرایند بهینه سازی متوقف میشود.	Positive integer {50} for ga, {100} for g amultiobj
	در صورت استفاده از <code>gaoptimset</code> <code>StallGenLimit</code> استفاده میشود.	
MaxStallTime	حد زمانی عدم تغییرات میانگین تابع هدف جهت توقف فرایند بهینه سازی	Positive scalar {Inf}
	در صورت استفاده از <code>gaoptimset</code> <code>StallTimeLimit</code> استفاده میشود.	
MaxTime	بیشترین زمان مجاز برای فرایند بهینه سازی	Positive scalar {Inf}
	در صورت استفاده از <code>gaoptimset</code> <code>TimeLimit</code> استفاده میشود.	
MigrationDirection	جهت مهاجرت زیرجمعیت ها به نسل قبل و بعد	'both' {'forward'}
MigrationFraction	کسری از جمعیت که مهاجرت میکند	Scalar {0.2}
MigrationInterval	تعداد نسل هایی که باید بگذرد تا یک انتقال یا مهاجرت نسل ها صورت بگیرد	Positive integer {20}

ویژگی های مربوط به تنظیمات

MutationFcn	تابعی که برای تولید فرزندان از والدین بر اساس mutation استفاده میشود	{'mutationgaussian'} for ga, {'mutationadaptfeasible'}* for gamultiobj 'mutationuniform' Custom mutation function
NonlinearConstraintAlgorithm	الگوریتم مورد استفاده برای قیود غیرخطی	{'auglag'} for ga, {'penalty'} for gamultiobj
	در صورت استفاده از NonlinConAlgorithm استفاده میشود.	
OutputFcn	تابعی که در پایان هر تکرار توسط ga فراخوانی میشود	Function handle or cell array of function handles {}
	در صورت استفاده از OutputFncs استفاده میشود.	
ParetoFraction	کسری از جمعیت که از فرانت اول برای فرانت بعد نگه داشته میشوند.	Scalar {0.35}
PenaltyFactor	پارامتر بروز رسانی جریمه	Positive scalar {100}
PlotFcn	برای ترسیم آرایه های مشخصی در فرایند بهینه سازی از این تابع استفاده میشود.	ga or gamultiobj: {} 'gaplotdistance' 'gaplotgenealogy' 'gaplotselection' 'gaplotscorediversity' 'gaplotscores' 'gaplotstopping' 'gaplotmaxconstr' Custom plot function
	در صورت استفاده از PlotFncs استفاده میشود.	ga only: 'gaplotbestf' 'gaplotbestindiv' 'gaplotexpectation' 'gaplotrange' gamultiobj only: 'gaplotpareto' 'gaplotparetodistance' 'gaplotrankhist' 'gaplotspread'

ویژگی های مربوط به تنظیمات

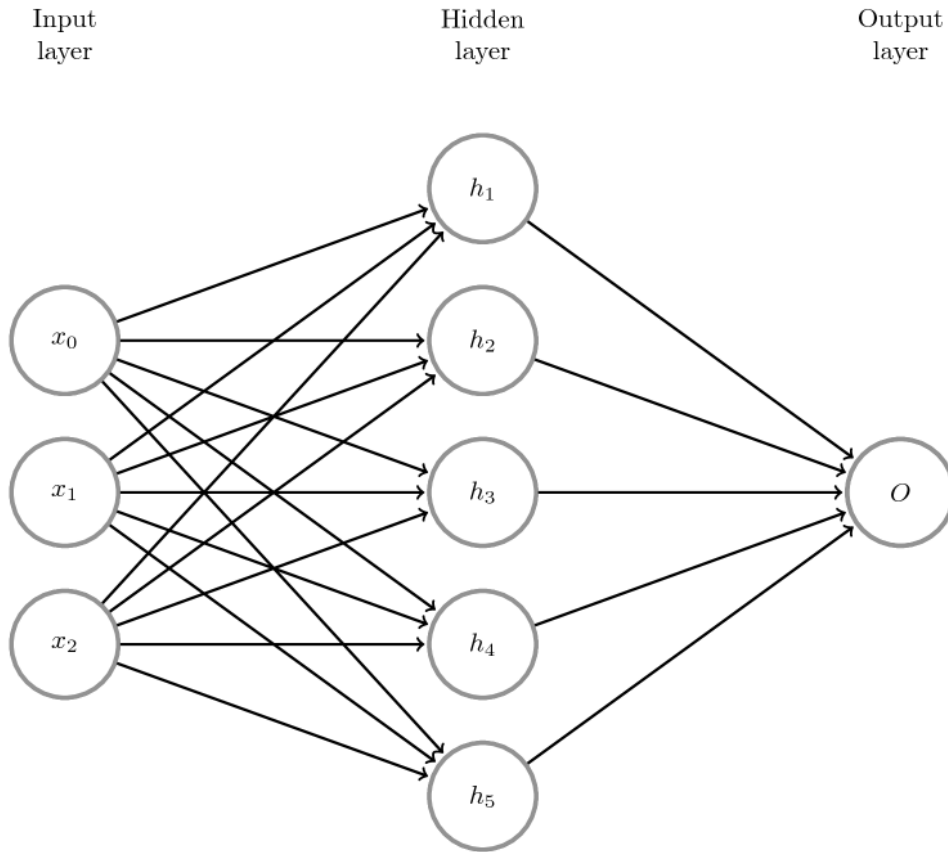
PlotInterval	تعداد نسل هایی که باید بگذرد تا تابع ترسیمی مجدداً فراخوتنی شود	Positive integer {1}
PopulationSize	اندازه ی جمعیت	Positive integer {50} when numberOfVariables <= 5, {200} otherwise {min(max(10*nvars,40),100)} for mixed-integer problems
PopulationType	نوع دیتا را برای جمعیت اولیه مشخص میکند	'bitstring' 'custom' {'doubleVector'} ga ignores all constraints when PopulationType is set to 'bitString' or 'custom'. See Population Options.
SelectionFcn	تعیین الگوریتم انتخاب	{'selectionstochunif'} for ga, {'selectiontournament'} for gamultiobj 'selectionremainder' 'selectionuniform' 'selectionroulette' Custom selection function
StallTest	شرایط توقف بر اساس تغییرات تابع هدف	'geometricWeighted' {'averageChange'}
UseParallel	محاسبات موازی تابع هدف و قیود غیرخطی	true {false}
UseVectorized	محاسبات تابع هدف به صورت برداری بر روی تمامی جمعیت یا به شکل فرد به فرد	true {false}

شبکه عصبی

مبحث شبکه های عصبی مربوط به شبیه سازی قوه یادگیری در انسان و پیاده سازی آن به صورت الگوریتم های کامپیوتری می باشد. قوه یادگیری در انسان عبارت از فراگیری الگوهای اطراف خود توسط تکرار می باشد. مغز به عنوان یک سیستم پردازش اطلاعات با ساختار موازی از میلیونها نرون تشکیل شده است. بافتی که عصب نامیده می شوند اجتماعی از نرون ها هستند که اطلاعات و پیامها را از یک قسمت بدن به قسمت دیگر منتقل می کنند.

