Session 1

3:08 PM Sunday, February 21, 2021

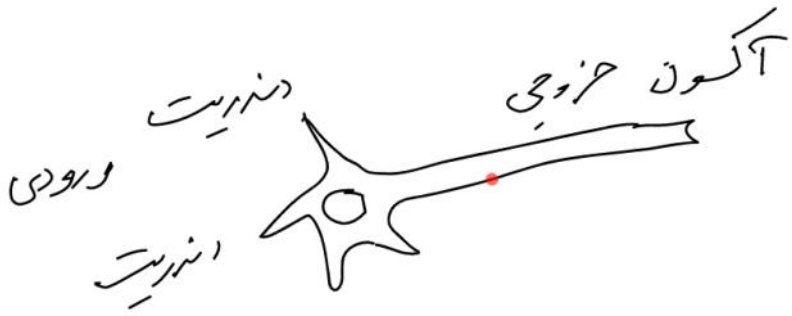
تعریف هوش: از دیدگاه فلسفی ← تطبیق با محیط. که به دو شکل خودش را نشان میدهد .
1) از طریق تکامل: بر فرض یک درخت بر اساس تکامل میتواند طوری تغییر کند که درخت های بعدی با یک محیط سرد تطابق داشته باشند. در الگوریتم های تکاملی در این مورد صحبت خواهد شد.
2) در لحظه: تنها انسان قابل این کار است. بر فرض با سرد شدن هوا برای خود لباس درست میکند.

هرچند درخت هوشمند است، اما این هوش بر اساس تطبیقی است که در تکامل به وجود آمده است و نسل فعلی نمیتواند از آن استفاده کند بلکه نسل های بعدی میتوانند خود را با محیط تطبیق دهند.

تفاوت هوش محاسباتی با هوش: در هوش محاسباتی بیشتر از ریاضیات استفاده می شود. ریاضیاتی اعم از: جبر، آمار و احتمال، منطق و ...

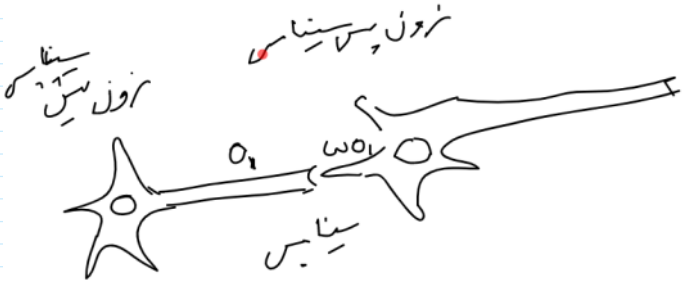
| شبکه عصبی

الهام گرفته از مغز انسان است.
در مغز انسان به سلول های عصبی نورون (Neuron) گفته میشود. در یک نورون ورودی هایی تحت عنوان دندریت دارد که معمولا 3 الی 4 تا می باشند. خروجی های آن نیز آکسون می گویند.

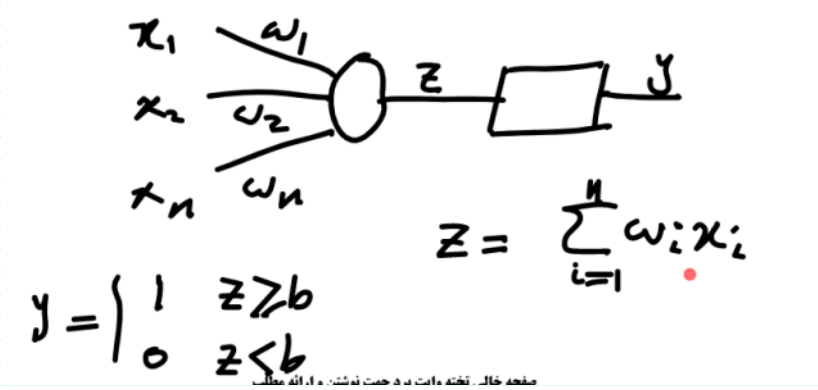


ارتباط بین ورودی و خروجی در یک نورون: فرض کنیم هر ورودی یک ولتاژ داشته باشد. اگر مجموع این ولتاژ ها از یک حد آستانه بیشتر شود باعث میشود آکسون (خروجی) یک شلیک انجام دهد. این خروجی به صورت پالس فرکانس می باشد.

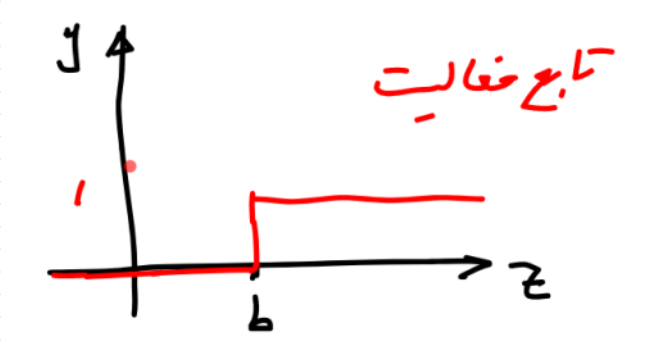
آکسون ها و دندریت ها از طریق یک اتصال شیمیایی به هم وصل می شوند. (اتصال فیزیکی نیست!) به این اتصال شیمیایی سیناپس میگویند.
وظایف متعدد سیناپس:
- تبدیل فرکانس به ولتاژ
- تقویت و تضعیف خروجی آکسون. یعنی اگر خروجی یک نورون O_1 باشد، ورودی نورون بعدی WO_1 می باشد.
به نورون قبلی نورون پیش سیناپسی و به نورون بعدی پس سیناپسی می گویند.



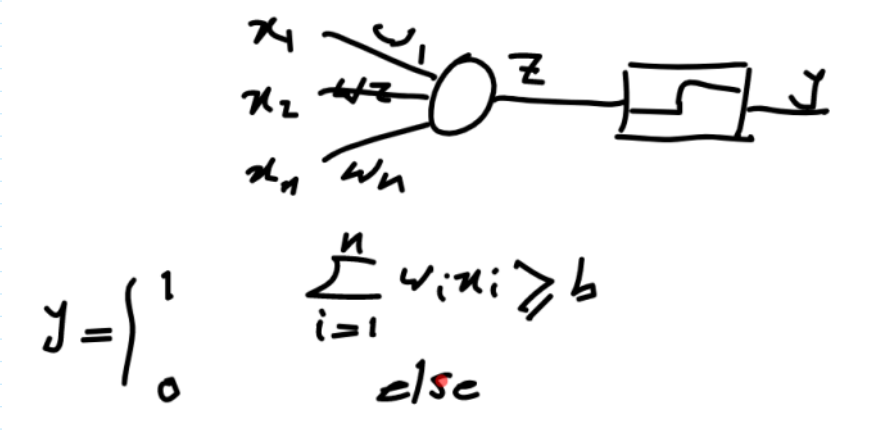
پرسپترون (Perceptron): یک مدل ریاضی ساده از یک نورون است. معمولا به شکل یک دایره کشیده می شود که دندریت x_1 تا x_n به آن وصل می شود. سیناپس ها به صورت w_1 تا w_n تعریف می شوند. و خروجی را Z می نامیم. خروجی طبق تعریف برابر است با مجموع وزن سیناپس در ورودی دندریت. خروجی نهایی y نامیده می شود که زمانی فعال میشود که از یک حد آستانه Threshold ای بیشتر شود.



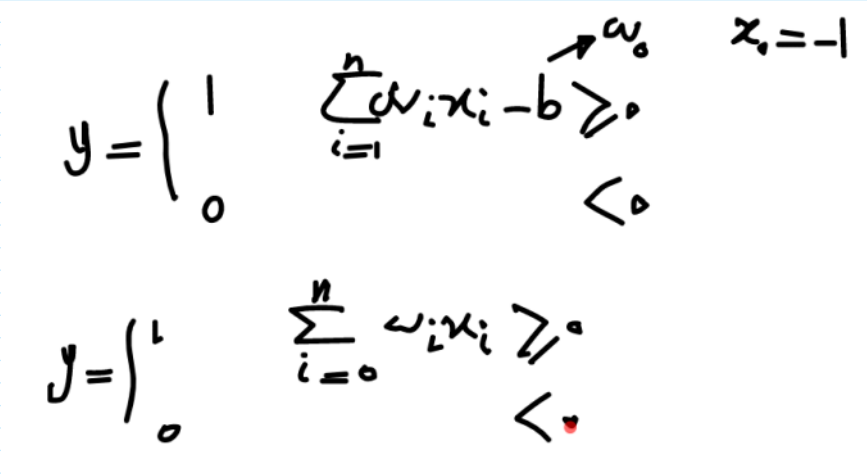
در واقع تابع بین z و y یک تابع پله می باشد. به طوری که تا زمانی که از b کوچکتر باشد خروجی 0 و اگر بزرگتر باشد 1 می باشد. به این تابع پله، تابع فعالیت (Activation Function) گفته می شود.



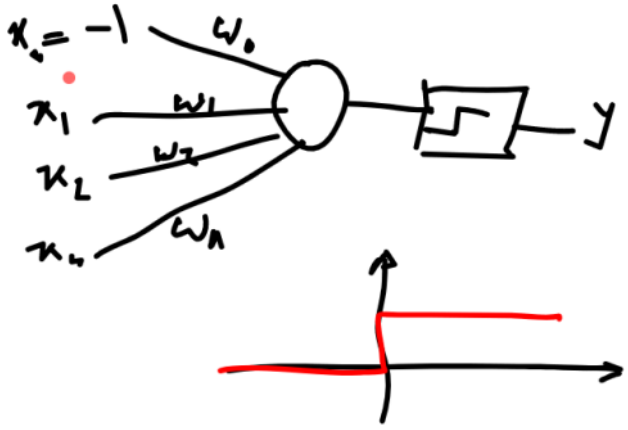
پس به طور کلی داریم:



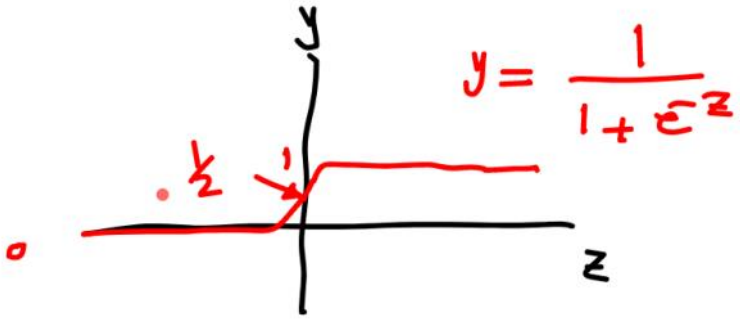
برای بهتر شدن شکل ریاضی، تلاش بر ساده تر کردن فرمول داریم. به طوری که رابطه y را تغییر میدهیم. ابتدا b را ابه سمت چپ نامساوی میبریم که b- میشود. b را به w_0 تغییر میدهیم و سپس یک x_0=-1 به آن اضافه میکنیم. در نهایت رابطه به شکل زیر تغییر میکند:



با تغییر رابطه فوق گویی داریم:



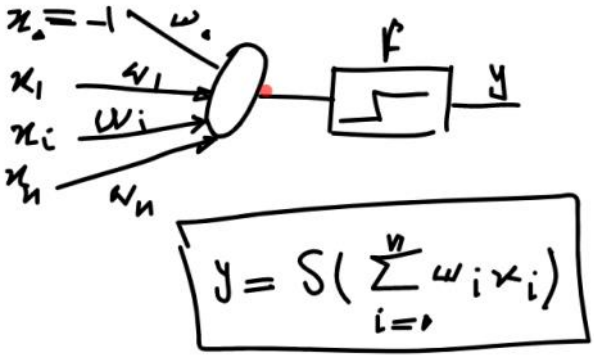
مشکل: تابع فعالیت اگر به صورت پله باشد مشتق پذیر نیست. پس برای مشتق پذیر بودن به جای استفاده از تابع پله از تابع سیگموئید (Sigmoid) استفاده می شود.



جلسه دوم

3:06 PM Tuesday, February 23, 2021

در جلسه قبل دیدیم که: پرسپترون ← مدل ریاضی ساده از نرون
مشاهده کردیم که مدل ساده پرسپترون به شکل زیر قابل نمایش است:



$$\boxed{y = S\left(\sum_{i=1}^{n} w_i x_i\right)}$$

که منظور از S تابع Sigmoid می باشد.

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right) \qquad \boxed{y = S\left(\sum_{i=1}^{n} w_i x_i\right)}$$

تابع فعالیت — تابع سیگموئید

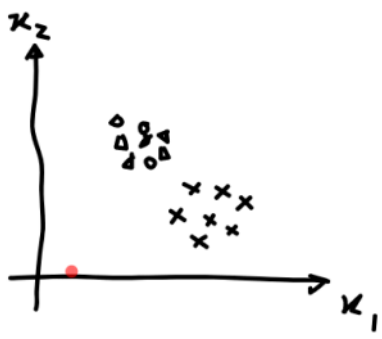
صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

تابع فعالیت لزوما سیگموئید نیست و بعدا خواهیم دید که بر فرض در شبکه های عمیق از تابع فعالیت رنو استفاده خواهد شد.

حال میخواهیم با یک مثال بررسی کنیم که پرسپترون چگونه در حل مسائل به ما کمک میکند. برای مثال فرض میکنیم سیب و پرتغال داریم که قطر و وزن آن ها اندازه گیری شده اند.

اگر y=1 را پرتغال و y=0 را سیب در نظر بگیریم. میدانیم که سیب نسبت وزن به قطر بیشتری نسبت به پرتغال دارد . در نمودار ضریدر ها پرتغال و باقی سیب هستند.

x_1	x_2	y
10	50	1
11	51	1
9	50	1
8	60	0
7	59	0
9	62	0



x_1 = قطر

x_2 = وزن

حال اگر بخواهیم خطی بکشیم که سیب و پرتغال را جدا کند، خط به صورت زیر می باشد:

حال اگر بخواهیم معادل این خط را بدست بیاوریم داریم:

$$-w_0 + w_1 x_1 + w_2 x_2 = 0$$

حال اگر بخواهیم وزن های این معادله خط را بدست بیاوریم. میدانیم در جایی که $x_1=1$ می باشد $x_2=0$ می باشد همچنین فرض میکنیم نقطه دیگر $x_1=10$ و $x_2=60$ می باشد. در این صورت داریم:

$$-w_0 + w_1 x_1 + w_2 x_2 = 0 \rightarrow -w_0 + w_1 = 0 \rightarrow w_1 = w_0$$

$$-w_0 + w_1 \times 10 + 60 w_2 = 0 \qquad 9w_1 = -60 w_2$$

در این حالت $w_2 = -3/20 w_0$ خواهد شد و داریم:

$$\div w_0 \quad -1 + x_1 - \frac{3}{20} x_2 = 0$$

$$-w_1 + w_0 x_1 - \frac{3}{20} w_0 x_2 = 0$$

با توجه به رابطه های فوق در میابیم که سمت چپ خط رسم شده $Z < 0$ و سمت راست آن $Z > 0$ می باشد. (برای صحت این موضوع میتوانیم نقطه (0, 0) را در معادله تست کنیم)

اگر بخواهیم این خط را به صورت ریاضی بدست آوریم، بایستی چه کار کنیم؟ میتوانیم از داده های آموزشی کمک بگیریم. تمامی x_1، x_2 و y^* های داده شده داده آموزشی می باشند.

$$-w_0 + w_1 x_1^l + w_2 x_2^l = y^{*l}$$

$$-w_0 + w_1 \times 10 + w_2 \times 60 = 1$$

$$\vdots$$

$$-w_0 + w_1 \times 4 + w_2 \times 62 = 0$$

سپس چند معادله فوق را میتوانیم به صورت ماتریس بنویسیم. به طوری که:

$$\underset{X}{\begin{bmatrix} -1 & 10 & 60 \\ -1 & 11 & 61 \\ -1 & 9 & 60 \\ -1 & 8 & 60 \\ -1 & 7 & 59 \\ -1 & 4 & 62 \end{bmatrix}} \underset{W}{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}} = \underset{\vec{y}^*}{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}$$

مجموع مربعات خطا برابر است با:

$$\text{مجموع مربعات خطا} = \sum_{l=1}^{6} \left(\sum_{i=0}^{n} w_i x_i^l - y^{*l} \right)^2 = (Xw - \vec{y}^*)^T \times (Xw - \vec{y}^*)$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

استاد توان 2 نگذاشت ولی بایستی در رابطه باشد. هم با باز کردن ضرب داخلی 2 ماتریس و هم با این منطق که ممکن است سیگما $w_i x_i$ از y^* کمتر شود و جاهایی بیشتر و منفی مثبت ها همدیگر را خنثی کنند ونتیجه صفر شود به این موضوع میتوان پی برد.

$$E = (Xw - \vec{y}^*)^T (Xw - \vec{y}^*)$$

سپس با مشتق گیری و برابر با صفر داریم:

$$|w_2|$$

$$\frac{dE}{dw} = 2X^T(Xw - y^*) = 0$$

$$X^TXw - X^Ty^* = 0$$

$$X^TXw = X^Ty^* \Rightarrow \boxed{w = (X^TX)^{-1}X^Ty^*}$$

صفحه خالی تخته وایت بورد جهت نوشتن و ارائه مطلب

نحوه مشتق گیری ماتریسی:

$$y = X^TX$$

$$\frac{dy}{dx} = 2X$$

$$y = (AX - B)^T(AX - B)$$

$$\frac{dy}{dx} = 2A^T(AX - B)$$

برای اثبات بایستی به فرم غیر برداری بنویسیم. سپس درایه به درایه مشتق گیری کنیم و سپس به حالت بردار در بیارویم و مشاهده میکنیم که چنین روابطی حاصل میگردند.

برای این که مطمئن شویم کجا بایستی ترنهاده گذاشت، بایستی اندازه ها با هم بخوانند. برفرض مثال داریم:

$$\underset{1\times 1}{y} = (\overset{1\times L}{AX} - \underset{L\times n\ m\times 1}{B})^T(\overset{L\times 1}{AX} - B)$$

$$\underset{m\times 1}{\frac{dy}{dx}} = 2\underset{m\times L}{A^T}\underset{L\times 1}{(AX - B)}$$

پس در نهایت w برابر است با :

$$E = (Xw - y^*)^T (Xw - y^*)$$

$$w = (X^T X)^{-1} X^T y^*$$

راه دیگر محاسبه نیز به روش زیر میباشد که در درس جبر خطی نیز مشاهده کرده ایم:

$$Xw = y^*$$

$$(2) \quad X^T X w = X^T y^*$$

$$(3) \quad w = (X^T X)^{-1} X^T y^*$$

طبق تئوری 14 فصل 6 کتاب جبرخطی و کاربرد های آن (David C Lay)، زمانی میتوان از (2) به (3) رسید که ستون های ماتریس X مستقل خطی باشند. یعنی اگر بر فرض مثالی داشته باشیم که در سیب ها و پرتغال ها $x_2 = 2x_1$ باشند، آنگاه دیگر ستون های ماتریس X مستقل خطی نخواهد بود و X^TX معکوس پذیر نیست. پس برای حل (2) در این موارد خاص بایستی از روش Row Reduction Algorithm کمک گرفت.

- توجه شود که روش توضیح شده برای حل مسائل Regression می باشد و خط مورد نظر مارا در مسئله Clustering تولید نخواهد کرد! در این مورد در جلسه بعد بیشتر صحبت خواهد شد

جلسه سوم

3:10 PM Sunday, February 28, 2021

چه ویژگی هایی برای حل یک موضوع مناسب تر هستند؟ 1) مستقل خطی باشند. 2) مرتبط به موضوع باشند.

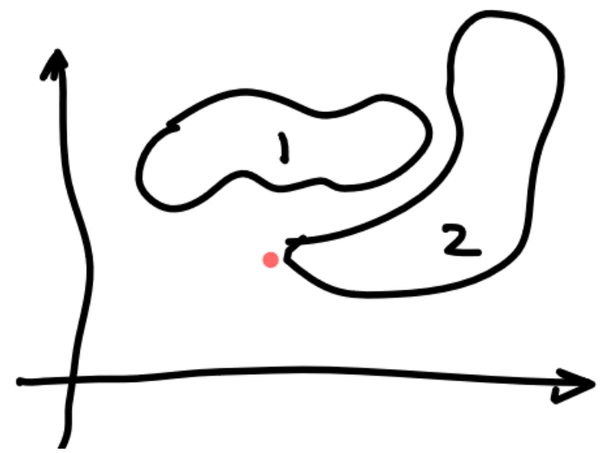
در مساله که حل کردیم 2 بعدی بود و خط جدا کننده است (با یک پرسپترون منحنی نمیتوان ساخت). در 3 بعدی صفحه و از 4 بعد به بالا ابر صفحه داریم.

انواع یادگیری:

- **یادگیری با نظارت (Supervised):** یادگیری از روی نمونه هایی که label دارند
- **یادگیری بدون نظارت (Unsupervised):** بر اساس شباهت. مثل خوشه بندی (Clustering) که داده های شبیه هم در یک گروه قرار میگیرند.
- **یادگیری تقویتی (Reinforcement) :** یادگیری بر اساس تجربه. مثلا از خانه به دانشگاه میرویم یک مسیری انتخاب میکنیم میبینیم طولانی شد. سپس یک مسیر دیگر میرویم و در آخر بر اساس تجربه مسیری با کوتاه ترین زمان به دانشگاه را انتخاب میکنیم (صحیح و خطا) البته کمی با صحیح و خطا از لحاظ رسیدن به جواب متفاوت است. از این جهت که بر فرض تا زمانی که به دانشگاه نرسیده ایم نمیدانیم که دچار خطا شده ایم یا نه ولی در در آزمون و خطا بالافاصله متوجه خطا شدن خواهیم شد.

شبکه های عصبی تمامی 3 یادگیری فوق را میتوانند انجام دهند. در این درس تمرکز بر شبکه های عصبی برای یادگیری با نظارت را داریم و روی باقی تمرکز چندانی نداریم.

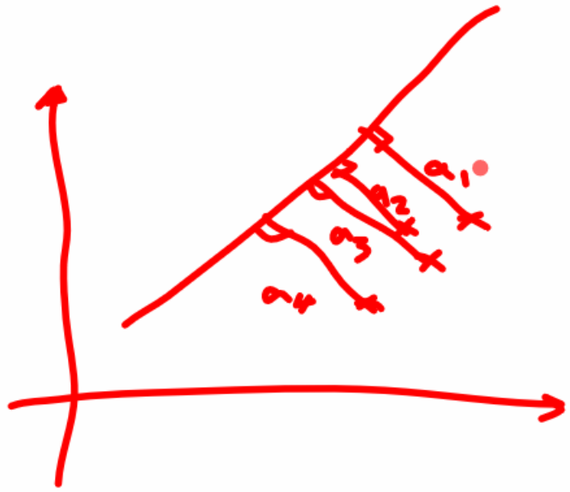
فرض کنید کلاسی داریم که به شکل خطی جدا نمی شوند:



برای جداسازی این کلاس ها با یک پرسپترون نمیتوان جداسازی کرد.

تفاوت رگرسیون و دسته بندی: در رگرسیون خروجی یک منحنی پیوسته است. ولی در دسته بندی خروجی گسسته است.
برای مثال از رگرسیون میتوانیم زمان با سرعت ماشین در زمان را در نظر بگیریم و میخواهیم سرعت ماشین در آینده را پیشبینی کنیم.
برای دسته بندی نیز بر فرض میتوانیم یک سری داده از آدم ها در نظر بگیریم که یا بیماری قلبی دارند یا خیر. سپس میخواهیم پیشبینی کنیم با وجود feature های یک آدم جدید آیا این فرد دارای بیماری قلبی می باشد یا خیر. مثال جلسه قبل یک مساله دسته بندی بود که با least squares error خواستیم دقیق بودن آن را مورد نظر قرار دهیم.
با پرسپترون هم میتوانیم هم regression هم classification را انجام دهیم.

least squares error در regression:



$$a_1^2 + a_2^2 + a_3^2 + a_4^2$$

تلاش بر این است که حاصل جمع نوشته شده را مینیمم کند.

کراس آنتروپی: اگر y را خروجی شبکه و خروجی خواسته شده *y باشد، هدف مینیمم کردن E می باشد که داریم:

$$y^1 \quad y^{*1}$$

$$\vdots$$

$$\min E = \sum_{l=1}^{L} (y^l - y^{*l})^2 = (y - y^*)^T (y - y^*)$$

$$y^L \quad y^{*L}$$

$$y \quad y^*$$

اگر خروجی پرسپترون را احتمال کلاس بگیریم. p*L یعنی کلاس داده L ام فرض پرتغال باشد (پرتغال 1 و غیر آن 0) و داریم:

$$\cdot\ \bigcirc^{P}$$

$$-\sum_{l=1}^{L}\left(p_l^{*}\log p_l+(1-p_l^{*})\log^{(1-p_l)}\right.$$

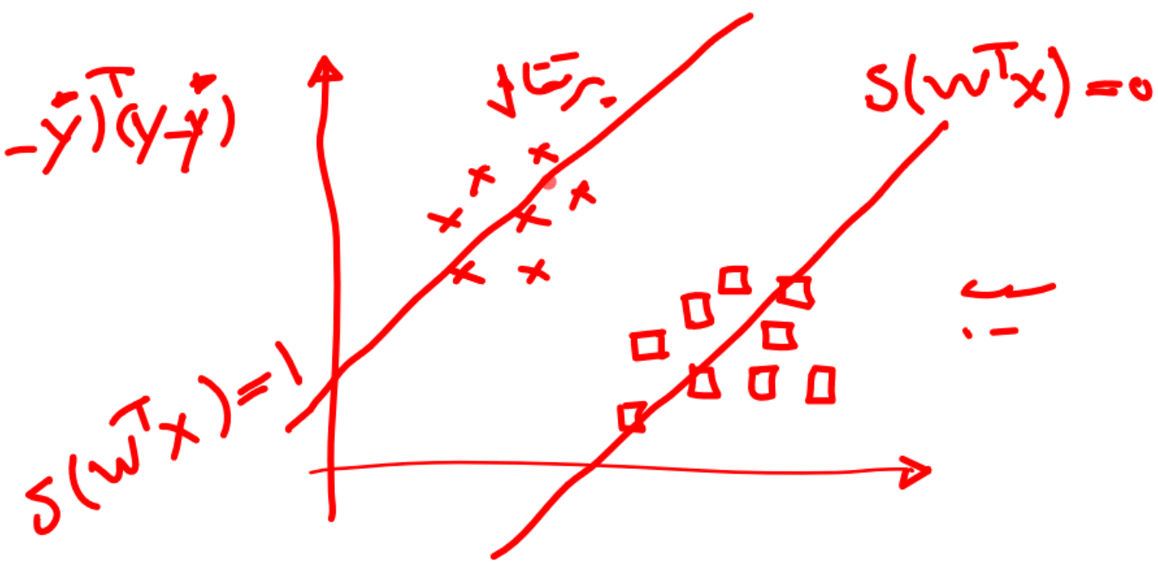
میتوان اثبات کرد وقتی کراس آنتروپی مینیمم میشود که در مثال ما کلاس هایی پرتغال هستند احتمال 1 و برای کلاس هایی که پرتغال نیستند احتمال صفر شود

$$E=-\sum_{l=1}^{L}\left(p_l^{*}\log p_l+(1-p_l^{*})\log(1-p_l)\right)$$

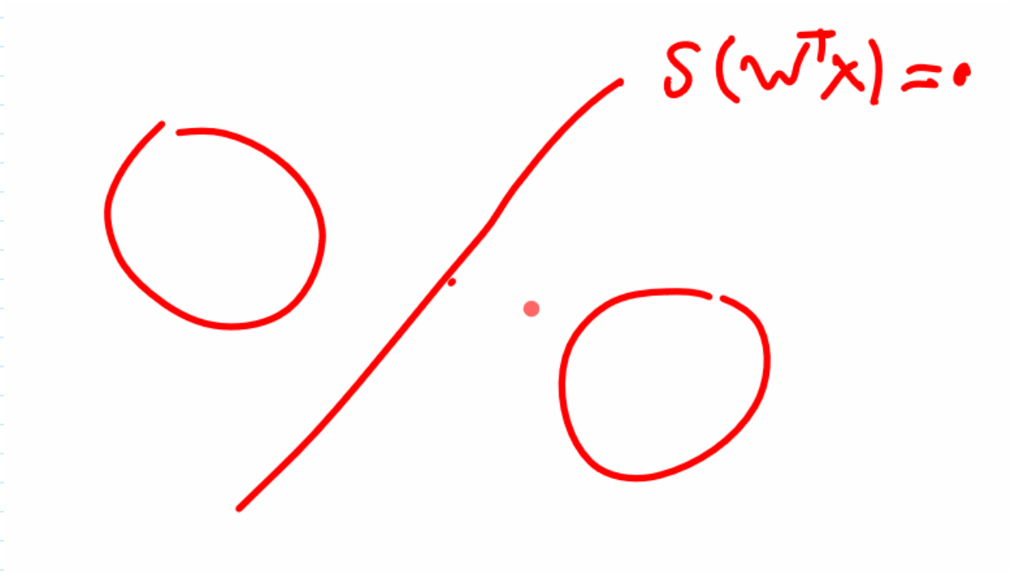
$$p_l=S\left(\sum_{i=0}^{n}w_i x_i\right)$$

در حل به کمک MSE (Mean Squared Error) صحیح میکنیم مربع خطات رو کم کنیم (همون که ریاضیش جلسه پیش بود، اون روش برای classification جواب نیست):

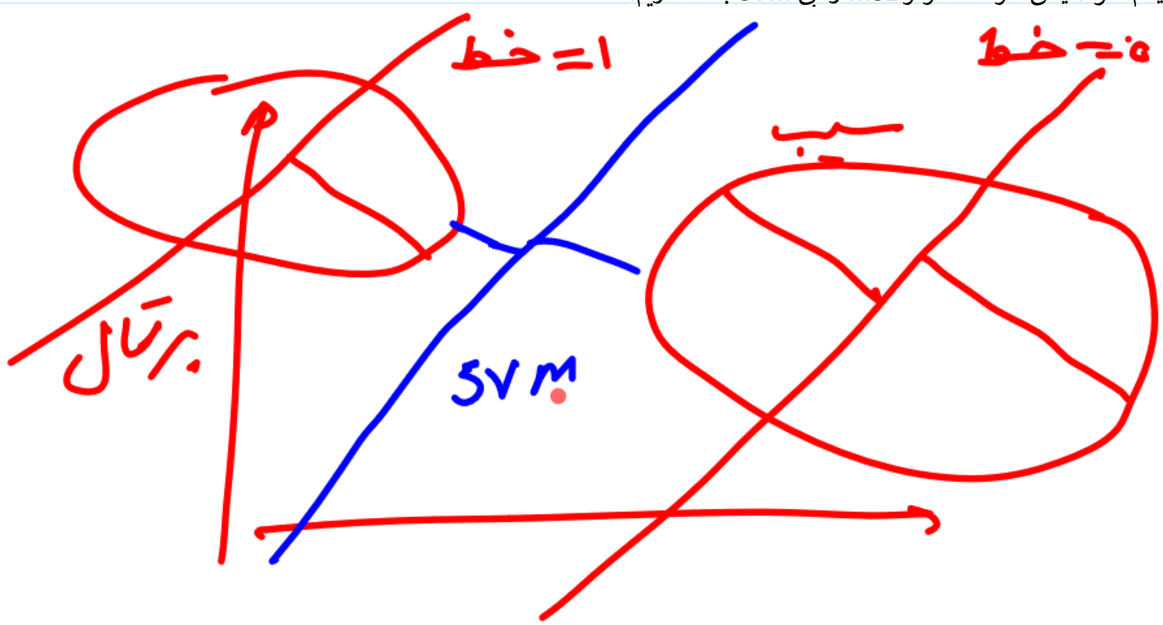
$$\min\ (y-\hat{y})^T(y-\hat{y})$$



اما در کراس آنتروپی میخواهیم خطی بدست آوریم که فاصله از داده های هردو کلاس به حداکثر برسد (برای مساله دسته بندی):

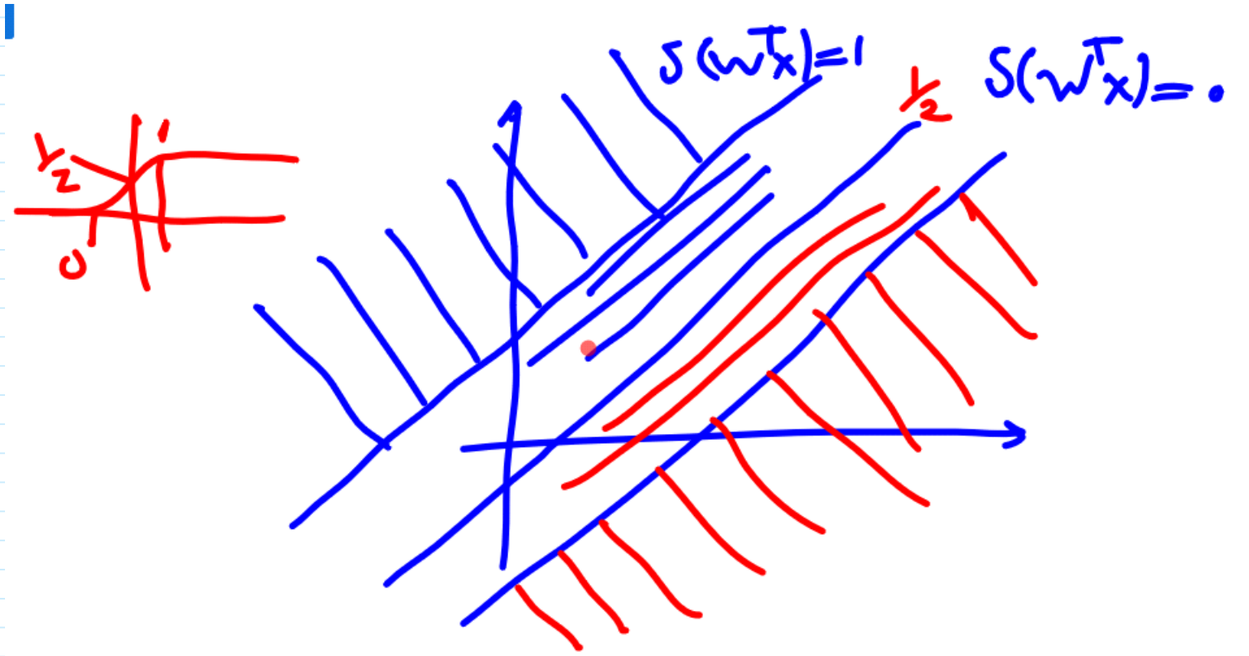


روش دیگر نیز SVM می باشد که خطی بدست می آید که فاصله تا هر دو داده ماکسیمم شود. یعنی اگر خط قرمز MSE و آبی SVM باشد داریم:



کراس آنتروپی نه به خوبی SVM می باشد و نه به بدی MSE (SVMدقیق تر است و در یادگیری ماشین تدریس می شود)
پس برای Regression از MSE و از SVM و کراس آنتروپی برای classification استفاده میکنیم.

سیگموید فضا را به 4 قسمت تقسیم میکند:



1) بخش بزرگتر از 1
2) بخش کوچکتر از 0
3) بخش بین 1/2 و 1
4) بخش بین 0 و 1/2

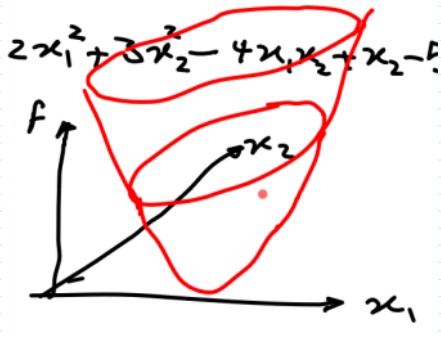
جلسه چهارم

3:03 PM Tuesday, March 2, 2021

میخواهیم به کمک گرادیان تابع هدف کراس آنتروپی را مینیمم کنیم. لذا یک مختصر توضیحی در مورد گرادیان خواهیم داشت تا بتوانیم وزن های پرسپترون را برای تابع هدف کراس آنتروپی بدست آوریم.

Hypothetis: $F(x_1,x_2) = 2x_1^2 + 3x_2^2 - 4x_1x_2 + x_2 - 5$

اگر بخواهیم روی مختصات رسم کنیم به صورت فرضی یک شکلی همانند شکل زیر خواهد داشت:



میخواهیم بردار گرادیان را بدست آوریم. علامت گرادیان را با یک مثلث برعکس نشان میدهیم و داریم:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

در نقطه $x_1 = 2$ و $x_2 = 3$ میخواهیم بردار گرادیان را مشخص کنیم. داریم:

$$\frac{\partial f}{\partial x_1} = 4x_1 - 4x_2 \Big|_{x_1=2,\, x_2=3} = -4$$

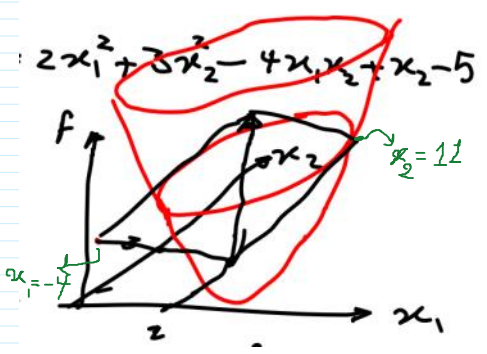
لذا درایه اول بردار گرادیان در نقطه مورد نظر 4- می باشد. برای محاسبه درایه دوم نیز داریم:

$$\frac{\partial f}{\partial x_2} = 6x_2 - 4x_1 + 1 = 11$$
$$18 - 8 + 1$$

پس بردار گرادیان در نقطه خواسته شده برابر است با:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{x_1=2,\, x_2=3} = \begin{bmatrix} -4 \\ 11 \end{bmatrix}$$

اگر بخواهیم بردار بدست آورده را در نقطه مورد نظر رسم کنیم، در جهت x_1 منهای 4 تا باید جلو برویم و در جهت x_2 نیز 11 تا جلو برویم و داریم:



این بردار در صفحه x_1-x_2است و در جهتی می باشد که بیشترین مقدار افزایش تابع را در آن جهت داشته باشیم. یعنی اگر بخواهیم ماکسیمم تابع را در کمترین مسیر بدست بیاوریم در جهت گرادیان حرکت میکنیم و اگر بخواهیم در کمترین مسیر به مینیمم تابع برسیم عکس جهت گرادیان حرکت میکنیم.

پس X^{new} برابر است با: (گاهی اوقات old را با t و new را با t+1 نمایش میدهند)

$$X^{new} = \begin{bmatrix} x_1^{new} \\ x_2^{new} \end{bmatrix} = \begin{bmatrix} x_1^{old} \\ x_2^{old} \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{X^{old}}$$

$$X^{t+1} = X^{t} - \eta \nabla f(X^{t})$$

یک ضریب اِتا نیز قبل گرادیان می آید و معنی اندازه قدم را دارد(البته در صورتی که بردار گرادیان را نرمال کنیم). یعنی مشخص میکند چه میزان خلاف جهت گرادیان باید حرکت کنیم تا به مینیمم برسیم.

مثال فرض کنیم تابع زیر را داریم و در نقطه x_0 گرادیان را بدست آورده ایم. سپس به سمت مینیمم حرکت میکنیم وبه x_1 میرسیم. این روند ادامه می یابد و مدت حرکتش وابسته به این است که گرادیان چه زمانی صفر می شود. اندازه اِتا اگر بزرگ باشد ممکن است بین 2 نقطه نوسان داشته باشیم (همون منحنی نیم دایره که کشیده) و هیچ موقع به مینیمم نرسیم.

همچنین یک مشکل گرادیان هم این است که ممکن است در مینیمم محلی بایستد.