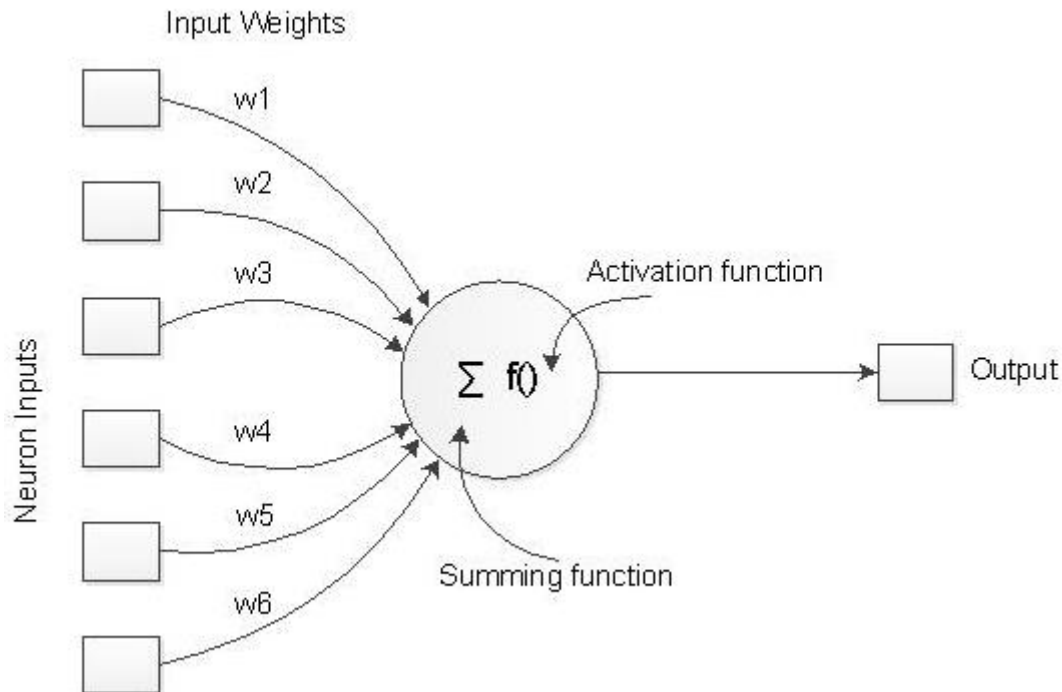
معماری شبکه

○ شبکه ی نورون ها



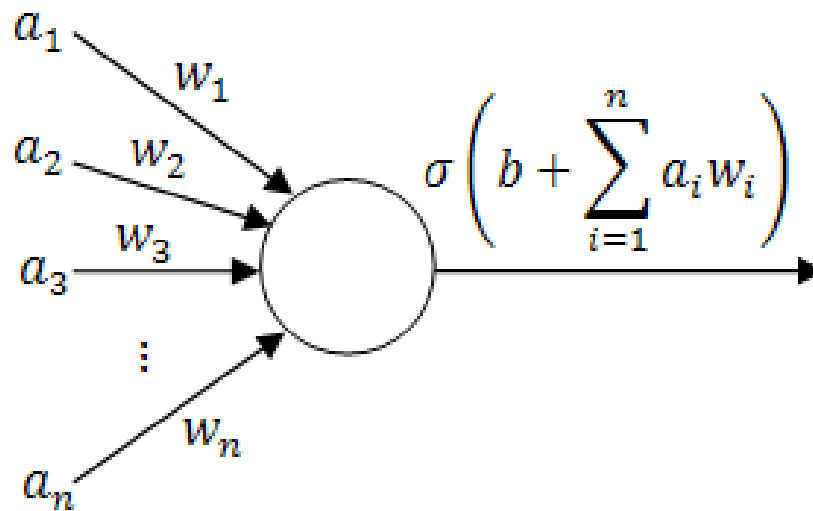
معماری شبکه

○ ساختار نرون



معماری شبکه

○ پرسپترون



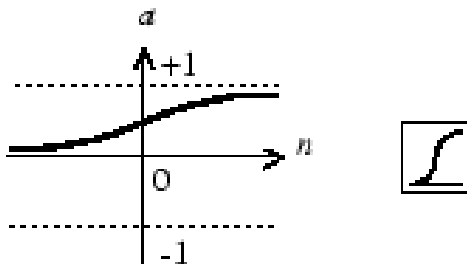
معماری شبکه

○ توابع فعال سازی

○ لگاریتم-سیگموئید

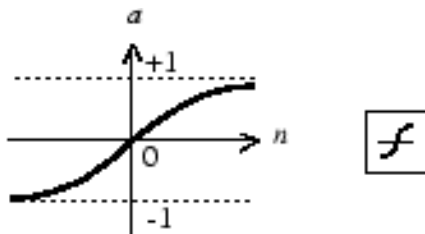
○ تانژانت-سیگموئید

○ خطی



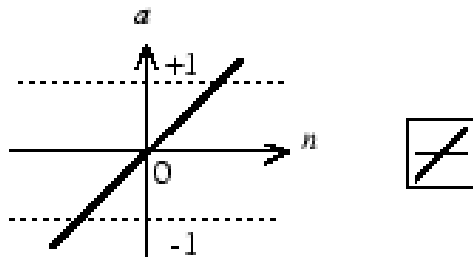
$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function



$$a = \text{tansig}(n)$$

Tan-Sigmoid Transfer Function

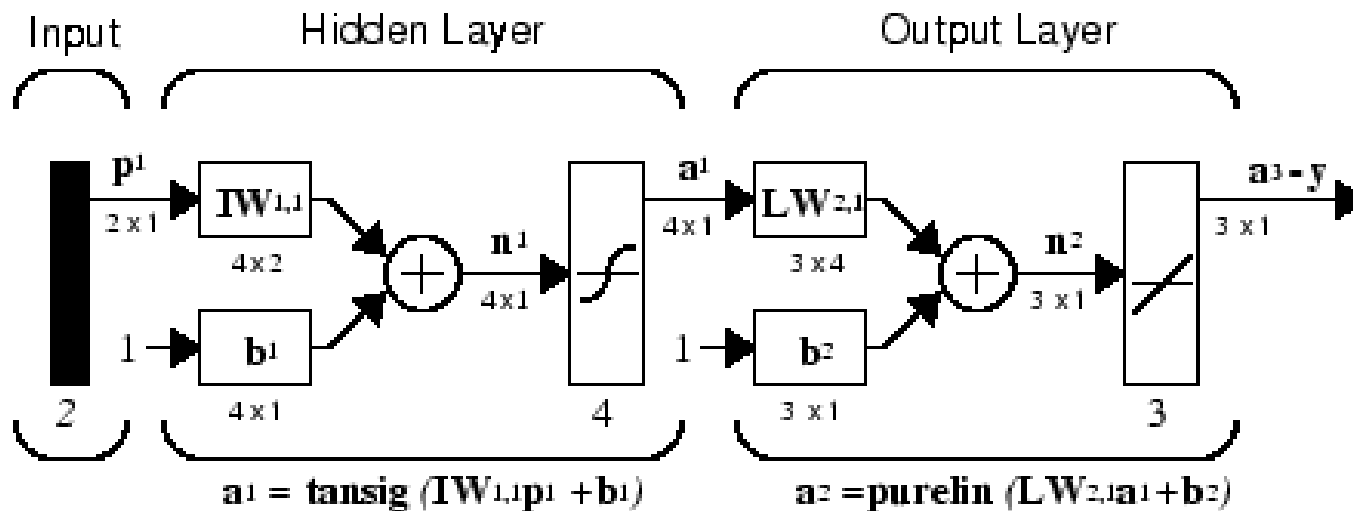


$$a = \text{purelin}(n)$$

Linear Transfer Function

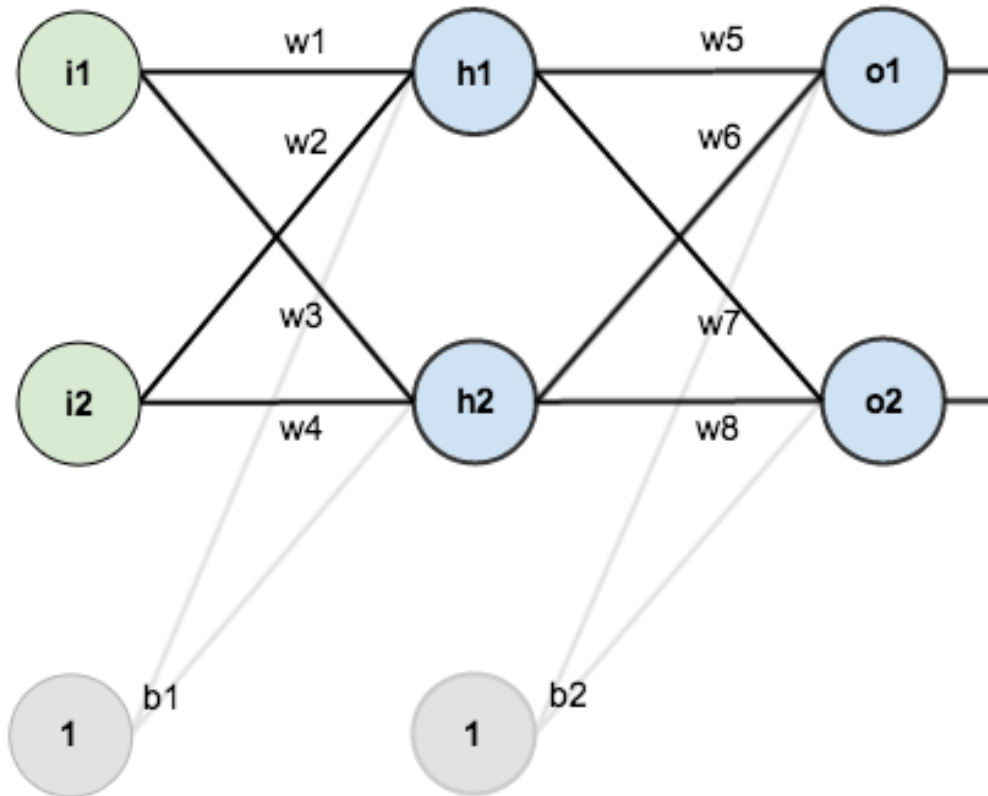
معماری شبکه

○ ساختار مدل ریاضی



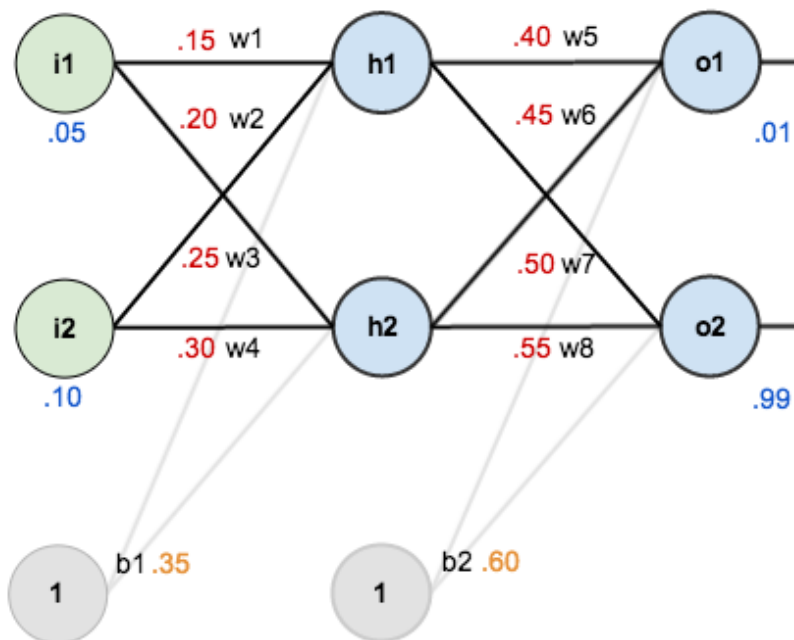
مثال عددی

○ مدل اولیه



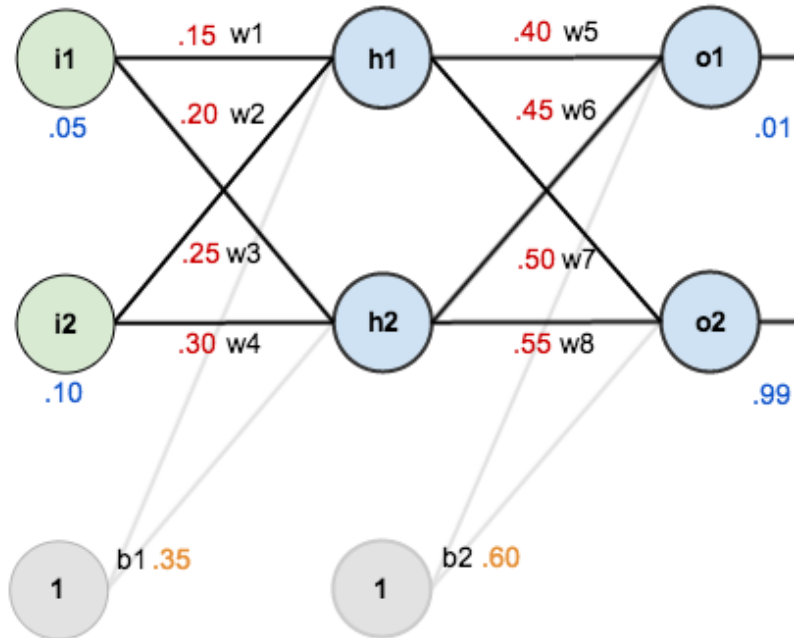
مثال عددی

○ مدل اولیه با مقادیر تصادفی برای وزن ها و بایاس ها



مثال عددی

مسیر پیشرو ○



$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

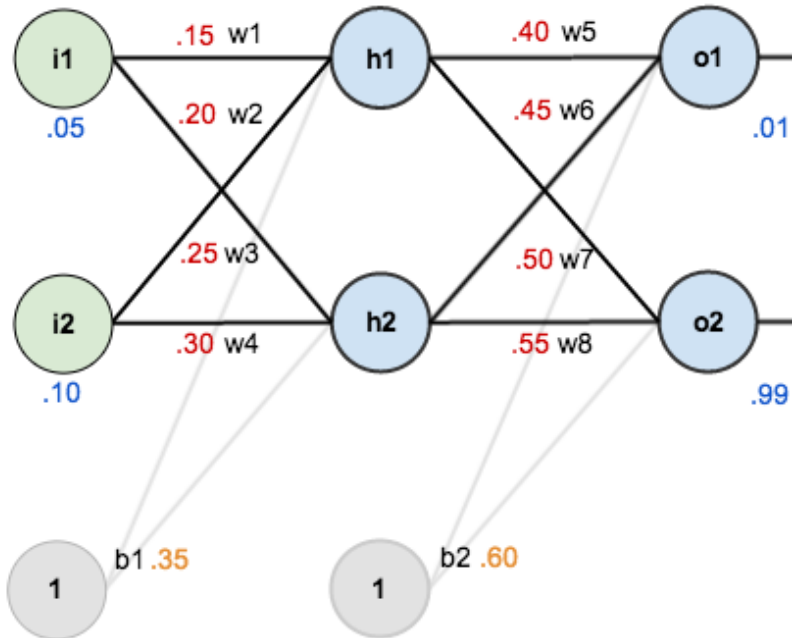
$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}} = \frac{1}{1 + e^{-0.3775}} = 0.593269992$$

$$out_{h2} = 0.596884378 \quad ???$$

مثال عددی

مسیر پیشرو



$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

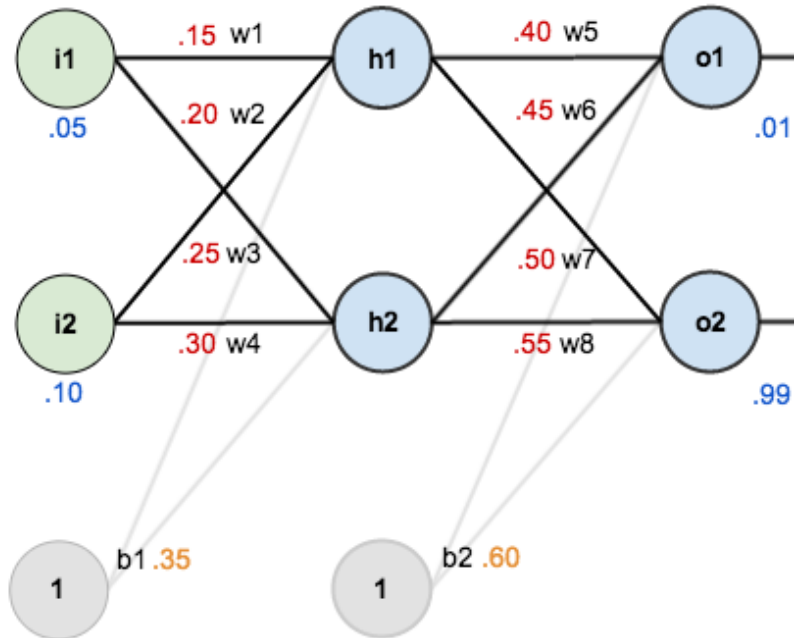
$$net_{o1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 * 1 = 1.105905967$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