حال اِتا را چه اندازه بگیریم! ؟ ابتدا اِتا را یه اندازه متوسطی در نظر میگیریم. اگر $\Delta f = f^{new} - f^{old}$ منفی بود. آنگاه یعنی خوب داریم پیش میریم پس اِتا را بزرگ میکنیم. مثلا در 1.1ضرب میکنیم. ولی اگر این Δf مثبت شود. یعنی به مینیمم نزدیک نمیشیم وبلکه داریم دورتر میشیم. پس باید اِتا را کوچکتر میکنیم یعنی مثلا در 0.9 ضرب میکنیم.

برای این که اِتا اندازه قدم باشد بایستی گرادیان در اندازه اش تقسیم شود تا بردار اندازه 1 داشته باشد:

اندازه قدم

$$X^{t+1} = X^{t} - \eta \frac{\nabla f(X^{t})}{\|\nabla f(X^{t})\|}$$

یک روش دیگر برای بدست آوردن مینیمم در توابع ریاضی این است که مشتق بگیریم و برابر با صفر قرار دهیم. با توجه به این که گرادیان ممکن است در مینیمم محلی گیر کند پس چرا از مشتق مساوی صفر استفاده نمیکنیم! ؟

علت: باید رُند تک تک x_i ها را برابر با صفر قرار بدهیم. یک دستگاه را حل کنیم و سپس تمامی x_iهای مختلف را امتحان کنیم ببینیم کدام مینیمم کمتری است. مشکل این است که حل دستگاه با ابعاد زیاد ممکن است راه حل بسته ای برای آن موجود نباشد.

آنتروپی: معنی پی نظمی است. مثلا وقتی بگوییم با داده هایی با احتمال 1 سیب و با احتمال 0 پرتغال و داده دیگر بالعکس ← قطعیت کامل داریم و پی نظمی = 0 است.
اگر اگر با احتمال 0.5 سیب و احتمال 0.5 پرتغال باشد پی نظمی بیشتر است.

کراس آنتروپی تا حد امکان میخواهد این آنتروپی را کم کند. یعنی تا حد امکان یک خط جدا کننده پیدا کند که برفرض بگوید یک داده ای با احتمال 1 سیب و 0 پرتغال است.

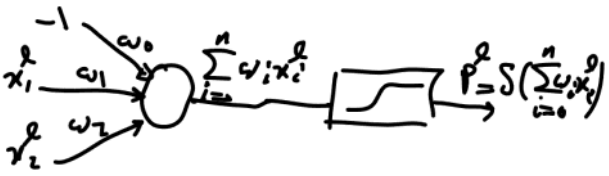
$$\text{کراس آنتروپی } E = -\sum_{l=1}^{L} \left(P_l^{*} \log P_l + (1-P_l^{*}) \log(1-P_l)\right)$$

$$\text{آنتروپی} = -\sum_{l=1}^{L} \left(P_l \log P_l + (1-P_l) \log(1-P_l)\right)$$

تفاوت کراس آنتروپی با آنتروپی در * است. در واقع * مقدار های واقعی است.

$$P_l = S\left(\sum_{i=0}^{n} w_i x_i\right)$$

یادآوری:



$$\omega_i^{t+1} = \omega_i^t - \eta \frac{\partial E}{\partial \omega_i}\Big|_t$$

Cross entropy

برای محاسبه رند کراس آنتروپی نیز داریم:

$$\frac{\partial E}{\partial \omega_i} = -\sum_{\ell=1}^{L}\left(P_\ell^* \frac{\partial P_\ell}{\partial \omega_i}\Big/ P_\ell - \frac{(1-P_\ell^*)}{1-P_\ell}\frac{\partial P_\ell}{\partial \omega_i}\right)$$

$$= \sum_{\ell=1}^{L}\left(\frac{1-P_\ell^*}{1-P_\ell} - \frac{P_\ell^*}{P_\ell}\right)\frac{\partial P_\ell}{\partial \omega_i}$$

$$P_\ell = S(\overbrace{\sum_{i=0}^{n} \omega_i x_i}^{z})$$

$$P_\ell = S(z) = \frac{1}{1+e^{-z}}$$

$$S'(z) = S(z)(1-S(z))$$

$$\frac{\partial P_\ell}{\partial \omega_i} = \frac{\partial P_\ell}{\partial z} \times \frac{\partial z}{\partial \omega_i} =$$

علت مشتق :

$$\underbrace{\frac{1}{1+e^{-z}}}_{S(z)}\overbrace{\left(1-\frac{1}{1+e^{-z}}\right)}^{1-S(z)} = \frac{e^{-z}}{(1+e^{-z})^2}$$

$$\frac{\cancel{1}+e^{-z}-\cancel{1}}{1+e^{-z}}$$

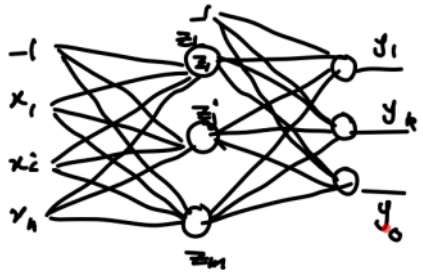
$$\omega_i^{t+1} = \omega_i^t - \eta \sum_{\ell=1}^{L}\left(\frac{1-P_\ell^*}{1-P_\ell} - \frac{P_\ell^*}{P_\ell}\right) S'\left(\sum_{i=0}^{n} \omega_i x_i^\ell\right) x_i^\ell$$

پس ما میتوانیم به این شکل وزن هارا از یک w_i تصادفی انتخاب کنیم و بعد با این رابطه w_i بعدی را بدست بیاوریم تا وقتی مینیمم نشده ادامه دهیم تا به جواب برسیم.

اگر نیاز به تشکیل منحنی برای جدا سازی داشته باشیم باید به سراغ MLP برویم. در شبکه ای از چندلایه پرسپترون که اگر x_1 تا x_n ورودی هایمان باشد، طبق ساده سازی قبلی x_0=-1 را نیز اضافه میکنیم . به لایه های وسطی لایه های پنهان میگویند.

MLP بایستی حداقل 3 پرسپترون داشته باشد، این که مقدار دقیق آن چه میزان می باشد میتوان با روش هایی نظیر صحیح و خطا حاصل کرد. به روش ریاضی اثبات شده است که یک شبکه عصبی پرسپترون که تابع فعالیت غیر خطی داشته باشد، اگر حداقل 3 لایه داشته باشد هر ابرمنحنی غیر خطی را میتواند تولید کند.
به لایه آخر لایه خروجی می گویند.
MLP بایستی حداقل 3 لایه داشته باشد.

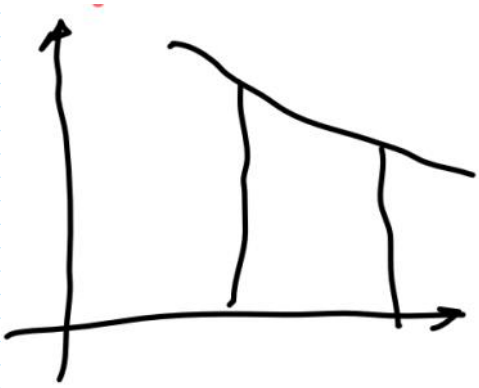
MLP



در مورد MLP جلسه بعدی بیشتر صحبت خواهیم کرد.

جلسه پنجم

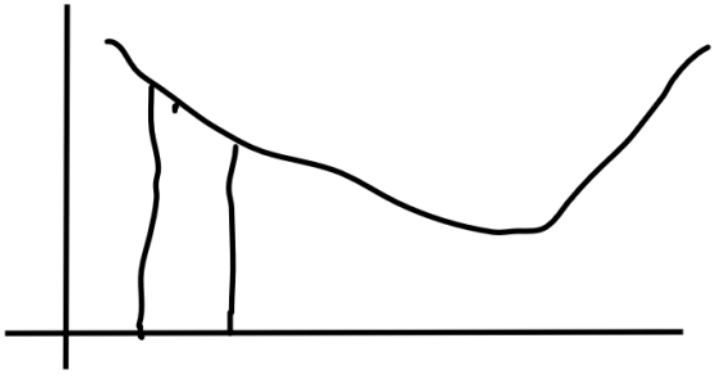
3:00 PM Sunday, March 7, 2021



در گرادیان وقتی شیب کنده باید قدم ها بزرگ باشه تا سریع تر به جواب برسیم

وقتی هم شیب اینطوری زیاده باید اندازه قدم کم باشه که نوسان نشه.

آیا شیب کم باشه لزوما به جواب میرسیم؟ نه



در این مثال در 2 نقطه مشخص شده شیب کم است ولی با جواب فاصله داریم.

در جلسه قبل دیدیم یک راه به روش زیر است:

$$w^{t+1} = w^{t} - \eta \frac{\nabla f(w^{t})}{\|\nabla f(w^{t})\|} \quad (1)$$

یک راه دیگر این است که علاوه بر نرمال کردن، از نرم 2 استفاده کنیم یعنی:

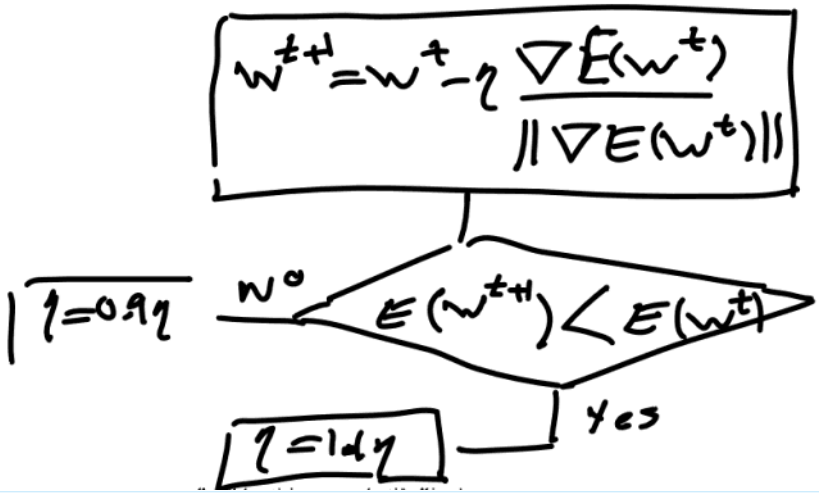
$$(2) \quad w^{t+1} = w^{t} - \eta \frac{\nabla f(w^{t})}{\|\nabla f(w^{t})\|^2}$$

اینطوری باعث میشه جایی که شیب زیاده اندازه قدم کم بشه جایی که شیب کمه اندازه قدم زیاد بشه.

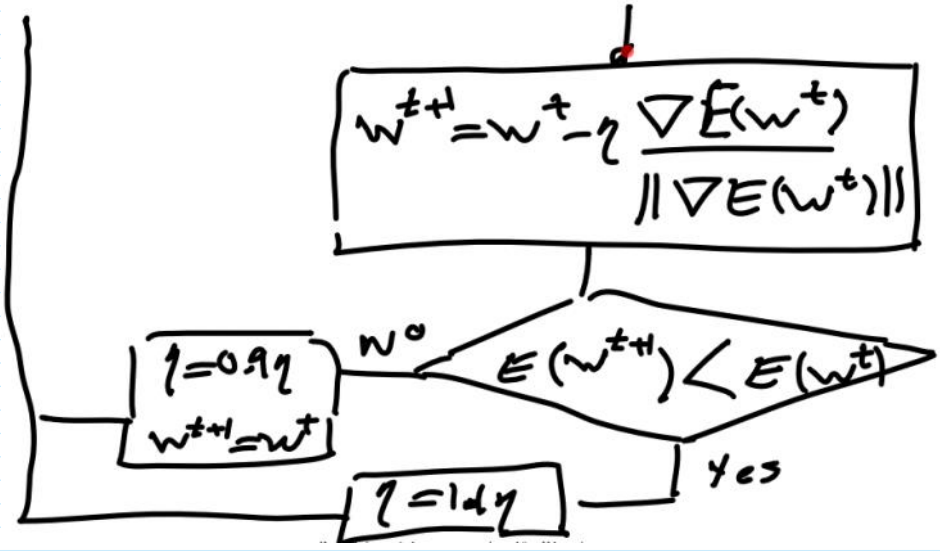
* وقتی به جواب رسیدیم که شیب خیلی کمه اون موقع باعث نمیشه که بپریم اونور از جواب خارج شیم؟ نه! اون موقع خود گرادیان خیلی کم میشه و اون کسر به 0 میل میکنه

* استاد پیشنهاد کرد که از فرمول (1) استفاده بشه که اتا 0.1 یا 0.01 در ابتدا قرار بگیره.

*همانطور که در جلسه پیش دیدیم، میتوانیم با فلوچارت زیر نیز ببینیم که اتا چگونه تغییر کند:



میشه وقتی اوضاع خراب شد w جدید را برابر w قدیم گذاشت، ولی خب میشه هم نذاشت چون با کم کردن اتا دوباره برمیگرده به وضعیت قبلی. در نهایت داریم:



| شبکه چند لایه ای پرسپترون (MLP)

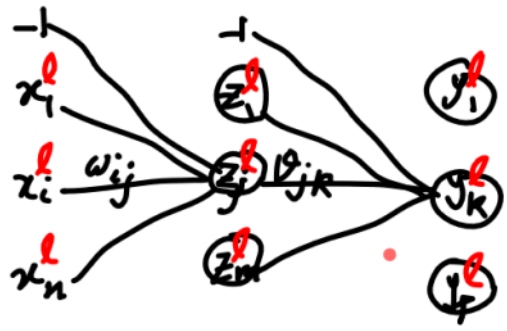
برای این که شلوغ نشه استاد فقط استاد از هر لایه فقط یک نرون رو رسم کرد شبکه اش رو (روابط برای داده | می باشد):

شبکه چند لایه پرسپترون

MLP

ورودی

مخفی



$$z_j^l = S\left(\sum_{i=0}^{n} w_{ij} x_i^l\right)$$

$$y_k^l = S\left(\sum_{j=0}^{m} v_{jk} z_j^l\right)$$

1- رگرسیون: از MSE یا SSE استفاده میشود.

$$E_{SE} = \frac{1}{2} \sum_{l=1}^{L} \sum_{k=1}^{P} \left(y_k^l - \hat{y}_k^l\right)^2$$

2- دسته بندی : از کراس آنتروپی و SVM. مثال از کراس آنتروپی:

$$E_{CE} = -\sum_{l=1}^{L} \sum_{k=1}^{P} \left(\hat{y}_k^l \log y_k^l + (1-\hat{y}_k^l)\log(1-y_k^l)\right)$$

* اگر داده ها را یک دفعه با هم یادگیری انجام دهیم batch learning و اگر هر داده جدید که میاد همون لحظه جداگونه یادگیری بشه Online learning می گویند. در batch learning وقتی هر batch که یاد میدهیم میشود یک epoch. این epoch در کل میتواند یک mini batch باشد یا batch یا کل داده ها.
* اگر در فرمول E_{SSE} ما سیگما اول رو حذف کنیم میشود Online learning. چون هر داده جدا خطاگیری انجام میشود. ولی وقتی Batch learning هست همگی با هم هستند.
* تفاوت Online و batch: در online learning هر داده جهت رو عوض میکند و نوسان میتواند بکند. ولی در batch learning متوسط جهت ها گرفته میشود و لذا نتیجه بهتری دارد. امروزه از mini batch learning استفاده میکنیم زیرا باعث میشود که از مینیمم های محلی فرار کنیم.
* در این درس فرض میکنیم کلا همه داده هارا با هم آموزش میدیم و کاری با batch و mini batch اینا نداریم.

فرض شود 1000 تا یا یک میلیون batch داده داریم (بسته به اندازه micro batch یا mini batch یا batch میگویند). همانطور که از جلسه قبل دیدیم خروجی هایی که میدونستیم چین یه * داشت. (درواقع اگر * داشته باشد desired value و اگر نداشته باشد predicted value یا observed value میگویند). داریم:

$$\begin{matrix} x_1^1 & \cdots & x_n^1 & \hat{y}_1^1 & \cdots & \hat{y}_p^1 \\ \vdots & & & & & \\ x_1^l & \cdots & x_n^l & \hat{y}_1^l & \cdots & \hat{y}_p^l \\ \vdots & & & & & \\ x_1^L & \cdots & x_n^L & \hat{y}_1^L & \cdots & \hat{y}_p^L \end{matrix}$$

با همین داده ها بخوایم online و batch را بررسی کنیم داریم:

$$\begin{bmatrix} x_1^1 & \cdots & x_n^1 & \bar{y}_1^1 & \cdots & \bar{y}_p^1 \\ \vdots & & & & & \\ x_1^l & \cdots & x_n^l & \bar{y}_1^l & \cdots & \bar{y}_p^l \\ \vdots & & & & & \\ x_1^L & \cdots & x_n^L & \bar{y}_1^L & \cdots & \bar{y}_p^L \end{bmatrix}$$

batch

$$E = \sum_{l=1}^{L=1000} \sum_{k=1}^{p}$$

$$E = \sum_{k=1}^{p}$$

on line

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

وقتی میخوایم مشتق بگیریم در رگرسیون دیگه سیگما دوم رو نداریم چون k رو فیکس کردیم و داریم:

$$v_{jk}^{new} = v_{jk}^{old} - \eta \frac{\partial E}{\partial v_{jk}} \qquad j = 0, 1, \ldots, m \quad k = 1, \ldots, p$$

$$① \quad \frac{\partial E_{SE}}{\partial v_{jk}} = \underbrace{\sum_{l=1}^{L} (y_k^l - \bar{y}_k^l)} \frac{\partial y_k^l}{\partial v_{jk}}$$

$$② \quad \frac{\partial E_{CE}}{\partial v_{jk}} = \underbrace{+\sum_{l=1}^{L} \left(\frac{1-\bar{y}_k^l}{1-y_k^l} - \frac{\bar{y}_k^l}{y_k^l} \right)} \frac{\partial y_k^l}{\partial v_{jk}}$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

در مقایسه بین 2 حالت، فقط قسمتی که زیرش خط کشیده ایم متفاوت است. اما مشتق جزئی ای که با رنگ سبز دورش خط کشیده ایم چطور محاسبه میشود؟
میتوان از chain rule کمک گرفت (جلسه پیش انجام شد) و مشتق سیگموید نسبت به U را نوشت. دیگر مشتق سیگموید نسبت به U را نمینویسم (چون جلسه قبل نوشتیم) و در کل داریم:

$$y_k^l = S\Big(\underbrace{\sum_{j=0}^{m} v_{jk} z_j^l}_{u}\Big)$$

$$\frac{\partial y_k^l}{\partial v_{jk}} = S'\Big(\sum_{j=0}^{m} v_{jk} z_j^l\Big) z_j^l$$

با توجه به مشتق های جزئی گرفته شده، نتیجه میگیریم وزن vjk در شبکه عصبی ابتدا بایستی یک مقدار رندم بگیرد و هربار به کمک روش گرادیان مطابق روابط زیر بروز شود تا مقدار بهتری را اخذ کند:

$$\text{SSE, MSE} \quad v_{jk}^{t+1} = v_{jk}^{t} - \eta \sum_{l=1}^{L} (y_k^l - \hat{y}_k^l)\, S'\Big(\sum_{j=0}^{m} v_{jk} z_j^l\Big) z_j^l$$

$$\text{CE} \quad v_{jk}^{t+1} = v_{jk}^{t} - \eta \sum_{l=1}^{L} \Big(\frac{1-\hat{y}_k^l}{1-y_k^l} - \frac{\hat{y}_k^l}{y_k^l}\Big) S'\Big(\sum_{j=0}^{m} v_{jk} z_j^l\Big) z_j^l$$

$$j = 0, \dots, m \qquad k = 1, \dots, p \qquad \text{وزن } (m+1)p$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

فرمول های فوق را میتوان به صورت ماتریسی نوشت تا در کامپیوتر راحت تر انجام گردند. مثال:

$$V^{t+1} = V^{t} - \eta M$$

حال اگر بخواهیم وزن wij که قبل تر است را به کمک گرادیان بهبود دهیم داریم:

$$w_{ij}^{t+1} = w_{ij}^{t} - \eta \frac{\partial E}{\partial w_{ij}}$$

$$\text{SSE} \quad \frac{\partial E}{\partial w_{ij}} = \sum_{l=1}^{L} \sum_{k=1}^{p} \underbrace{(y_k^l - \hat{y}_k^l)}_{\text{خطا}} \frac{\partial y_k^l}{\partial w_{ij}}$$

$$\text{CE} \quad \frac{\partial E}{\partial w_{ij}} = \sum_{l=1}^{L} \sum_{k=1}^{p} \Big(\frac{1-\hat{y}_k^l}{1-y_k^l} - \frac{\hat{y}_k^l}{y_k^l}\Big) \frac{\partial y_k^l}{\partial w_{ij}}$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

توجه شود که هنگام مشتق جزئی گرفتن دیگر سیگمای دوم حذف نشده است. این علت را به دو روش میتوان نشان داد:

- **روش شهودی:** در این روش میتوان گفت w_{ij} در شکل شبکه عصبی رسم شده، روی تمامی y_k ها میتواند اثر بگذارد و لذا سیگمای آن حذف نمیشود. در حالی که v_{jk} تنها روی y_k اثر میگذارد و دیگر سیگما نداریم.
- **روش ریاضی:** هنگامی که نسبت به یک متغیری رند میگیریم، باقی متغیر ها دیگر به چشم متغیر حساب نمی آیند و همگی 0 میشوند و لذا از سیگمای دوم در هنگامی که v_{jk} داریم فقط y_k به شکل متغیر می باشد و بیرون می آید و باقی 0 میشوند. در حالی که در w_{ij} همگی به چشم یک ضریب می باشند و حذف نمی شوند.

حال بایستی قسمتی که با دایره آبی مشخص شده اند را معلوم کنیم. با توجه به کمبود وقت این جلسه وقت نشد محاسبه شود و در جلسه بعد مشخص میکنیم.

جلسه ششم

3:00 PM Tuesday, March 9, 2021

یادآوری و انجام بخش آخر باقی مانده: در جلسه پیش راجع به مشتق گیری صحبت میکردیم. مشتق وزن های لایه خروجی را بدست آوردیم. ولی وزن های ورودی لایه مخفی فرصت به اتمام نرسید.
در محاسبه خطای SSE. با توجه به این که 1/2میتواند در ضریب اتا برود خیلی اهمیتی ندارد. ولی به هر حال داریم:

$$E_{SSE} = \frac{1}{2} \sum_{l=1}^{L} \sum_{k=1}^{P} \left(y_k^l - \hat{y}_k^l \right)^2$$

$$E_{CE} = - \sum_{l=1}^{L} \sum_{k=1}^{P} \left(\hat{y}_k^l \log y_k^l + (1-\hat{y}_k^l) \log (1-y_k^l) \right)$$

نسبت به v مشتق گرفتیم، با توجه به این که داریم:

$$y_k^l = S\left(\sum_{j=0}^{m} v_{jk} z_j^l \right)$$

میتوان گفت:

$$\frac{\partial E_{SSE}}{\partial v_{jk}} = \sum_{l=1}^{L} \boxed{\left(y_k^l - \hat{y}_k^l \right)} \frac{\partial y_k^l}{\partial v_{jk}} \qquad \boxed{\left(y_k^l - \hat{y}_k^l \right)} \to e_k^l$$

$$\frac{\partial E_{CE}}{\partial v_{jk}} = \sum_{l=1}^{L} \boxed{\left(\frac{1-\hat{y}_k^l}{1-y_k^l} - \frac{\hat{y}_k^l}{y_k^l} \right)} \frac{\partial y_k^l}{\partial v_{jk}} \qquad \boxed{\left(\frac{1-\hat{y}_k^l}{1-y_k^l} - \frac{\hat{y}_k^l}{y_k^l} \right)} \to e_k^l$$

با تغییر فوق 2 تاش یکی میشه و متوجه میشویم فرق کراس آنتروپی و SSE تنها در e^L_k می باشد.

$$\frac{\partial E_{SSE}}{\partial v_{jk}} = \sum_{l=1}^{L} \boxed{\left(y_k^l - \hat{y}_k^l \right)} \frac{\partial y_k^l}{\partial v_{jk}} \qquad \boxed{\left(y_k^l - \hat{y}_k^l \right)} \to e_k^l$$

$$\frac{\partial E_{CE}}{\partial v_{jk}} = \sum_{l=1}^{L} \boxed{\left(\frac{1-\hat{y}_k^l}{1-y_k^l} - \frac{\hat{y}_k^l}{y_k^l} \right)} \frac{\partial y_k^l}{\partial v_{jk}} \qquad \boxed{\left(\frac{1-\hat{y}_k^l}{1-y_k^l} - \frac{\hat{y}_k^l}{y_k^l} \right)} \to e_k^l \qquad \frac{\partial y_k^l}{\partial v_{jk}} \to S'(\,)\, z_j^l$$

در نتیجه می توانیم بنویسیم:

$$v_{jk}^{t+1} = v_{jk}^{t} - \eta \sum_{l=1}^{L} e_k^l \, S'\left(\sum_{j=0}^{m} v_{jk} z_j^l\right) z_j^l$$

میتوانیم فرمول فوق رو یکم قشنگ ترش کنیم. به طوری که خطا ضربدر مشتق را دلتای ال و کا میذاریم:

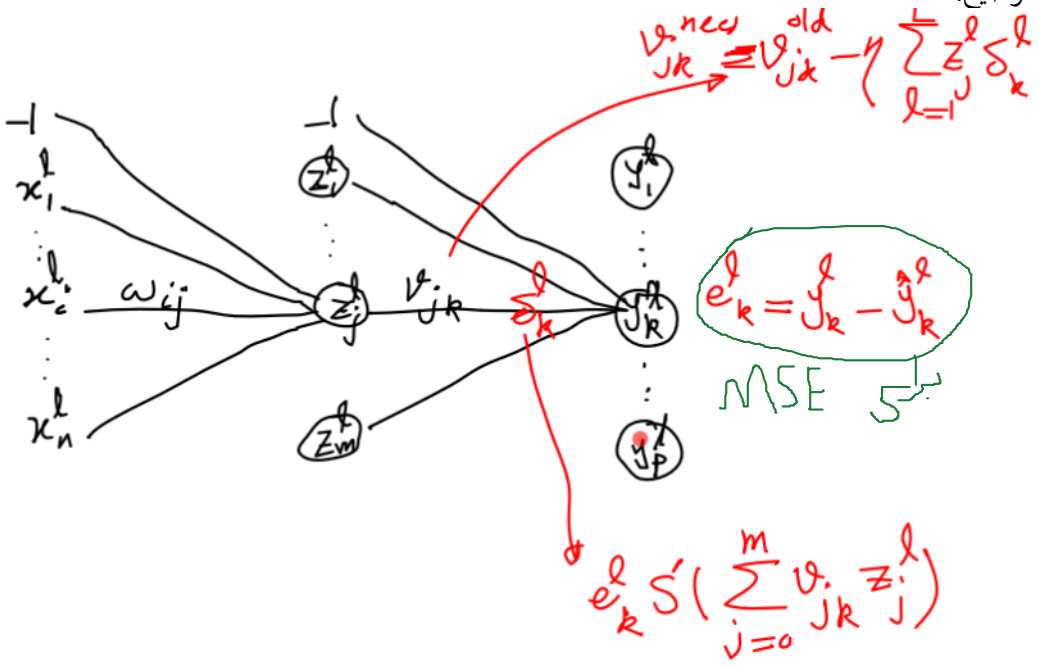
$$v_{jk}^{t+1} = v_{jk}^{t} - \eta \sum_{l=1}^{L} \underbrace{e_k^l \, S'\left(\sum_{j=0}^{m} v_{jk} z_j^l\right)}_{\delta_k^l} z_j^l$$

$S'(u) = S(u)(1-S(u))$

و داریم:

$$v_{jk}^{t+1} = v_{jk}^{t} - \eta \sum_{l=1}^{L} \delta_k^l z_j^l$$

توضیح:



یعنی چی؟ یعنی الان ما e_k^l رو داریم که خطای تابع سیگموید در خروجیه. این خطا خطای بعد از اعمال تابع سیگموید هستش. اگه بخوایم خطای قبلش رو بدست بیاریم باید چیکار کنیم؟ باید اون خطا رو ضرب کنیم توی مشتق تابع سیگموید تا δ_k^l حاصل بشه.

حالا برگردیم به ادامه جلسه قبل که ناقص موند. خواستیم مشتق y_k^l نسبت به w_{ij} رو حساب کنیم. به کمک chain rule داریم:

$$\frac{\partial E}{\partial w_{ij}} = \sum_{l=1}^{L} \sum_{k=1}^{P} e_k^l \frac{\partial y_k^l}{\partial w_{ij}} \rightarrow S'\left(\sum_{j=0}^{m} v_{jk} z_j^l\right) v_{jk} \frac{\partial z_j^l}{\partial w_{ij}}$$

$$y_k^l = S\Big(\overbrace{\sum_{j=0}^{m} v_{jk} z_j^l}^{u}\Big)$$

$$z_j^l = S\Big(\underbrace{\sum_{i=0}^{n} w_{ij} x_i^l}_{u}\Big)$$

$$\frac{\partial z_j^l}{\partial w_{ij}} = \frac{\partial z_j^l}{\partial u} \times \frac{\partial u}{\partial w_{ij}}$$

$$S'\left(\sum_{i=0}^{n} w_{ij} x_i^l\right) \qquad x_i^l$$

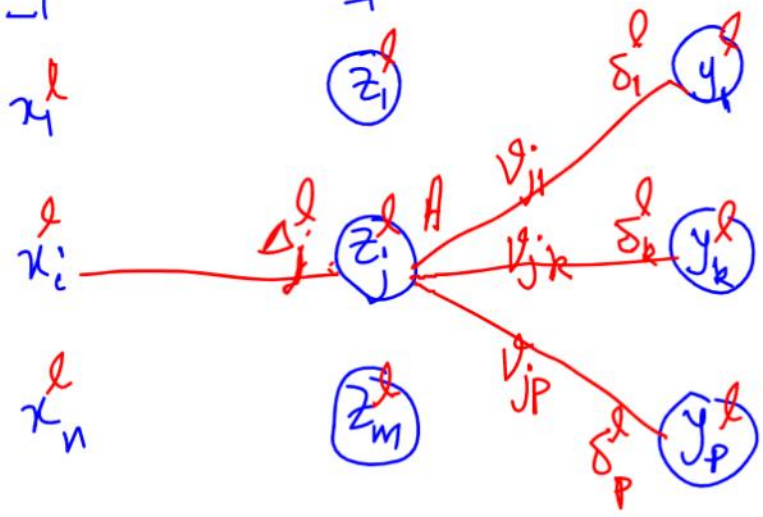
پس بخوایم w_{ij} جدید رو حساب کنیم. داریم:

$$w_{ij}^{t+1} = w_{ij}^{t} - \eta \sum_{l=1}^{L} \underbrace{\sum_{k=1}^{P} \delta_k^l v_{jk} S'\left(\sum_{i=0}^{n} w_{ij} x_i^l\right)}_{\Delta_j^l} x_i^l$$

$$\boxed{w_{ij}^{t+1} = w_{ij}^{t} - \eta \sum_{l=1}^{L} \Delta_j^l x_i^l}$$

در بیان شهودی داریم:

$$w_{ij}^{new} = w_{ij}^{old} - \eta \sum_{l}^{L} x_i^l \Delta_j^l$$



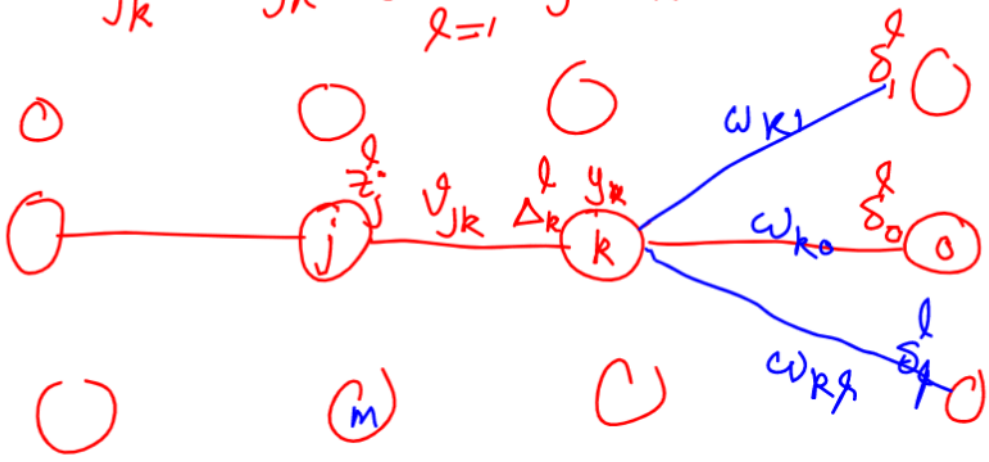
$$A = \sum_{k=1}^{p} v_{jk} \delta_k^l$$

$$\Delta_i^l = A\, \delta'\left(\sum_{i=0}^{n} w_{ij} x_i^l\right)$$

با توجه به این که خطا را از لایه آخر به عقب میبریم، به این روش Error Backpropagation می گویند.

اگر همین مفهوم را بسط بدهیم، در صورتی که حتی 1000 تا لایه هم داشته باشیم، میتوانیم خطای لایه قبل با وجود لایه بعدی را به روش زیر بدست آوریم:

$$v_{jk}^{new} = v_{jk}^{old} - \eta \sum_{l=1}^{L} z_j^l \Delta_k^l$$



$$\Delta_k^l = \delta'\left(\sum_{j=0}^{m} v_{jk} z_j^l\right) \sum_{o=1}^{q} w_{ko} \delta_o^l$$

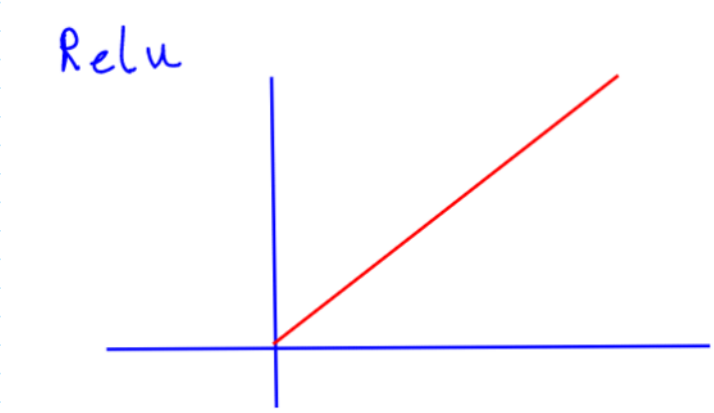
پس ما همواره خطای لایه آخر را بدست می آوریم. همینطوری میریم عقب میرسیم به اولی که میشه یک Epoch. بعد این کار را به تعداد زیادی دفعه انجام میدیم تا اوکی شه.

* هرچند که از لحاظ ریاضی اثبات شده است که با 3 لایه میتوان هر ابرمنحنی مورد نظر را تولید کرد. اما این کار نیاز است به صورت عمودی تعداد زیادی پرسپترون داشته باشیم. امروزه به این نتیجه رسیده اند هرچه لایه ها بیشتر باشند (به صورت افقی بیشتر داشته باشیم) میتوانیم تابع های پیچیده ای را به راحتی مدل کنیم چرا که وزن های زیاد

تری داریم. البته یک مشکلی هم که داریم اینه که وقتی تعداد لایه ها زیاد میشه برای overfit نشدن نیاز داریم داده های زیادی (حدود میلیون تا) را داشته باشیم.

* برای شبکه های عمیق وقتی لایه زیاد داریم تابع فعالیت sigmoid مناسب نیست. زیرا خطای لایه قبل که بر حسب بعدی میخواد بدست بیاید همینطوری روی مشتق سیگموئید ضرب میشود. از طرفی مشتق سیگموئید یک عددی بین 0 و 1 میشود و وقتی روی هم ضرب میشن به 0 میل میکند و عملا گویی یادگیری نخواهیم داشت.

راه حل برای شبکه های عمیق: استفاده از تابع ReLU (Rectified linear unit): تابعی که در x>0 به صورت y=x و در x<0 به صورت y=0 می باشد:

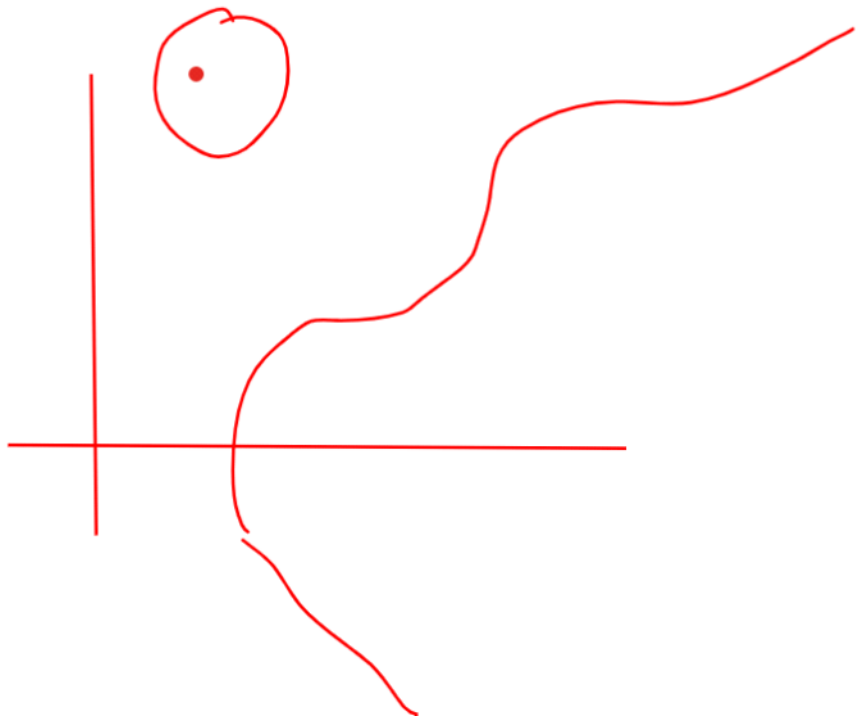


با این کار مشتق در x>0 برابر با 1 می باشد و مشکل سیگموئید را نخواهد داشت. البته با توجه به این که قسمت منفی y=0 می باشد تابع ReLU نیز بعضی جاها داستان میشه و یه طوری تغییر میدن که در قسمت منفی هم یک مقداری داشته باشد.

چرا سراغ تابع فعالیت رفتیم!؟ مهم ترین وظیفش این بود که غیر خطی میکنه. اگر تابع فعالیت خطی باشه 100 تا شبکه هم داشته باشیم معادل با 1 پرسپترون میشه. چون خطی هستند و گویی انگار شبکه نداریم.

برای توجیه دیگر مثلا میتوان گفت تابع ReLU یک قسمت را کلا 0 میکند و برای classification خوبه که تقسیم بندی صورت بگیره.

* شبکه عمیق بسته به نوع تابع فعالیت میتواند هر شکلی را تولید کند:



سوال: Overfitting چه می باشد؟

ما در شبکه عصبی 2 مساله داریم.:

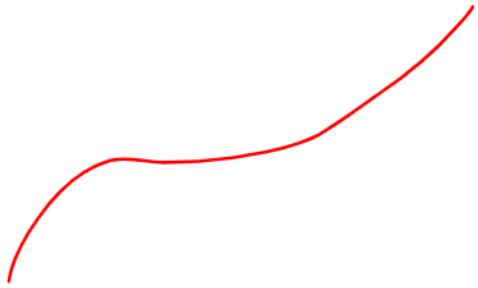
- یکی این که آیا شبکه عصبی ما قابلیت و پیچیدگی لازم برای تولید تابع غیر خطی ما رو در حالت رگرسیون داره یا

نه. ←بایاس. شبکه هرچه لایه های بیشتری داشته باشد و وزن های بیشتری داشته باشد میتواند شکل های پیچیده تری را تولید کند. پس بایستی تصمیم بگیریم تابع مورد نظر ما برای مساله ما تا چه حد پیچیده باشد

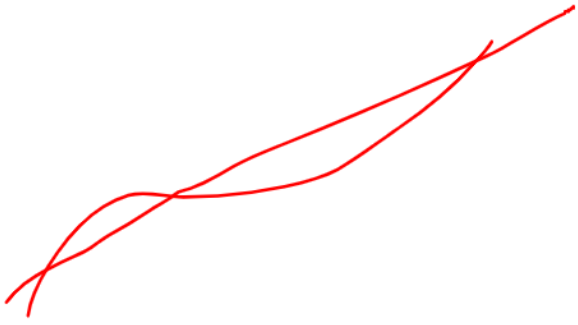
- مشکل دیگه اینه که اگر خیلی پیچیده باشه، اگر داده های آموزشی کم باشه فقط برای اونا خوب جواب میده. (Overfitting) یعنی چون یه چیز پیچیده داریم و پارامتراش زیاده برای اون داده های آموزشی کم یه طوری تنظیم میشه که خطای اونارو صفر میکنه و برای داده هایی که نداریم خطا خیلی زیاد میشه. ← واریانس.