$$out_{o2} = 0.772928465$$

مثال عددی

○ محاسبه ی خطا



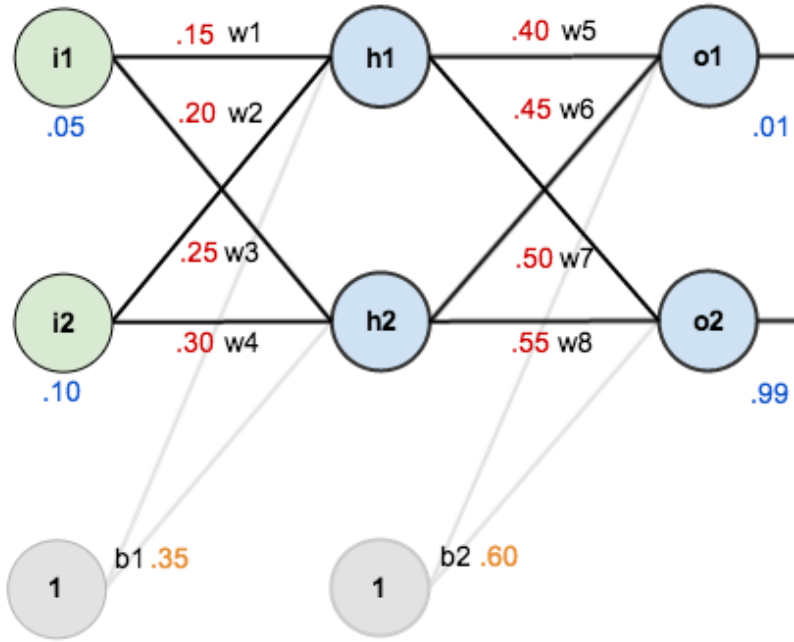
$$E_{total} = \sum \frac{1}{2} (target - output)^2$$

$$E_{o_1} = \frac{1}{2} (target_{o_1} - out_{o_1})^2 = \frac{1}{2} (0.01 - 0.75136507)^2 = 0.274811083$$

$$E_{o_2} = 0.023560026 \quad \leftarrow \text{Target}=0/99$$

$$E_{total} = E_{o_1} + E_{o_2} = 0.274811083 + 0.023560026 = 0.298371109$$

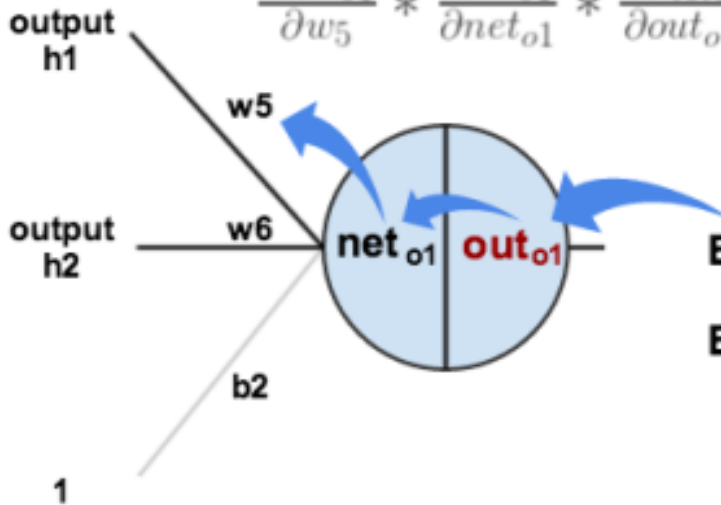
مثال عددی



○ مسیر پسرو و اصلاح وزن ها

○ لایه خروجی و لایه مخفی

$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$

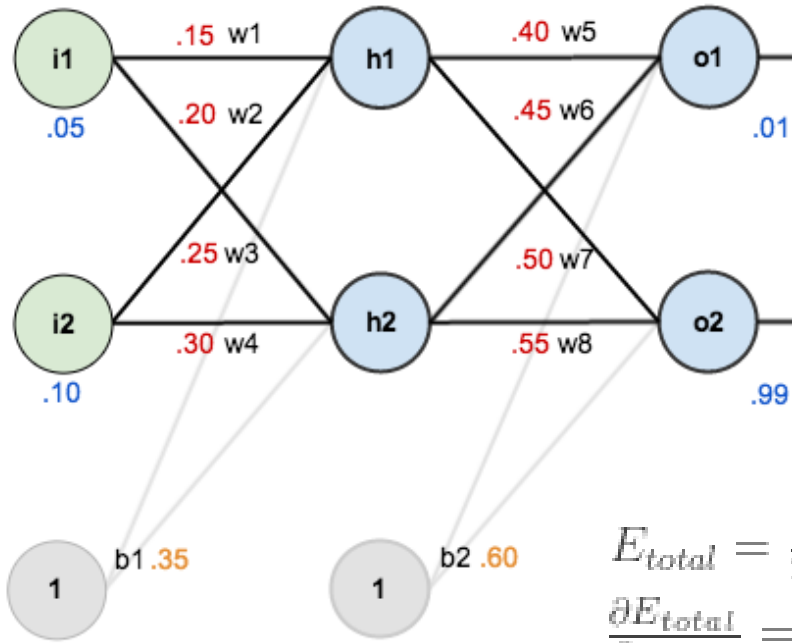


$$E_{o1} = \frac{1}{2}(\text{target}_{o1} - \text{out}_{o1})^2$$

$$E_{total} = E_{o1} + E_{o2}$$

$$\frac{\partial E_{total}}{\partial w_5}$$

مثال عددی



○ مسیر پسرو و اصلاح وزن ها

$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1})^{2-1} * -1 + 0$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = -(target_{o1} - out_{o1}) = -(0.01 - 0.75136507) = 0.74136507$$

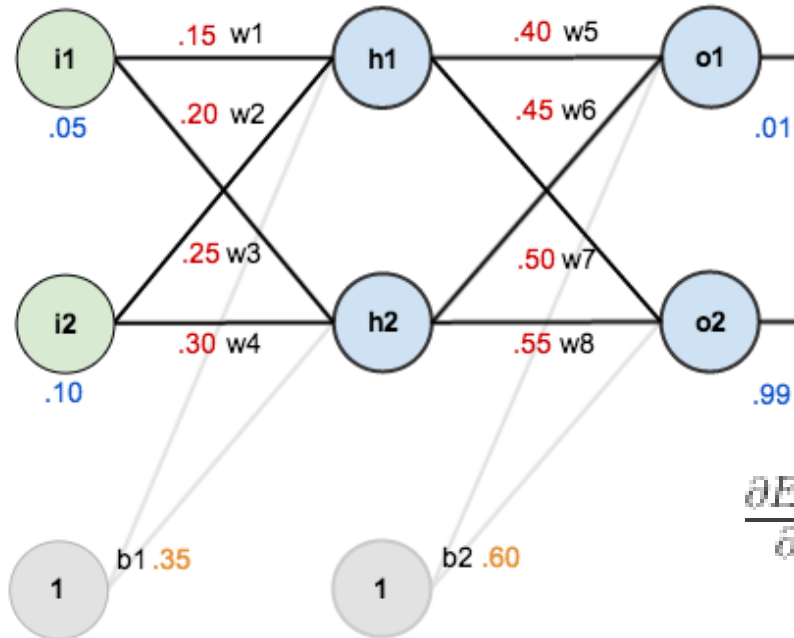
$$out_{o1} = \frac{1}{1 + e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1}) = 0.75136507(1 - 0.75136507) = 0.186815602$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$$