در نتیجه ما اگر تعداد داده هامون زیاد باشه میتوانیم از شبکه های عمیق استفاده کنیم و مشکل overfit نداشته باشیم. ولی اگر تعداد داده ها به اندازه کافی زیاد نباشه اون موقع بایستی سطح پیچیدگی به میزانی کم بشه که overfit نشه.

مثال: فرض شود میخواهیم همچین شکلی را مدل کنیم:

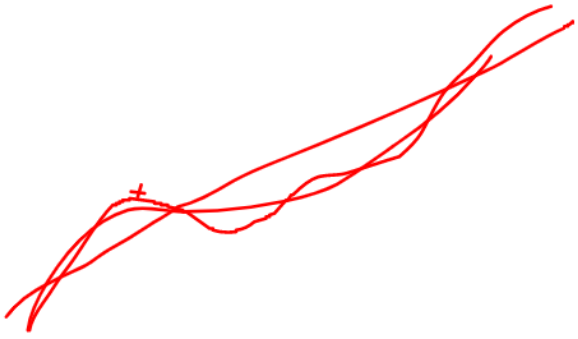


اگر با یک پرسپترون بخواهیم مدل کنیم یک خط میشود:



در این صورت این خط برای داده هایی که در داده های آموزشی مشاهده نشده خطای زیادی نخواهد داشت.

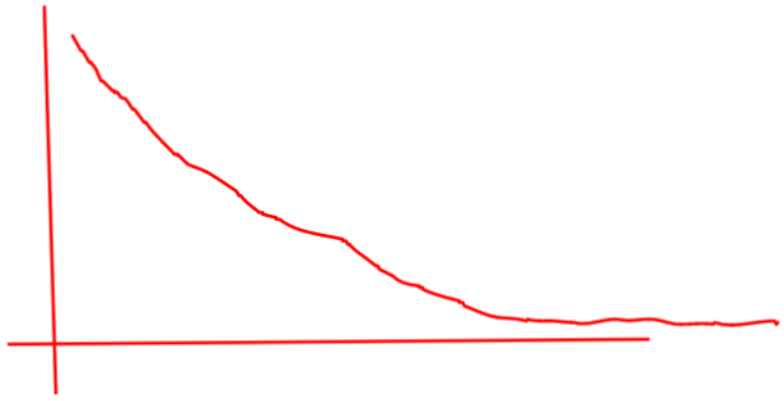
ولی اگر از یک شبکه پیچیده تری استفاده کنیم که داده های کم و حتی noise دارد، آنگاه به طوری روی آن ها fit میشود که با داده های مورد نظر مشاهده نشده فاصله میگیرد و خطا زیاد می شود:



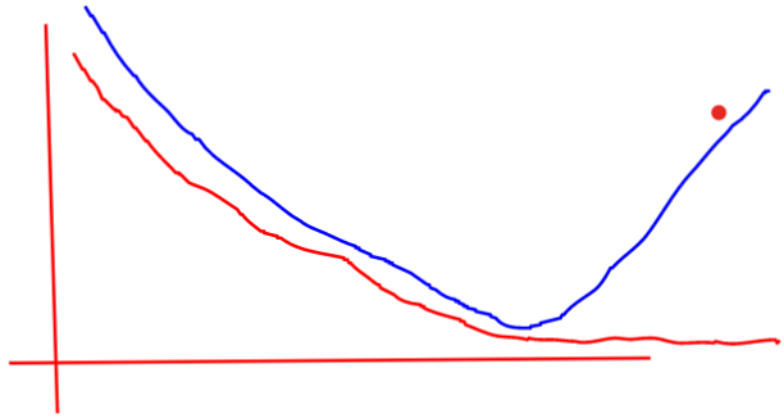
پس اگر داده کافی نداریم نمیتوانیم از شبکه های پیچیده استفاده کنیم.

اگر داده ها به اندازه کافی نباشد و بخوایم شبکه عمیق استفاده کنیم چیکار کنیم؟ داده ها رو به 3 دسته تقسیم میکنیم. دسته اول برای آموزش شبکه. دسته دوم برای تعیین کردن این که چند Epoch انجام بگیره خوبه (Validation). هدفمون در دسته دوم اینه که خیلی به سمت مینیمم نزدیک نشیم تا overfit صورت بگیره.

در واقع اگر ما یه منحنی داشته باشیم که محور x تعداد epoch ها و محور عمودی خطا باشد. داریم:



مشاهده میکنیم هرچه epoch بیشتر میشود خطا رفته به رفته کم می شود. ولی وقتی داده های validation رو میاریم تو کار، میبینم وقتی epoch رو از یه جا به بعد بیشتر کنیم خطا نه تنها کمتر نمیشه بلکه بیشترم میشه:



از نقطه ای که دیگر validation کم نمیشه و بیشتر میشه نقطه ایست که داره overfitting صورت میگیره و باید تعداد epoch هارا دست نگه داریم

و در نهایت از داده های سوم تحت عنوان تست استفاده میشود تا خطای کار اندازه گیری شود.

*ما از روی داده های validation میتوانیم مقادیر دیگری نظیر تعداد نورون ها، لایه ها و ... (به طور کلی Hyperparameters) را نیز تنظیم کنیم. به نحوی که Overfitting صورت نگیرد.

جلسه هفتم

3:01 PM Sunday, March 14, 2021

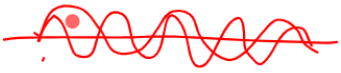
توابع پایه

توابعی که بشه بقیه توابع رو بر اساس آنها نوشت.

$$f(x) = \sum_{i=1}^{\infty} w_i \, \underbrace{b_i(x)}_{\text{توابع پایه}}$$

به عنوان مثال تبدیل فوریه، لاپلاس، تبدیل سینوس، کسینوس و .. را میتوان توابع پایه تلقی کرد.
به عنوان مثال در تبدیل کسینوس داریم:

$$f(x) = \sum_{i=1}^{\infty} w_i \cos(2\pi w_i)$$



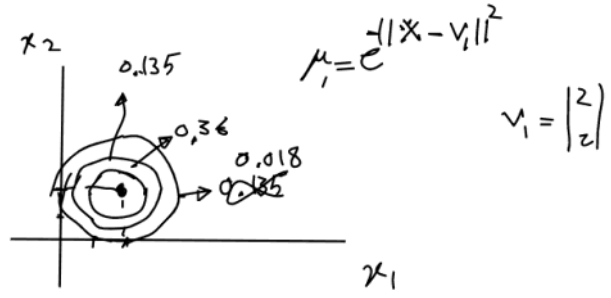
معنیش میشه این که داریم یک تابع رو به صورت ترکیب کسینوس هایی با فرکانس های مختلف در میاریم. (که حالا استاد مثلا 2 تاشو کشید)

در حالت کلی مثلا در تبدیل لاپلاس تابع نمایی مختلط می باشد. در تبدیل فوریه هم مشاهده کردیم نمایی ساده است.
چرا به اینا تبدیل فوریه یا لاپلاس میگن؟ فضای فرکانس میشه اون w_iها. یعنی با توجه به این w_i ها اگر w یک کسینوس با فرکانس بالا بزرگ باشه میگیم تابع در این ضریب دارای فرکانس های بالاست. وقتی میریم به فضای فرکانس فقط به w_i ها نگاه میکنیم. اگه تابع تغییرات زیادی داشته باشه فرکانس های بیشتری داره اگر تغییرات کمی داشته باشه فرکانس کم بیشتری داره و از طریق w_i ها متوجه میشویم تابع تغییراتش چقدر است.
با توجه به این که تابع تبدیل معکوس پذیر هم هستش بعد از w ها بر میگردیم به x ها. اگر بخواهیم مثلا فرکانس های بالا رو حذف کنیم w هایی که برای فرکانس های بالا هست رو صفر میکنیم و بعد f(x) رو دوباره بازسازی میکنیم و تغییرات ناگهانی تابع در فرکانس های بالا حذف خواهند شد. یا مثلا میشه تابع رو فقط با w ها توصیف کرد که جنبه فشرده سازی دارد.

از نظر ریاضی اثبات می شود که هر تابعی را میتوان به صورت زیر نوشت (X به جاx نوشتیم چون ممکنه ورودی ابعاد زیادی داشته باشه):

$$f(x) = \sum_{i=1}^{\infty} w_i \, \underbrace{e^{-\|X - V_i\|^2}}_{\text{توابع پایه شعاعی}}$$

چرا بهش میگیم توابع پایه شعاعی؟ برای مثال فرض شود که $v_1 = (2, 2)$ می باشد. اگر کانتور هاش رو رسم کنیم:

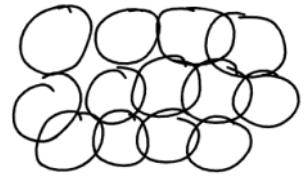


دایره ای به مرکز V_1 و شعاع R_1 $\rightarrow$ $\|X - V_1\|^2 = R_1^2$

در واقع کانتور ها در توابع شعاعی میشوند دایره هایی که مقدار تابع پایه ای در محیط این دایره ها با هم برابر هستش.

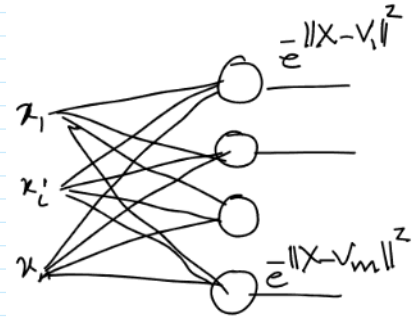
در نقطه V_1 مقدار تابع μ برابر با 1 می باشد. سپس اگر توان e را 1 کنیم . (دایره به شعاع 1) مقدار μ برابر با 0.36 خواهد شد. یعنی داخل دایره به شعاع یک مقدار μ از 1 تا 0.36 تغییر

خواهد کرد. در صورتی که شعاع دایره 2 باشه تا 0.018 کم میشه.
میتوان گفت توابع شعاعی تا شعاع رادیکال 2 که به مقدار 0.135 میرسد را پوشش میدهد و پس از آن مقدار ناچیزی خواهد داشت.
توابع شعاعی فضای مارا به صورت دایره هایی تقسیم بندی میکنند، بسته به این که مراکز به چه صورت می باشند ممکن است این توابع Overlap داشته باشند یا نداشته باشند

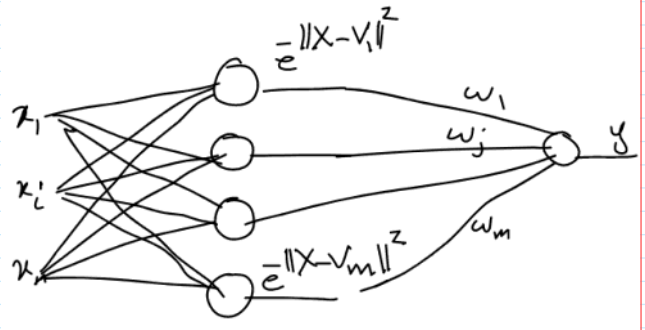


|شبکه عصبی RBF (Radial Basis Function) - توابع پایه شعاعی

برای ورودی ها وزن گرفته نمیشه و خروجی به صورت یک تابع پایه شعاعی خواهد شد:



سپس در یک وزن ضرب میشوند و خروجی را تولید میکنند:

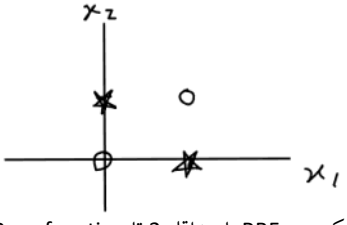


در نتیجه اگر بخواهیم فرمول y را بنویسیم:

$$y = \sum_{j=1}^{m} w_j e^{-\|X-V_j\|^2}$$

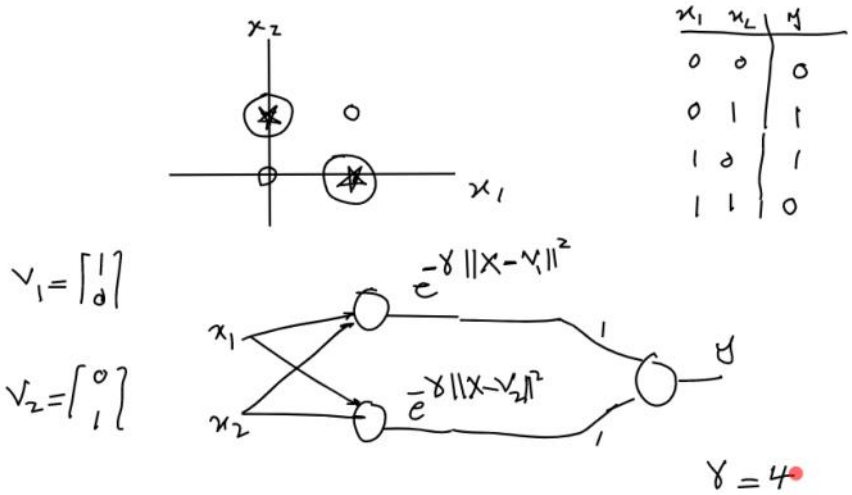
فرقش با توابع پایه اینه که سیگما تا بینهایت نیست و تا یه ثابت m هستش.
* در شبکه RBF یک لایه بیشتر نداریم. هرچند بعد ها روش هایی اومده به صورت سلسله مراتبی میشه چندتا لایه درست کرد.

برای توجیه بهتر توابع شعاعی، میتوان مساله XOR را در نظر گرفت. اگر در نمودار صفر هارو با دایره و 1 هارو با ستاره نمایش دهیم داریم:

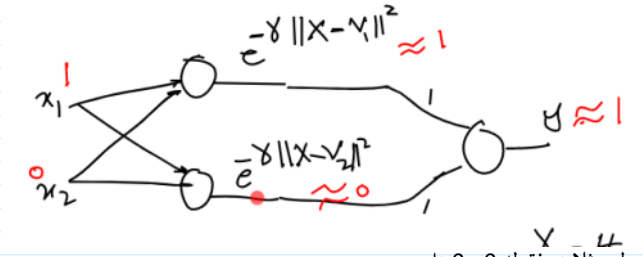


x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

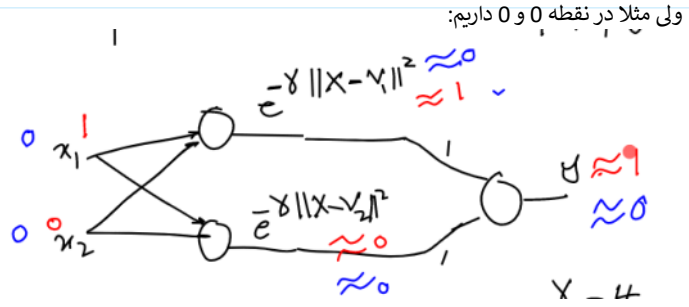
با یک پرسپترون نمیتوان این مساله را مدل کرد. در RBF با حداقل 2 تا Base function میتوان مدل کرد:



با این مدل در نواحی که ستاره داریم خروجی ما 1 و غیر آن صفر است:

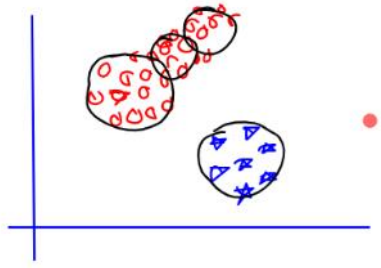


ولی مثلا در نقطه 0 و 0 داریم:



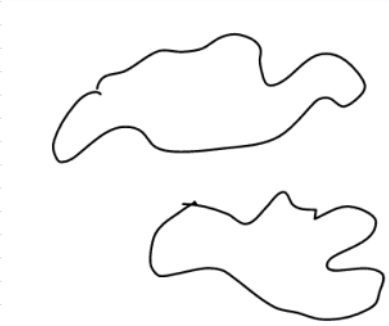
* گاما رو هرچی بیشتر بگیریم فضایی که دایره اشغال میکند کوچکتر می شود و بالعکس با کاهش گاما شعاع دایره بزرگتر میشه

در مثال پرتغال و سیب نیز اگر بخواهیم از RBF کمک بگیریم، base function ها اگر دایره های مشکی باشند داریم:

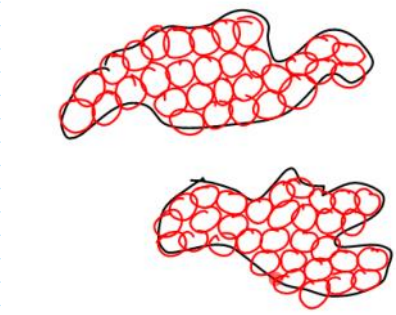


هرچه این دایره های مشکی کوچکتر باشند شکل های دقیق تری را میتوان توصیف کرد.

برای مثال اگر داشته باشیم:



برای این که بتوانیم فضا رو پوشش دهیم تعداد زیادی base function نیاز داریم با شعاع ثابت:



هرچه این دایره ها شعاع بیشتری داشته باشند دقت کمتری داریم و کمتر باشند دقت بیشتری داریم.

حالا باید تصمیم بگیریم چطوری این مراکز رو یاد بگیریم.

خطای SSE به روش زیر محاسبه می شود:

$$E_{SSE} = \frac{1}{2}(Y-\hat{Y})^T(Y-\hat{Y})$$

$$\begin{array}{c} x_1^1 \cdots x_n^1 \\ \vdots \\ x_1^l \cdots x_n^l \\ \vdots \\ x_1^L \cdots x_n^L \end{array} \begin{bmatrix} \hat{y}^1 \\ \\ \hat{y}^l \\ \\ \hat{y}^L \end{bmatrix} \rightarrow \hat{Y}$$

$$Y = \begin{bmatrix} y^1 = \sum_{j=1}^{m} w_j e^{-\gamma \|X^1 - V_j\|^2} \\ \\ y^L = \sum_{j=1}^{m} w_j \cdot e^{-\gamma \|X^L - V_j\|^2} \end{bmatrix} \qquad X^l = \begin{bmatrix} x_1^l \\ \vdots \\ x_n^l \end{bmatrix}$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

$$y^l = \sum_{j=1}^{m} w_j \mu_j^l \qquad \mu_j^l = e^{-\gamma \|x^l - v_j\|^2}$$

$$W = \begin{bmatrix} w_1 \\ w_j \\ w_m \end{bmatrix}$$

$$y^l =$$

$$Y = AW \qquad A = \begin{bmatrix} \mu_1^1 & \mu_m^1 \\ \mu_1^L & \mu_m^L \end{bmatrix}$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

$$Y = AW \qquad \tilde{y} = \begin{bmatrix} \tilde{y}^1 \\ \tilde{y}^l \\ \tilde{y}^L \end{bmatrix}$$

$$\begin{bmatrix} y^1 \\ y^l \\ y^L \end{bmatrix} = \begin{bmatrix} \mu_1^1 \cdots \mu_j^1 \cdots \mu_m^1 \\ \mu_1^l \cdots \mu_j^l \cdots \mu_m^l \\ \mu_1^L \cdots \mu_j^L \quad \mu_m^L \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_j \\ \vdots \\ w_m \end{bmatrix} \qquad \mu_j^l = e^{-\gamma \|x^l - v_j\|^2}$$

$$\frac{dE_{SSE}}{dW} = \frac{d}{dW}\left[(AW - \tilde{y})^T (AW - \tilde{y})\right]$$

$$A^T(AW - \tilde{y}) = 0 \rightarrow A^T A W = A^T \tilde{y}$$

$$W = (A^T A)^{-1} A^T \tilde{y}$$

برای استفاده از کراس آنتروپی، روی خروجی بایستی تابع سیگموید بذاریم که بیاد روی 0 و 1. چون توی حالت کلی خروجی y میتواند هر عددی باشد.

$$E_{CE} = -\sum_{l=1}^{L} \left(\tilde{p}^l \log p^l + (1 - \tilde{p}^l) \log(1 - p^l)\right)$$

$$p^l = S(y^l) = S\left(\sum_{j=1}^{m} w_j \mu_j^l\right)$$

به عنوان تمرین سعی کنید به صورت برداری بنویسید و مشتق برداری بگیرید

اما در کلاس با همون صورت ک نوشتیم پیش بریم داریم:

$$E_{CE} = -\sum_{l=1}^{L}\left(\bar{p}^{l}\log p^{l} + (1-\bar{p}^{l})\log(1-p^{l})\right)$$

$$p^{l} = S(y^{l}) = S\Big(\overbrace{\sum_{j=1}^{m} w_j \mu_j^{l}}^{y^{l}}\Big)$$

$$\frac{\partial E_{CE}}{\partial w_j} = \sum_{l=1}^{L}\left(\frac{1-\bar{p}^{l}}{1-p^{l}} - \frac{\bar{p}^{l}}{p^{l}}\right)\frac{\partial p^{l}}{\partial w_j} \longrightarrow S'(y^{l})\frac{\partial y^{l}}{\partial w_j}$$

$$\mu_j^{l}$$

$$w_j^{new} = w_j^{old} - \eta\sum_{l=1}^{L}\left(\frac{1-\bar{p}^{l}}{1-p^{l}} - \frac{\bar{p}^{l}}{p^{l}}\right) S'(y^{l})\,\mu_j^{l}$$

$$w_j^{t+1} = w_j^{t} - \eta\sum_{l=1}^{L}\left(\frac{1-\bar{p}^{l}}{1-p^{l}} - \frac{\bar{p}^{l}}{p^{l}}\right) S'(y^{l})\,\mu_j^{l}$$

* دقت شود با استفاده از سیگموید RBF بودن از بین نمیرود. بلکه انگار آخر اون شکل RBF که y داریم، اون y بره توی سیگموید و P رو درست کنه.

از روی همین فرمول هم میشه رابطه ماتریسی اش را نوشت. به طوری که :

$$W^{t+1} = W^{t} - \eta\left\{\left[(1-\bar{Q})./(1-Q) - \bar{Q}./Q\right].\star S'(y)\right\} A$$

$$\bar{Q} = \begin{bmatrix}\bar{p}^{1}\\ \bar{p}^{l}\\ \bar{p}^{L}\end{bmatrix} \qquad Q = \begin{bmatrix}p^{1}\\ p^{l}\\ p^{L}\end{bmatrix} \qquad \bar{Q}./Q = \begin{bmatrix}\bar{p}^{1}/p^{1}\\ \vdots\\ \bar{p}^{l}/p^{l}\\ \vdots\\ \bar{p}^{L}/p^{L}\end{bmatrix}$$

صفحه خالی تخته وایت برد جهت نوشتن و ارائه مطلب

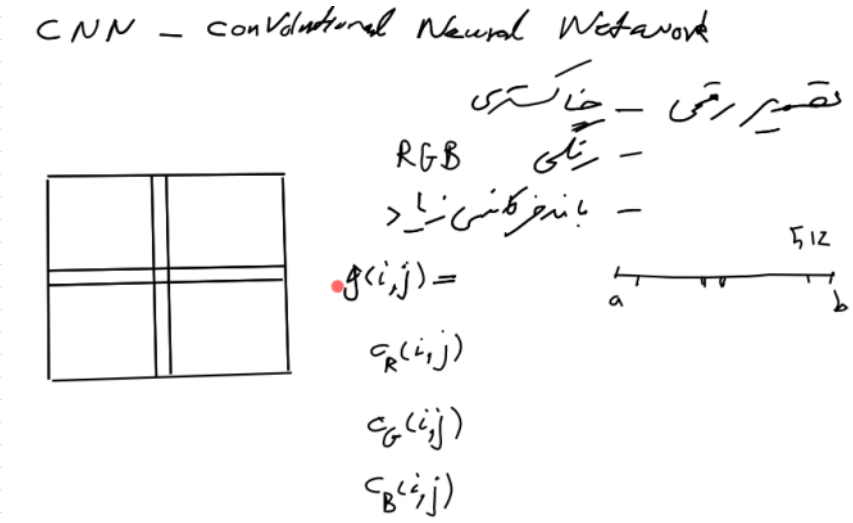
* برای یادگیری مراکز توابع شعاعی یکی از روش ها خوشه بندی هستش که بعدا در این مورد صحبت خواهد شد. یا این که میتوان به صورت تصادفی انتخاب میشوند که دقت کمتری نسبت به خوشه بندی دارد. گرادیان به خوبی برای یادگیری مراکز جواب نمیدهد.

جلسه هشتم

3:09 PM Tuesday, March 16, 2021

هر تصویر یک ماتریس 2 بعدی است که تعداد سطر و ستون ها رزولوشن را تعیین میکنند. به هر نقطه از ماتریس را پیکسل میگویند.
تصاویر به 4 نوع تقسیم می شوند:

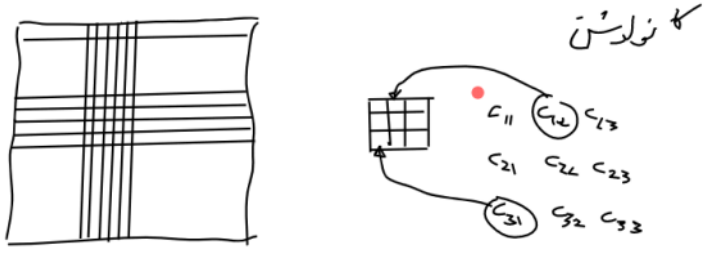
- **خاکستری (Grayscale):** هر پیکسل یک عدد بین 0 تا 255. به طوری که هرچه به 0 نزدیک باشد مشکی تر و هرچه به 255 نزدیک تر باشد سفید تر است. با $g(i,j)$ میتوانیم میزان خاکستری بودن پیکسل در سطر i و ستون j را مشخص کنیم.
- **باینری:** تنها با صفر و یک. به طوری که صفر مشکی و 1 سفید
- **رنگی:** یک نوعش RGB می باشد. که از سه کانال قرمز، سبز و آبی تشکیل شده است و هرکدام در یک ماتریس نمایان میشوند (سه ماتریس داریم). که با $c_R(i, j)$، $c_G(i, j)$ و $c_B(i, j)$ نمایش میدهیم.
- **باند فرکانسی زیاد:** تصاویر هم با دوربین های مخصوص گرفته میشود (مثل تصاویر ماهواره ای) که علاوه بر RGB، فرکانس های قابل دید را بسته به این که تصویر ماهواره ای چه میزان باند دارد، که مثلا 64 باند، 256 باند و یا 512 باند دارند می آیند مثلا اگر از a تا b را اشغال کند و به 512 بخش تقسیم می کنند، 512 تا ماتریس حاصل می شود که هریک شدت نور متفاوتی دارند. به کمک پردازش های تصویر ماهواره ای مثلا مشخص میشود خاک از چه نوعی است، سنگ از چه نوعی است و یا سطح زیر کشت چه چیزی است و ...



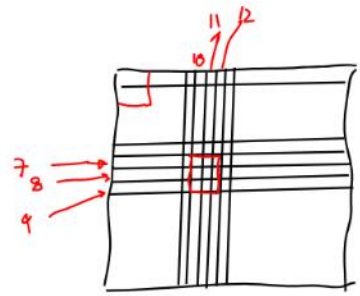
معمولا در Remote Sensing از تصاویر ماهواره ای استفاده میشود. در کاربرد های معمولی ما یا RGB ای هستیم یا grayscale. در RGB با توجه به این که به صورت (0-255, 0-255, 0-255) می باشد ما 256^3 رنگ را میتوانیم توصیف کنیم.

| کانولوشن

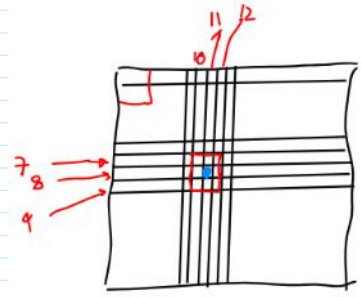
یک عملگر که بسته به این که عمگراش چی باشه میتونه عکس رو تار کنه یا لبه هارو نمایان تر کنه و ...
فرض کنید که یک تصویر 32×32 پیکسل داریم (مربع سمت چپ در تصویر پایین)
در کانولوشن یه ماتریس کوچکتر (به اندازه سایز پنجره کانولوشن که معمولا 3×3 یا 5×5 یا 7×7 در نظر میگیرند. معمولا فرد است) در یک ماتریس بزرگتر حرکت میکند.
فرض شود که یک ماتریس 3×3 داریم که مقادیر عملگر ما (ضرایب کانولوشن) را مشخص میکند.



به این ماتریس کوچکتره پنجره کانولوشنی هم میگیم.
در این پنجره کانولوشنی ما یک سری وزن وجود دارد که مشخص میکند ضرایب کانولوشن ما چه چیزی می باشد.
این ماتریس کوچک در تصویر اصلی ما در پیکسل های 1 تا3 در سطر و ستون قرار میگیرد و یدونه یدونه راست حرکت میکند تا به انتها برسد. سپس میاد پایینتر و یکدونه یکدونه میره چپ تا برسه آخر. (یعنی از راست به چپ جارو میکند میره پایین بعد به چپ چارو میکند). وقتی این ماتریس کوچیک رو روی تصویر اصلی حرکت میدیم یه تصویر جدید حاصل میشه. به صورتی که ضرایب کانولوشنی با پیکسل های تصویری ضرب میشه بعد با هم جمع میشن و پیکسل جدید وسطی رو میدن.
برای درک بهتر مثلا فرض شود پنجره کانولوشنی برسه به وسط:



اگر تصویر اولی رو با g نشون بدیم. و تصویر که با اثر کانولوشن بدست میاد 'g نشون بدیم. پیکسل وسطی یعنی (8,11) تغییری که میکنه برابر است با:



$$g'(8,11) = c_{11}\, g(7,10) + c_{12}\, g(7,11) + c_{13}\, g(7,12) + c_{21}\, g(8,10) + c_{22}\, g(8,11) + c_{23}\, g(8,12) + c_{31}\, g(9,10) + c_{32}\, g(9,11) + c_{33}\, g(9,12)$$

سوال : اینطوری اکه که پیکسل های سطر و ستون اول و آخر که بدون تغییر باقی میمونه ← برای راه حل یک سطر اول و آخر، یک ستون اول و آخر به تصویر اضافه میشود. از لحاظ مقادیر هم سطر اول جدید مقدار سطر اول قدیم را میگیرد. سطر آخر جدید مقدار سطر آخر قدیم را میگیرد و ...

در شبکه عصبی یک تابع فعالیت هم روش اعمال میشود (مثل ReLU) تا اگر از یه حدی کمتر باشد صفر بگیریم:

$$g'(8,11) = A\big(c_{11}\, g(7,10) + c_{12}\, g(7,11) + c_{13}\, g(7,12) + c_{21}\, g(8,10) + c_{22}\, g(8,11) + c_{23}\, g(8,12) + c_{31}\, g(9,10) + c_{32}\, g(9,11) + c_{33}\, g(9,12)\big)$$

صفحه خالی. تخته وایت برد جهت نوشتن و ارائه مطلب

حال این کانولوشن چه تغییری میتونه بده؟ برای مثال فرض شود که پنجره کانولوشنی برابر است با:

0	0	0
-1	0	1
0	0	0

در این حالت مشتق افقی میگیرد. به صورتی که پیکسل راست هرپیکسل را منهای چپی میکند ودر خود میگذارد. با این کار مثلا شکل چپی به راستی تبدیل میشه:

یا مثلا اگر پنجره کانولوشنی برابر با زیر باشد: (هرچند به نظرم آخرین درایه هم باید صفر باشه)

0	-1	0
0	0	0
0	1	1

اون موقع خط های عمودی حذف میشه و شکل اینطوری میشه:

یا مثلا کانولوشن زیر:

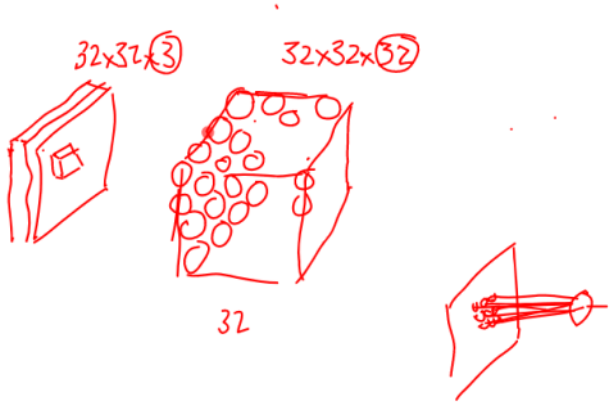
0	0	10
0	10	0
10	0	0

خط های مورب رو پر رنگ تر میکند.

به پنجره کانولوشنی Filter هم میگن.

| CNN:

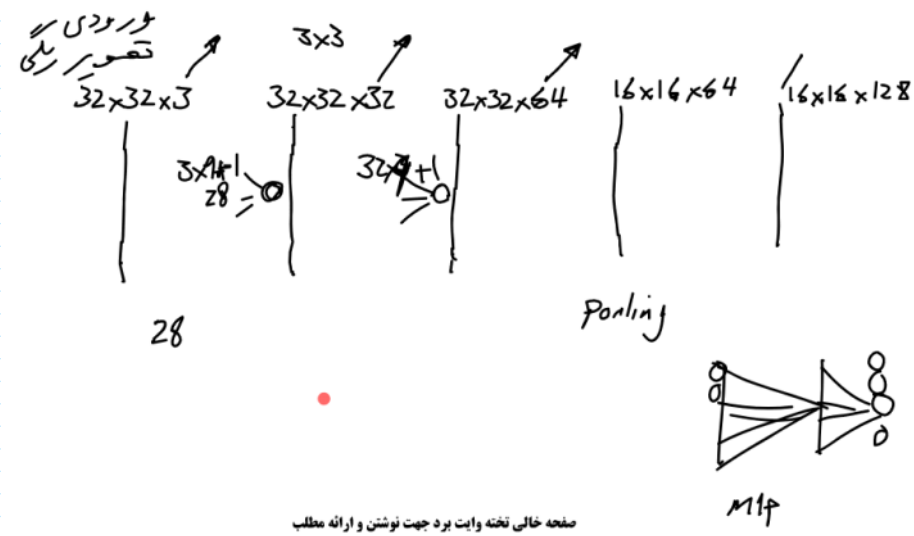
ورودی یک تصویر است. اگر تصویر را رنگی در نظر بگیریم که 32×32 می باشد. با توجه به این که 3 کانال داریم به صورت 3×32×32 می شود.



در تصویر بالا پایین سمت راست یک فیلتر را نورون را میبینیم که ورودی اش 9 تا پیکسل می باشد. این حالت graysacle می باشد. در صورتی که RGB باشد ورودی یک مکعب می باشد. 9 تا برای Red. 9 تا برای Green و 9 تا برای Blue و لذا 27 تا ورودی میگیرد.

هر نورون در CNN معادل یک فیلتر می باشد. در مثال فوق اگر ما 32 تا فیتلر (32 تا پنجره کانولوشنی با ضرایب کانولوشنی متفاوت) داشته باشیم. آنگاه هر کدام از این فیلتر ها بایستی یک عکس 3×32×32 را اختیار کند. پس طبق توضیحات پاراگراف فوق بایستی هر فیلتر 27 تا ورودی بگیرد و یک عکس کاملا جدید با اعمال کانولوشن تولید کند. از طرفی با توجه به این که 32 تا فیلتر داریم پس بایستی 32 تا عکس تولید کند و لذا 32×32×32 بُعد لایه پنهان می شود.

پس متوجه شدیم هر فیلتر روی یک پیکسل اعمال میشه و یک تغییراتی رو اعمال میکنه. 32×32 تا پیکسل داشتیم و 32 تا فیلتر که نتیجش یک 32×32×32 شد. که هر فیلتر بسته به ضرایبی که داخلش داره یک feature خاصی رو توجه میکنه. مثلا یک فیلتر به خط های عمودی توجه داره. یکی افقی. یکی اُریب و ...



چرا ورودی لایه اول 28 تا شد؟ چون پنجره کانولوشنی 3×3 بود و 9 تا شد. 3 تا کانال RGB هم داشتیم که شد 27 تا. یدونه هم bias شد 28 تا

* معمولا در CNN های استاندارد اینطوریه که لایه اول 32 باشه بعد 64 بعد 128 و ...

* در لایه سوم هم ورودی ها همونطور که تو شکل مشخص شده 9*32 برای پنجره کانولوشنی است و + 1 برای بایاس

همینطور که جلو میریم عمق زیاد میشه. برای این که از لحاظ خروجی به مشکلی بر نخوریم کاری که میکنن اینه که بعد یکی 2بار انجام فیلتر میان طول و عرض رو کاهش میدهند. به این کار Pooling گفته میشود.

ما 2 نوع Pooling مشهور داریم: متوسط گیری و ماکسیمم گیری:

- **متوسط گیری:** 4 تا پیکسل رو متوسط میگیریم میکنیم یک پیکسل
- **ماکسیمم گیری:** ماکسیمم 4تا پیکسل رو تحت یک پیکسل در نظر میگیریم.

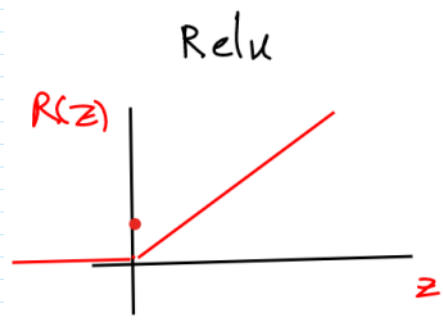
پس از Pooling در شکل میبینیم یک فیلتر دیگر اعمال شده است. پس از انجام تعدادی فیلتر، در انتها 2 لایه Fully Connected می گذارند. یعنی همه خروجی ها به همه نورون ها وصل می شوند. یعنی 2 لایه آخر مشابه MLP می شود. در لایه آخر مثلا اگر مساله Classficiation باشد به تعداد کلاس ها خروجی خواهیم داشت.

حالا ما وزن های فیلتر ها و وزن های 2 لایه آخر رو چطوری بدست بیاریم؟ ← **گرادیان تصادفی**

$$w^{new} = w^{old} - \eta \sum_{l=1}^{L} x \delta$$

مشابه قبل باید به کمک backpropagation خطا رو انتقال بدیم به عقب و محاسبه صورت گیرد تا وزن ها اصلاح گردند.

همانطور که از قبل توضیح داده شد، در صورت استفاده از سیگموید برای تابع فعالیت چون تعداد لایه ها زیاد است سیگموید گرادیان رو صفر میکند (چون همش بین 0 و 1 عه و تو هم ضرب میشن به صفر میل میکنه) و یادگیری نخواهیم داشت. بنابر این معمولا از ReLU استفاده میشود:



$$R(z) = \begin{cases} z & z > 0 \\ 0 & z < 0 \end{cases}$$

تابع ReLU از لحاظ فیلتری هم یک معنی شهودی میدهد. این که اگر فیلتری که روی تصویر اعمال میکنیم مقدارش مثبت بود ارزشمنده و در نتیجه خودشو نشون میده ولی اگر کوچیک بود دیگر به درد نمیخوره و صفرش میکنیم.

گرادیان تصادفی

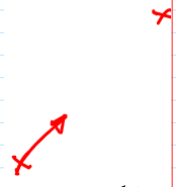
به جای استفاده از کل داده از mini batch استفاده میشود. فرمول یادگیری همانطور که دیدیم برابر است با:

$$w^{new} = w^{old} - \eta \cdot \sum_{l=1}^{L}$$

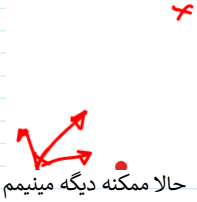
این که L چقدر باشد در گرادیان تصادفی بدین شکل است که اگر مثلا 10 هزار تا داده داشته باشیم میایم به صورت تصادفی و یکنواخت هزار تا رو انتخاب میکنیم میشه سایز mini batch. سپس در هر Epoch از این هزار تا برای آپدیت کردن وزن ها استفاده میکنیم.

مزیت: فرار کردن از مینیمم محلی. چون وقتی شبکه عمیق میشه تعداد مینیمم محلی هم زیاد میشه و وقتی از گرادیان معمولی استفاده کنیم در مینیمم محلی گیر میکنیم ولی گرادیان تصادفی شانس فرار کردن بیشتری دارد. علتشم اینه که چون از همه داده ها استفاده نکردیم جهتش خیلی دقیق نمیشه که به سمت مینیمم بره و تا حدی به سمتی میبرد که خطا کمتر شود.

برای درک این موضوع با شکل فرض کنیم اگر از همه داده ها استفاده کنیم اینطوری میریم به مینیمم محلی:



حالا که از بخشی از داده ها استفاده میکنیم جهتش به چپ یا راست منحرف میشه:



حالا ممکنه دیگه مینیمم محلی رو رد کنیم و به مینیمم اصلی منحرف شویم.

* در شبکه های عمیق بایستی تعداد داده ها نیز زیاد باشد. مثلا برای CNN با هزار تا تصویر نمیشه آموزش داد و به یک میلیون تا 10 میلیون تصویر نیاز داریم.

* برای افزایش سرعت از "گرادیان تصادفی موازی" نیز استفاده میشود که در این درس به توضیح آن پرداخته نمیشود.

جلسه نهم (بعد عید)

3:09 PM Sunday, April 4, 2021

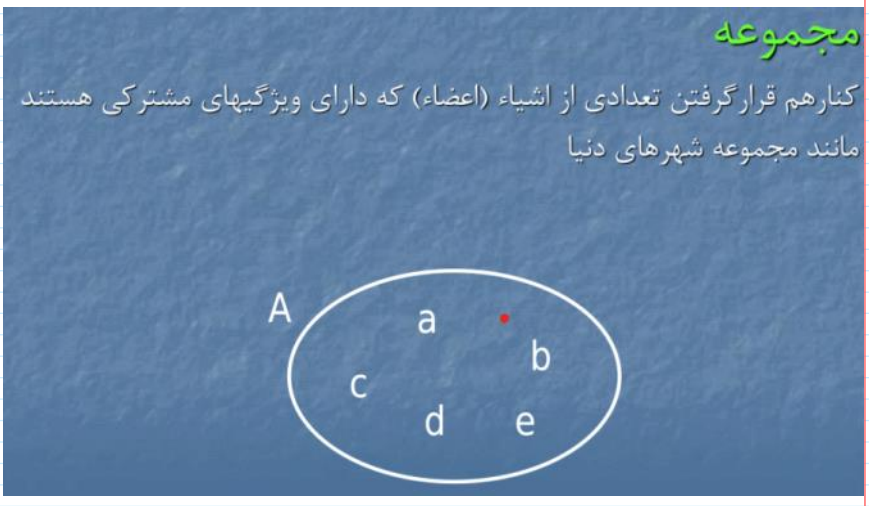
فرق بین متغیر قطعی و تصادفی: ما در متغیر تصادفی با خود متغیر سر و کار نداریم بلکه با احتمال سر و کار داریم. که در احتمال متغیر رو به فضای فرکانس یا فراوانی میبره که تعداد رخداد رو نشون میده.
در متغیر قطعی فضا عوض نمیشه و با مقدار خود متغیر سر و کار داریم.

- مثال از متغیر قطعی: قد علی 180 سانتی متره. 180 سانتی متر مقدار دقیق قد علی رو نشون میده.
- مثال از متغیر تصادفی: احتمال این که فردا بارون بیاد 60 درصده. این که احتمالش 60 درصده یعنی این که اگر ما شرایط آب و هوایی مشابه فردا رو داشته باشیم. تجربه و آمار نشان میدهد از تعداد دفعاتی که آب و هوا مشابه فردا بوده، اگر 1000 روز بوده 600 روزش بارون اتفاق افتاده.

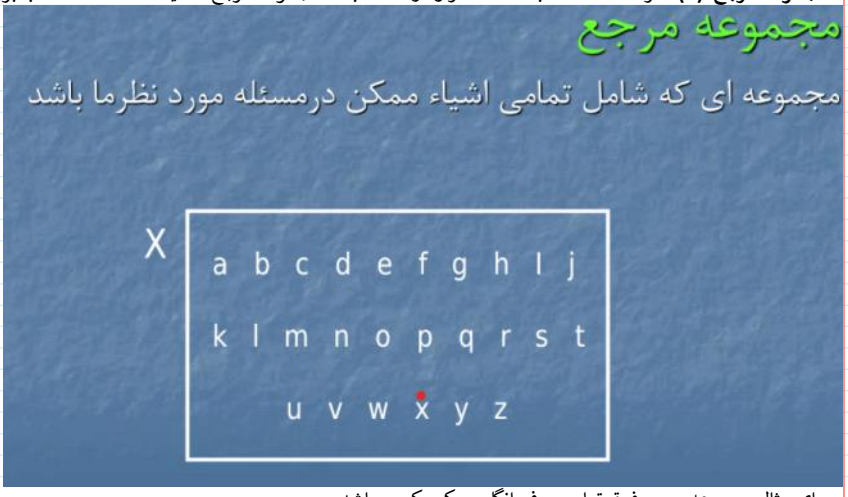
* زمانی میتوان از روش های آماری و احتمال استفاده کرد که تعداد نمونه ها زیاد باشه. مثلا من 1 میلیون بار تاس بندازم این که نشون بدم احتمال این که 2 بیاد 1/6 هستش بهتر میتونم نشون بدم تا این که 10 بار تاس بندازم. در یادگیری میتوان از روش آماری استفاده کرد چون تعداد های زیادی میتوانیم داده داشته باشیم.

متغیر نادقیق (فازی): متغیری که مقدار دقیقش رو نمیدونیم. مثلا بگیم "قد کوتاه". ما از یک بازه قد برای قد کوتاه استفاده میکنیم مثل 100 تا 130 سانتی متر. دقیق نمیشه گفت مقدار "قد کوتاه" چند سانتی متر است.
دقت شود این مفهوم با متغیر تصادفی متفاوت است. در متغیر تصادفی ما همواره مقدار رو میبریم تو فضای فرکانس و درمورد فراوانیش صحبت میکنیم ولی در متغیر نادقیق مقدار به صورت دقیق قابل تشخیص نیست و لذا یک بازه تعریف میشود.
برای این که داخل بازه بتونیم بین 100 تا 130 تفاوت قائل بشیم، اگر اهمیت بین قد 130 و 100 متفاوت باشد، میایم یک مقدار تعلق تعریف میکنیم. میگیم 130 سانتی متر مثلا 0.4 تعلق به قد کوتاه تعلق داره و 100 سانتی متر 1 تعلق داره به قد کوتاه. (مقدار تعلق همیشه بین 0 و 1 هستش و مجموعشون لزوما 1 نیست).
در بعضی موارد که برامون تفاوتشون اهمیت نداره تعلق همه رو 1 در نظر میگیریم. ← متغیر فازی بازه ای
اگر تفاوت داشته باشند ← متغیر فازی استاندارد

فازی تعمیم نظریه مجموعه هاست. مجموعه های کلاسیک میشه حالتی از مجموعه فازی. مثل این که اعداد مختلط رو میخونیم اول اعداد حقیقی رو میخونیم بعد که مختلط رو خوندیم میفهمیم اعداد حقیقی حالتی از اعداد مختلط هستند.

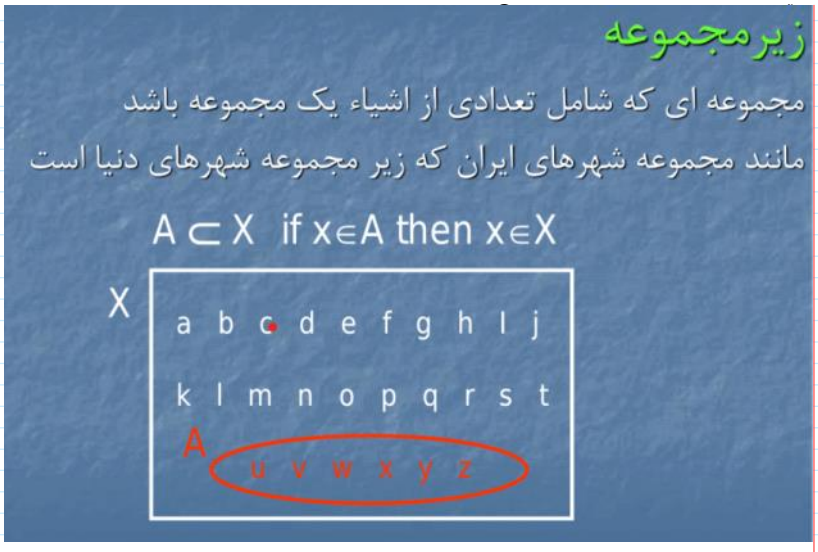


مجموعه مرجع (X): اگر مثلا اعداد ما با 1 تا 10 سر و کار داشته باشه مجموعه مرجع ما میشه اعداد 1 تا 10. بنابراین این مساله است که مشخص میکند مجموعه مرجع ما چه باشد.



برای مثال مجموعه مرجع فوق تمام حروف انگلیسی کوچک می باشد.

زیرمجموعه: همواره داخل مجموعه مرجع



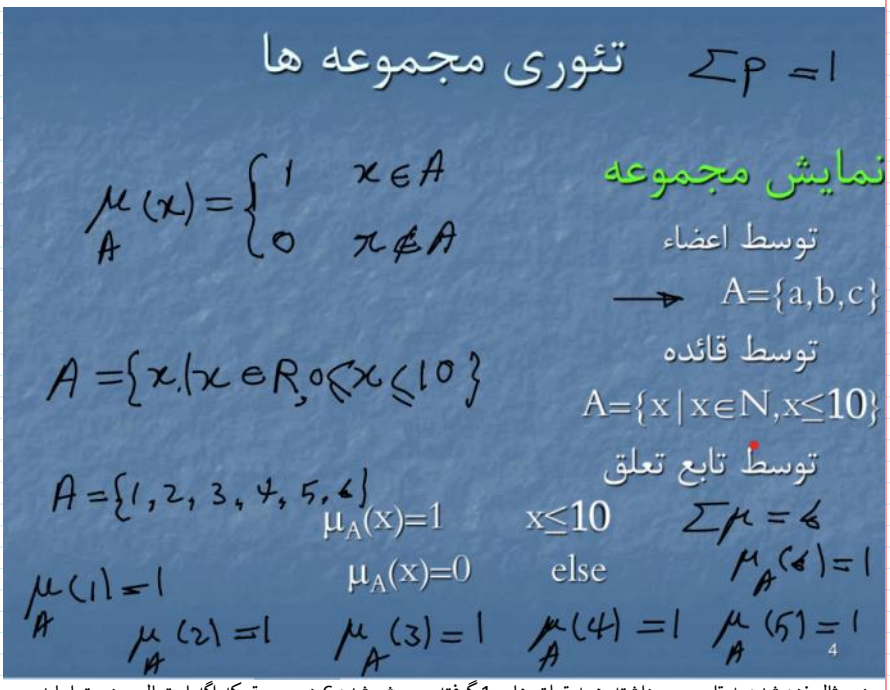
نمایش مجموعه:

- اگر تعداد اعضا محدود ← توسط اعضا
- اگر تعداد اعضا زیاد، مثل اعداد حقیقی بین 0 تا 10 ← توسط قائده
- **توسط تابع تعلق:** مشابه با روش احتمال که فضای متغیر رو میبرد توی فضای فرکانس در تابع تعلق فضای متغیر به فضای تعلق برده می شود. تابع تعلق را با μ نمایش میدهند. در $\mu_A(x)$، A اسم مجموعه فازی و x متغیر ما می باشد. در حالت کلاسیک میتوان تابع تعلق را برابر با زیر تعریف کرد:

$$\mu_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

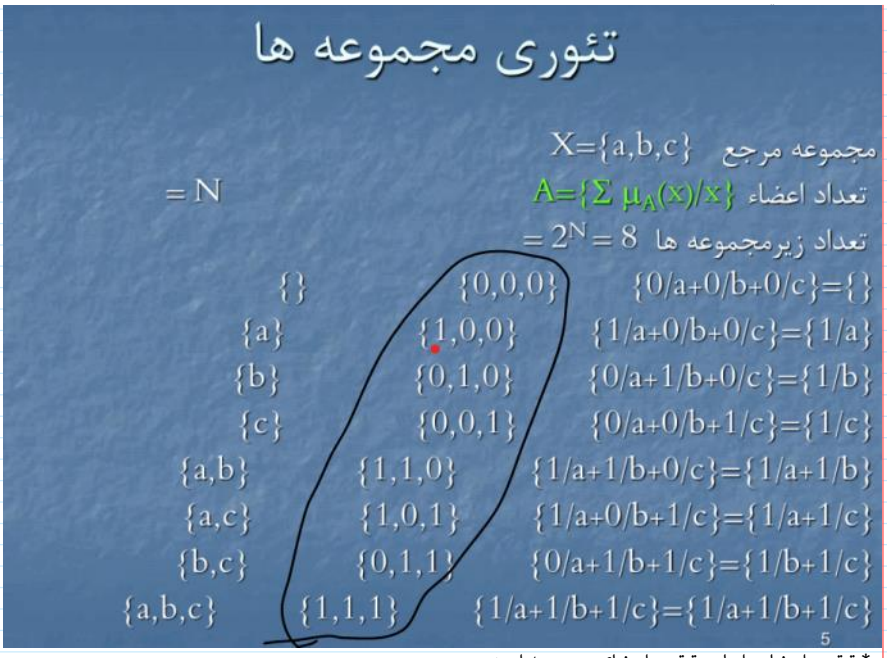
	این که متغیر ما تا چه حد به بازه تعلق دارد فضای تعلق را مشخص میکند.

* جمع فراوانی ها میشه فراوانی کل. یعنی مثلا 10 بار تاس رو بندازیم تعداد دفعه های تکرار 1 تا 6 رو جمع بزنیم باید 10 بشه. ولی در فضای تعلق چون فراوانی نداریم جمع تعلق ها حتما نباید 1 بشه.



در مثال زده شده یه تاس رو برداشته همه تعلق هارو 1 گرفته جمعش شده 6 در صورتی که اگه احتمال بود حتما باید جمعش 1 میشد.
با تعمیم تعریف فوق، این که تابع تعلق علاوه بر 0 و 1 مقداری بین 0 و 1 نیز اختیار میتواند کند میتوانیم نظریه مجموعه های کلاسیک را بسط بدهیم.

اگر بخواهیم مجموعه ها را با تابع تعلق نمایش بدهیم، یک راهش نمایش برداریه. یعنی مثلا اگر مجموعه ما {a, b, c} باشه. اون قسمتی که استاد دورش خط کشیده میشه روش برداریه. جاهایی که 1 شده یعنی تعلق داره جاهایی که 0 شده تعلق نداره:



* ترتیب اعضا بر اساس ترتیب اعضای مجموعه است.

* اینجا پاورپوینت هم یکم به هم ریخته داره میگه اگه تعداد اعضا N = باشه تعداد زیر مجموعه میشه 2^N

در روش غیر برداری تعلق های صفر رو نمایش نمیدهیم. اونایی که تعلق دارن رو 1 میذاریم منتهی یه خط کسری داریم و مخرج خود عضو رو مشخص میکنیم:

$$A=\left\{\frac{1}{a}+\frac{1}{b}\right\}$$

که حالا به نحو کلی ترش داریم:

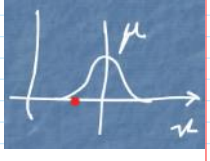
$$A=\left\{\frac{\mu_A(a)}{a}+\frac{\mu_A(b)}{b}\right\}$$

* جدا کردن اعضا در روش غیر برداری با "," یا "+" انجام میشه.

مقدار تعلق لزوما برای اعضای گسسته نیست. بر فرض میتوان برای مجموعه پیوسته نوشت:

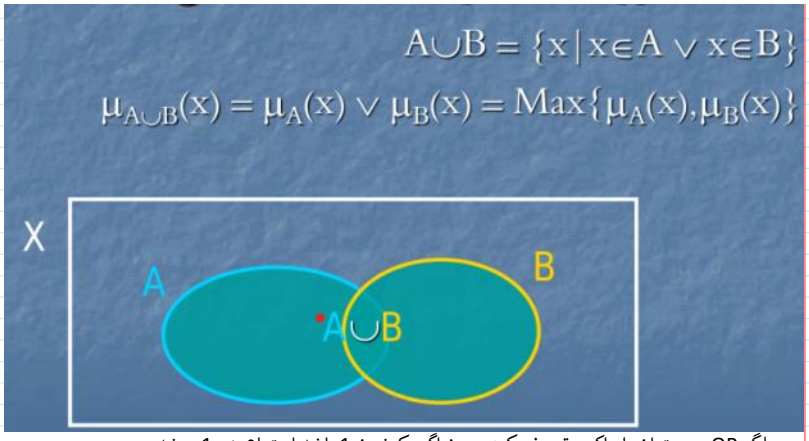
$$\mu_A(x)=e^{-x^2}$$

=



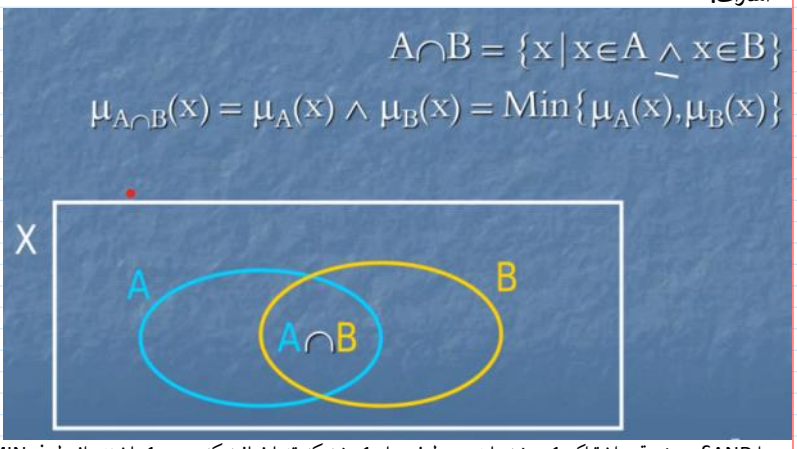
در حالت کلاسیک یک سری عملیات روی مجموعه داریم که حالا میخوایم با تابع تعلق تعریفشون کنیم.

- اجتماع:



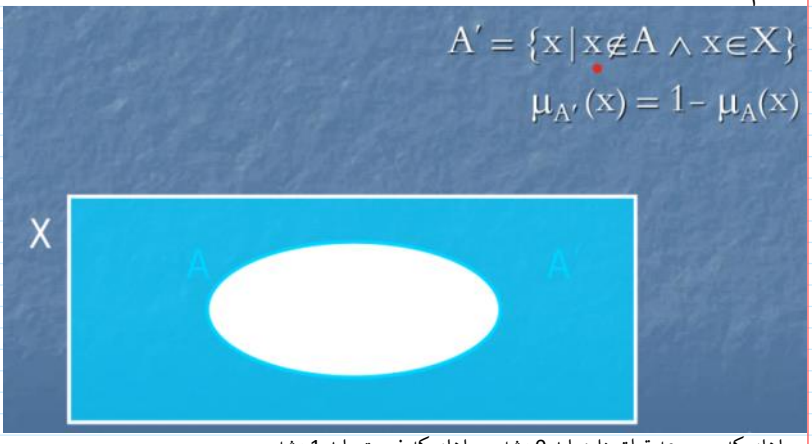
عملگر OR رو میتوان با ماکس تعریف کرد. چون اگر یکیشون 1 باشه اجتماع هم 1 میشه.

- اشتراک:



چرا AND؟ چون وقتی اشتراک 1 میشه باید وسط نمودار 1 بشه که تنها زمانیه که هردو 1 باشند. از طرفی MIN میشه چون خروجی MIN تنها زمانی 1 میشه که هردوشون 1 باشن.

-متمم



جاهایی که مجموعه تعلق داره باید 0 بشه و جاهایی که نیست باید 1 بشه.

خواص مجموعه ها:

عملیات روی مجموعه ها – خواص

جابجایی $A \cup B = B \cup A$

$A \cap B = B \cap A$

شرکت پذیری $A \cup (B \cup C) = (A \cup B) \cup C$

$A \cap (B \cap C) = (A \cap B) \cap C$

توزیع پذیری $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

دمرگان $(A \cup B)' = A' \cap B'$

$(A \cap B)' = A' \cup B'$

عملیات روی مجموعه ها – خواص

$A \cap A = A \cup \varnothing = A \cap X = A \quad A \cup A =$

$A \cap \varnothing = \varnothing \qquad A \cup X = X \qquad (A')' = A$

$A \cap A' = \varnothing \qquad A \cup A' = X$

If $A \subseteq B \subseteq C$ then $A \subseteq C$

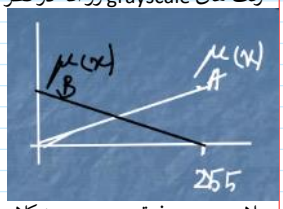
جلسه دهم

3:00 PM Tuesday, April 6, 2021

از لحاظ زبانی وقتی در مورد اعضا صحبت میکنیم (به صورت جمله ای) از اجتماع و اشتراک استفاده میکنیم. وقتی که در راجع به تعلق صحبت میکنیم، لغت به کار آمده AND و OR می باشد.

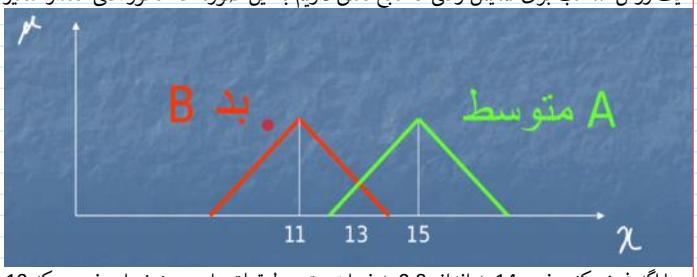
حالا میخوایم ببینیم آیا تعلق بین 0 و 1 باشه اشکالی برای نظریه مجموعه ها پیش می آید یا نه. اگر نیاد میتونیم نظریه مجموعه هارا تعمیم بدیم.

رنگ های grayscale رو اگه در نظر بگیریم. در باز 0 تا 255 هرچه به 255 نزدیک میشیم تعلق به سفید بیشتر میشه هرچه به 0 نزدیک میشیم تعلق به سیاه بیشتر میشه:



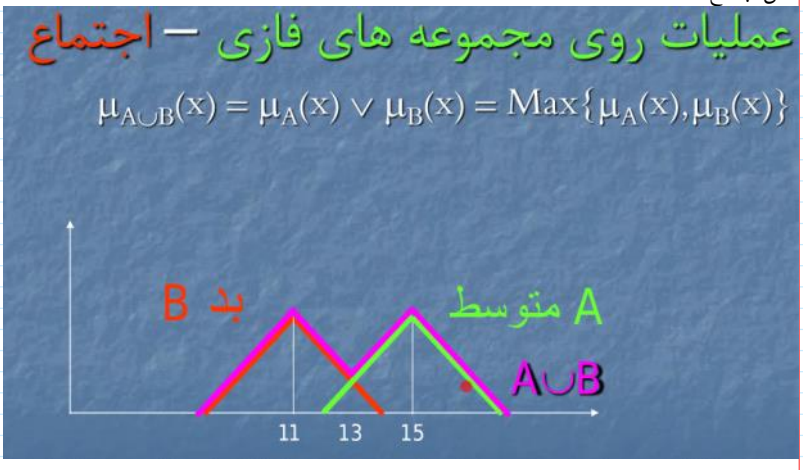
علاوه بر مورد فوق، در مجموعه کلاسیک نمیتوان سرعت زیاد، قد کوتاه یا زیبایی رو مدل کرد. با توجه به این که انسان نیز از این موارد استفاده میکند برای مدل کردن انسان بایستی از مجموعه فازی کمک گرفت.

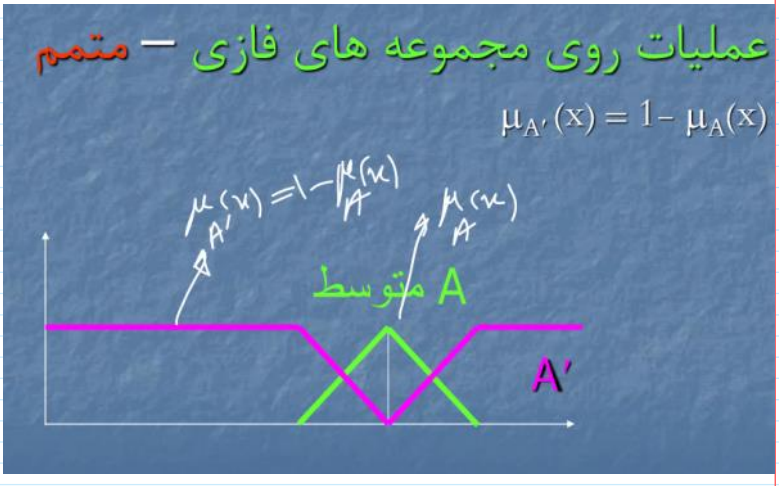
یک روش مناسب برای نمایش وقتی که تابع تعلق داریم به این صورته که محور افقی مقدار متغیر ها و محور عمودی مقدار تعلق باشد:

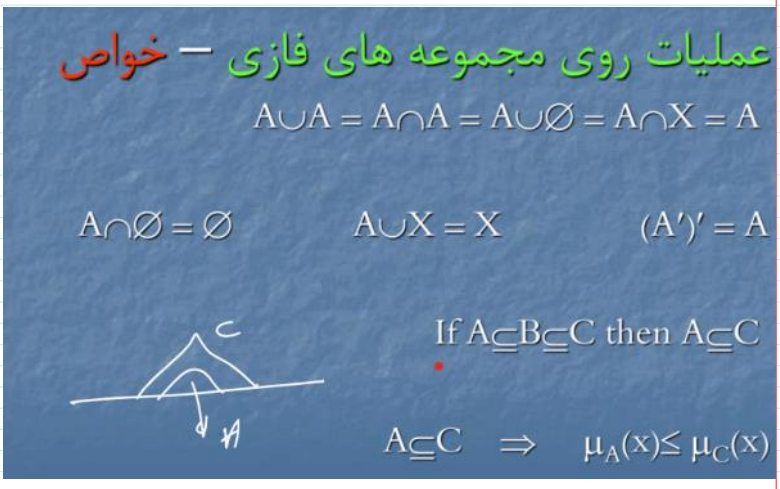


ما اگه فرض کنیم نمره 14 به اندازه 0.8 به نمرات متوسط تعلق داره، معنیش این نیست که 10 بار امتحان بدیم 8 بارش 14 میشه (معنی احتمال نمیده). بلکه ارزش اون نمره رو نشون میده.
جمع مقدار تعلق به میزان Overlap ربط داره. اگر Overlap زیاد باشه از 1 بیشتر میشه جمعش. اگر کیپ بشن جمعش 1 میشه و اگر فاصله داشته باشن از 1 کمتر میشه.

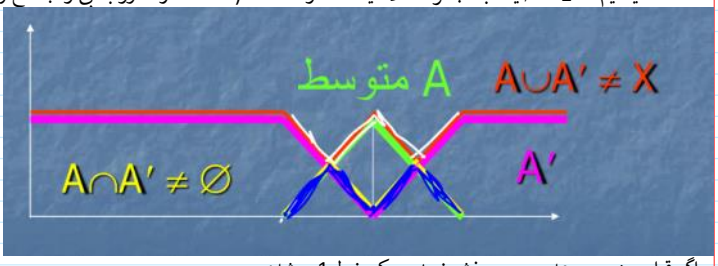
تعلق اجتماع:



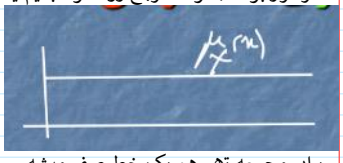




مشاهده میکنیم که 2 خاصیت با مجموعه کلاسیک متفاوت است (استاد اشتراک رو با آبی و اجتماع رو با سفید کشید):



اگر قرار بود مجموعه مرجع رو نشون بدیم یک خط 1 میشه:



برای مجمعه تهی هم یک خط صفر میشه

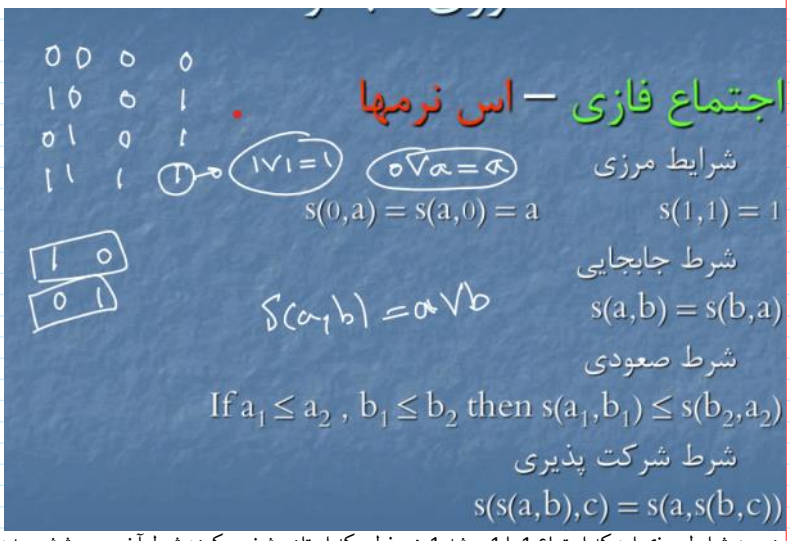
با توجه به این که این 2 خاصیت، از خواص های اصلی مجموعه نمی باشد (خاصیت های اصلی دمورگان و ... می باشند) میتوانیم بگوییم مجموعه فازی تعمیم مجموعه های کلاسیک می باشد.

چرا اجتماع رو MAX و اشتراک رو MIN و متمم رو 1 - مجموعه تعریف میکنیم؟ نمیشه جور دیگه تعریف کرد؟ درصورتی میتوان داشت که 2 خاصیت زیر را ارضا کند:

- در صورتی که 4 خاصیت اصلی رو داشته باشه (جابه جایی، شرکت پذیری، توزیع پذیری و دمورگان)
- شرایط مرزی: اگر مقادیر تعلق 0 و 1 شد، اجتماع و اشتراک و .. مثل حالت استاندارد باشد. (چون میخواهیم تعمیم باشد) یعنی مثلا در شکل زیر رو در نظر بگیریم. اگر 2 ستون اول ورودی هامون باشه و ستون سوم اشتراک و ستون آخر اجتماع باید داشته باشیم:

همچنین برای متمم هم باید داشته باشیم:

ما روی اجتماع و اشتراک ها میتونیم یه سری شرایط بذاریم که اگر این شرایط برقرار باشه اون موقع میتونه این خاصیت هارو خراب نکنه و شرایط مرزی هم داشته باشه در نتیجه نظریه مجموعه رو خراب نکنه. این اجتماع های جدید رو اس نرم و اشتراک هارو تی نرم میگیم.



در بین شرایط مرزی این که اجتماع 1 با 1 میشه 1 همونطور که استاد مشخص کرده شرط آخر رو پوشش میده و اشتراک a با 0 میشه a اون 3 شرط باقی رو پوشش میده چون تو هرکدوم از اون 3 تا حداقل یدونه صفر رو داره و اون یکی a تلقی میشه.

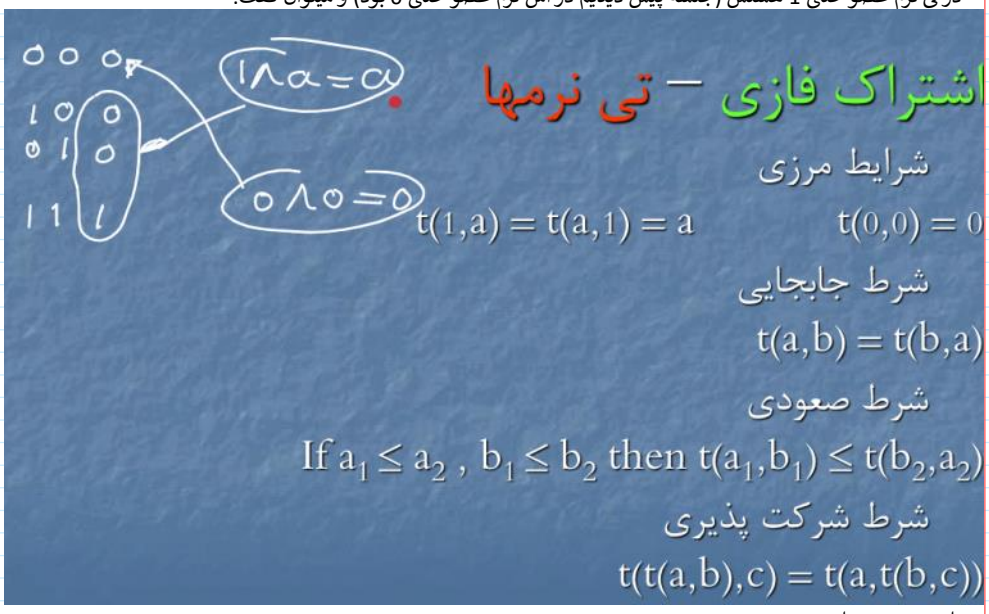
* دقت شود s(1,a) لزوما برابر با 1 نیست. با استفاده از اون دو شرط مرزی موجود در اسلاید فوق میتوان 0 0، 1 0، 0 1، و 1 1 رو مدل کرد ولی میشه یک نوع فرمول اجتماع غیر از ماکسیمم گیری (که حالا در اسلاید های جلسه بعد هست) تعریف کرد که اجتماع a و 1 لزوما 1 نشه

جلسه یازدهم

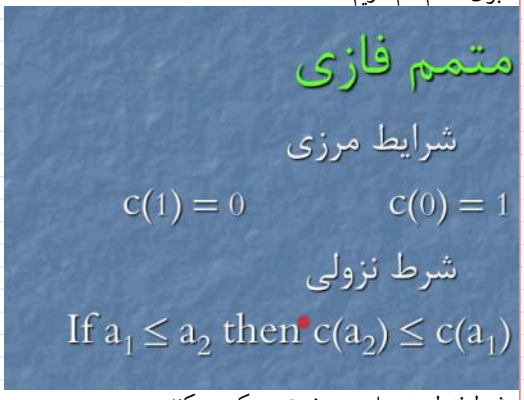
2:58 PM Sunday, April 11, 2021

این که مثلا میایم اجتماع رو میگیریم MAX و اشتراک رو MIN به این میگن یک کلاس نرمال. دانشمندان دیگری اومدن کلاس های نرمال دیگری هم ساختن که توی این جلسه بهشون می پردازیم.

* در تی نرم عضو خنثی 1 هستش (جلسه پیش دیدیم در اس نرم عضو خنثی 0 بود) و میتوان گفت:



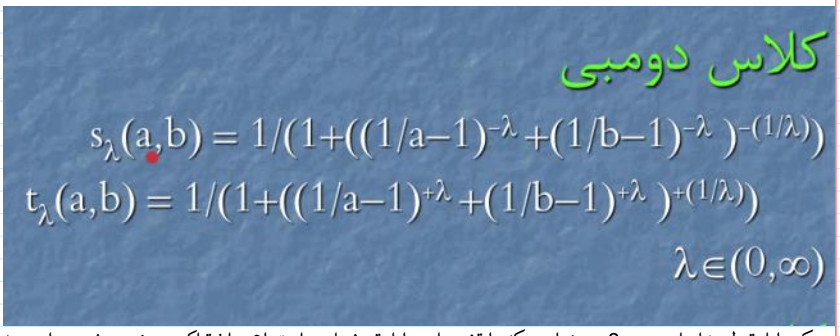
برای متمم هم داریم:



شرط نزولی هم داریم چون متمم عکس میکنه.

* برای کلاس نرمال متمم باید خاصیت دمورگان رو از بین نبرد.

حالا بریم کلاس های نرمال رو بپردازیم (نیازی به حفظ نیست صرفا میخوایم به گوشمون بخوره):



یک پارامتر لمبدا داریم بین 0 و بینهایت که با تغییر این پارامتر خواص اجتماع و اشتراک عوض میشود. راجع به این که با تغییر پارامتر اشتراک و اجتماع چطوری عوض میشن در جلوتر توضیح داده میشه.

برای کلاس هایی که متمم تعریف نمیشه متمم همون متمم استانداردِه

کلاس یاگر

$$s_w(a,b) = \mathrm{Min}(1,(a^w+b^w)^{1/w})$$

$$t_w(a,b) = 1-\mathrm{Min}(1,((1-a)^w+(1-b)^w)^{1/w})$$

$$c_w(a) = (1-a^w)^{1/w}$$

$$w\in(0,\infty)$$

جمع وضرب دراستیک

$$s_{ds}(a,b) = \{a \text{ if } b=0,\ b \text{ if } a=0,\ 1 \text{ else}\}$$

$$t_{dp}(a,b) = \{a \text{ if } b=1,\ b \text{ if } a=1,\ 0 \text{ else}\}$$

22

جمع و ضرب دراستیک اجتماع و اشتراک خوبی نیست. برای این تعریف شده که دو خاصیتی که برقرار نبود برقرار بشن. یعنی میتوان گفت:

$A \cup A' = X$

$A \cap A' = \phi$

جمع وضرب دراستیک

$$s_{ds}(a,b) = \{a \text{ if } b=0,\ b \text{ if } a=0,\ 1 \text{ else}\}$$

$$t_{dp}(a,b) = \{a \text{ if } b=1,\ b \text{ if } a=1,\ 0 \text{ else}\}$$

22

وقتی که این خاصیتا بخوان اتفاق بیافتن خود اجتماع و اشتراک ها جالب نمیشن. که در اسلاید های بعدی مشاهده میکنیم.

جمع وضرب اینشتین

$$s_{es}(a,b) = (a+b)/(1+ab)$$

$$t_{ep}(a,b) = ab/(2-(a+b-ab))$$

جمع وضرب جبری

$$s_{as}(a,b) = a+b-ab$$

$$t_{ap}(a,b) = ab$$

کلاس سوگنو

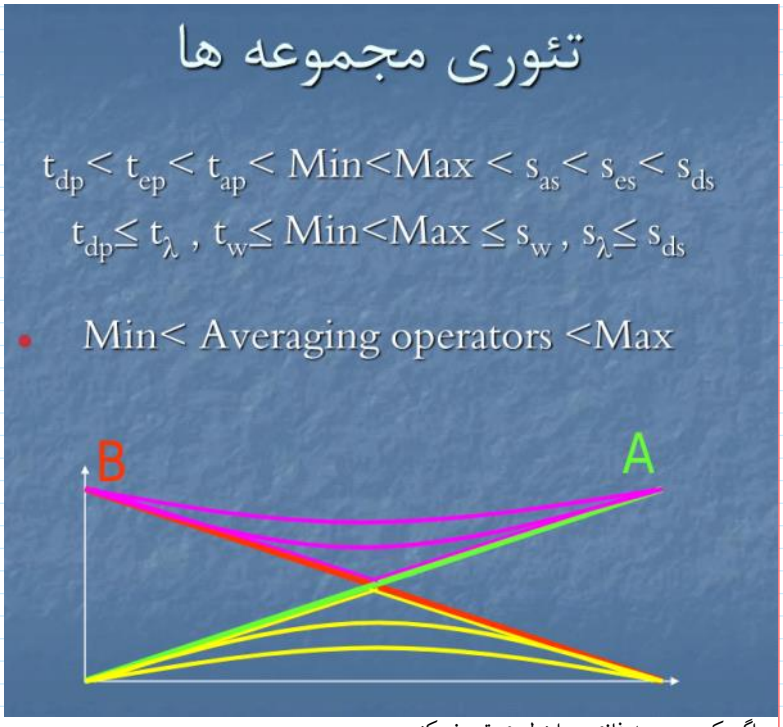
$$c_\lambda(a) = (1-a)/(1+\lambda a)$$

جمع و ضرب جبری معادل اجتماع و اشتراک در احتمال است وقتی که مستقل باشند:

$$P(A\cup B) = P(A)+P(B)-P(A\cap B)$$

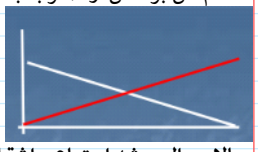
$$P(A\cap B) = P(A)\underbrace{P(B|A)}_{P(B)}$$

حالا بریم ببینیم چه فرق دارن:



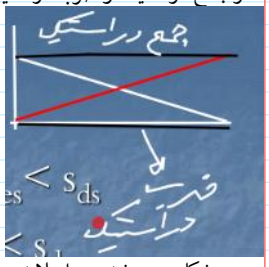
اگر یک مجموعه فازی رو اینطوری تعریف کنیم:

متمم اش برعکس از صفر به بالا میره.