مثال عددی



مسیر پسرو و اصلاح وزن ها

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

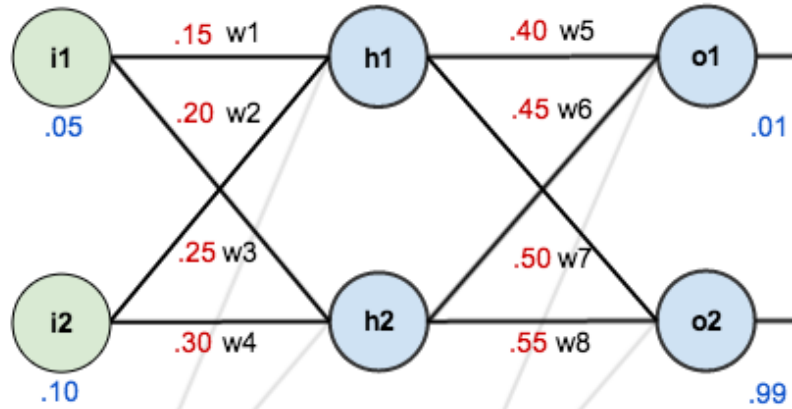
$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 * 0.082167041 = 0.35891648$$

$$w_6^+ = 0.408666186 \quad w_8^+ = 0.561370121$$

$$w_7^+ = 0.511301270$$

مثال عددی



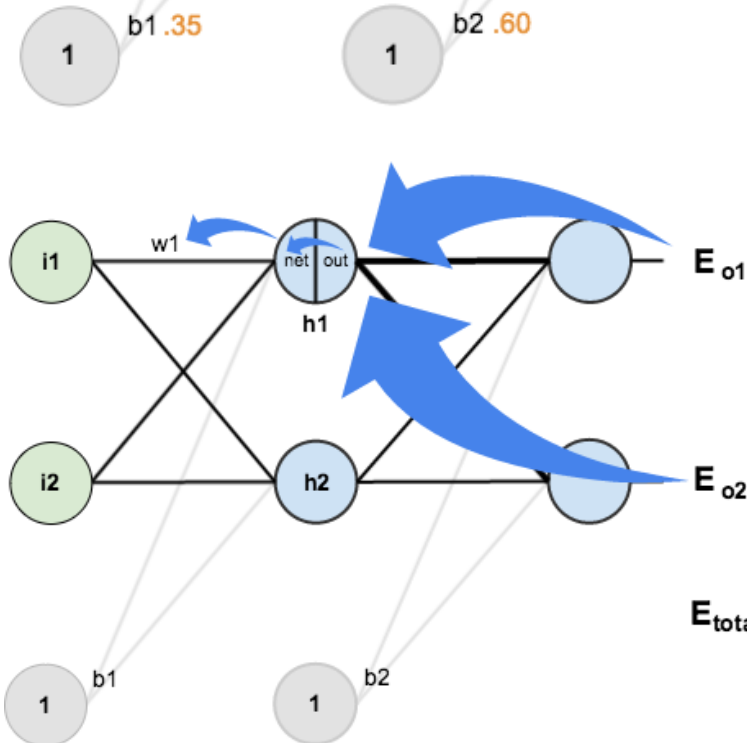
○ مسیر پسرو و اصلاح وزن ها

○ لایه مخفی و لایه ورودی

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

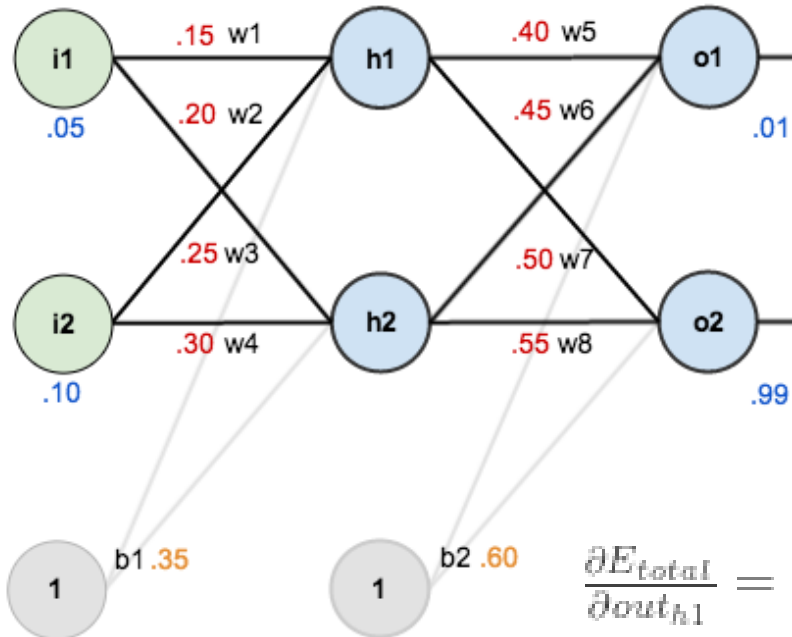
$$\downarrow$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$E_{total} = E_{o1} + E_{o2}$$

مثال عددی



○ مسیر پسرو و اصلاح وزن ها

○ لایه مخفی و لایه ورودی

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}}$$

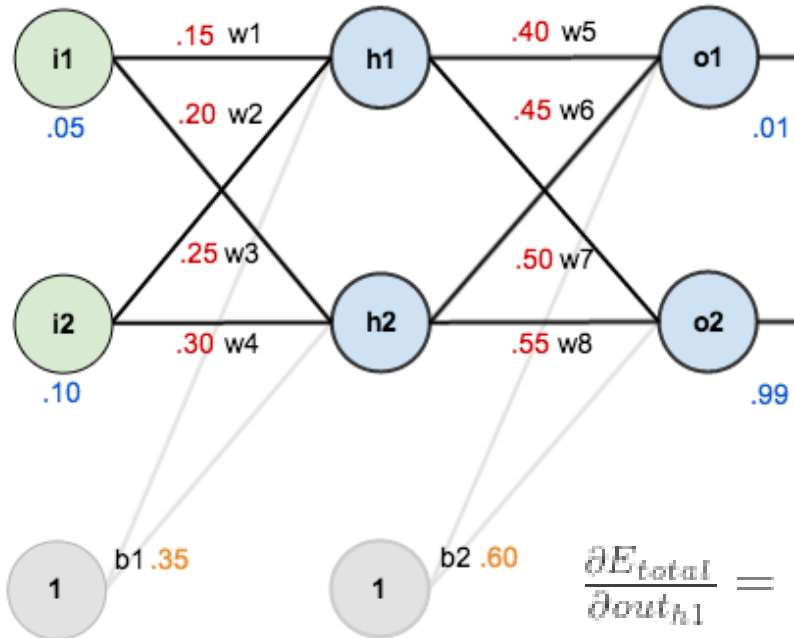
$$\frac{\partial E_{o1}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} = 0.74136507 * 0.186815602 = 0.138498562$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial out_{h1}} = w_5 = 0.40$$

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}} = 0.138498562 * 0.40 = 0.055399425$$

مثال عددی



○ مسیر پسرو و اصلاح وزن ها

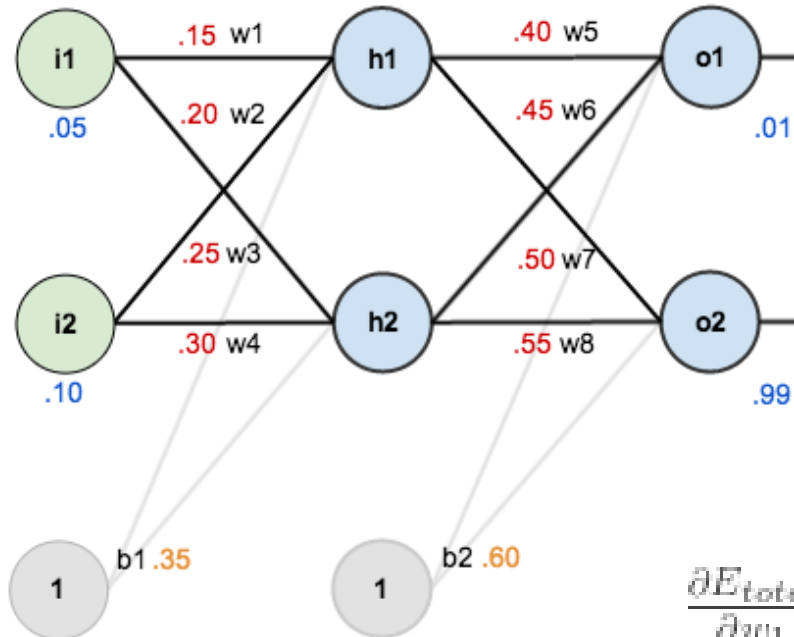
○ لایه مخفی و لایه ورودی

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

$$\frac{\partial E_{o2}}{\partial out_{h1}} = -0.019049119$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} = 0.055399425 + -0.019049119 = 0.036350306$$

مثال عددی



○ مسیر پسرو و اصلاح وزن ها

○ لایه مخفی و لایه ورودی

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}}$$

$$\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1 - out_{h1}) = 0.59326999(1 - 0.59326999) = 0.241300709$$

$$\frac{\partial E_{total}}{\partial w_1} = 0.036350306 * 0.241300709 * 0.05 = 0.000438568$$

$$w_1^+ = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.000438568 = 0.149780716$$

$$w_2^+ = 0.19956143$$

$$w_3^+ = 0.24975114$$

$$w_4^+ = 0.29950229$$

سیستم های فازی

○ منطق فازی

در منطق فازی گزاره ها کاملا درستیا کاملا غلط نیستند. آنها در عوض دارای درجه درستی می باشند.