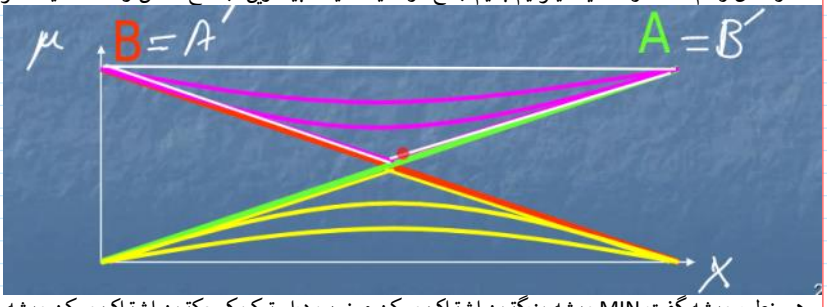


حالا سوال میشه اجتماع و اشتراک چطوری عمل میکنه !؟

در جمع دراستیک و ضرب دراستیک اجتماعش یک خط y=1 و اشتراکش y=0 میشه:

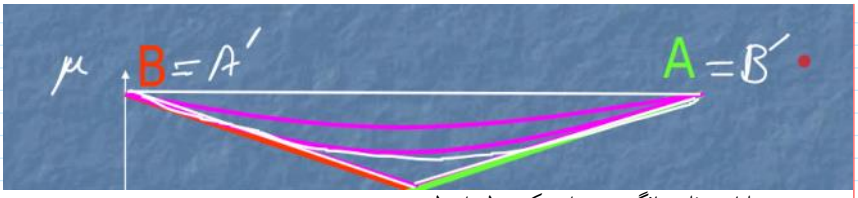


در شکل رسم شده در اسلاید میتونیم بگیم جمع دراستیک میشه بیشترین اجتماع ممکن و MAX میشه کوچکترین اشتراک ممکن (استاد با رنگ سفید کشید)

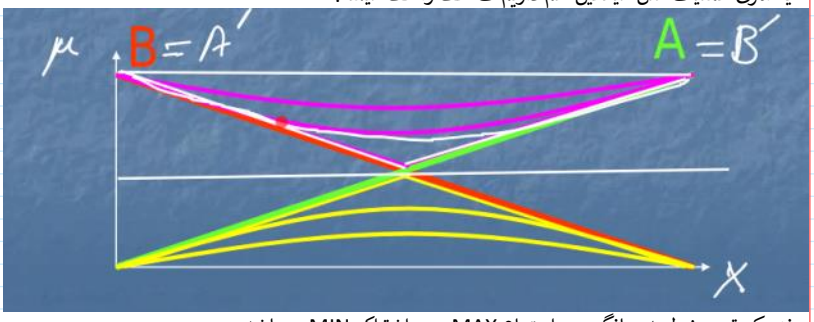


همینطور میشه گفت MIN میشه بزرگترین اشتراک ممکن و ضرب دراستیک کوچکترین اشتراک ممکن میشه.

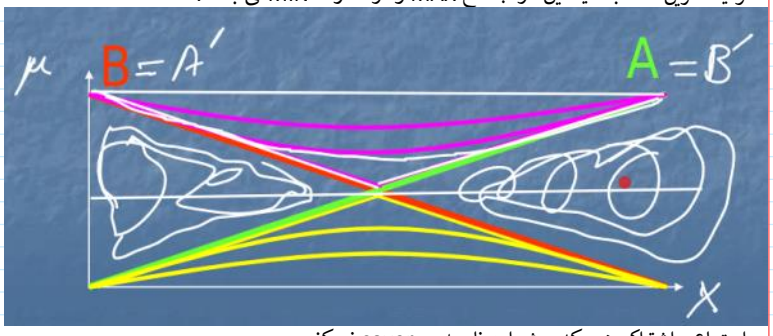
جمع جبری یک منحنی ما بین میشه:



یه سری عملیات مثل میانگین هم داریم ک خط واسط میشه:



نزدیک ترین خط به میانگین در اجتماع MAX و در اشتراک MIN می باشد.



اجتماع و اشتراک هیچکدومش این ناحیه رو cover نمیکنن.

---اتمام اسلاید

تمرین:

حجم کم $= A = \left\{ \frac{0.2}{1} + \frac{0.9}{2} + \frac{0.5}{3} \right\}$ نرخ کم $= B = \left\{ \frac{0.4}{1} + \frac{0.8}{2} + \frac{0.3}{3} \right\}$

$$A \cap B = \left\{ \frac{0.2}{1} + \frac{0.8}{2} + \frac{0.3}{3} \right\}$$

تصویر فوق غلطه! اینا با هم اشتراک ندارن. چرا؟ چون اینا واحداشون فرق داره:

حجم کم $= A = \left\{ \frac{0.2}{1} + \frac{0.9}{2} + \frac{0.5}{3} \right\}$ نرخ کم $= B = \left\{ \frac{0.4}{1} + \frac{0.8}{2} + \frac{0.3}{3} \right\}$

cm^3 cm^3 cm^3 یکا یکا یکا

$$A \cap B = \cancel{\left\{ \frac{0.2}{1} + \frac{0.8}{2} + \frac{0.3}{3} \right\}}$$

به شرطی میتوان اشتراک گرفت که اینا هم مرجع بشن. مثلا فرض کنیم یک C داریم که حجم متوسطه، چون واحدا یکیه اگه اشتراک بگیریم داریم:

حجم کم $= A = \left\{ \frac{0.2}{1} + \frac{0.9}{2} + \frac{0.5}{3} \right\}$ (cm3, cm3, cm3)

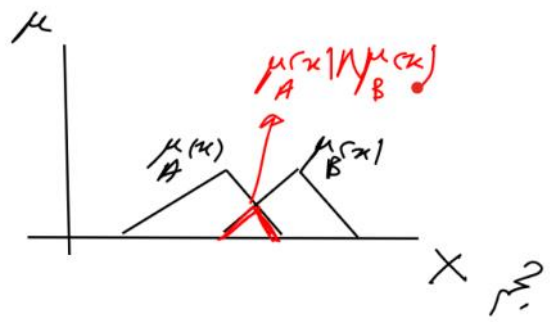
فشار کم $= B = \left\{ \frac{0.4}{1} + \frac{0.8}{2} + \frac{0.3}{3} \right\}$ (پسکال، پسکال، پسکال)

$\cancel{A \cap B = \left\{ \frac{0.2}{1} + \frac{0.8}{2} + \frac{0.3}{3} \right\}}$

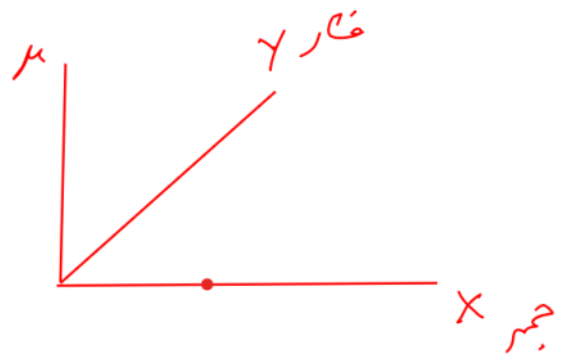
$A \cap C = \left\{ \frac{0.4}{3} \right\}$

حجم متوسط $= C = \left\{ \frac{0.4}{3} + \frac{0.8}{4} + \frac{0.2}{5} \right\}$

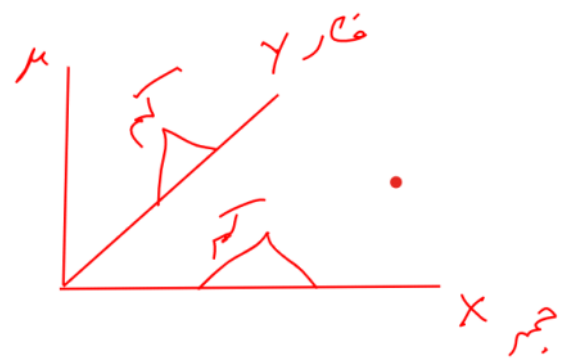
ولی اگر پیوسته باشند میتونیم توی شکل نشون بدیم:



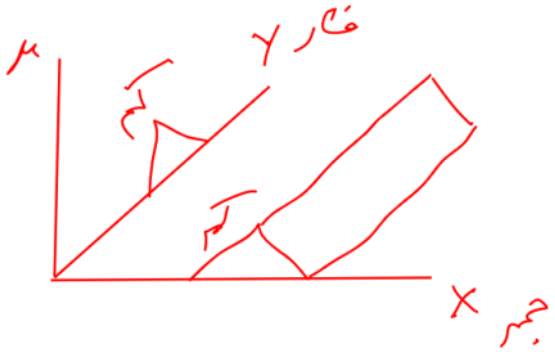
فرض کنیم که محور هامون 3 تا باشه.



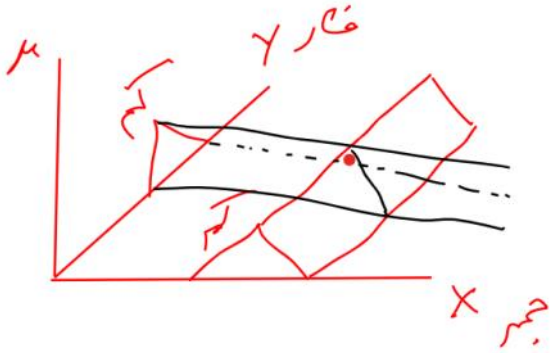
اگر فشار کم و حجم کم داشته باشیم:



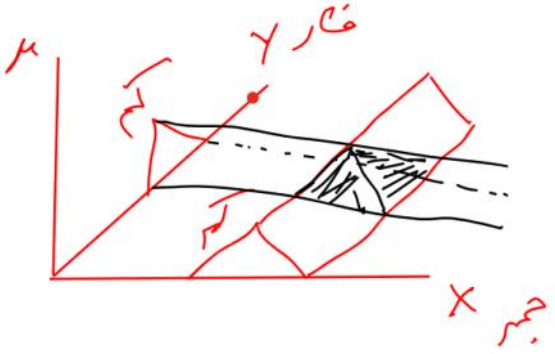
اینا هیچ اشتراکی ندارند. ولی برا این که اشتراک داشته باشن، میتوانیم از عملگر توسعه استوانه ای استفاده کنیم. به این صورت که به مجموعه حجم کم مجموعه فشار رو اضافه میکنه و مجموعه رو دو بعدی میکنه



اگر به فشار هم مجموعه مرجع حجم رو اضافه کنیم با توسعه استوانه ای:



در این وسط یک هرم به وجود میاد که میشه اشتراک فشار کم و حجم کم.



--- شروع اسلاید

ضرب کارتزین رو میشه بر اساس توسعه استوانه ای تعریف کرد یا این که میشه مستقل تعریف کرد و توسعه استوانه ای رو بر حسب ضرب کارتزین تعریف کنیم. مثل داستان مرغ و تخم مرغه چون هیچکدوم رو نمیشه اول گفت هرکدوم یک دور دارن.

دور این که اول توسعه استوانه ای روتعریف کنیم کمتره. اول ضرب کارتزین رو تعریف میکنیم بعد توسعه استوانه ای رو و بار بعدی برعکس

ضرب کارتزین در حالت کلاسیک به چه درد میخوره؟ اگر اعضای کلاس هوش محاسباتی رو با ویژگی اسم بیان کنیم و بگیم که حسن، علی، نقی، تقی. **این چه مشکلی داره؟** ما ممکنه نتونیم همه افراد رو از هم جدا کنیم. ممکنه 2 تا نقی و 2 تا تقی داشته باشیم. اگه با اسم و فامیل بریم شناسایی راحت تر میشه و در حالت کلی هرچی ویژگی بیشتر بشه شناسایی راحت تره.

مجموعه از تعدادی اشیا هستش. توی جلسه های اول سیب و پرتقال رو با قطر و وزن نمایش دادیم. اگر قطر و وزن نتونن اینارو خوب توصیف کنن میشه یک ویژگی دیگه اضافه کرد که بهتر جدا کنه.

اشیا در مجموعه ها میتونن بیشتر از یک ویژگی داشته باشند. ضرب کارتزین مجموعه مرجع چند بعدی رو ایجاد میکنه. اگر مجموعه مرجع اسامی رو داشته باشیم، مجموع مرجع فامیلا هم داشته باشیم. حاصل ضرب کارتزین اشون مجموع مرجع افراد رو با 2 ویژگی اسم و فامیل تولید میکنه.

البته در ضرب کارتزین ترتیب مهمه. $V \times U$ با $U \times V$ دو مجموعه مرجع متفاوت میسازند

ضرب کارتزین

$U=\{u_1,u_2\}$

$V=\{v_1,v_2,v_3\}$

$U\times V=\{(u,v) \mid u\in U, v\in V\}$

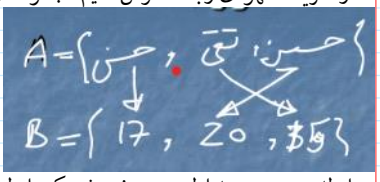
$U\times V=\{(u_1,v_1),(u_1,v_2),(u_1,v_3),(u_2,v_1),(u_2,v_2),(u_2,v_3)\}$

تعداد اعضاء $= N_U \cdot N_V$

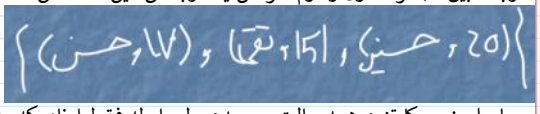
$U_1\times U_2\times\ldots\times U_n=\{(u_1,u_2,\ldots,u_n) \mid u_1\in U_1,\ldots,u_n\in U_n\}$

اگر مجموعهU، n تا عضو و V هم m تا عضو داشته باشه مجموعه حاصل از ضرب کارتزین m×n تا عضو داره.

رابطه: یک تعریف شهودی و یک تعریف ریاضی داره. در تعریف ریاضی میشه زیر مجموعه از یک حاصل ضرب کارتزین.
خود ضرب کارتزین یک مجموعه است ولی مجموعه چند بعدیه.
در تعریف شهودی رابطه، فرض کنیم مجموعه A,B زیر رو داریم:

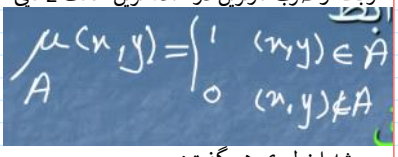


رابطه بین مجموعه اول و دوم خودش یک رابطس. این که حسن 17 گرفته پس میتوانیم یک مجموعه دو بعدی در نظر بگیریم که حسن و 17 یک عضوش بشه. وهمینطور داریم:

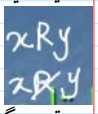


حاصل ضرب کارتزین همه حالت رو میده. ولی رابطه فقط اونایی که ما میخوایم رو میده.
چرا این میشه زیر مجموعه از ضرب کارتزین؟ چون ضرب کارتزین همه حالت رو تولید میکنه و رابطه ما نمیتونه خارجش باشه و هر رابطه که تعریف کنیم داخلش هستش.

رابطه و ضرب کارتزین در ساده ترین حالت 2 تایی هستش. پس اگه بخوایم با تابع تعلق نشون بدیم باید تابع تعلق 2 بعدی باشه. یعنی مثلا:



میشه اینطوری هم گفت:



وقتی میگیم (x,y)εA یعنی x با y رابطه داره پس xRy. برعکسشم که مشخصه.

در نمایش ماتریسی رابطه هم جاهایی که رابطه داریم 1 جاهایی که نداریم صفر میذاریم:

نمایش ماتریسی رابطه

$R_1(u,v)=\{(u_1,v_1),(u_1,v_2)\}$

		V		
		v_1	v_2	v_3
	u_1	1	1	0
U	u_2	0	0	0

رابطه کامل و رابطه تهی

$$\begin{array}{cc|ccc} & & & V & \\ & & v_1 & v_2 & v_3 \\ \hline & u_1 & 1 & 1 & 1 \\ U & u_2 & 1 & 1 & 1 \\ & u_3 & 1 & 1 & 1 \end{array} \qquad \begin{array}{cc|ccc} & & & V & \\ & & v_1 & v_2 & v_3 \\ \hline & u_1 & 0 & 0 & 0 \\ U & u_2 & 0 & 0 & 0 \\ & u_3 & 0 & 0 & 0 \end{array}$$

* رابطه یک مجموعه دو بعدیه

رابطه همسایگی

U={ایران,فرانسه}

V={پاکستان,آلمان}

$$\begin{array}{cc|cc} & & & V \\ & & v_1 & v_2 \\ \hline & u_1 & 1 & 0 \\ U & u_2 & 0 & 1 \end{array}$$

$R_1(u,v)=\{(u_1,v_1),(u_2,v_2)\}$

$R_1(u,v)=\{(\text{آلمان},\text{فرانسه}),(\text{پاکستان},\text{ایران})\}$

اگر بخوایم با notation جدید بنویسیم هم:

$R_1(u,v)=\{\overbrace{(u_1,v_1)}^{1},\overbrace{(u_2,v_2)}^{1}\}$

•

$R_1(u,v)=\{\overbrace{(\text{آلمان},\text{فرانسه})}^{1},\overbrace{(\text{پاکستان},\text{ایران})}^{1}\}$

عملیات روی رابطه ها – اجتماع

$R \cup S = \{U \mid U \in R \vee U \in S\}$

$\mu_{R\cup S}(U) = \mu_R(U) \vee \mu_S(U) = \mathrm{Max}\{\mu_R(U), \mu_S(U)\}$

عملیات روی رابطه ها – اشتراک

$R \cap S = \{U \mid U \in R \wedge U \in S\}$

$\mu_{R\cap S}(U) = \mu_R(U) \wedge \mu_S(U) = \mathrm{Min}\{\mu_R(U), \mu_S(U)\}$

عملیات روی رابطه ها – متمم

$$\mu_{R'}(U) = 1 - \mu_R(U)$$

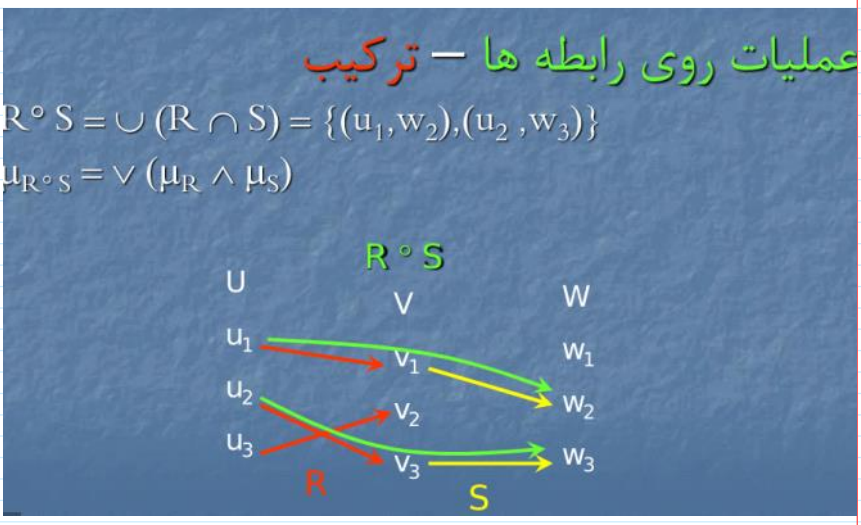
عملیات روی رابطه ها – زیرمجموعه

$$R \subseteq S \Rightarrow \mu_R(U) \le \mu_S(U)$$

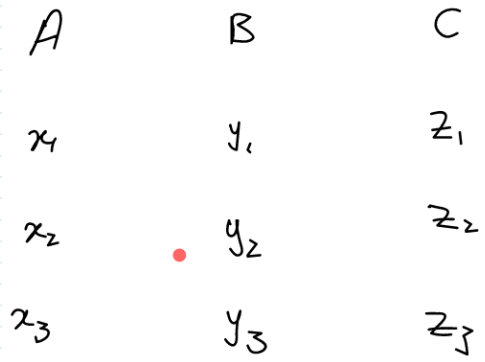
یک عملکرد دیگه هم داریم که ترکیبه که جلسه بعد در موردش حرف میزنیم.

جلسه دوازدهم

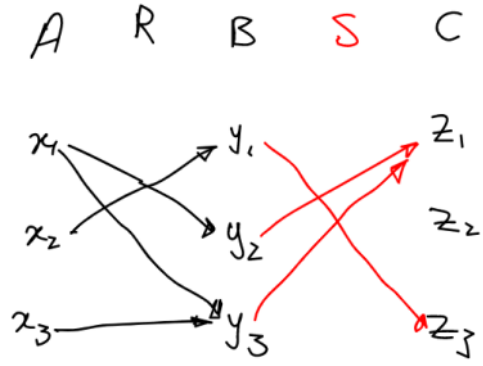
3:02 PM Tuesday, April 13, 2021



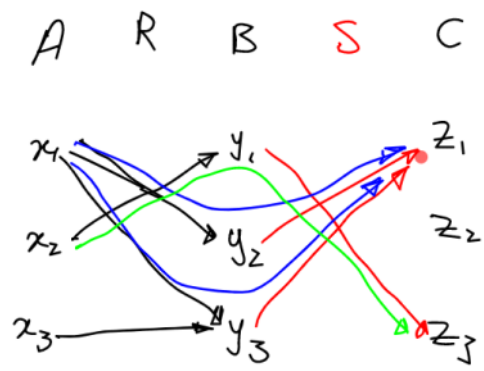
ترکیب: فرض کنیم مجموعه Aو Bو C داریم. به طوری که:



حالا فرض کنیم A و B با رابطه R و B و C هم با رابطه S در ارتباط هستند:



ترکیب به ما میگه رابطه بین Aو C چیه. یعنی ما اگه رابطه AوB و همچنین رابطه Bو C داشته باشیم ترکیب به ما میگه رابطه AوC چیه.



پس میتوان گفت :

$$(x_1, z_1) \in R \circ S$$

$$(x_2, z_3) \in R \circ S$$

باید حداقل یک مسیر وجود داشته باشه تا بتونیم رابطه رو برقرار کنیم. اگر بخوایم بر حسب تابع تعلق بیان کنیم میتونیم بگیم:

$$\mu_{R\circ S}(x_1, z_1) = \overset{0}{\left(\mu_R(x_1, y_1) \wedge \mu_S(y_1, z_1)\right)} \vee \overset{1}{\left(\mu_R(x_1, y_2) \wedge \mu_S(y_2, z_2)\right)} \vee \overset{1}{\left(\mu_R(x_1, y_3) \wedge \mu_S(y_3, z_3)\right)}$$

چون بین x_1و y1 در R رابطه ای نیست مقدارش 0 میشه ولی برای 2 تای دیگه برقراره و 1 میشه.

میشه به نحو زیر هم نوشت:

$$\mu_{R\circ S}(x, z) = \bigvee_{y \in Y} \left(\mu_R(x, y) \wedge \mu_S(y, z)\right)$$

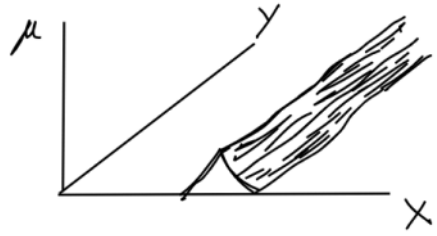
* در توسعه استوانه ای هدفمون اضافه کردن یک بُعد به یک مجموعه است و باعث میشه یک مجموعه یک بعدی رو به 2 بعدی تبدیل کنیم. کارش هم این بود که تعلق های یک مجموعه رو برای همه اعضای مجموعه دوم تکرار میکنه:

$A' =$ توسعه استوانه ای A

$$X = \left\{\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\right\} \quad A = \left\{\frac{0.4}{a} + \frac{0.3}{b}\right\}$$

$$Y = \left\{\frac{1}{1} + \frac{1}{2} + \frac{1}{3}\right\}$$

$$A' = \left\{\frac{0.4}{a,1} + \frac{0.4}{a,2} + \frac{0.4}{a,3} + \frac{0.3}{b,1} + \frac{0.3}{b,2} + \frac{0.3}{b,3}\right\}$$



حالا ما 'A رو میتونیم اینطوری تعریف کنیم:

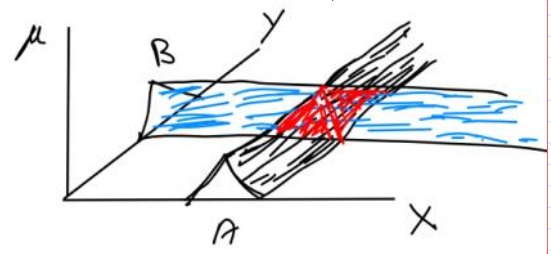
$$A' = A \times Y$$

و همون داستان مرغ و تخم مرغ بین توسعه استوانه ای و حاصل ضرب کارتزین به وجود میاد.

$$\mu_{A'}(x,y) = \mu_A(x) \wedge \cancel{\mu_{A'}(y)}$$

$$\mu_{A'}(x,y) = \mu_A(x)$$

از اشتراک این 2 تا یک هرم به وجود میاد:



این هرم حاصل ضرب کارتزین A و B میشه.

حال اگر مجموعه A با مجموعه مرجع X و مجموعه B با مجموعه مرجع Y داشته باشیم:

$$A \quad X \qquad\qquad A' = A \times Y \rightarrow X \times Y$$

$$B \quad Y \qquad\qquad B' = X \times B \rightarrow X \times Y$$

$$A' \cap B' = (A \times Y) \cap (X \times B) = A \times B$$

$$A \cap B = A \times B =$$ اشتراک توسعه استوانه A با توسعه استوانه B

در صورتی که مجموع مرجع یکسان نباشد

حال اگر مجموعه AوB داشته باشیم ک هردو مرجع X رو داشته باشند. مجموعه C هم مرجع X و D مرجع Y داشته باشد داریم:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) \qquad\qquad \begin{matrix} A & X \\ B & X \end{matrix}$$

$$\mu_{C \cap D}(x,y) = \mu_C(x) \wedge \mu_D(y) \qquad\qquad \begin{matrix} C & X \\ D & Y \end{matrix}$$

میبینیم روابط یکسان شد. چون این فرمول اگر ما یک مرجع یکسان نیست متغیر هاش رو متفاوت فرض کنیم. خودش به طور اتوماتیک اون توسعه استوانه ای رو برای ما انجام میده و لازم نیست ما توسعه استوانه ای رو انجام بدیم.

مثال)

روش سوم

$$\mu_{A\times B}(x,y)=\mu_A(x)\wedge\mu_B(y) \qquad \mu_{A\times B}(a,1)=\mu_A(a)\wedge\mu_B(1)=0.2\wedge 0=0$$

$$A=\left\{\frac{0.2}{a}+\frac{0.8}{b}\right\} \qquad X=\left\{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}\right\}$$

$$B=\left\{\frac{0.7}{2}+\frac{0.3}{3}\right\} \qquad Y=\left\{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}\right\}$$

روش اول

$$A\cap B=A\times B=\left\{\frac{0.2}{a,2}+\frac{0.2}{a,3}+\frac{0.7}{b,2}+\frac{0.3}{b,3}\right\}$$

روش دوم

$$A\times Y=\left\{\frac{0.2}{a,1}+\frac{0.2}{a,2}+\frac{0.2}{a,3}+\frac{0.8}{b,1}+\frac{0.8}{b,2}+\frac{0.8}{b,3}\right\}$$

$$X\times B=\left\{\frac{0.7}{a,2}+\frac{0.7}{b,2}+\frac{0.7}{c,2}+\frac{0.3}{a,3}+\frac{0.3}{b,3}+\frac{0.3}{c,3}\right\}$$

$$A\times B=(A\times Y)\cap(X\times B)=\left\{\frac{0.2}{a,2}+\frac{0.2}{a,3}+\frac{0.7}{b,2}+\frac{0.3}{b,3}\right\}$$

-- برگشت به اسلاید

ترکیب هارو به صورت ضرب ماتریس هم میشه در نظر گرفت:

روابط

Max Min

	F	v_1	v_2	v_3
D	R	C	T	E
u_1	.1	0	0	1
u_2	1	0	0	1
u_3	10	1	1	0

	P	w_1	w_2	w_3
F	S	.1	1	10
v_1	C	1	0	0
v_2	T	0	1	1
v_3	E	0	0	1

	P	w_1	w_2	w_3
D	R°S	.1	1	10
u_1	.1	0	0	1
u_2	1	0	0	1
u_3	10	1	1	1

نکته که ازش موجوده اینه که بین 2 روابط باید یک مجموعه مشترک که این جا v_i ها هستن موجود باشه وگرنه نمیشه ترکیب کرد.

روابط

عملیات روی روابط – خواص

جابجایی

$R \cup S = S \cup R$

$R \cap S = S \cap R$

شرکت پذیری

$R \cup (S \cup T) = (R \cup S) \cup T$

$R \cap (S \cap T) = (R \cap S) \cap T$

توزیع پذیری

$R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$

$R \cup (S \cap T) = (R \cap S) \cup (R \cap T)$

دمرگان

$(R \cup S)' = R' \cap S'$

$(R \cap S)' = R' \cup S'$

روابط

ضرب کارتزین فازی

$U=\{\mu_{u1}/u_1, \mu_{u2}/u_2\}$

$V=\{\mu_{v1}/v_1, \mu_{v2}/v_2\}$ $\mu_{U\times V} = \mu_U \wedge \mu_V$

$U\times V=\{\mu_{uv}/(u,v) \mid \mu_{uv}= \mathrm{Min}(\mu_u, \mu_v), u\in U, v\in V\}$

$U\times V=\{\mu_{11}/(u_1,v_1), \mu_{12}/(u_1,v_2), \mu_{21}/(u_2,v_1), \mu_{22}/(u_2,v_2)\}$

تعداد اعضاء $= N_U \cdot N_V$

$U_1\times U_2\times\ldots\times U_n =\{\mu_{u1,\ldots,un}/(u_1,u_2,\ldots,u_n) \mid$

$\mu_{u1,\ldots,un} = \mathrm{Min}(\mu_{u1},\ldots,\mu_{un}), u_1\in U_1,\ldots,u_n\in U_n \}$

ضرب کارتزین فازی میشه اشتراک توسعه استوانه ای ها. این اشتراک توسعه استوانه ای ها معادل با این هست که بگیم زوج هایی که زوج اول برای مجموعه اول و زوج دوم برای مجموعه دوم و تعلقش اشتراک بین تعلق ها باشه.

روابط

رابطه بین مجموعه های فازی

رابطه بین مجموعه های فازی زیرمجموعه ای از ضرب کارتزین فازی آن مجموعه ها است

$R(u_1,u_2,\ldots,u_n) \subset U_1\times U_2\times\ldots\times U_n$

$\mu_R \leq \mu_{U1\times U2\times\ldots\times Un}$

$U\times V=\{\mu_{11}/(u_1,v_1), \mu_{12}/(u_1,v_2), \mu_{21}/(u_2,v_1), \mu_{22}/(u_2,v_2)\}$

$R=\{\mu_{R11}/(u_1,v_1),\mu_{R12}/(u_1,v_2),\mu_{R21}/(u_2,v_1),\mu_{R22}/(u_2,v_2)\}$

$\mu_{Rij} \leq \mu_{ij}$

روابط

رابطه فازی دوری

U={ایران,فرانسه}

V={پاکستان,آلمان}

روابط فازی را می توان هم برای متغیرهای زبانی (دوری) وهم برای متغیرهای دقیق (فاصله) بکار برد

	U	u_1	u_2
V	دوری	فرانسه	ایران
v_1	آلمان	.1	.8
v_2	پاکستان	.9	.2

اگه بیشترین فاصله مرجع رو در نظر بگیریم و فاصله رو تقسیم بر اون بکنیم دوری بدست میاد.

روابط

عملیات روی رابطه های فازی – اجتماع

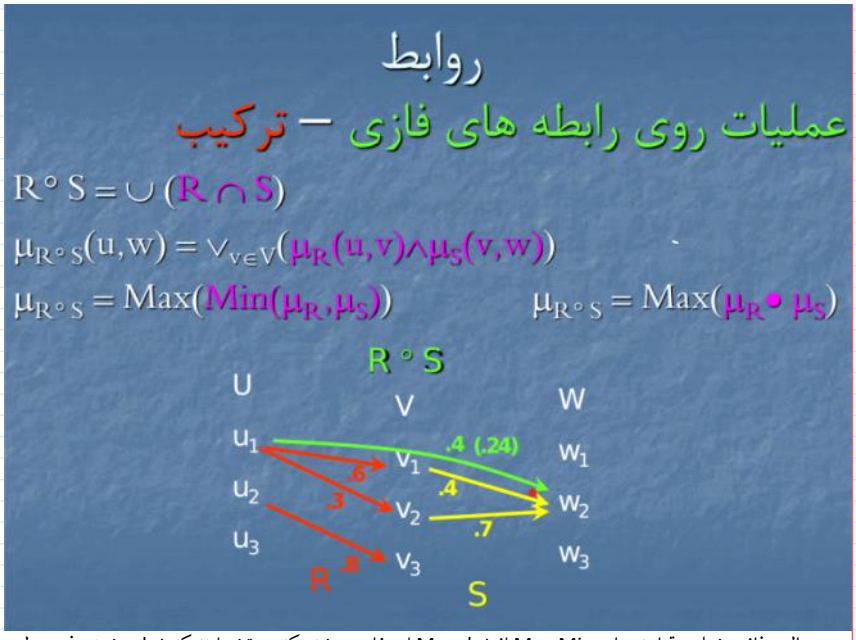
$R \cup S = \{U \mid U \in R \vee U \in S\}$

$\mu_{R \cup S}(U) = \mu_R(U) \vee \mu_S(U) = \mathrm{Max}\{\mu_R(U), \mu_S(U)\}$

عملیات روی رابطه های فازی – اشتراک

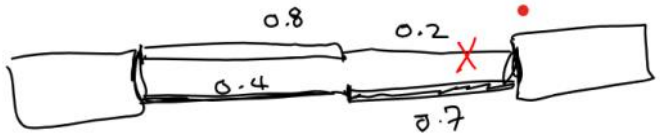
$R \cap S = \{U \mid U \in R \wedge U \in S\}$

$\mu_{R \cap S}(U) = \mu_R(U) \wedge \mu_S(U) = \mathrm{Min}\{\mu_R(U), \mu_S(U)\}$



در حالت فازی خیلی وقتا به جای Max Min از Max dot استفاده میشه. که مزیتش اینه که dot مشتق پذیره ولی minمشتق پذیر نیست.
اگر از MaxMin بریم بین 0.3 و 0.7 مینیمم بگیریم میشه 0.3 . بین 0.4 و 0.6 هم میشه 0.4. در آخر max(0.4, 0.3) = 0.4
اگرم از Max dot بریم مسیر بالایی میشه 0.24 مسیر پایینی میشه 0.21 که max اشون 0.24

فرض شود ماشین خراب شده و میخوایم بوکسور کنیم. یک سری طناب کوتاه با قطر های متفاوت داریم. اگر قطر هارو نرمال کنیم فرض کنیم یکیش 0.8 باشه یکی 0.2 یکی 0.4 و یکی دیگه 0.7. فرض کنیم 2 ماشین رو مطابق به شکل زیر وصل میکنیم:

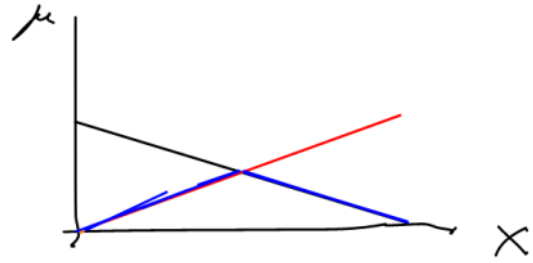


اولین طنابی که پاره میشه مسلما همین 0.2 عه. بعدش 0.4 که پاره شد ارتباط از بین میره. پس تا وقتی که بین min ها Maximum برقرار است ارتباط بین 2 ماشین برقرار است. این میشه توجیه فیزیکی Max Min. یعنی تا وقتی که Max Min این طناب ها یعنی 0.4 پاره نشده ارتباط برقراره.

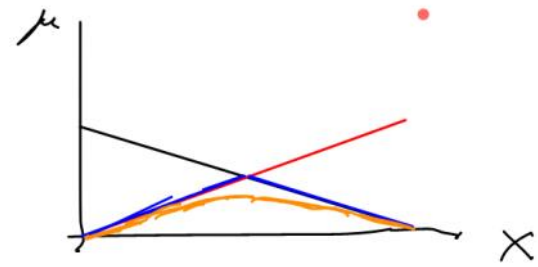
جلسه سیزدهم

3:03 PM Sunday, April 18, 2021

وقتی از مینیمم برای اشتراک استفاده کنیم شبیه شکل آبیه به دست میاد که مشتق پذیر نیست:



ولی وقتی از ضرب استفاده میکنیم همچین منحنی حاصل میشه که هموار تره و مشتق پذیر میشه:



* نزدیک ترین اشتراک به مینیمم همین ضرب هست لذا تقریب خوبی برای مینیمم هستش با توجه به این که مشتق پذیرم هست.

وقتی از مینیمم برای اشتراک استفاده میکنیم هیچ تفاوتی قائل نمیشه بزرگه چیه:

$0.2 \wedge 0.8 = 0.2$

$0.2 \wedge 0.5 = 0.2$

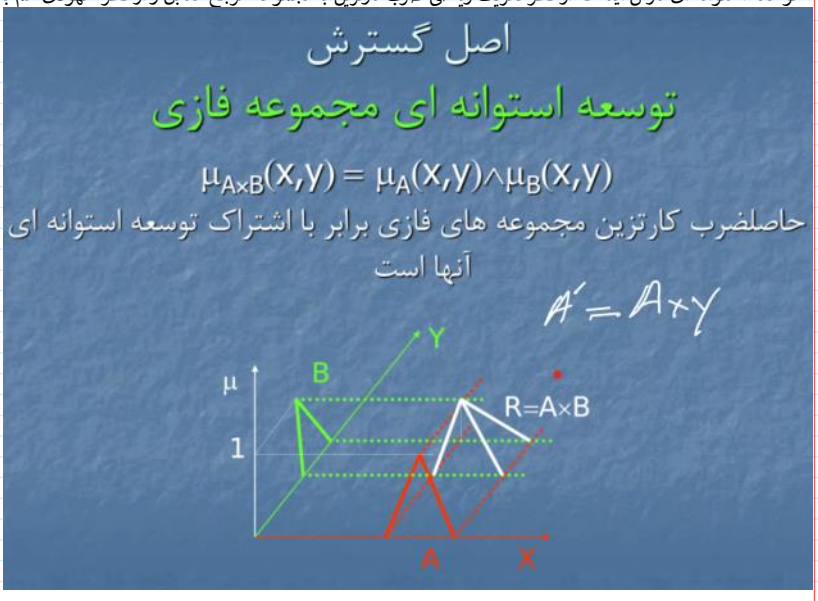
ولی وقتی از ضرب استفاده کنیم تفاوتش قابل توجه است:

$0.2 \wedge 0.8 = \cancel{0.16}\ 0.16$

$0.2 \wedge 0.5 = 0.1$

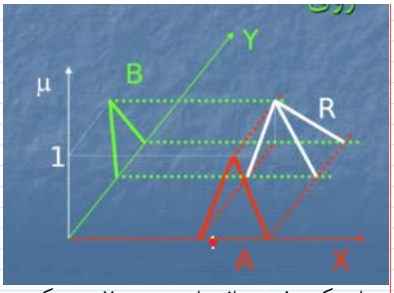
--اسلاید:

توسعه استوانه ای کارش اینه که از نظر تعریف ریاضی ضرب کارتزین با مجموعه مرجع مقابل و از نظر شهودی هم با تکرار مقدار تعلق در بعد جدید یک مقدار اضافه میکرد:

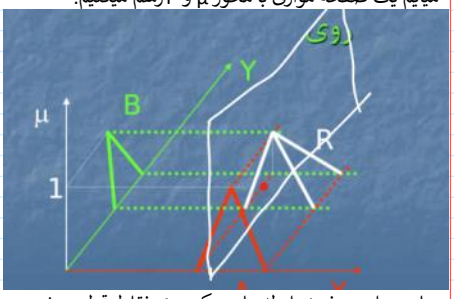


حالا چطوری بُعد رو کم کنیم؟ میتونیم تصویر کنیم. مثل این که سایه خود رو روی دیوار ببینیم بسته به این که در کدوم جهته تصویر میشه روی اون بُعد.