○ استنتاج فازی

استنتاج فازی روشی برای تفسیر مقادیر بردار ورودی است که بر اساس مجموعه های از قوانین، مقادیری برای بردار خروجی مقداردهی می کند.

منطق فازی

- کیف من در خانه است.
- کتاب من در کیفم است.
- برادر من از من بزرگتر است.

این جملات یا کاملاً درست هستند یا کاملاً غلط!

منطق فازی

○ علی بلند است.

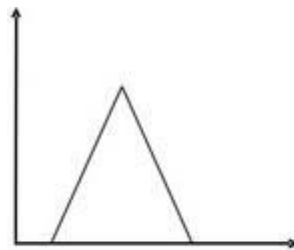
○ هوا گرم است .

○ کتاب من بزرگ است.

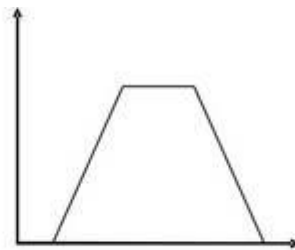
در منطق فازی گزاره ها کاملا درست یا کاملا غلط نیستند. آنها در عوض دارای درجه درستی می باشند. اگر درست را با ۱ و غلط را با ۰ نشان دهیم درستی هر جمله می تواند یک عدد بین ۰ و ۱ باشد.

مفاهیم پایه در منطق فازی

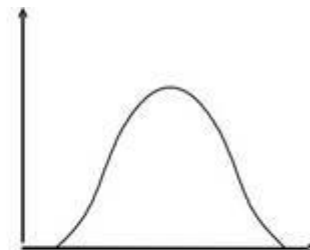
- یک تابع عضویت منحنی است که نشان می دهد یک نقطه در فضای ورودی چگونه به مقدار عضویت در خروجی نگاشت می شود، این مقدار عضویت عددی بین ۰ تا ۱ است.



Triangular membership function



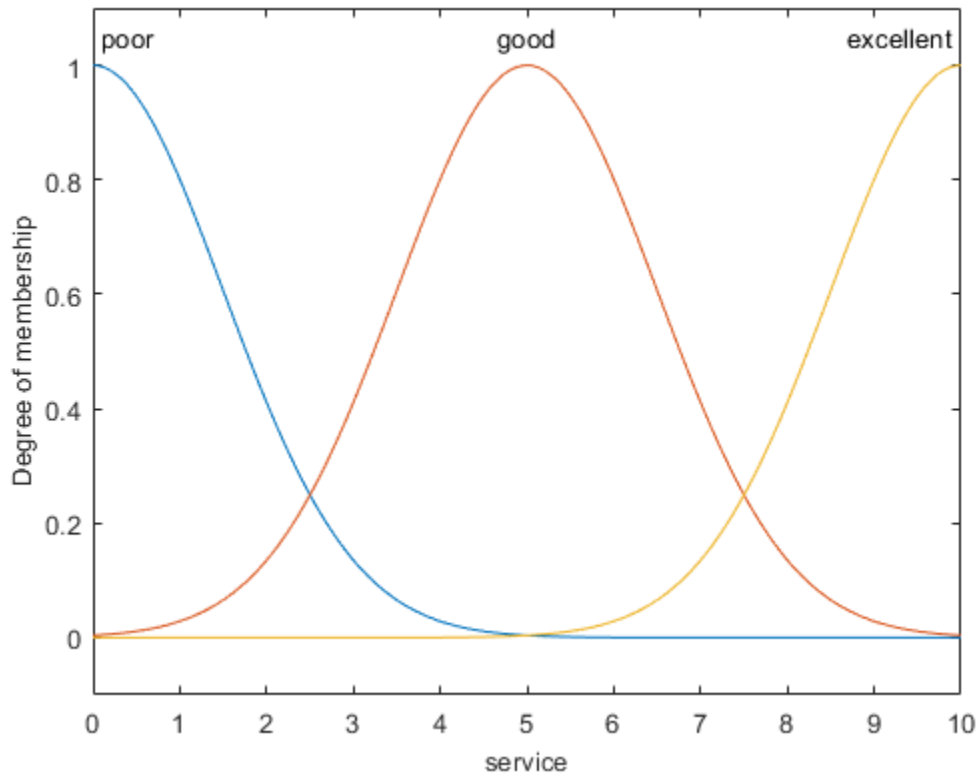
Trapezoidal membership function



Gaussian membership function

مفاهیم پایه در منطق فازی

○ درجه ی عضویت:



استنتاج فازی

○ مراحل طراحی سیستم استنتاج فازی:

1. فازی سازی

2. ساخت پایگاه قوانین و داده

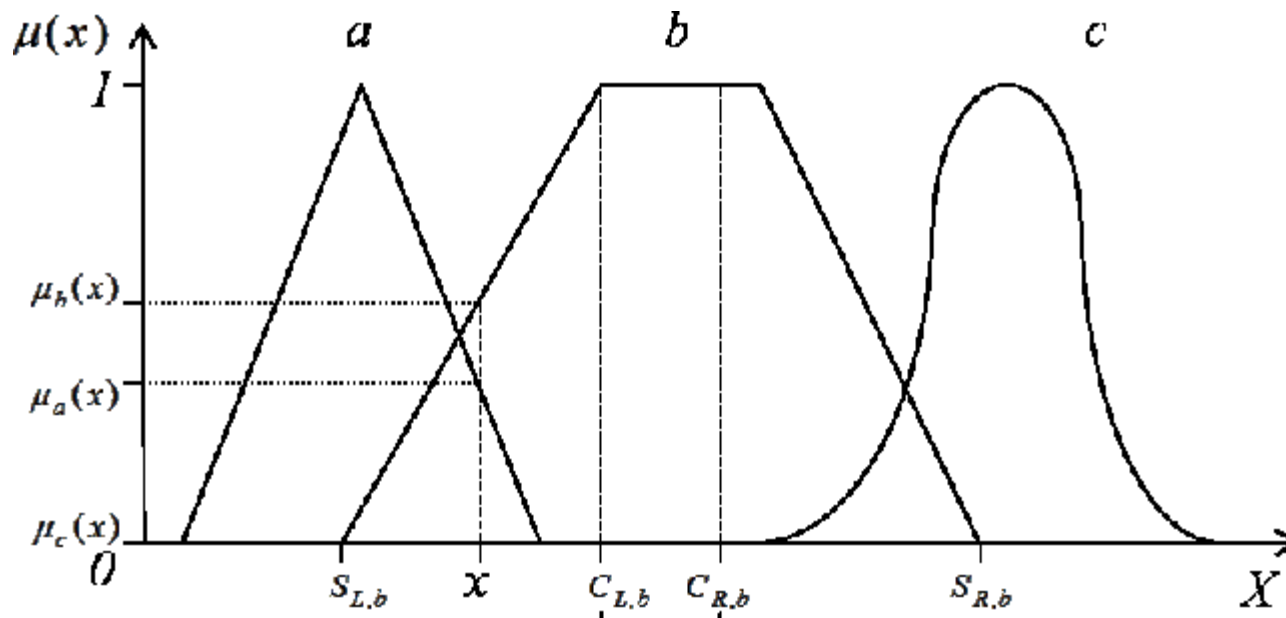
3. استنتاج

4. نافازی سازی

استنتاج فازی

○ فازی سازی: تعیین درجه عضویت ورودی ها به مجموعه های

فازی با کمک توابع عضویت فازی



استنتاج فازی

○ ساخت پایگاه قوانین و داده ها

if x_1 is A_1^r and x_2 is A_2^r and ... and x_n is A_n^r then y is B^r

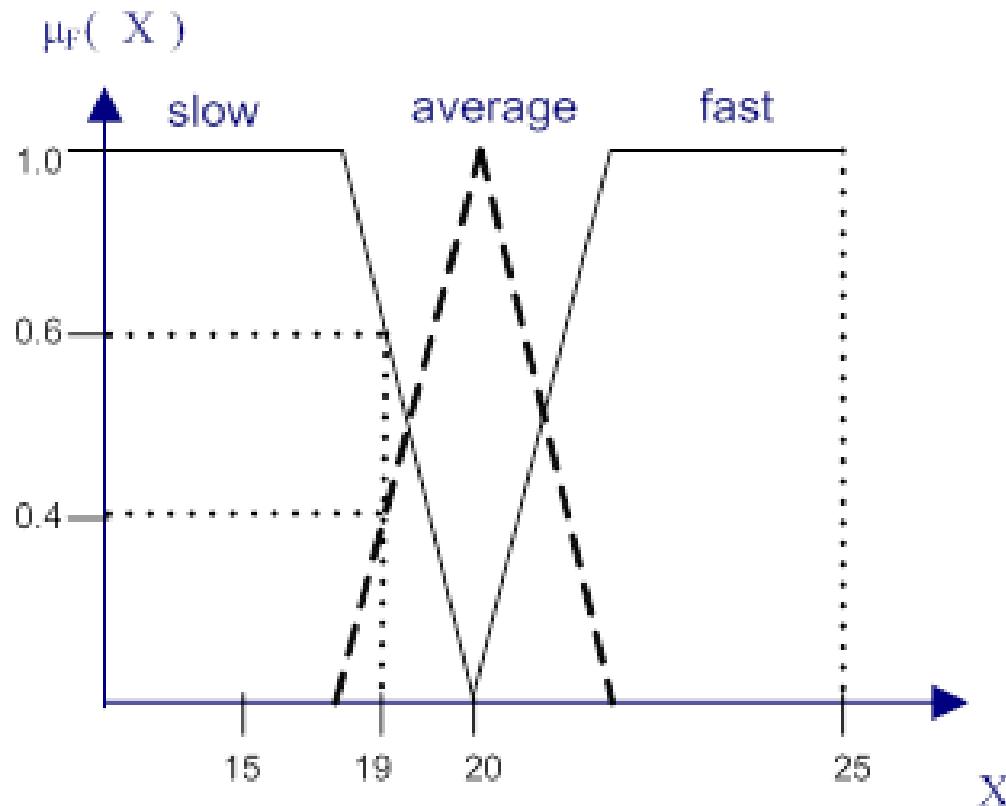
○ Xها ورودی های سیستم

○ A و B مجموعه های فازی که با توابع عضویت نشان میدهیم

○ Y برابر خروجی سیستم

استنتاج فازی

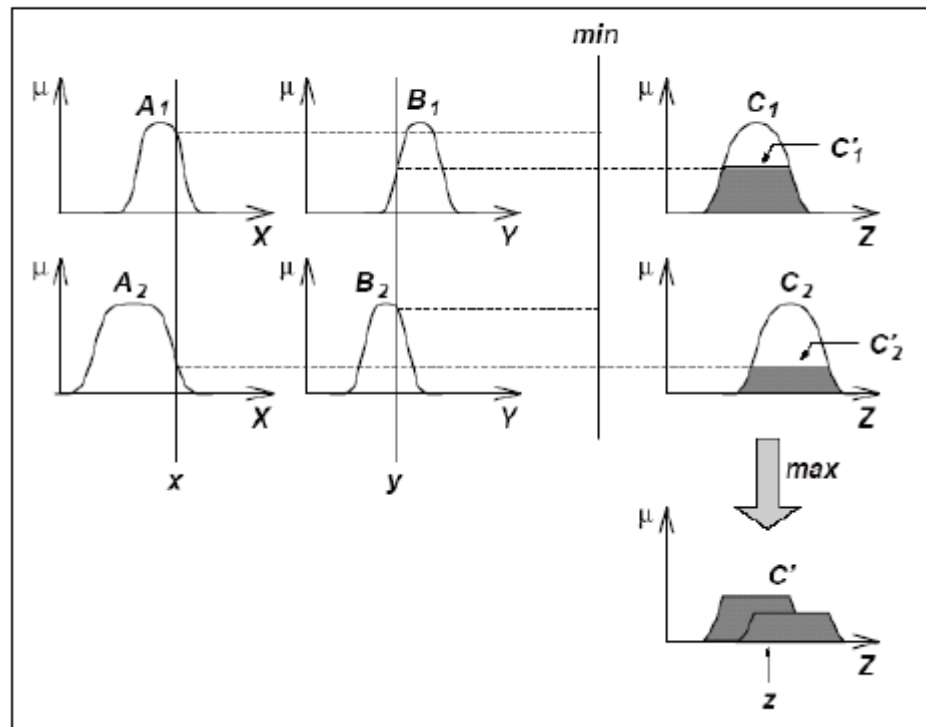
○ اعمال عملگرها بر روی قوانین در بخش ورودی ها



استنتاج فازی

○ یافتن تابع عضویت خروجی (Implication)

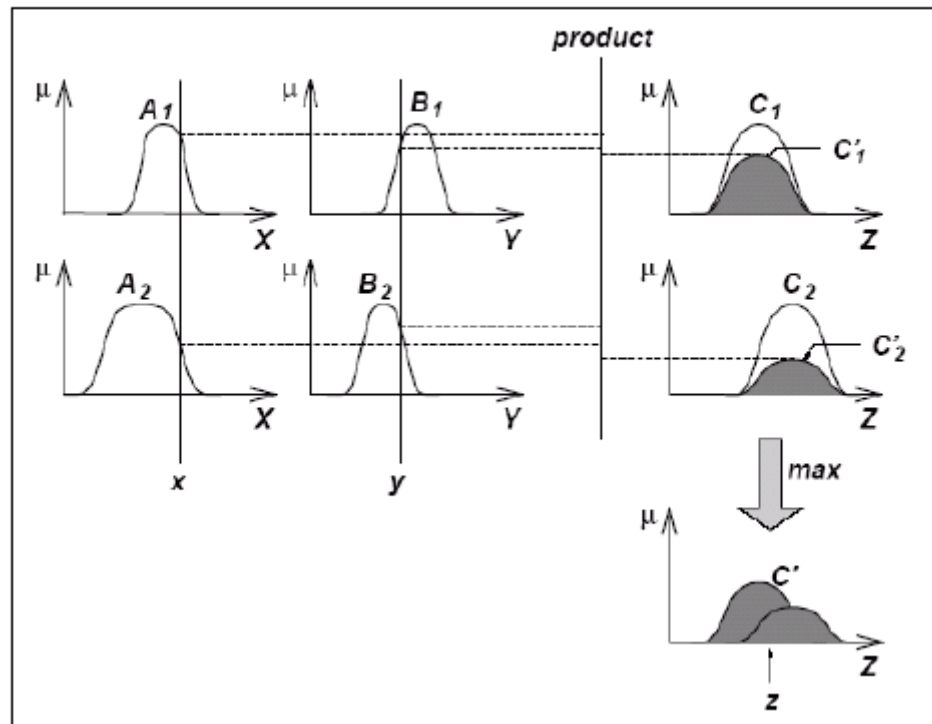
Max-Min Composition is used.