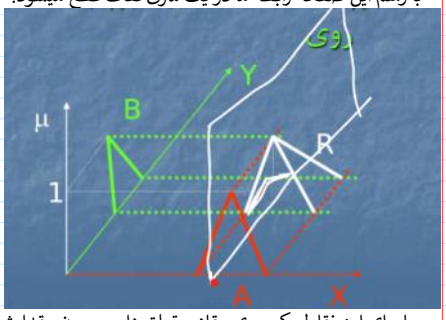
از لحاظ ریاضی فرض شود بخواهیم در یک نقطه مثل نقطه زیر در محور x تصویر رو بدست بیاریم:



میایم یک صفحه موازی با محور μ و Yرسم میکنیم:



با رسم این صفحه رابطه ما در یک سری نقاط قطع میشود:



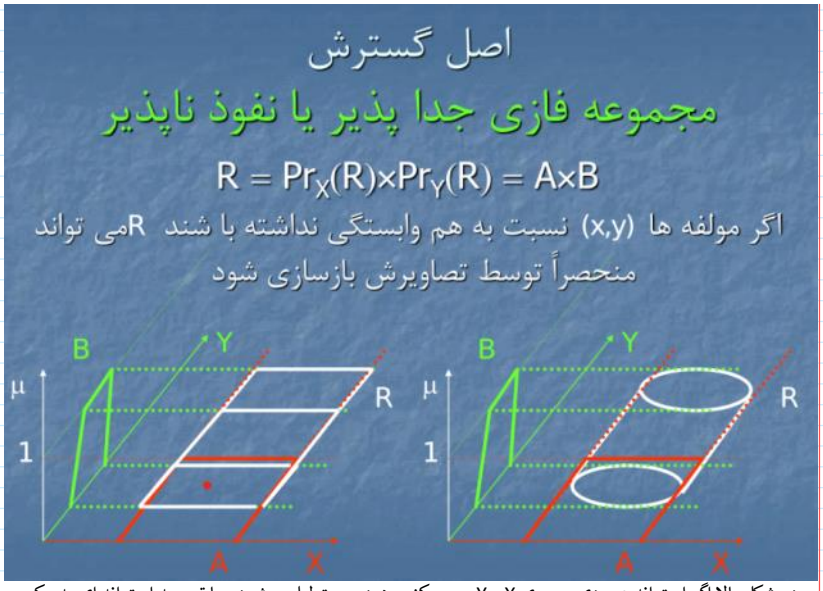
ما برای این نقاط یک سری مقادیر تعلق داریم. چون مقدارشون بیش از یدونس ما مقدار تعلق رو در این نقطه برابر با اجتماع میگیریم و اگر اجتماع رو ماکس بگیریم میشه Max این مقدار های تعلق.

اگر رابطه R داشته باشیم تصویرش روی X میشه:

$$R_x = \bigcup_{y \in Y} R$$

$$\mu_{R_x}(x) = \bigvee_{y \in Y} \mu_R(x,y)$$

* همونطور که در گرافیک اگر مثلا یک مکعب داشته باشیم که از وسطش یک استوانه حذف شده باشد تصویر کنیم روی محور x و y میشه دو تا مربع و نمیشه از تصویرشون شکل اصلی رو ساخت (چون دیگه حفره معلوم نیست)، در رابطه ها هم زمان هایی که بین X و Y همبستگی موجود باشد نمیتوان از تصویر رابطه را بدست آورد و تنها زمانی میشه روابط رو بدست آورد که همبستگی نداشته باشند.



در شکل بالا اگر استوانه عمودی رو روی X و Y محور کنیم هردو مستطیل میشود و با توسعه استوانه ای به مکعب میرسیم که با شکل اصلی متفاوت است. در این گونه موارد میگیم رابطه اولیه جداپذیر نیست ولی اگر بتونیم رابطه اولیه رو بسازیم میگیم جداپذیر است. طبق تعریف استاد داریم:

در صورتیکه بتوان یک رابطه را از روی اشتراک توسعه استوانه ای تصاویرش بازسازی نمود آن رابطه جداپذیر است

که به جای اشتراک توسعه استوانه ای ضرب کارتزین هم میشه گفت چون یکین.

$R \qquad R_X \quad R_Y$

$R = R_X \times R_Y$ جداپذیر

پس الان استوانه مثلا میشه جدانپذیر ولی مثلا مکعب یا هرم میشه جداپذیر.

* دقت شود ما الان در مورد مجموعه فازی مرتبه اول داریم صحبت میکنیم و حق نداریم یک مکعب یا استوانه مایل داشته باشیم چون اونموقع برای یک متغیر 2 تا مقدار تعلق داره که میشه مجموعه فازی مرتبه دوم.

$A \quad B$

$\mu_A(x) \quad \mu_B(y)$ x, y مرامیتیه

$x^2+y^2=R$

$\mu_R(x,y) = \mu_A(x) \wedge \mu_B(y)$ جداناپذیر

ولی اگر وابسته نباشند:

$\mu_R(x,y) = \mu_A(x) \wedge \mu_B(y)$ جداپذیر می‌شود

$R_X = A$

$R_Y = B$

ولی در مواقعی که x,y وابسته اس دیگه R_x !=A یا R_y !=B یا هردو.

مشابه اش رو توی احتمالات هم داشتیم:

$$p(x,y)=p(x)\,p(y|x)=p(x)\,p(y) \quad \text{مستقل}$$

$$p(x)=\int p(x,y)\,dy \equiv R_x=\bigcup_{y\in Y} R$$

$$p(y)=\int p(x,y)\,dx \equiv R_y=\bigcup_{x\in X} R$$

$$\mu_{R_x}(x)=\bigvee_{y\in Y}\mu_R(x,y)$$

$$\mu_{R_y}(y)=\bigvee_{x\in X}\mu_R(x,y)$$

$$R=R_x\times R_y \quad \text{جدایی پذیر}$$

$$\mu_R(x,y)=\mu_{R_x}(x)\wedge\mu_{R_y}(y)$$

در احتمالات هم تنها اگر مستقل باشن میشه از ضریشون به اصلیه رسید.

همونطور که در جدول میبینیم وقتی خواستیم تصویر بگیریم برای هرکدوم ماکس گرفتیم در جهت اون فلش ها

اصل گسترش

توسعه استوانه ای مجموعه فازی

حاصلضرب کارتزین برابر با اشتراک توسعه استوانه ای است

		v_1	v_2	v_3
		.4	.8	1
u_1	.7	.7 .4	.7 .8	.7 1
u_2	.9	.9 .4	.9 .8	.9 1
u_3	1	1 .4	1 .8	1 1

		v_1	v_2	v_3
	R	.4	.8	1
u_1	.7	.4	.7	.7
u_2	.9	.4	.8	.9
u_3	1	.4	.8	1

الان توسعه استوانه ای میشه تکرار هرکدوم از V_i ها در ستون مورد نظر و تکرار U_j ها در سطر متناظر بعد بینشون اشتراک یعنی مینیمم بگیریم اشتراک توسعه استوانه ای حاصل میشه.
با توجه به این که رابطه به دست اومده (جدول راستی) با رابطه اصلی (اسلاید قبلی) یکی نیست ← جداپذیر نیست.

اصل گسترش

توسعه استوانه ای مجموعه فازی

اگر یک مجموعه فازی توسط توسعه استوانه ای تصاویرش قابل بازسازی باشد مجموعه جداپذیر می باشد

جدا پذیر

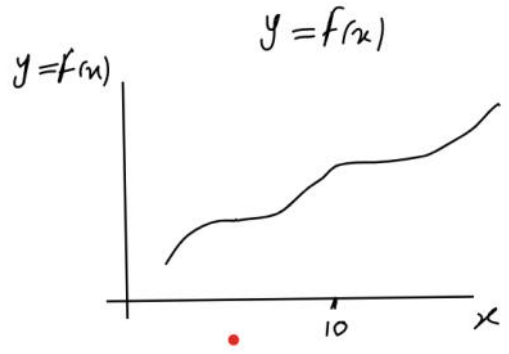
		v_1	v_2	v_3
	R	.4	.8	1
u_1	.7	.4	.7	.7
u_2	.9	.4	.8	.9
u_3	1	.4	.8	1

جدا ناپذیر

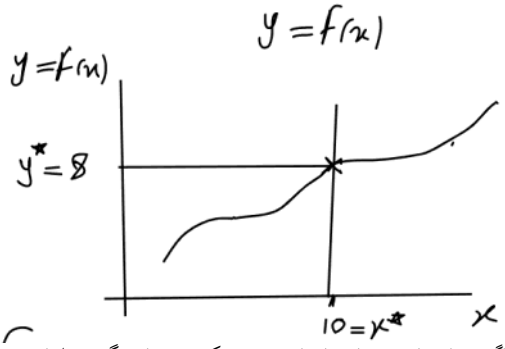
		v_1	v_2	v_3
	R	.4	.8	1
u_1	.7	.2	.7	.5
u_2	.9	.3	.8	.9
u_3	1	.4	.1	1

در جدول راستی در ستون v_1 ماکسیمم اش شد 0.4 و در سطر u_1 ماکسیمم اش میشه 0.7 و اشتراک اینا میشه 0.4 که با 0.2 برابر نیست پس جداپذیر نیست.

فرض شود y=f(x) رو داریم. حالا ما به صورت جبری روابطشو نداریم ولی به صورت هندسی همچین منحنی داره:

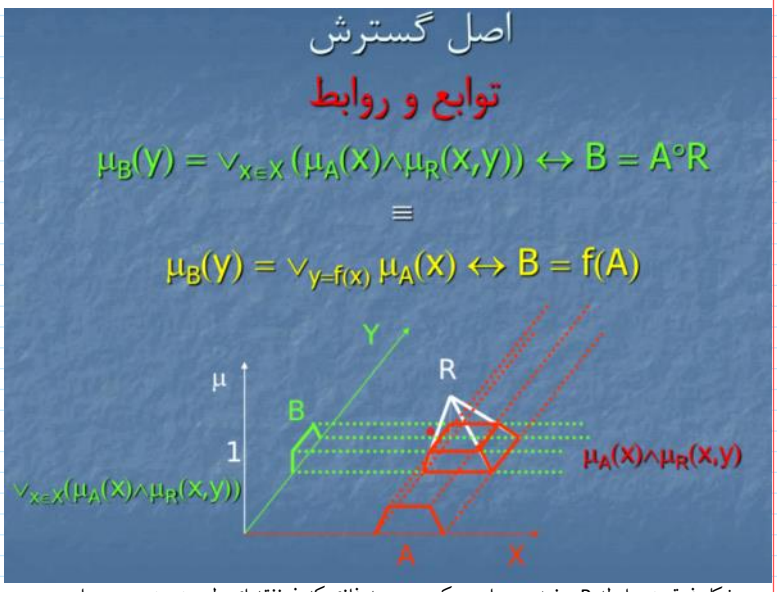


اگه بخوایم ببینیم در x=10 ببینیم مقدارش چی میشه باید از x=10 یک خط عمود رسم کنیم تلاقیش رو ببینیم چی میشه:

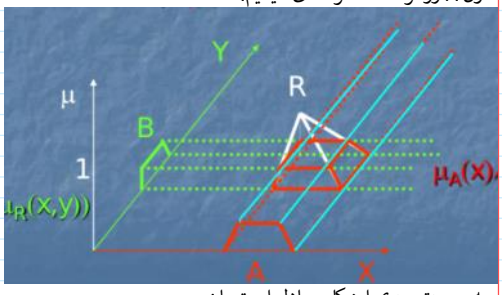


حالا اگر بخوایم با توسعه استوانه ای و تصویر کردن بخوایم بگیم. مثل این میمونه که بگیم :

1. توسعه استوانه ای x^* را روی محور y بدست می آوریم
2. اشتراک توسعه استوانه ای بدست آمده در مرحله قبل با منحنی(رابطه) بدست می آوریم.
3. تصویر اشتراک بدست آمده در مرحله قبل را روی محور y بدست می آوریم.



در شکل فوق هم رابطه R سفیده رو داریم. یک مجموعه فازی که ذوزنقه ای طور هم هست رو داریم.
اول A رو توسعه استوانه ای میدیم:

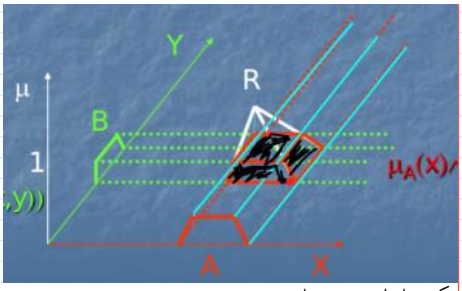


به صورت جبری این کار معادل است با :

$$A \times Y$$

$$\mu_A(x, y)$$
$$y \in Y$$

در مرحله بعد اشتراک باید حساب بشه. اشتراک این توسعه استوانه ای با رابطه میشه این تیکه سیاه رنگ که مشخص شده:



که بر اساس جبری داریم:

$$(A \times Y) \cap R$$

$$\underset{y \in Y}{\mu_A(x,y)} \wedge \mu_R(x,y)$$

حالا تصویر این اشتراک رو روی محور Y بدست میاریم. یعنی:

$$B = \bigcup_{x \in X} (A \times Y) \cap R$$

$$\mu_B(y) = \bigvee_{x \in X} \underset{y \in Y}{\mu_A(x,y)} \wedge \mu_R(x,y)$$

اگر رابطه R و مجموعه فازی A را داشته باشیم و خروجی B را بخواهیم ساده

۱- توسعه استوانه A $\quad A \times Y$

۲- اشتراک توسط استوانه‌ای با رابطه $(A \times Y) \cap R$

۳- تصویر اشتراک روی محور y $\quad B = \bigcup_{x \in X} (A \times Y) \cap R$

که این کار معادل ترکیب A و R می باشد

$$B = \bigcup_{x \in X} (A \times Y) \cap R = A \circ R$$ ترکیب

A	B	R
ورودی	خروجی	رابط بین مرحله‌ی ورودی و خروجی

$A \quad R \qquad B = A \circ R$

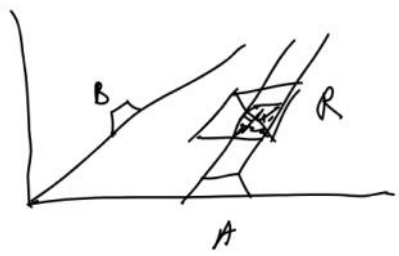
$B \quad R \qquad A = R \circ B$

$A \quad B \qquad R = A \times B$

دقت شود R در آخرین خط نوشته شده تنها یکی از روابط می باشد.

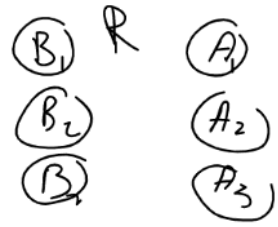
جلسه چهاردهم

در جلسه پیش دیدیم که:

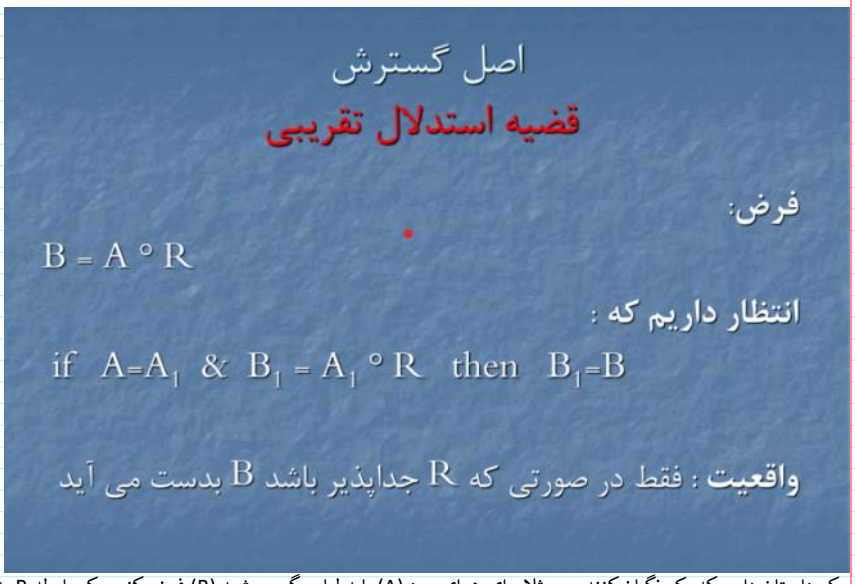


$$A \circ R = B$$
$$R \circ B = A$$

اگر یک سری مجموعه فازی داشته باشیم مثلا فرض شود اگر هوا سرد باشد (A_1) آنگاه باید باید لباس گرم پوشید (B_1). بعد یک سری دیگه قاعده هم داشته باشیم در جلوتر یاد میگیرم چطوری ازشون رابطه Rرو بسازیم:



اگر R بدست بیاریم انگار یک شبکه عصبی به دست آوردیم که با هر ورودی میتونیم خروجی مورد نظر رو بدست بیاریم.



یک داستان داریم که یکم نگران کنندس. مثلا برای هوای سرد (A) باید لباس گرم پوشید (B) فرض کنیم یک رابطه R بدست بیاریم. برای برخی قانون ها R میشه A×B. انتظار ما اینه وقتی A رو با R ترکیب کنیم B بشه ولی همیشه اینطوری نیست.

تنها در صورتی ممکنه که R جداپذیر باشه. در غیر این صورت یک B_1 ای میشه مثلا. اما این B_1 یک تقریب خیلی خوبی از B است.

$A \quad B \qquad \boxed{R = A \times B}$

R

$A \circ R \neq B$

$A \circ R = B$ اگر R جداپذیر باشد

$A \circ R = B_1$

میتوان اثبات کرد اگر A_1ای را با R ترکیب کنیم و B_1 بدست بیاد. هرچی A_1 به A نزدیک تر بشه B_1 هم به B نزدیک میشه. اگر A_1 برابر با A باشه B_1 تقریبا برابر با B میشه و با هیچ روش دیگری تقریب به این خوبی نمیشه بدست آورد.

$A_1 \circ R = B_1 \qquad \boxed{A_1 \circ R = B_1}$

$A_1 \to A \qquad B_1 \to B$

$A_1 = A \qquad B_1 \approx B$

همین حرفای استاد رو اگه در اسلاید ببینیم:

اصل گسترش

قضیه استدلال تقریبی

می توان اثبات کرد: اگر R جداپذیر هم نباشد، هرچه ورودی به A نزدیکتر شود، خروجی هم به B نزدیکتر خواهد شد

If A1 ⟶ A then B1 ⟶ B

نتیجه: ترکیب ورودی و R در هر حالت بهترین تقریب ممکن را ارائه خواهد کرد

اصل گسترش

تعمیم برای توابع چند متغیره

$\mu_B(y) = \vee(\mu_{A1\times\ldots\times An}(x) \wedge \mu_R(x_1,\ldots x_n,y)) \leftrightarrow B = (A_1\times\ldots\times A_n)^\circ R$

$\mu_B(y) = \vee_{y=f(x1,\ldots,xn)}\, \mu_{A1\times\ldots\times An}(x) \leftrightarrow B = f(A_1\times\ldots\times A_n)$

$A_1=\{.2/1,1/2,.1/3\}$ $A_2=\{.1/4,1/5,.2/6\}$

$y=x^2$ $B = ?$

$A_1\times A_2 =\{.1/(1,4),.2/(1,5),.2/(1,6),.1/(2,4),1/(2,5),.2/(2,6),$
$.1/(3,4),.1/(3,5),.1/(3,6)\}$

$y=x_1+x_2$ $B = ?$

حالا بخوایم یک مثال بزنیم ببینیم چطور میشه استفاده کرد. فرض شود ما یک $y = f(x) = x^2 - 2x + 2$ داریم. اگر x برابر با مجموعه فازی A باشه که :

$$y = f(x) = x^2 - 2x + 2$$

$$x = A = \left\{\frac{0.2}{1} + \frac{1}{2} + \frac{0.7}{3}\right\}$$

$$B = f(A) \qquad A^\circ R = B$$

حالا R رو چطوری بدست بیاریم؟ میتونیم تابع رو به صورت یک رابطه بنویسیم. با توجه به این که A در 1 و 2 و 3 هستش میریم جایگذاری میکنیم در f(x) و 1 و 2و 5 بدست میاد و داریم:

$$y = f(x) = x^2 - 2x + 2$$

$$x = A = \left\{\frac{0.2}{1} + \frac{1}{2} + \frac{0.7}{3}\right\}$$

$$B = f(A) \qquad A^\circ R = B$$

$$\begin{array}{c|ccc} & 1 & 2 & 5 \\ \hline 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{array}$$

و B برابر است با:

$$[0.2 \quad 1 \quad 0.7]\begin{bmatrix} \cancel{1}\,0.2 & 0 & 0 \\ \cancel{0.1}\,0 & 1 & 0 \\ \cancel{0}\,0 & 0 & 1 \end{bmatrix} =$$

$$B = \left\{\frac{0.2}{1} + \frac{1}{2} + \frac{0.7}{5}\right\}$$

حال اگر بخوایم مساله رو یکم پیچیده ترش بکنیم، میتونیم بگیم:

$$f(x_1, x_2) = x_1^2 x_2 - x_1 x_2^3 + x_1 - x_2$$

$$x_1 = A_1 = \left\{ \frac{0.2}{-1} + \frac{0.4}{0} + \frac{0.5}{1} \right\} \qquad x_2 = A_2 = \left\{ \frac{0.8}{0} + \frac{0.7}{1} \right\}$$

$$y = f(A_1, A_2)$$

برای راه حل اول باید A رو بدست بیاریم:

$$f(x_1, x_2) = x_1^2 x_2 - x_1 x_2^3 + x_1 - x_2$$

$$x_1 = A_1 = \left\{ \frac{0.2}{-1} + \frac{0.4}{0} + \frac{0.5}{1} \right\} \qquad x_2 = A_2 = \left\{ \frac{0.8}{0} + \frac{0.7}{1} \right\}$$

$$B = y = f(A_1, A_2) \qquad B = A \circ R \qquad A = A_1 \times A_2$$

$$A = A_1 \times A_2 = \left\{ \frac{0.2}{-1,0} + \frac{0.2}{-1,1} + \frac{0.8}{0,0} + \frac{0.7}{0,1} + \frac{0.5}{1,0} + \frac{0.5}{1,1} \right\}$$

چون میخوایم به روش ماتریسی حل کنیم، رابطه ای باید بسازیم که ستون های A ما با سطر های R ما هم جنس باشن.

$$f(x_1, x_2) = x_1^2 x_2 - x_1 x_2^3 + x_1 - x_2$$

$$\left\{ \frac{0.2}{-1,0} + \frac{0.2}{-1,1} + \frac{0.8}{0,0} + \frac{0.7}{0,1} + \frac{0.5}{1,0} + \frac{0.5}{1,1} \right\}$$

$$A = A_1 \times A_2$$

	-1	0	1
-1,0	1	0	0
-1,1	0	1	0
0,0	0	1	0
0,1	1	0	0
1,0	0	0	1
1,1	0	1	0

$$B = A \circ R \qquad R$$

$$F(x_1,x_2) = x_1^2 x_2 - x_1 x_2^3 + x_1 - x_2$$

$$\left\{ \frac{0.2}{-1,0} + \frac{0.2}{-1,1} + \frac{0.8}{0,0} + \frac{0.2}{0,1} + \frac{0.5}{1,0} + \frac{0.5}{1,1} \right\}$$

$$A = A_1 \times A_2$$

	-1	0	1
-1,0	0.2	0	0
-1,1	0.2	1	0
0,0	0.8	1	0
0,1	0.2	0	0
1,0	0.5	0	1
1,1	0.5	1	0

$$B = \left\{ \frac{0.7}{-1} + \frac{0.8}{0} + \frac{0.5}{1} \right\}$$

$$B = A^{\circ} R$$

R

==========

در حالت کلی داریم:

$$B = A^{\circ} R$$

$$B_1 \times B_2 \times \cdots \times B_m$$

$$A_1 \times A_2 \times \cdots \times A_n$$

R $\quad A_1 \times \cdots A_n \quad B_1 \times \cdots \times B_m$

ولی مشکل اینجاس اگر ابعاد خیلی بزرگ باشه R زیاد میشه. مثلا اگر هرکدوم از A_i ها و B_i ها 10 تا داشته باشد رابطه R $10^n \times 10^m$ تا عضو داره.

اگر پیوسته باشه دیگه از این روش نمیشه حل کرد. در حالت پیوسته 2 حالت داریم:

- ورودی خروجی گسسته بشه: میشه همین که تا الان بررسی کردیم. ولی این هزینه محاسباتی بسیار زیادی داره
- مقدار تعلق گسسته بشه: هزینه محاسباتی خیلی میاد پایین. بهش میگن برش lambda که بعیده تو این درس در موردش صحبت بشه

منطق:

موضوع صحبت منطق، راجع به درستی و نادرستی گزاره ها می باشد. اگه درست باشد 1 و غیر این صورت 0 است.
مثلا در یک گزاره ساده میگویی "حسن برادر علی است".

$$T(p) = \begin{cases} 1 & p \text{ درست باشد} \\ 0 & p \text{ نادرست باشد} \end{cases}$$

p : حسن برادر علی است

برای هر گزاره میتوان یک مجموعه و یک عضو تعریف کرد و گزاره رو تحت عنوان تعلق داشتن یک عضو به یک مجموعه بیان کرد.

p : حسن برادر علی است

$p: a \in A$

$A =$ برادران علی

$a =$ حسن

$$T(p) = \mu_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

پس منطق حرف جدیدی نمیزند و منطق خودش همون مجموعه است.

جلسه پانزدهم

گزاره های منطقی

منطق کلاسیک

P: x is in A Q: y is in B

If $x \in A$ then $T(P)=1$ else $T(P)=0$

If $y \in B$ then $T(Q)=1$ else $T(Q)=0$

$P \vee Q : x \in A \text{ or } y \in B \equiv T(P \vee Q) = \mathrm{Max}(T(P),T(Q))$

$P \wedge Q : x \in A \text{ and } y \in B \equiv T(P \wedge Q) = \mathrm{Min}(T(P),T(Q))$

$P' : x \notin A \equiv T(P') = 0$

$P \rightarrow Q : x \notin A \text{ or } y \in B \equiv T(P \rightarrow Q) = T(P' \vee Q)$

$P \leftrightarrow Q : \equiv T(P \leftrightarrow Q)=1 \text{ if } T(P)=T(Q) \text{ else } T(P \leftrightarrow Q)=0$

دیدم یک گزاره به صورت تعلق یک عضو به یک مجموعه است که درست بودنش وقتیه که عضو هست

قانون های منطق یا شرطی هستن یا دو شرطی. شرطی رو بخوایم تو مجموعه بگیم:

$$P \Rightarrow Q$$
$$A \Rightarrow B$$

که همین با اجتماع بخوایم بگیم میشه:

$$A' \cup B$$

و دو شرطی هم بخوایم بگیم:

$$P \Leftrightarrow Q \qquad (A' \cup B) \cap (A \cup B')$$

گزاره های منطقی

رابطه قانون شرطی کلاسیک

$P \rightarrow Q \equiv$ If P: $x \in A$ Then Q: $y \in B \equiv A' \cup B$

$R=(A \times B) \cup (A' \times Y) \equiv$ If A Then B

$R=(A \times B) \cup (A' \times C) \equiv$ If A Then B Else C $\equiv$

If A Then B And If A′ Then C

در واقع P آنگاه Q یک رابطه است. (چرا؟)

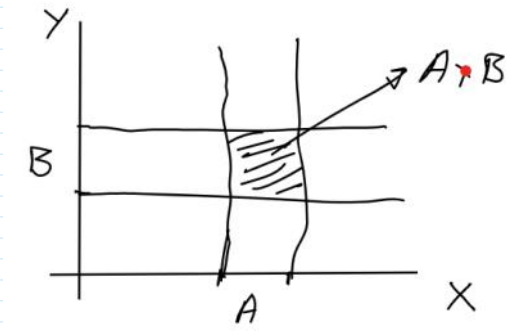
به عبارت دیگه وقتی میگیم A آنگاه B همان:

$$A \Rightarrow B \equiv A' \cup B$$

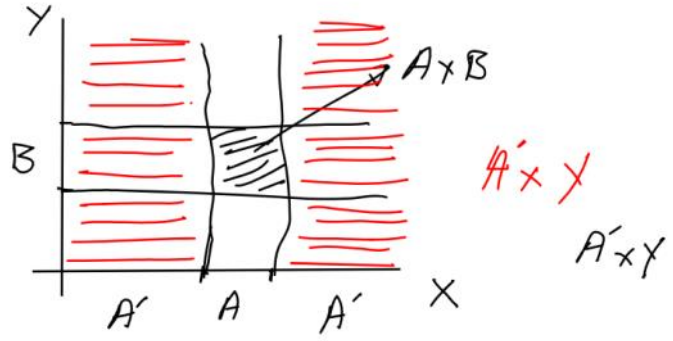
هستش. پس چرا یک رابطه درست میشه؟ چون مجموعه مرجع A و B متفاوت است. اگر 2 مجموعه داشته باشیم با هم اشتراک بگیریم اگه مجموعه مرجع یکی باشه بعد تغییر نمیکنه ولی اگه مجموعه مرجع اشون متفاوت باشه چون هم مرجع میکنیم ابعاد اضافه میکنه و یک رابطه میشه و رابطه هم زیر مجموعه از حاصل ضرب کارتزین بود.

$$A' \cup B = (A' \times Y) \cup (X \times B)$$

یکی از روش های بدست آوردن رابطه بر اساس توجیهاتی است که از یک سری قانون میکنیم. مثلا اگر قانون "اگر هوا سرد باشد آنگاه لباس گرم میپوشیم"، اگر بهش تحت عنوان منطق ارسطویی نگاه کنیم معنیش میشه که یا هوا سرد است یا نیست. اگر باشه توجیه قانون اینه که لباس گرم بپوشیم. یعنی اگر A برقرار باشه توسعه استوانه ای میدیم. B هم که لباس گرم مپوشیم رو توسعه استوانه ای میدیم و اشتراکشون میشه:



حالا اگر هوا سرد نبود چیکار کنیم؟ اون موقع نمیتونیم بگیم چیکار میشه کرد، بلکه هرکار بخوایم میتونیم بکنیم. یعنی میشه این قسمتا ک قرمز میشه کرد:



بنابراین رابطه برابر میشه با:

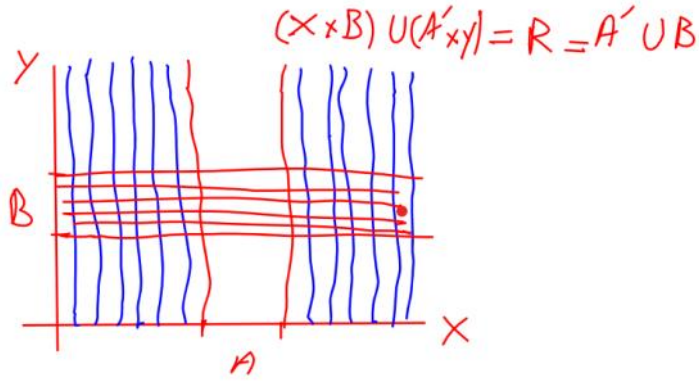
$$R = (A \times B) \cup (A' \times Y)$$

پس این رابطه ای است که بر اساس A آنگاه B در منطق کلاسیک بهش رسیدیم. یک رابطه دیگه هم از تعریف A آنگاه B بود:

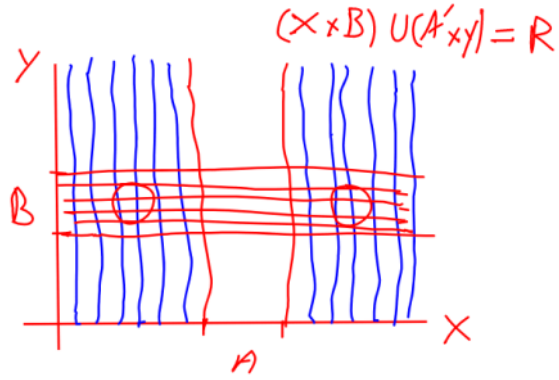
$$R_1 = (A \times B) \cup (A' \times Y)$$
$$R_2 = A' \cup B = (A' \times Y) \cup (X \times B)$$

حالا آیا به نظرتون رابطه اول و دوم با هم تفاوت دارن یا ندارن؟ برای این که بهتر بفهمیم این موضوع رو میریم R_2 هم نمودارشو میکشیم:

۱



تفاوت این 2 تا شکل چیه؟ یک فرق کوچولو فقط دارن. این که نگاه کنین این 2 تا جا که دایره کشیدیم هم قرمز داریم هم آبی:



ولی توی قبلی همه جا یدونه بیشتر نیست. یعنی در اجتماع بعضی جاها شده aVa و چون در حالت کلاسیک aVa = a این 2 تا شکل فرق ندارن.
پس توی حالت کلاسیک برابرن، توی بعضی حالت های نرمال هم برابرن مثل Max که بگیریم باز a میشه. ولی توی بعضی جاها مثل جمع و ضرب جبری داریم:

$$aVb = a+b-ab$$

$$aVa = 2a - a^2$$

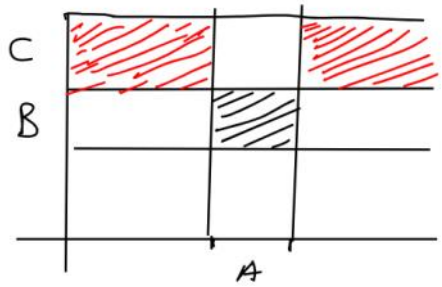
پس اینجاها این 2 رابطه ها با هم یکی نیستن. در عمل هیچکدوم مناسب نیستن. چرا؟ چون مثلا اگر بخوایم یک ربات بسازیم ک بر اساس این منطق استفاده بکنه داریم میگیم وقتی برقرار نبود هرکاری میخوای بکن ولی این آیا قابل قبول هست!؟

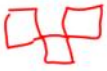
بنابراین 2 قانون دیگه تعریف میکنیم.
اول این که وقتی شرط برقرار نبود باید بگیم چیکار کنه

$$A \Rightarrow B \text{ else } C$$

حالا اگر بخوایم به صورت شکلی اینو در بیاریم و رابطه جبریشم بنویسیم اینطوری میشه که:

$A \Rightarrow B$ else C $\qquad R_3 = (A \times B) \cup (A' \times C)$

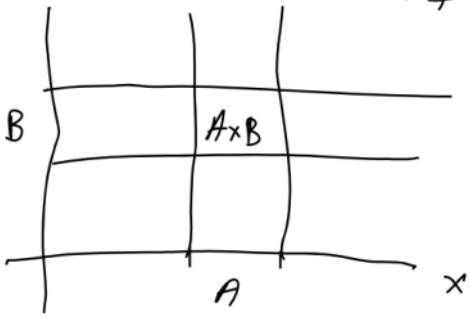




اگر قانون رو به صورت درکی که روزمره ازش داریم استفاده داریم، اینطوریه که اگر شرط برقرار بود انجام بده ولی اگر نبود هیچ کار نکن، این که هیچ کار نکن میشه اجتماع با تهی که میشه:

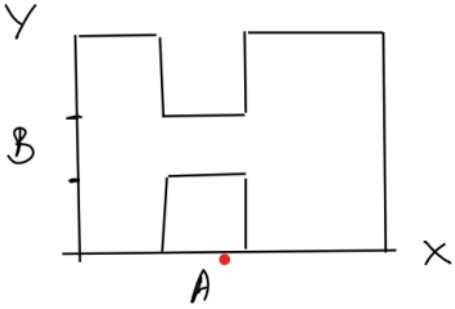
$R_4 = (A \times B) \cancel{\cup (A' \times \phi)}$

$R_4 = A \times B$



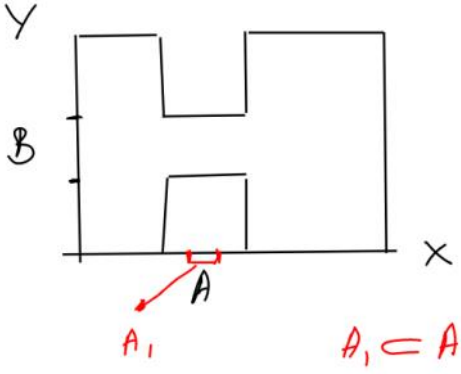
* این حرفا همشون به حالت انیمیشینی در اسلاید ها موجوده

تا الان دیدیم که:



$R_1 = R_2$

حالا میخوایم ببینیم بسته به ورودی خروجی چی میشه. فرض شود که یک A_1 داشته باشیم:



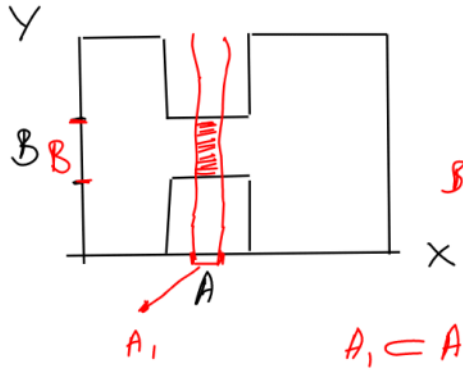
$A_1 \subset A$

اگر A_1 ورودی باشه خروجی میشه به 2 روش محاسبه کرد:

یکیش اینه:

$$B_1 = A_1 \circ R$$

پس اول A_1 رو توسعه استوانه ای میدیم که با این کار انگار داریم در y ضرب میکنیم. بعد اشتراک با B رو میگیریم و در آخر تصویر میکنیم رو B



$$R_1 = R_2 = R$$

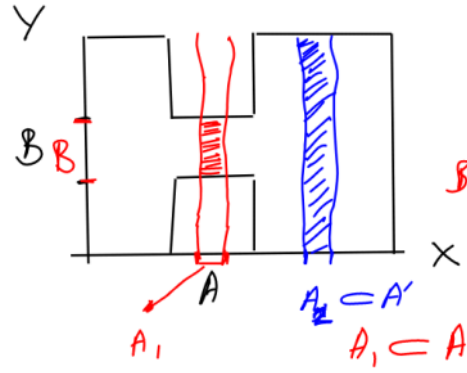
$$B = \bigcup (A_1 \times y) \cap R$$

$$A_1 \subset A$$

$$B_1 = A_1 \circ R = B$$

پس B_1 برابر شد با B. نتیجه اخلاق این چیه؟ در حالت کلاسیکی حتی ما زیر مجموعه از A رو داشته باشیم باز میتونیم B رو بدست بیاریم و داستان جداپذیر و جداناپذیر بودن رو نداریم.

الا اگر A_2 ما زیرمجموعه از A' باشه خروجی چی میشه؟



$$R_1 = R_2 = R$$

$$B = \bigcup (A_1 \times y) \cap R$$

$$A_2 \subset A'$$

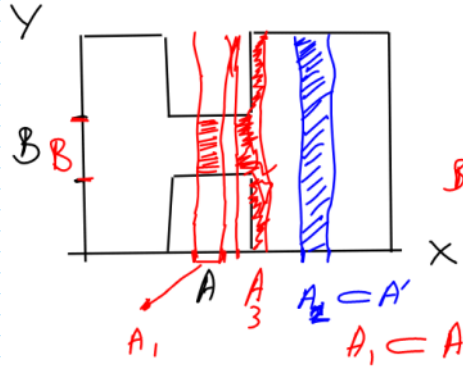
$$A_2 \circ R = Y$$

$$A_1 \subset A$$

$$B_1 = A_1 \circ R = B$$

میشه مجموعه مرجع y، یعنی وقتی شرط برقرار نبود هرکاری میخوای بکن.

اگر A_3 هم داشته باشیم مرزش بیافته، اون موقع B_3 میشه y:



$$R_1 = R_2 = R$$

$$B = \bigcup (A_1 \times y) \cap R$$

$$A_2 \subset A'$$

$$A_2 \circ R = Y$$

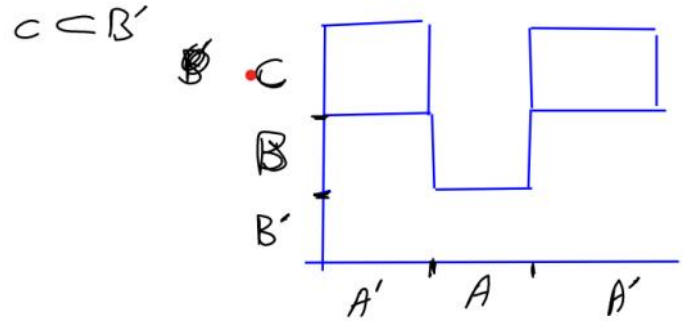
$$A_1 \subset A$$

$$B_1 = A_1 \circ R = B$$

$$A_3 \circ R = Y$$

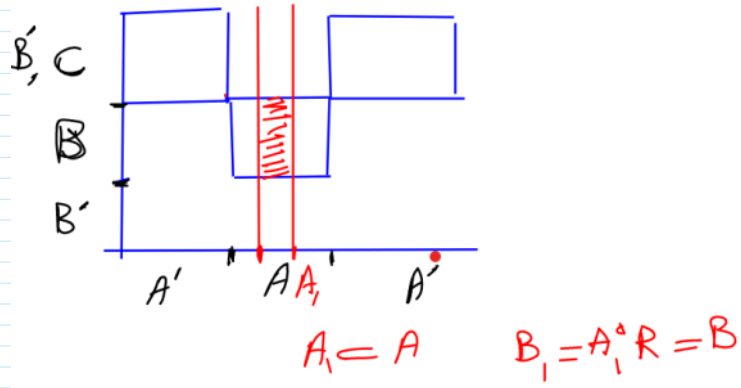
در منطق فازی سه بعدی میشه. اگه حالا بر اساس جداپذیر بودن یا نبودن داستان فرق میکرد که جلسات قبل دیدیم.

حالا برگردیم سراغ رابطه دوم که گفتیم اگر A آنگاه B در غیر این صورت C:

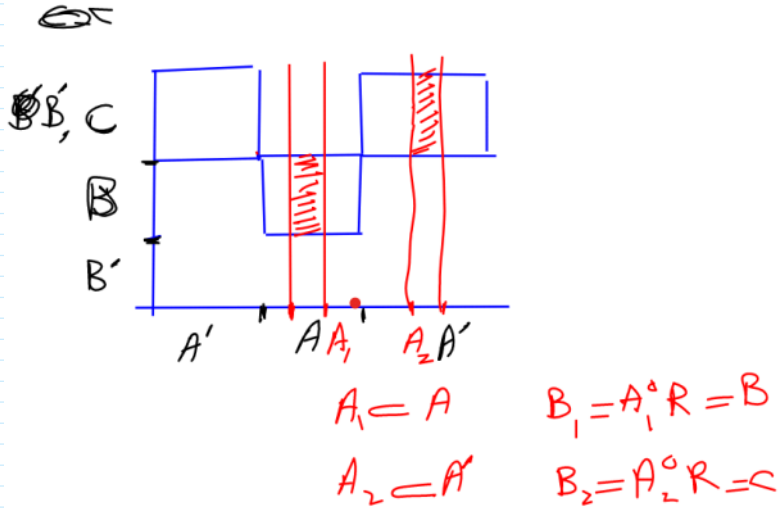


حالا اینجا بیایم چند حالت چک کنیم:

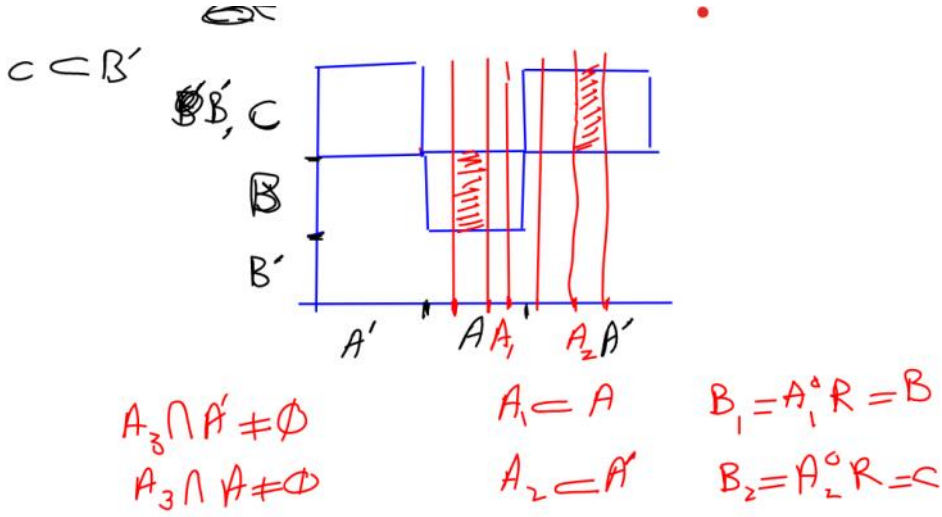
حالت اول این که A_1 زیر مجموعه A باشه. در این صورت B_1 چی میشه؟



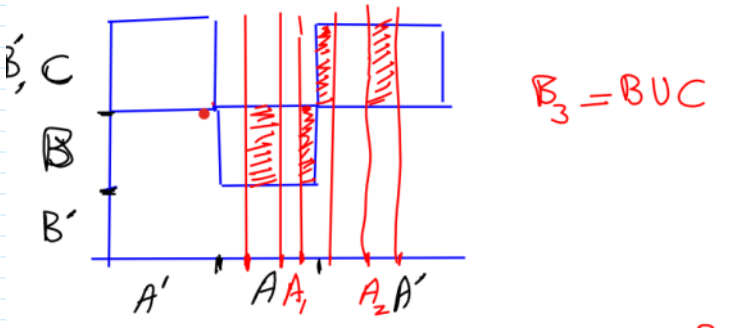
حالا اگر A_2 ما زیر مجموعه A' باشه، B_2 میشه:



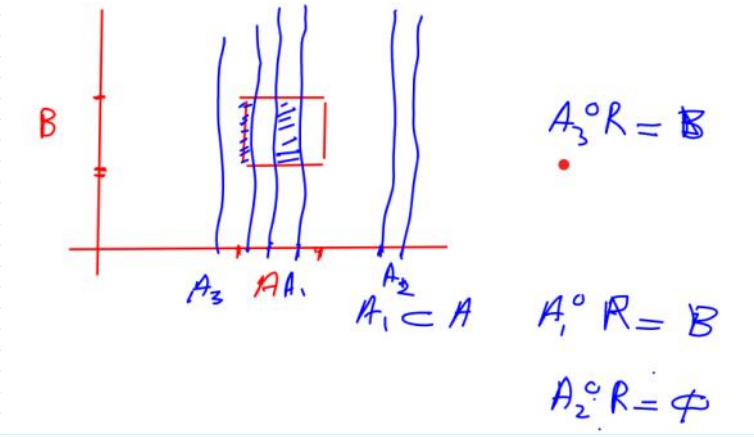
حالا فرض کنیم یک A_3 هم تعریف کنیم که توی مرز بیافته، اون موقع داریم:



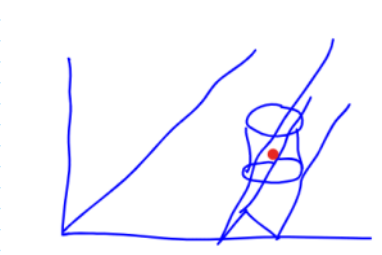
در این اصورت توسعه استوانه ای A_3 رو انجام بدیم و بعد تصویر کنیم B_3 میشه:



و در آخر حالتی رو در نظر بگیریم که حالت پیش فرضمون بود، این که وقتی شرط برقرار نبود هیچکار نکنیم. در این حالت داریم:



در حالت فازی هم روش همینه فقط شکلا سه بعدیه. مثلا رابطمون استوانه باشه و ورودیمون یک مثلث که توسعه استوانه ای بدیم نبشی بشه:



اشتراکشو بگیریم میشه مخروط ولی اگه همینو تصویر کنیم اونور میشه مستیطیل. این رابطه الان جداپذیر نیست چون تصویرارو توسعه استوانه ای بدیم استوانه نمیشه.

جلسه شانزدهم

گزاره های منطقی

منطق فازی

P: x is A Q: y is B

$T(P)=\mu_A(x)$ $T(Q)=\mu_B(y)$

$P \vee Q$: x is A or y is B $\equiv T(P \vee Q)=Max(T(P),T(Q))$

$P \wedge Q$: x is A and y is B $\equiv T(P \wedge Q)=Min(T(P),T(Q))$

P' : $T(P')=1-T(P)$

$P \rightarrow Q$: x is A then y is B $\equiv T(P \rightarrow Q) = T(P' \vee Q)$

در حالت فازی هم مثل قبلیه فقط مقدار بین 0 و 1 داره

در جلسه قبل دیدیم:

$$A \Rightarrow B$$

$$R_1 = A' \cup B$$

$$R_2 = (A \times B) \cup (A' \times y)$$

$$R_3 = A \times B$$

$$A \Rightarrow B \text{ else } C$$

$$R_4 = (A \times B) \cup (A' \times C)$$

اگر بخوایم با مقدار تعلق بگیم:

رابطه اول:

$$\mu_{R_1}(x,y) = \mu_{A'}(x) \vee \mu_B(y)$$

که اگه اجتماع رو با max تعریف کنیم میشه

$$Max\{1-\mu_A(x), \mu_B(y)\}$$

در رابطه دوم هم:

$$\mu_{R_2}(x,y) = (\mu_A(x) \wedge \mu_B(y)) \vee \mu_{A'}(x)$$

چون y میشه 1 وقتی ضرب بشه دیگه میره

که اینم برابره با:

$$\mu_{R_2} = Max\{Min\{\mu_A(x), \mu_B(y)\}, 1-\mu_A(x)\}$$

برای رابطه سوم:

$$\mu_{R_3}(x,y) = \mu_A(x) \wedge \mu_B(y) = Min(\mu_A(x), \mu_B(y))$$

برای رابطه چهارم:

$$\mu_{R_4}(x,y) = (\mu_A(x) \wedge \mu_B(y)) \vee (\mu_A(x) \wedge \mu_C(y))$$

$$\mu_{R_4} = Max\{Min(\mu_A(x), \mu_B(y)), Min(1-\mu_A(x), \mu_C(y))\}$$

مثال:

اگر لب ها زیاد و کثیف باشند آنگاه آب باید داغ باشد

A_1 A_2 B

$$A_1 \times A_2 \Rightarrow B$$

برای نوشتن رابطه این قانون باید از R3 استفاده بشه. **چرا؟**چون نمیتونیم بگیم وقتی شرط برقرار نبود هرکاری میخوای بکن. پس میفهمیم بسته به کاربرد باید تصمیم بگیریم از چه فرمولی استفاده کرد.

$$A_1 \times A_2 \Rightarrow B \qquad R = A_1 \times A_2 \times B$$

$$\mu_R(x_1, x_2, y) = Min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \mu_B(y)\}$$

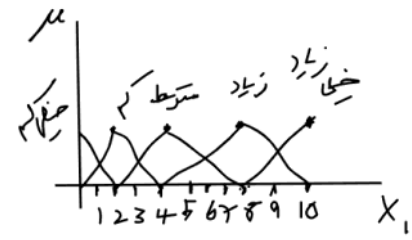
اگر به صورت مینیمم فرض کنیم مشابه فوق میشه. اگر به صورت ضرب فرض کنیم:

$$\mu_R(x_1, x_2, y) = Min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \mu_B(y)\}$$
$$= \mu_{A_1}(x_1)\, \mu_{A_2}(x_2)\, \mu_B(y)$$

پس در مساله به طور کلی داریم:

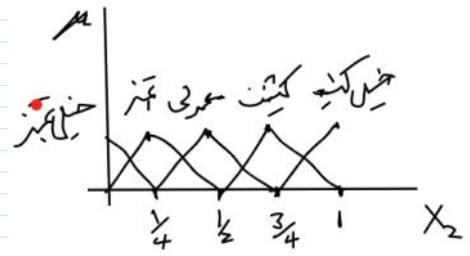
A_1 زیاد A_2 کثیف B داغ $A_1 \times A_2 \Rightarrow B$

اگر برای وزن لباس ها، خیلی زیاد، زیاد، معمولی و کم داشته باشیم، میشه گفت مثلا از 8 تا 10 کیلو خیلی زیاده. از 4 تا 10 بشه زیاد. از 2 تا 8 متوسط. 0 تا 4 کم و 0 تا 2 خیلی کم باشه.



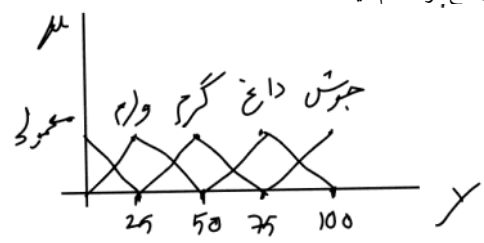
این نقاط توسط افراد خبره با آمارگیری باید تعیین بشه.

به همین ترتیب برای کثیفی میتوان گفت:



نقاط کلیدی که اون نقطه ها اون بالای نمودار هستن با آنتروپی میشه مشخص کرد.

برای داغ کردن هم میشه گفت:



حالا میایم میگیم که:

اگر لباس نسبتاً زیاد و نسبتاً کثیف باشند آب چگونه باید باشد

A_1^* A_2^* B^*

برای بدست آوردن آب که *B هست داریم:

$$B^* = A^* \circ R$$

$$B^* = (A_1^* \times A_2^*) \circ (A_1 \times A_2 \times B)$$

اگر برای نسبتا زیاد بگیم 6 تا 8 باشه. با توجه به نمودار های رسم شده و مقدارایی که دارن میشه گفت:

$$A_1^* = \left\{ \frac{0.3}{6} + \frac{1}{7} + \frac{0.4}{8} \right\}$$

برای نسبتا کثیف هم اگر داشته باشیم:

$$A_2^* = \left\{ \frac{0.1}{0.625} + \frac{0.8}{0.75} \right\}$$

حالا ابتدا بایستی حاصل ضرب کارتزین A_1^* و A_2^* رو بدست بیاریم. (دقت بشه توی قانون استار نداره ولی برای ورودی استار داره و تو امتحان حواسمون باشه اشتباه نکنیم)
سپس با مینیمم گرفتن داریم:

$$A_1^* \times A_2^* = \left\{ \frac{0.1}{6,0.625} + \frac{0.1}{7,0.625} + \frac{0.1}{8,0.625} + \frac{0.3}{6,0.75} + \frac{0.8}{7,0.75} + \frac{0.4}{8,0.75} \right\}$$

حالا میریم مقدار خروجی رو گسسته سازی میکنیم. بعد رابطه رو بر حسب مقدار تعلق مینویسم. چون در رابطه دیگه بر حسب قانونه دیگه استار نباید بذاریم! داریم:

	55	65	75	85	95
6, 0.625					
7, 0.625					
8, 0.625					
6, 0.75					
7, 0.75					
8, 0.75					

$$\mathrm{Min}\left(\mu_{A_1}(7), \mu_{A_2}(0.625), \mu_B(85)\right)$$

$$\mu_R(x_1, x_2, y) = \mathrm{Min}\left\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \mu_B(y)\right\}$$

حالا مقدار تعلق چطوری بدست بیاد؟ مثلا در خونه اول جدول رو نگاه میکنیم. $A_1 = 6$ هستش $A_2 = 0.625$ و $B = 55$. میریم تو اون نمودارا که کشیدیم میبینیم در نقطه 8 داریم $A_1 = 1$ و در 4 هم $A_1 = 0$ هستش. چون خطی داره میره بالا اون نقطه وسطش که در 6 هستش میشه 0.5. به همین ترتیب برای بقیه هم بدست میاریم و جدول رو پر میکنیم.

*برای بدست آوردن 55 در B اینطوری عمل کرد که گفت از بین 50 تا 75 ما 25 تا مقدار داریم. 55 میشه 5 از اون 25 که 5/25 میشه 25 از 100 که حالا 0.2 در نظر گرفتن.

		55 (0.2)	65 (0.6)	75 (1)	85 (0.6)	95 (0.2)
0.5	0.5	6, 0.625	0.2	0.5	0.5	0.2
0.75	0.5	7, 0.625	0.2	0.5	0.5	0.5	0.2
1	0.5	8, 0.625	0.2	0.5	0.5	0.5	0.2
0.5	1	6, 0.75	0.2	0.5	0.5	0.5	0.2
0.75	1	7, 0.75	0.2	0.6	0.75	0.6	0.2
1	1	8, 0.75	0.2	0.6	1	0.6	0.2

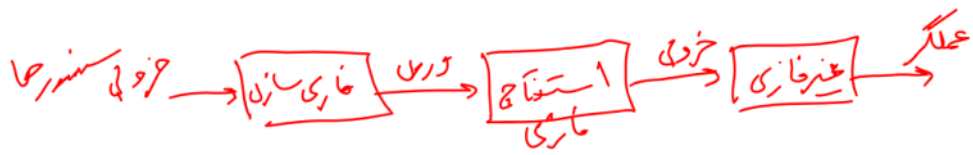
		55 (0.2)	65 (0.6)	75 (1)	85 (0.6)	95 (0.2)
0.5 0.5	6, 0.625	0.2 (0.1)	0.5 (0.1)	0.5	0.5	0.2
0.75 0.5	7, 0.625	0.2 (0.1)	0.5 (0.1)	0.5	0.5	0.2
1 0.5	8, 0.625	0.2 (0.1)	0.5 (0.1)	0.5	0.5	0.2
0.5 1	6, 0.75	0.2 (0.3)	0.5 (0.5)	0.5	0.5	0.2
0.75 1	7, 0.75	0.2 (0.8)	0.6	0.75	0.6	0.2
1 1	8, 0.75	0.2 (0.4)	0.6 (0.4)	1 (0.4)	0.6	0.2
		0.2	0.6	0.75	0.6	0.2

در نتیجه:

*

$$\tilde{B} = \left\{ \frac{0.2}{55} + \frac{0.6}{65} + \frac{0.75}{75} + \frac{0.6}{85} + \frac{0.2}{95} \right\}$$

(در جواب به پرسش بچه ها، به طور کلی اتفاق زیر قراره بیافته:

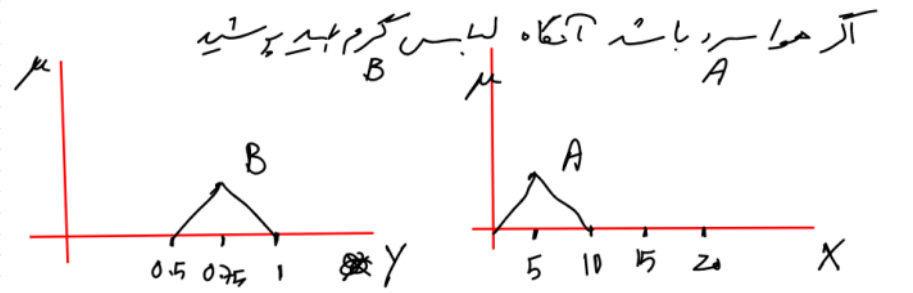


(

جلسه هفدهم

فرض کنیم قانون داریم "اگر هوا سرد باشد، آنگاه لباس گرم باید پوشید"

میایم درجه هوا رو تا 20 درجه در نظر میگیریم و برای لباس هم ضخامت رو در نظر میگیریم و نرمال میکنیم که ضخیم ترینش 1 باشه. حالا داریم:



حالا برای این قانون، رابطه اش رو با حاصل ضرب کارتزین میریم چون وقتی هوا سرد نبود نمیخوایم کاری کنیم. پس داریم:

$$A \Rightarrow B \qquad R = A \times B \qquad \mu_R(x,y) = \mathrm{Min}\{\mu_A(x), \mu_B(y)\}$$

حالا فرض شود بهمون بگن هوا 6 درجه است. با این قانون میخوایم ببینیم چه لباسی باید پوشید. **چیکار کنیم؟!** اگر ورودی A* بگیریم و خروجی B* رو بخوایم بدست بیاریم داریم:

$$B^* = A^* \circ R$$

ولی ما الان A* رو نداریم. **چطوری بدست بیاریم؟** میایم μ براش تعریف میکنیم. به شکلی که:

$$\mu_{A^*}(x^*) = \begin{cases} 1 & x^* = 6^\circ \\ 0 & x^* \neq 6^\circ \end{cases}$$

سپس داریم:

$$B^* = A^* \circ R$$

$$\mu_{B^*}(y) = \bigvee_{x \in X} \left(\mu_{A^*}(x^*) \wedge \mu_R(x,y) \right) \qquad \mu_{A^*}(x^*) = \begin{cases} 1 & x^* = 6^\circ \\ 0 & x^* \neq 6^\circ \end{cases}$$

$$\mu_{B^*}(y) = \mathrm{Max}_{x \in X} \left\{ \mathrm{Min}\{\mu_{A^*}(x^*), \mu_R(x,y)\} \right\}$$

حالا 2 حالت داریم. یا این که $\mu(A^*)$ امون 1 هست یا 0 . تو حالتی که صفر داریم مینیمم میشه 0 و تو حالتی که 1 داریم مینیمم میشه خود اون μ. پس داریم:

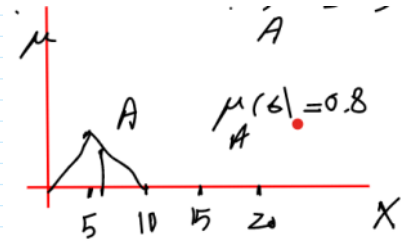
$$\mu_{B^*}(y) = \mathrm{Max}_{x \in X} \left\{ \underbrace{\mathrm{Min}\{\mu_{A^*}(x^*), \mu_R(x,y)\}} \right\} = \mathrm{Max}\{0, \mu_R(6,y)\}$$

$$\mu_{B^*}(y) = \mu_R(6,y)$$

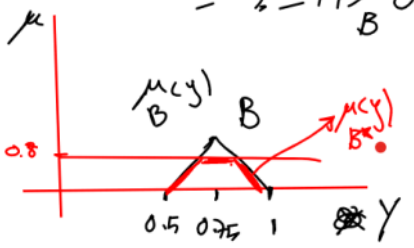
خود μ_R هم برابره با:

$$\mu_{B^*}(y) = \mu_R(6,y) = \mathrm{Min}\{\mu_A(6), \mu_B(y)\}$$

برای بدست آوردن مقادیر بر میگردیم به شکلا:



ولی باید مینیمم بگیریم با $\mu_B(y)$ که میشه:



پس ما با یک ورودی غیر فازی یک خروجی فازی بدست آوردیم. حالا بخوایم این خروجی فازی رو غیر فازی کنیم مثلا میشه مرکز ثقل رو حساب کرد که چون ذوزنقه و متقارن هست میشه 0.75 در نظر گرفت.پس:

$$\mu_{B^*}(y) = \mathrm{Min}\{\mu_A(6), \mu_B(y)\}$$

$$y^* = 0.75$$

یک نکته جالب در این مثال اینه که اگر ورودی به جای مجموعه فازی یک مقدار دقیق باشه میشه ماکسیسمم رو حذف کرد. چون بین 0 و یک مقدار باید ماکسیمم گیری بشه و دیگه ماکسیمم حذف میشه. **دلیلش چیه؟** چون برای اشتراک از مینیمم استفاده کردیم و اگه ضرب استفاده بشه داستان متفاوت میشه.

$$\mathrm{Max}\{0.2, 0.8\} = 0.8$$

$$0.3 \quad 0.8 = 0.8$$

$$x \cdot y \quad \begin{cases} a \cdot b \\ a + b - ab \end{cases}$$

وقتی از مینیمم و ماکسیمم برای اجتماع و اشتراک استفاده کنیم دیگه تفاوتی بین 0.2 و 0.3 ایجاد نمیشه و نتیجه نهایی هم یکی میشه و برای این که یک تفاوتی قائل بشیم مثلا باید از ضرب یا a+b-ab استفاده کرد.

حالا اگر به جای یک قانون تعداد قانونامون زیاد بود چیکار کنیم؟ مثلا اگر اینا قانون هامون باشن (با فرض این که A_1 تا A_n مجموعه مرجع یکسانی داشته باشن):

$$A_1 \Rightarrow B_1$$

$$\vdots$$

$$A_i \Rightarrow B_i$$

$$\vdots$$

$$A_N \Rightarrow B_N$$

حالا برای این که اشتباه برداشت نشه اینارو اون بالا بذاریم که با ابعاد اشتباه نگیریم:

$$A^1 \Rightarrow B^1$$
$$\vdots$$
$$A^i \Rightarrow B^i$$
$$\vdots$$
$$A^N \Rightarrow B^N$$

و بهمون A^* بدن بگن B^* چی میشه. **چیکار کنیم!؟** ما برای هرقانون یک رابطه ای داریم که B^* داخل هرکدوم به این روش بدست میاد:

$$A^1 \Rightarrow B^1 \qquad R^1 \qquad B^{*1} = A^* \circ R^1$$
$$\vdots$$
$$A^i \Rightarrow B^i \qquad R^i \qquad B^{*i} = A^* \circ R^i$$
$$\vdots$$
$$A^N \Rightarrow B^N \qquad R^N \qquad B^{*N} = A^* \circ R^N$$

که هریک از B^{*i} هایی که بدست میاد یک خروجی جزئی هستش.
برای بدست آوردن خروجی کلی چیکار کنیم؟ بستگی داره بین قوانین چی باشه. اگر بین قوانین اشتراکه میایم اشتراک میگیریم و اگه بینش اجتماعه میایم اجتماع میگیریم.

مثلا اگر بگیم: "اگر هوا سرد باشد لباس گرم بپوشیم، اگر گرم باشه لباس خنک." **این قانون بینش اشتراکه یا اجتماع!؟** بینشون اجتماع هست چون هرکدوم حالت خاصی رو پوشش میدن.
ولی وقتی بگیم برای این که لباس گرم بپوشیم باید هوا سرد باشه و تو خونه نباشیم. اون موقع دیگه بینشون اشتراک میشه

حالا اگر فرض کنیم بین قوانین اجتماع باشه. اون موقع خروجی کلی میشه:

$$B^* = B^{*1} \cup \dots \cup B^{*N}$$

اگر بین بعضی ها اجتماع و برخی اشتراک باشه هم:

$$\boxed{B^* = (B^{*1} \cup B^{*2}) \cap B^{*3}}$$

قانون دوشرطی زیر رو در نظر بگیریم:

$$A \Leftrightarrow B$$

تفسیر این قانون یعنی این که:

$$A \Leftrightarrow B \equiv (A' \cup B) \cap (B' \cup A)$$
$$\equiv (A \Rightarrow B) \cap (B \Rightarrow A)$$

پس دو قانون داریم بینش اشتراکه. و میتوان گفت:

$$\equiv \underset{R^1}{(A \Rightarrow B)} \cap \underset{R^2}{(B \Rightarrow A)}$$

$$B^{*1} = A^* \circ R^1 \qquad B^{*2} = A^* \circ R^2$$

$$B^* = B^{*1} \cap B^{*2}$$

سوال: آیا میشه بر اساس رابطه هر قانون یک رابطه نهایی تولید کرد که ورودی فقط یک بار با رابطه نهایی ترکیب بشه و هزینه محاسباتی کمتر بشه!؟ جواب این سوال بسته به اینه که بینشون اشتراکه یا اجتماع.

اگر بینشون اجتماع باشه میشه ولی اشتراک باشه نمیشه (چرا!؟)

علت:

$$R^1 \qquad B^{*1} = A^* \circ R^1$$

$$\vdots$$

$$R^i \qquad B^{*i} = A^* \circ R^i$$

$$R^N \qquad B^{*N} = A^* \circ R^N$$

$$B^* = \bigcup_i (A^* \circ R^i)$$

$$= \bigcup_i \left(\bigcup_{x \in X} A^* \cap R^i \right)$$

سپس اجتماع بیرونی رو میشه داخل و میشه نوشت:

$$= \bigcup_{x \in X} A^* \cap \left(\bigcup_i R^i \right)$$

$$= \bigcup_{x \in X} A^* \circ \left(\bigcup_i R^i \right)$$

و بنابراین رابطه ای میشه تعریف کرد و داریم:

$$\mathbb{R} = \bigcup_i R^i$$

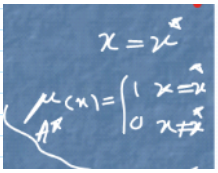
$$B^* = A^* \circ \mathbb{R}$$

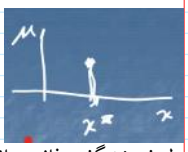
ولی اگه اشتراک باشه نمیشه این کار رو کرد که حالا میشه تو خونه تمرین کرد بررسی شه که چرا نمیشه.

$$B^* = \bigcap_i (A^* \circ R^i)$$

حالا بریم سراغ روش های فازی سازی:

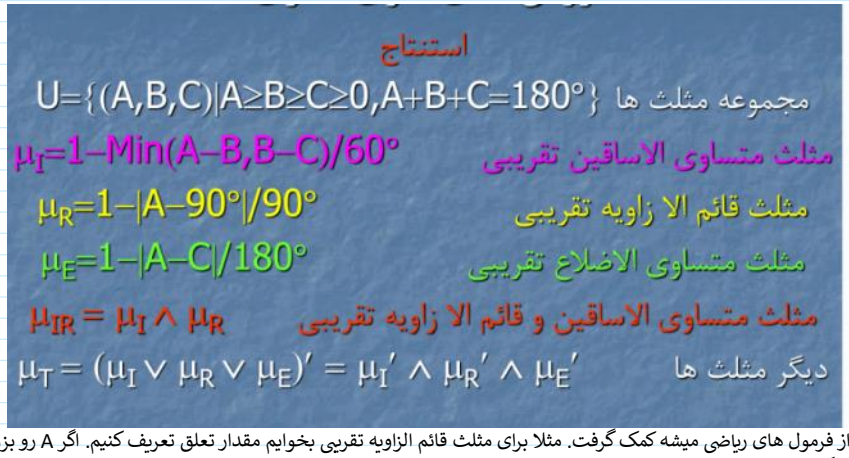
یکیش اینه که با تعلق بگیم. مثلا اگر یک x داشته باشیم که برابر با x^* باشه (مثل مثال امروز که برابر با 6 بود) بیایم بگیم:





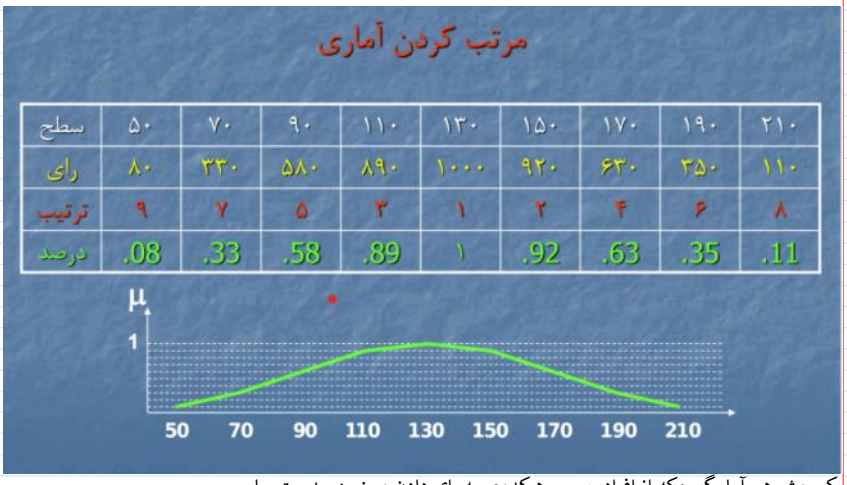
ولی نمیشه گفت فازی سازی شده فقط نمایش رو با تابع تعلق بیان کردیم.

روش دیگه اینه به صورت شهودی نقاط کلیدی رو مشخص کنیم که یک شخص خبره یا آمارگیری تعیین میکرد بعد اینارو طوری به صورت خطی وصل میکنیم که در هرنقطه جمعشون بشه 1



از فرمول های ریاضی میشه کمک گرفت. مثلا برای مثلث قائم الزاویه تقریبی بخوایم مقدار تعلق تعریف کنیم. اگر A رو بزرگترین زیاویه بعد B بعد C در نظر بگیریم. بزرگترینش باید 90 بشه پس اون رابطه زرد رنگه میگه هرچی A به 90 نزدیک تر بشه تعلق به 1 نزدیکتر میشه و بالعکس هرچی از 90 دور کنیم به 0 نزدیکتر میشه

از منطق هم میشه تعریف کرد مثلا برای 3 تا مثلث تعریف کردیم برای بقیه مثلثا میشه متمم اونا.



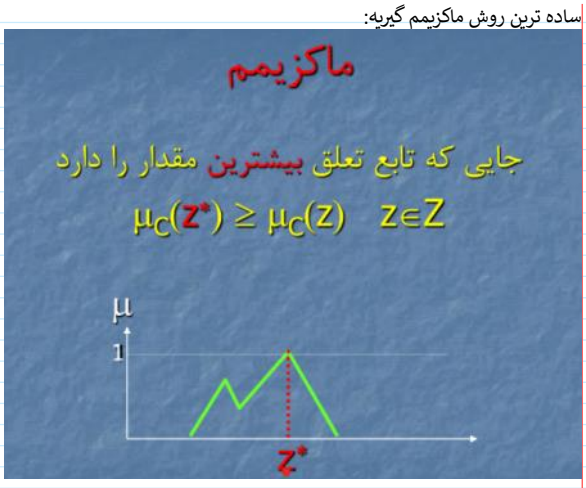
سطح	۵۰	۷۰	۹۰	۱۱۰	۱۳۰	۱۵۰	۱۷۰	۱۹۰	۲۱۰
رای	۸۰	۳۳۰	۵۸۰	۸۹۰	۱۰۰۰	۹۲۰	۶۳۰	۳۵۰	۱۱۰
ترتیب	۹	۷	۵	۳	۱	۲	۴	۶	۸
درصد	.08	.33	.58	.89	۱	.92	.63	.35	.11

یکی روش هم آمار گیریهکه از افراد بپرسیم هرکدوم چه رای دادن و منحنی بدست بیاریم

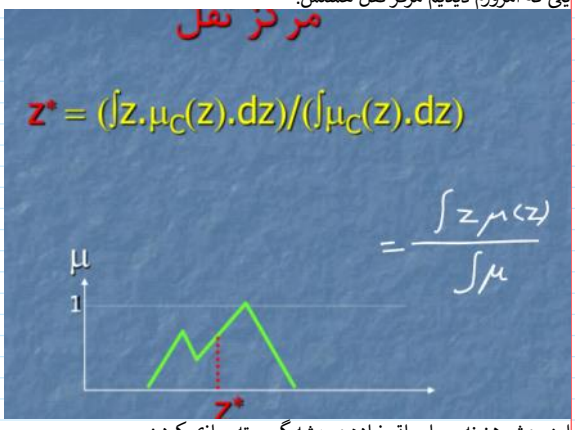
از آنترپی و شبکه عصبی و الگوریتم های تکاملی هم میشه که استاد گذر کرد چون وقت نمیشه.

حالا بریم سراغ غیر فازی سازی:

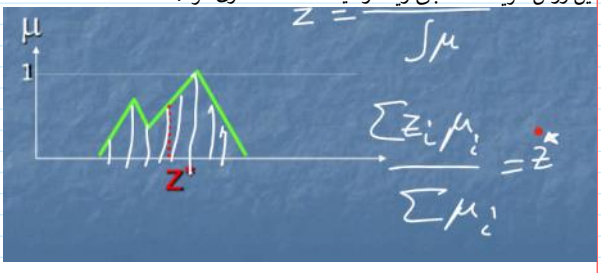
ساده ترین روش ماکزیمم گیریه:



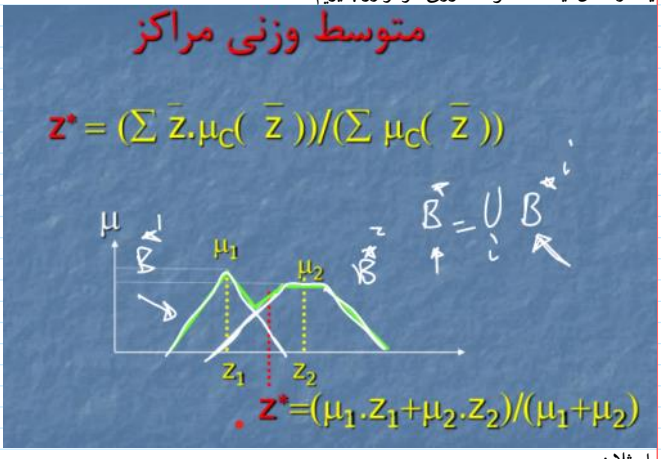
یکی که امروزم دیدیم مرکز ثقل هستش:



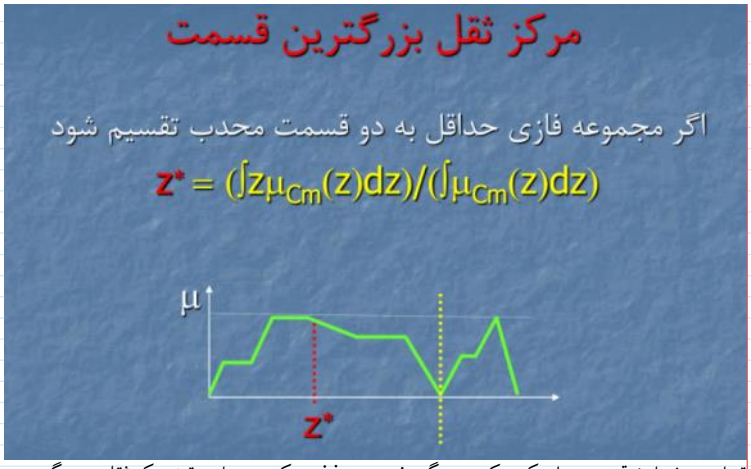
این روش هزینه محاسباتی زیاده و میشه گسسته سازی کرد :



یک راه حل اینه که متوسط وزنی مراکز رو بگیریم:



یا مثلا :



تو این روش اون قسمت های کوچیک رو میگیم نویزه و حذف میکنیم و برای بقیه مرکز ثقل رو میگیریم.