استنتاج فازی

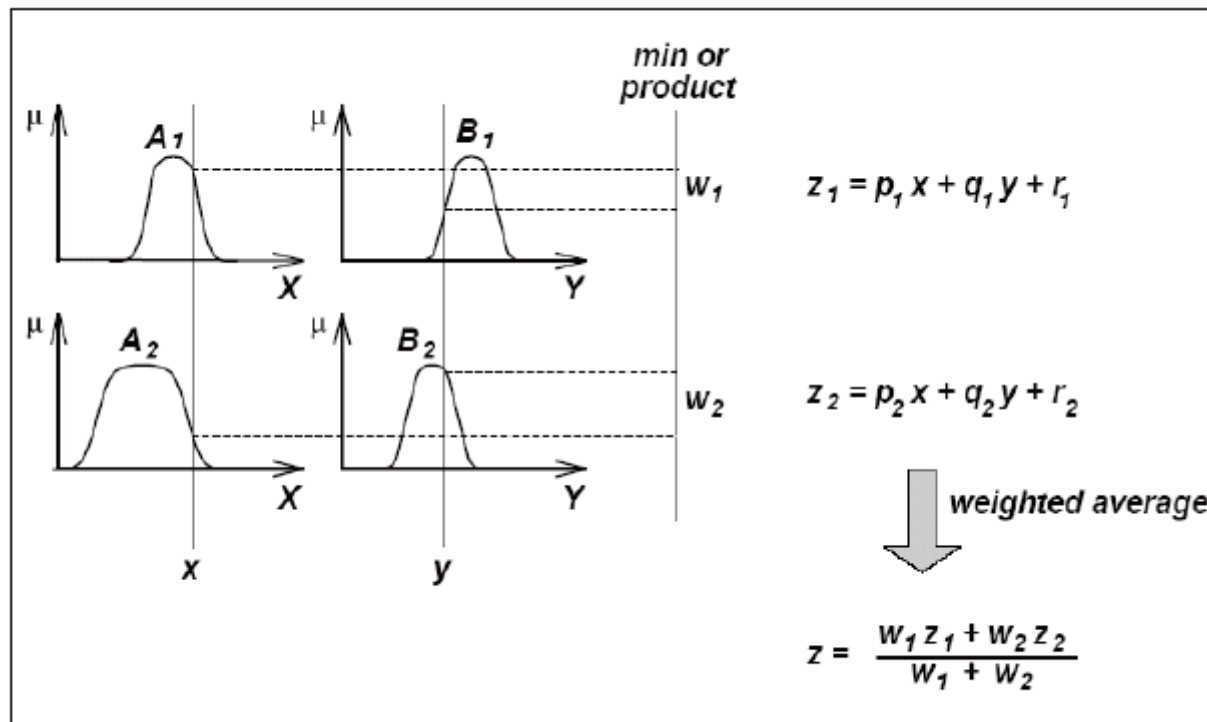
○ یافتن تابع عضویت خروجی (Implication)

Max-Product Composition is used.



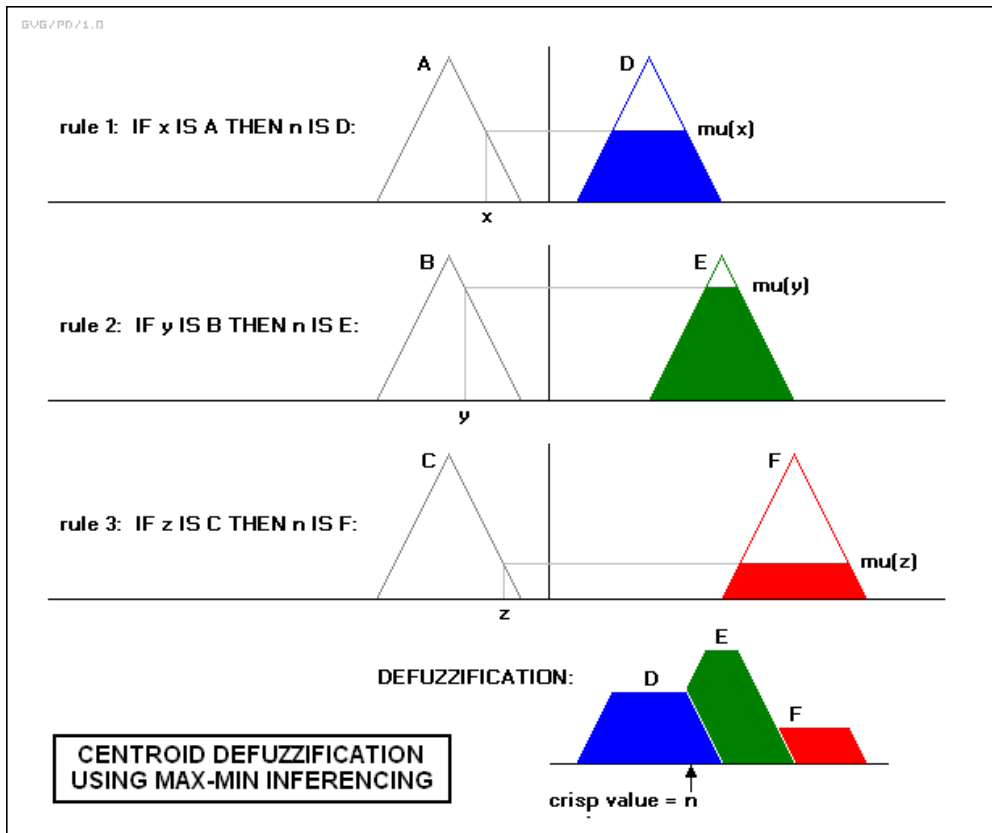
استنتاج فازی

○ یافتن تابع عضویت خروجی (Implication)



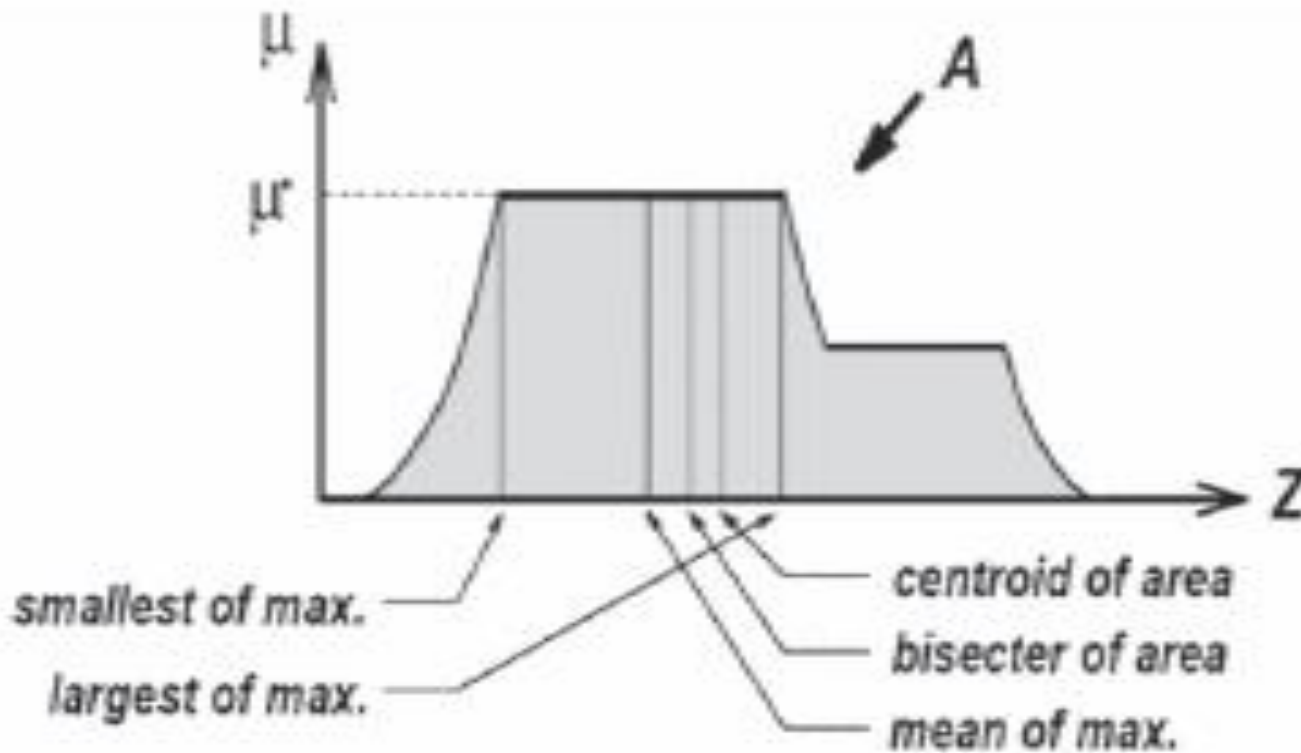
استنتاج فازی

○ استنتاج و تجميع خروجی ها (Aggregate)



استنتاج فازی

○ نافیازی سازی (defuzzification)



استنتاج فازی

○ انواع سیستم های فازی (بیان خروجی ها در قوانین):

○ Mamdani

خروجی ها به شکل فازی بیان شود

○ Sogeno

خروجی ها به شکل ترکیب خطی از ورودی ها با ضرایب مجهول یا به شکل ثابت

تعریف شود

سیستم استنتاج فازی-عصبی تطبیقی

«تنظیم توابع عضویت یک سیستم استنتاج فازی جهت تطابق

ورودی ها و خروجی های سیستم با یک مجموعه داده ی

آموزشی»

نوع سیستم: Sugeno

فرایند آموزش، ترکیبی حداکثر کاهش پس انتشار- کاهش

گرادیان