جلسه هجدهم

سیستم خبره فازی

برای تشکیل یک حالت استاندارد برای سیستم خبره فازی یک سری اصطلاحات و استاندارد پیش میاد که اول میایم اینارو تعریف کنیم:

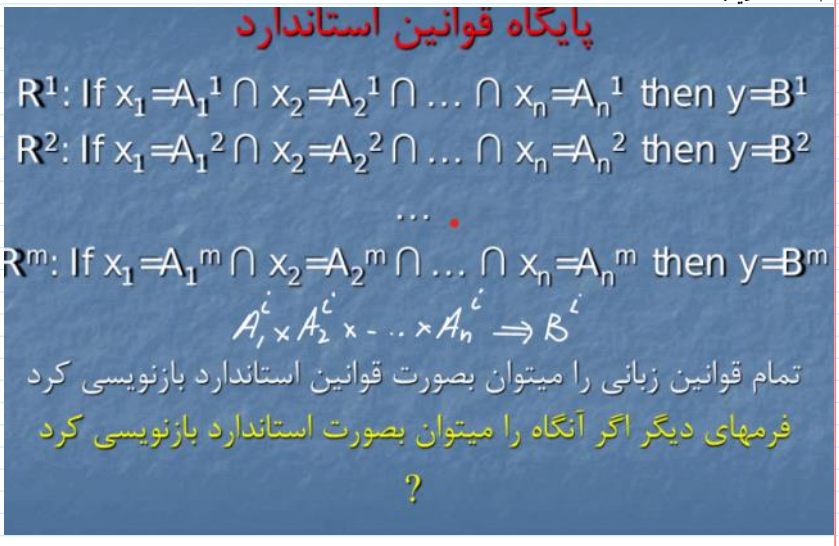
- **ترم های اتمیک:** ترم هایی که نمیتوان به قسمت های کوچکتری تقسیم کرد مثل جوان، پیر، آهسته و... . برای اینا میشه مجموعه فازی هم تشکیل داد. معمولا مقدارشون در یک بازه هستش و مقدار دقیق هم ندارن.
- **صفت ها و قید های تصحیح کننده:** قبل و بعد ترم های اتمیک میاد. مثلا خیلی جوان یا کمی جوان یا ... که با علامت ریاضی میشه بدستش آورد.

- **قانون استاندارد:** یک قانونی رو استاندارد میگن اگه همه متغیر های ورودی در شرط باشن و قسمت نتیجه هم فقط یکی باشه. به عبارت دیگه میشه اینطوری نوشتش:

قانون i ام:

$$A_1^i \times A_2^i \times \dots \times A_n^i \Rightarrow B^i$$

صفحه اسلاید:



اگر یک قانونی داشته باشیم که استاندارد نباشه میشه تبدیل کرد به چند قانون استاندارد. مثال:

In The Name Of God

$$X_1 \quad X_2 \qquad Y_1 \qquad Y_2$$

$$A_1 \cup A_2 \Rightarrow B_1 \text{ else } B_2$$

در این موقع باید به یکی از دو شکل زیر در بیاریم:

$$\boxed{X_1 \times X_2 \Rightarrow Y_1}$$
$$\boxed{X_1 \times X_2 \Rightarrow Y_2}$$

برای این کار:

$$(A_1 \cup A_2 \Rightarrow B_1)$$
$$\cap\ ((A_1 \cup A_2)' \Rightarrow B_2)$$

علت اشتراک:

$$A \Rightarrow B \ \ \text{یا} \ \ C$$
$$(A' \cup B) \cap (A \cup C)$$

اگر اجتماع باشه A اجتماع با 'A میشه مرجع و مرجع هم وقتی داشته باشیم با هرچی اجتماع بشه میشه همواره True

حالا برگردیم به مساله امون:

$$(A_1 \cup A_2 \Rightarrow B_1) \quad (1)$$
$$\cap\ ((A_1 \cup A_2)' \Rightarrow B_2) \quad (2)$$
$$(2) \quad A_1' \cap A_2' \Rightarrow B_2$$

$$\boxed{X_1 \times X_2 \Rightarrow Y_1}$$
$$\boxed{X_1 \times X_2 \Rightarrow Y_2}$$

قانون دوم استاندارد هستش چو نمیشه حالت دومی که داشتیم ولی قانون اول استاندارد نیست چون بینشون اجتماعه. پس تبدیلش میکنیم به 2 قانون:

$$(1) \quad (A_1 \Rightarrow B_1) \ (a)$$
$$\cup\ (A_2 \Rightarrow B_1) \ (b)$$

حالا a و b آیا استانداردن؟ نه! چون در a فقط مرجع X_1 رو داریم و برای b هم فقط مرجع X_2 داریم و حاصل ضرب کارتزین نداریم. پس باید هم مرجع کنیم ← توسعه استوانه ای

$$A_1 \times X_2 \Rightarrow B_1 \qquad X_1 \times A_2 \Rightarrow B_1$$

پس قوانین استخراج شده استاندارد در آخر برابرند با:

$$\underbrace{(A_1' \times A_2' \Rightarrow B_2)}_{R^1} \cap \Big(\underbrace{(X_1 \times A_2 \Rightarrow B_1)}_{R^2} \cup \underbrace{(A_1 \times X_2 \Rightarrow B_1)}_{R^3} \Big)$$

$$R = R^2 \cup R^3$$

و با توجه به نکته جلسه قبل که اگر اجتماع داشتیم میتونستیم بین رابطه ها اجتماع بگیریم ولی اگر اشتراک داشتیم نمیتونستیم اگر بهمون ورودی بدن داریم:

$$A_1^* \times A_2^* = A^*$$

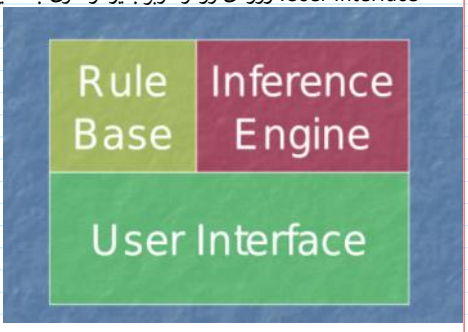
$$B^{1*} = A^* \circ R^*$$

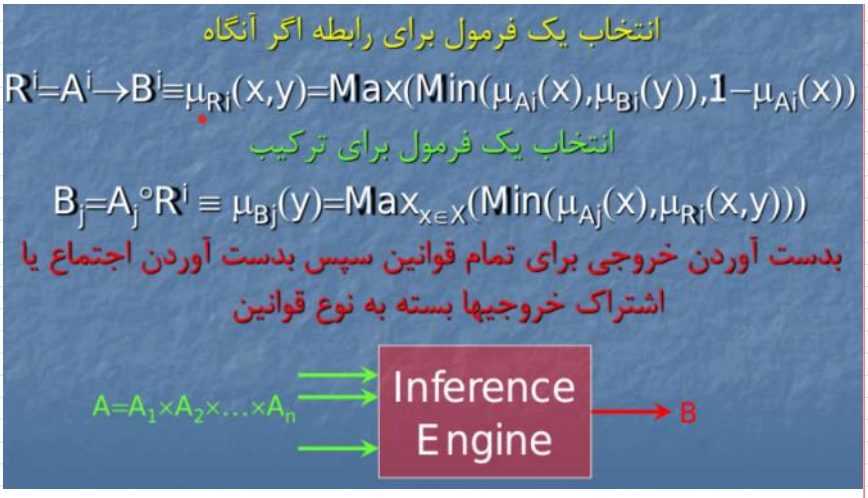
$$B^{2,3*} = A^* \circ R$$

$$B^* = B^{1*} \cap B^{2,3*}$$

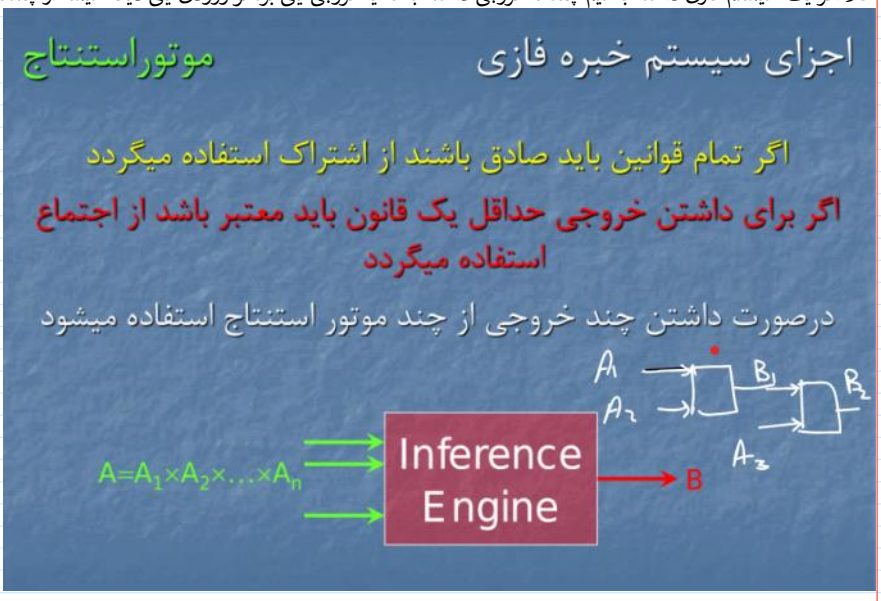
حالا برگردیم به سیستم خبره فازی. سیستم خبره فازی از سه بخش تقسیم شده:

- پایگاه قوانین
- موتور استنتاج: ورودی رو با رابطه قوانین ترکیب میکنه و خروجی جزئی بدست میاره. و خروجی های جزئی رو با هم اجتماع اشتراک میکنه خروجی نهایی بدست میاد
- User Interface: ورودی رو از کاربر بگیره و فازی بکنه یا به صورت تابع تعلق در بیاره و بده به موتور استنتاج و خروجی نهایی رو بگیره و غیر فازی کنه و بده به کاربر

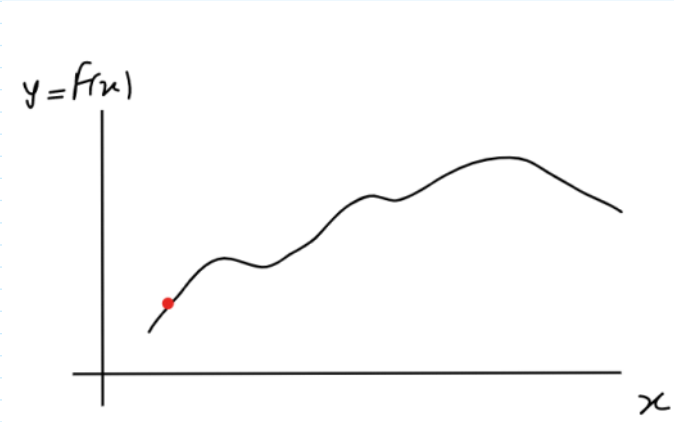




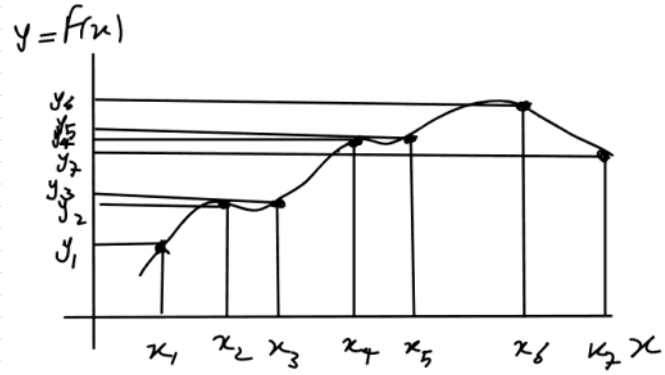
حالا اگر یک سیستم فازی داشته باشیم چند تا خروجی داشته باشه یا خروجی یکی بره تو ورودی یکی دیگه میشه از چندتا موتور استنتاج استفاده کرد:



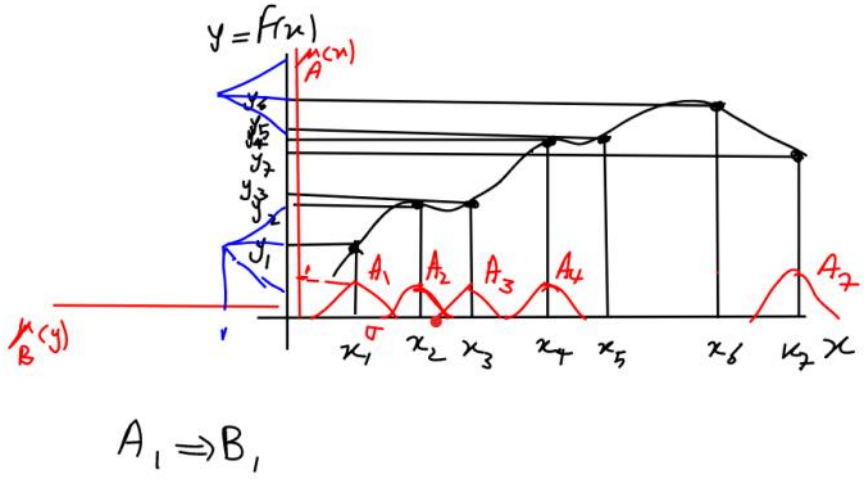
فرض کنید که شما یک سری داده دارید. میخوایم یه چیزی شبکه سیستم خبره فازی درست کنیم که مشابه شبکه عصبی کار کنه. فرض کنیم یک منحنی فرضی موجوده:



ما فقط یک سری نقاط این از این منحنی رو تحت عنوان train data داریم:



حالا میخوایم یک سری قوانین درست کنیم بر اساس این داده ها که داریم.



پس ما اولین کاری که کردیم اینه که قانون تعریف کردیم. چطوری؟ به ازای داده هایی که داریم اون داده ها بشن مرکز داده های فازی ورودی و خروجی و اطراف هم یک مقدار پوشش بدن.
حالا از بین رابطه های زیر کدومو انتخاب کنیم؟:

$$R_1 = A' \cup B$$

$$R_2 = (A \times B) \cup (A' \times y)$$

$$R_3 = A \times B$$

$$R_4$$

مسلما R_3 چون وقتی رابطه برقراره که میدونیم چیکار بکنیم ولی برقرار نباشه هیچ کاری نباید بکنیم.

$$A^1 \Rightarrow B^1 \qquad R^1 = A^1 \times B^1$$

$$A^i \Rightarrow B^i \qquad R^i = A^i \times B^i \qquad \mu_{R^i}(x,y) = \mu_{A^i}(x)\,\mu_{B^i}(y)$$

$$\vdots$$

$$A^N \Rightarrow B^N \qquad R^N = A^N \times B^N$$

برای ترکیب هم فرض مکنیم از ماکس دات استفاده میکنیم

حالا بهمون یه ورودی میدن میگن خروجیش چی میشه. **چیکار کنیم؟** باید فازیش کنیم. جلسه پیشم دیدیم یه راه سادش اینه که:

$$x = x^* \qquad A^* \qquad \mu_{A^*}(x) = \begin{cases} 1 & x = x^* \\ 0 & x \neq x^* \end{cases}$$

$$y = ?$$

$$B^{*i} = A^* \circ R^i$$

$$\mu_{B^{*i}}(y) = \mathrm{Max}\left(\mu_{A^*}(x) \cdot \mu_{R^i}(x,y)\right)$$

$$= \mathrm{Max}\begin{pmatrix} x = x^* & \mu_{R^i}(x^*,y) \\ x \neq x^* & 0 \end{pmatrix} = \mu_{R^i}(x^*,y)$$

$$B^{*i} = A^* \circ R^i$$

$$\mu_{B^{*i}}(y) = \mathrm{Max}\left(\mu_{A^*}(x) \cdot \mu_{R^i}(x,y)\right)$$

$$\mu_{B^{*i}}(y) = \mu_{A^i}(x^*)\,\mu_{R^i}(x)$$

که داریم:

$$\mu_{B^{*i}}(y) = \mu_{A^i}(x^*)\,\mu_{B^i}(y)$$

$$\mu_{B^*}(y) = \mu_{B^{*1}}(y) \vee \cdots \vee \mu_{B^{*N}}(y)$$

اگر از متوسط وزنی مراکز استفاده کنیم:

مرکز

$$\mu_{B^{*i}}(y) = \overbrace{\mu_{A^i}(x^*)}\,\overbrace{\mu_{B^i}(y)}$$

مرکز y_i

مرکز y_i

مقدار تعلق در مرکز $\mu_{A^i}(x^*)$

$$\mu_{B^*}(y) = \mu_{B^{*1}}(y) \vee \cdots \vee \mu_{B^{*N}}(y)$$

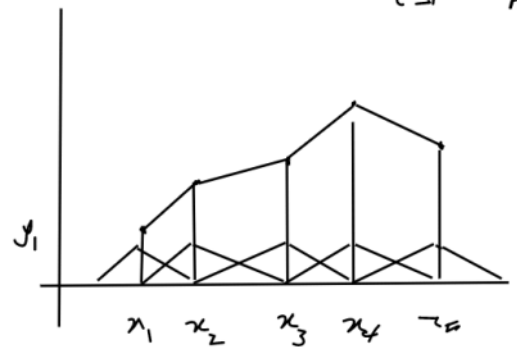
$$z^* = \frac{\sum_i z_i \mu_i}{\sum_i \mu_i} \qquad y^* = \frac{\sum_i y_i \mu_{A^i}(x^*)}{\sum_i \mu_{A^i}(x^*)}$$

پس سیستم خبری فازی برابر میشه با :

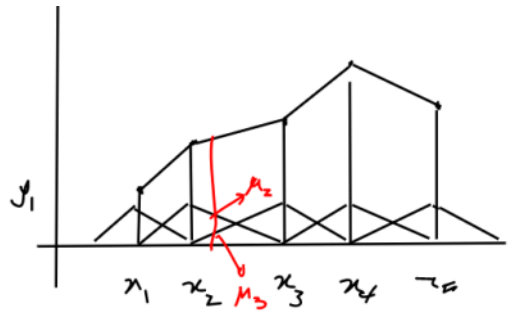
$$\boxed{f(x^*) = y^* = \frac{\sum_{i=1}^{N} y_i \mu_{A^i}(x^*)}{\sum_{i=1}^{N} \mu_{A^i}(x^*)}}$$

اگر بخوایم روی شکل متوجه بشیم معنی این فرمول چیه داریم:
ما یک سری نقاط داریم که روشون مجموعه فازی تعریف کردیم. اگر مجموعه فازی رو طوری تعریف کنیم که جمع تعلق ها در یک نقطه 1 بشه فقط صورت کسر میمونه و مخرج 1 میشه پس ساده میشه:

$$f(x) = \sum_{i=1}^{N} y_i \mu_{A^i}(x)$$



در مراکز مقدار تعلق برای اون مرکز 1 هستش برای بقیه 0 هستش پس تو مراکز خروجی میشه y_i ولی در بقیه جاها میشه یه همچین چیزی:



$$y = \frac{\mu_2 y_2 + \mu_3 \mu_3}{1 = \mu_2 + \mu_3}$$

و میتوان نوشت برابراست با:

$$y = \mu_2 y_2 + (1-\mu_2)\, y_3$$

ولی لزومی نداره که جمع مقدار تعلقا 1 بشه. اگه جمعشون 1 بشه اون شکلی که بدست میاد قطعه قطعه خطی هستش ولی وقتی جمعشون 1 نباشه خطی نمیشه.
وقتی تعداد داده های آموزشی بیشتر باشه شکلمون دقیق تر بدست میاد حالا چه قطعه قطعه خطی باشه چه قطعه قطعه غیر خطی.

دوره جلسه پیش:

In The Name Of God

سیستم خبره فازی برای تقریب تابع (رگرسیون) وقتی که مقدار از نقاط تابع را داشته باشیم

$$\mu_{R^i}(x,y) = \mu_{A^i}(x)\,\mu_{B^i}(y)$$ ترکیب Max Dot

فازی سازی ورودی $\mu_{A^*}(x) = \begin{cases} 1 & x = x^* \\ 0 & x \neq x^* \end{cases}$

غیرفازی سازی متوسط وزنی مراکز $y^* = \dfrac{\sum \text{مرکز} \times \text{تعلق مرکز}}{\sum \text{تعلق مراکز}}$

* در رابطه آخر نوشته شده سیگما تعلق در مرکز ضربدر مرکز تقسیم بر سیگمای تعلق مراکز

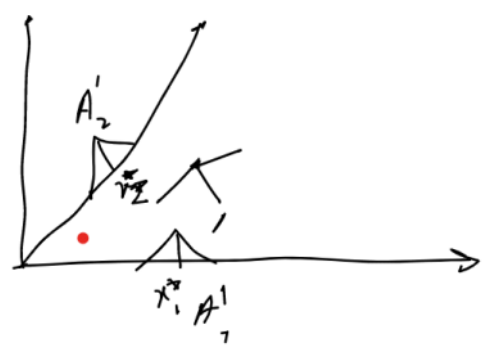
$$f(x) = \frac{\sum_{i=1}^{N} y_i \mu_{A^i}(x)}{\sum_{i=1}^{N} \mu_{A^i}(x)}$$

حالا فرض شود برای چند بُعد بخوایم فرمول رو بنویسیم. اگر حاصل ضرب کارتزین رو با ضرب مدل کنیم:

$$f(x_1,\dots,x_n) = \frac{\sum_{i=1}^{N} y_i \prod_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{i=1}^{N} \prod_{j=1}^{n} \mu_{A_j^i}(x_j)}$$

$\mu_{A_j^i}(x_j)$ → مجموعه فازی برای عدد زام و قانون iام

معنیش رو اگر بخوایم با شکل نشون بدیم:



حالا میتوان ایده های شبکه عصبی و سیستم فازی رو با هم قاطی کنیم میتونیم یک شبکه عصبی فازی درست کنیم. به این صورت که:

$$y = \sum_{i=1}^{N} y_i \frac{\prod_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{k=1}^{N} \prod_{l=1}^{n} \mu_{A_l^k}(x_l)}$$

سپس داریم:

$$y = \sum_{i=1}^{N} y_i \underbrace{\frac{\prod_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{k=1}^{N} \prod_{l=1}^{n} \mu_{A_l^k}(x_l)}}_{=r_i} \qquad r_i' \to \prod_{j=1}^{n} \mu_{A_j^i}(x_j), \quad \sum_{k=1}^{N} r_k' \to \sum_{k=1}^{N} \prod_{l=1}^{n} \mu_{A_l^k}(x_l)$$

$$y = \sum_{i=1}^{N} y_i r_i$$

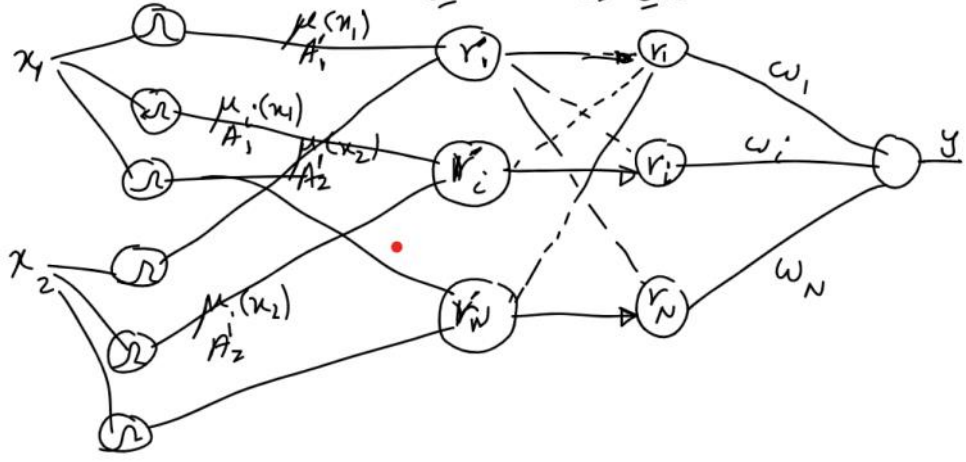
حالا فرض کنیم y_i ها هم در حالت شبکه عصبی بلد نیستیم و اسمشو میذاریم w_i:

$$y = \sum_{i=1}^{N} y_i r_i = \sum_{i=1}^{N} w_i r_i$$

پس میتونیم به این شکل درست کنیم که یک سری مجموعه فازی داریم برای هر بُعد، اینا با هم حاصل ضرب کارتزین میشن قوانین رو درست میکنن. این قوانین نرمال میشن و در یک سری وزن ضرب میشن و خروجی رو میسازن:

$$r_i = \frac{r_i'}{\sum r_k'} \qquad \sum r_i = 1 \qquad y = \sum_{i=1}^{N} w_i r_i$$

قوانین نرمال قوانین



فرق شبکه عصبی فازی که اینجا گفتیم با سیستم خبره فازی چی هست؟ فرقش اینکه:

- در سیستم خبره فازی ما w_i هارو بر اساس خروجی داده هایی که داشتیم بدست می آوردیم. ولی در شبکه عصبی فازی وزن ها یادگیری می شوند.
- مجموعه های فازی که در سیستم خبره فازی برای قوانین تعریف میکردیم بر اساس هر داده بود ولی در شبکه عصبی فازی مراکز قوانین رو هرطور که دوست داشتیم انتخاب میکنیم.

شبکه عصبی فازی 2 نوع یادگیری میتونه داشته باشه:

1. مراکز رو یاد بگیریم و وزن های خروجی رو
2. مراکز رو ثابت فرض کنیم و فقط وزن های خروجی رو بدست بیاریم.

به این نوع شبکه عصبی فازی میگیم TSK0 که خیلی شبیه به RBF هستش فقط این که در RBF ما نرمال سازی نداریم اینجا داریم.

$$y = \sum w_i r_i(X) = w^T R(X) \qquad \text{TSK0}$$

$$y = w^T R(X) \qquad w = \begin{bmatrix} w_1 \\ \\ w_N \end{bmatrix}$$

$$y^l = w^T R(X^l)$$

$$R(X) = \begin{bmatrix} r_1(X) \\ \\ r_N(X) \end{bmatrix}$$

در تصویر بالا مشاهده میکنیم که چطوری میشه رابطه رو به شکل برداری نوشت. y^l هم داره رابطه رو برای داده L ام میگه. یعنی اگر M تا داده داشته باشیم:

$$\begin{cases} y^1 = R^T(x^1) w \\ y^l = R^T(x^l) w \\ y^M = R^T(x^M) w \end{cases} \qquad Y = \bar{R} w$$

$$\bar{R} = \begin{bmatrix} R^T(x^1) \\ \\ R^T(x^M) \end{bmatrix}$$

اگر y واقعی رو y^* در نظر بگیریم مجموع مربعات خطا رو داریم:

$$E = \frac{1}{2}(Y - Y^*)^T (Y - Y^*) \qquad Y \to \bar{R} w$$

$$\frac{dE}{dw} = \bar{R}^T (Y - Y^*) = \bar{R}^T \bar{R} w - \bar{R}^T Y^* = 0$$

$$w = (\bar{R}^T \bar{R})^{-1} \bar{R}^T Y^*$$

اما بعضی موقع ها ممکنه اون ترمی که معکوسش کردیم full rank و معکوس پذیر نباشه پس میایم یک ترمی بهش اضافه میکنیم:

$$E = \frac{1}{2}(y - \hat{y})^T (y - \hat{y}) - \lambda w^T w$$

$$\hat{y} \rightarrow \bar{R} w$$

$$\frac{dE}{dw} = \bar{R}^T (y - \hat{y}) = \bar{R}^T \bar{R} w - \bar{R}^T y = 0$$

$$w = (\bar{R}^T \bar{R})^{-1} \bar{R}^T y$$

$$w = (\bar{R}^T \bar{R} + \lambda I)^{-1} \bar{R}^T y$$

یک راه دیگش هم گرادیان گرفتن هستش:

$$y^l = \sum_{i=1}^{N} w_i r_i^l$$

$$E = \frac{1}{2} \sum_{l=1}^{M} (y^l - \hat{y}^l)^2$$

$$\boxed{\Delta w = -\eta \sum_{l=1}^{M} (y^l - \hat{y}^l) r_i^l}$$

یکی از روش هایی که میشه مراکز رو تعیین کرد روش خوشه بندی هستش.

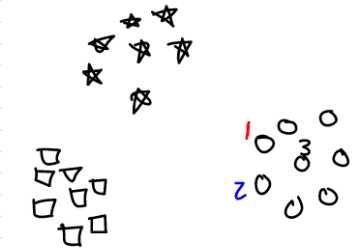
K-Means & C-means

فرض کنیم یک سری داده داریم:

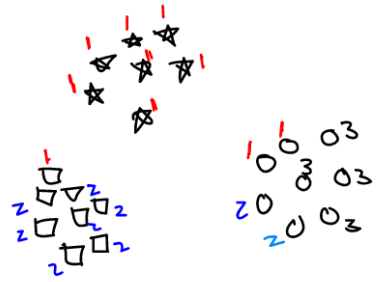
فرض میکنیمk=c=3

* یکی از مشکلات K-means و C-means اینه که تعداد خوشه هارو باید خودمون انتخاب کنیم. اگر این تعداد رو خوب تعیین نکنیم ممکنه خوب جواب نده (مخصوصا اگر داده ها تو هم باشن) هرچند که اگر تو هم نباشن جوابش معمولا خوب میشه.

فرض میکنیم ابتدا به صورت تصادفی 3 تا مرکز رو انتخاب میکنیم. برای این که نشان دهیم وقتی این 3 تا داده از هم جدا باشن به راحتی جواب میده در این مثال هر 3 تا مرکز رو بد انتخاب میکنیم:



بعد نگاه میکنیم هر داده به هر مرکز که هستش لیبل داده رو با لیبل مرکز یکی در نظر میگیرم:

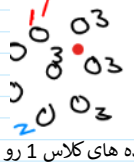


حالا که لیبل زدیم برای داده ها، میایم مقدار تعلق هم اینطوری در نظر میگیریم:

$$u_{ik} = \begin{cases} 1 & X_k \in C_i \\ 0 & X_k \notin C_i \end{cases}$$

برای مثال اگر یکی از داده ها رو داده دهم در نظر بگیریم داریم:

$$u_{310} = 0$$
$$u_{210} = 0$$
$$u_{110} = 1$$



حالا اگر مرکز داده های کلاس 1 رو بخوایم محاسبه کنیم :

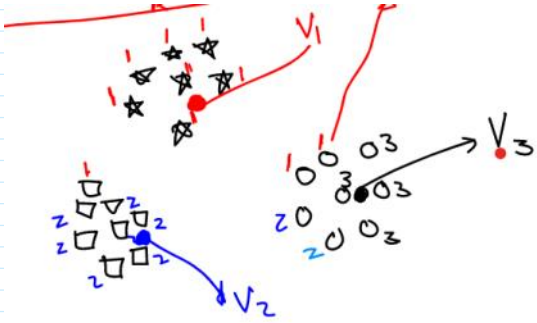
$$V_1 = \frac{\sum_k u_{1k} X_k}{\sum_k u_{1k}}$$

مخرج کسر که میشه تعداد داده هایی که لیبل 1 دارن صورتش هم میشه جمع مقدار اونایی که لیبل 1 دارن.

در حالت کلی هم داریم:

$$\boxed{V_i = \frac{\sum_k u_{ik} X_k}{\sum_k u_{ik}}}$$

حالا اگر به صورت شهودی مرکز هارو نشون بدیم یه همچین چیزی میشن:



حالا که مرکز جدید تعیین شد بر اساس مراکز جدید لیبل های جدید درست میکنیم و U_i هارو عوض میکنیم:

یکی 2 بار دیگه تکرار کنیم دیگه لیبل های داده ها دقیق معلوم میشه. به این روش میگیم K-means و C-means (تو کامپیوتر تا وقتی تکرار میشه که یا u_i ها تغییراتشون از یک حد آستانه کمتر بشه یا مراکز تغییراتشون از یک حد آستانه کمتر بشه)

اگر بخوایم مساله خوشه بندی رو به شکل یک مساله بهینه سازی بنویسیم:
تابع هدفش چی میشه (چه خوشه بندی ای خوشه بندی خوبی هست)؟ داده های داخل یک خوشه تا حد امکان به مرکز خوشه نزدیک باشند. به زبان ریاضی یعنی:

$$E = \sum_k \sum_i u_{ik} \|X_k - V_i\|^2$$

ما میخوایم این رو مینیمم کنیم. **آیا میشه مشتق بگیریم برابر با صفر بذاریم؟** نه. چون u_i هامون 0 و 1 هستن و گسسته اند نمیشه مشتق گرفت.

پس میایم u_i هارو فازی فرض میکنیم. یعنی مقداری بین 0 و 1 در نظر بگیریم. این از نظر شهودی اشکالی ندارد. چرا؟ بر فرض مثال زیر رو در نظر بگیرید:

داده که دورش رو قرمز کشیدیم چرا باید تعلق 1 به قرمز داشته باشه تعلق 0 به سیاه یا بالعکس؟ فاصله اش از 2 مرکز تقریبا مساویه و میشه گفت 0.5 تعلق داره به سیاه 0.5 به قرمز

با این کار u_i ها پیوسته و مشتق پذیر میشن.

اگر نسبت به u مشتق بگیریم:

$$\frac{\partial E}{\partial u_{ik}} = \|X_k - V_i\|^2 = 0$$

$$V_i = X_k$$

که اینم امکان پذیر نیست چون نمیتونیم برای هر داده یک مرکز داشته باشیم. پس اینم مشکل داره و باید تابع هدف تغییر پیدا کنه:

$$E = \sum_i \sum_k u_{ik}^m \|X_k - V_i\|^2$$

که مشتق بگیریم:

$$E = \sum_i \sum_k u_{ik}^m \|x_k - v_i\|^2$$

$$\frac{\partial E}{\partial u_{ik}} = m\, u_{in}^{m-1} \|x_k - v_i\|^2 = 0$$

$$u_{ik} = 0 \qquad x_k = v_i$$

بازم این جواب به درد نمیخوره. هرچند که u_{ik} رو 0 بذاریم جواب درستی هست و مینیمم میکنه ولی به درد بخور نیست. اگر اون 2 تا ترم هم برابر بذاریم باز مینیمم میشه ولی باز به درد ما نمیاد.
برای این که یه کاری بکنیم که u_{ik} ها صفر نشن باید یک محدودیتی بذاریم که نتونه صفر بکنه.

$$\sum_{i=1}^{c} u_{ik} = 1$$

$$k = 1, \dots, N$$

با این محدودیت اگر یک x_1 داشته باشیم با 2 تا خوشه. اگر برای یک خوشه بشه 0.4 برای اون یکی باید بشه 0.6:

① x_1

② $u_{11} = 0.4$

$u_{21} = 0.6$

حالا جلوتر میبینیم این محدودیت چه مشکلی ایجاد میکنه ولی فعلا با این محدودیت بریم جلو ببینیم چی میشه.

روش لاگرانژ

$$E = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^m \|x_k - v_i\|^2 - \sum_{k=1}^{N} \lambda_k \left(\sum_{i=1}^{C} u_{ik} - 1 \right)$$

توی روش لاگرانژ میگه تابع هدف رو منهای یک سری ضرایب لاگرانژ ضریدر محدودیت ها بکنیم. محدودیت برای درست کردن محدودیت ها هم هر رابطه که داریم همرو ببریم یک سمت مساوی صفر بکنیم و اون ترمی که سمت چپ تساوی هست رو میاریم. برای محدودیت ما این شکلی میشد:

$$\sum u_{ik} - 1 = 0$$

در روش لاگرانژ یک بار نسبت به متغیر ها و یکبار نسبت به ضرایب لاگرانژ مشتق میگیریم. سپس 3 تا معادله بدست میاد که باید توشون بیایم ضرایب لاگرانژ رو حذف کنیم و یک سری رابطه بدست بیاریم. که جلسه بعد این کار رو انجام میدیم

$$\frac{\partial E}{\partial u_{ik}} = 0 \qquad \frac{\partial E}{\partial v_i} = 0 \qquad \frac{\partial E}{\partial \lambda_k} = 0$$

جلسه بیستم

طبق جلسه پیش داریم:

$$E = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^{m} \|X_k - V_i\|^2 - \sum_{k=1}^{N} \lambda_k \left(\sum_{i=1}^{C} u_{ik} - 1 \right)$$

حالا اگر نسبت به u_{ik} مشتق بگیریم چون هم i مشخصه هم k مشخصه دیگه برای بقیه ترم ها صفر میشه پس سیگما حذف میشه و داریم:

$$\frac{\partial E}{\partial u_{ik}} = m\, u_{ik}^{m-1} D_{ik} - \lambda_k = 0$$

با مساوی صفر قرار دادن داریم:

$$u_{ik}^{m-1} = \frac{\lambda_k}{m} \times \frac{1}{D_{ik}}$$

که با رادیکال به قوه m-1 نداریم:

$$u_{ik} = \left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \frac{1}{D_{ik}^{\frac{1}{m-1}}}$$

اگر رابطه اصلی را نسبت به λ_k هم مشتق بگیریم داریم:

$$\frac{\partial E}{\partial \lambda_k} = \sum_{i=1}^{C} u_{ik} - 1 = 0 \rightarrow \sum_{j=1}^{C} u_{jk} = 1$$

با جایگذاری مقدار u_{ik} بدست آمده در مشتق اول در رابطه دوم داریم:

$$\boxed{\sum_{j=1}^{C} \left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \frac{1}{D_{jk}^{\frac{1}{m-1}}} = 1} \quad (2)$$

و اسم رابطه بدست آمده در مشتق اول هم میذاریم 1:

$$\boxed{u_{ik} = \left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \frac{1}{D_{ik}^{\frac{1}{m-1}}}} \quad (1)$$

سپس داریم:

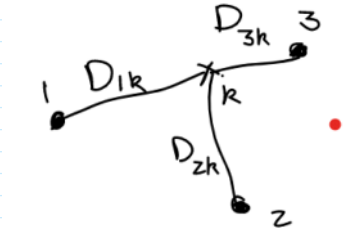
$$\left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \times \frac{1}{D_{ik}^{\frac{1}{m-1}}} = u_{ik} \qquad (1)$$

$$\left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \sum_{j=1}^{c} \frac{1}{D_{jk}^{\frac{1}{m-1}}} = 1 \qquad (2)$$

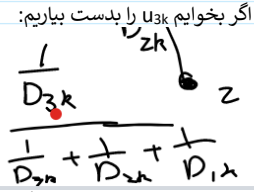
$$\frac{(1)}{(2)}$$

$$\boxed{\frac{\frac{1}{D_{ik}^{\frac{1}{m-1}}}}{\sum_{j=1}^{c} \frac{1}{D_{jk}^{\frac{1}{m-1}}}} = u_{ik}}$$

پس مقدار تعلق داده k به مرکز خوشه i با رابطه فوق حاصل می شود. **معنیش چی میشه؟**



حالا اگر بخوایم u_{3k} را بدست بیاریم:



$$u_{3k} = \frac{\frac{1}{D_{3k}}}{\frac{1}{D_{3k}} + \frac{1}{D_{2k}} + \frac{1}{D_{1k}}}$$

در رابطه فوق توان m-1 رو نادیده گرفتیم چون موقع تابع هدف به توان m رسونده بودیم. حالا نکته ای که میخوایم بهش برسیم اینه که u_{3k} میشه عکس فاصله تقسیم بر مجموع عکس فاصله ها، با این کار وقتی u_{ik} ها با هم جمع بشن صورت و مخرج یکی میشه و برابر با 1 میشن و نرمال میشن (همون محدودیتی که اعمال کردیم).

حالا بریم ببینیم فرمول نسبت به مرکز چی میشه:

$$E = \sum_{i,k} u_{ik}^{m} (X_k - V_i)^T (X_k - V_i) - \cdots$$

$$\frac{\partial E}{\partial V_i} = -2 \sum_{k} u_{ik}^{m} (X_k - V_i) = 0$$

$$\sum_{k} u_{ik}^{m} X_k = V_i \sum_{k} u_{ik}^{m}$$

$$\boxed{V_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} X_k}{\sum_{k=1}^{N} u_{ik}^{m}}}$$

m که مخالف 1 هستش چه نقشی داره؟ معمولا یک عددی بزرگتر از 1 تا 2 در نظر میگیرین. هرچی توان از 1 بیشتر میشه عدد تعلق رو به 0 نزدیک تر میکنه و حالت فازی رو کمتر میکنه و هرچی به 1 نزدیک تر میشه میزان فازی بودن بیشتری خواهیم داشت.

$$u_{ik} = \frac{\left(\frac{1}{\|X_k - V_i\|}\right)^{\frac{2}{m-1}}}{\sum_{j=1}^{c}\left(\frac{1}{\|X_k - V_j\|}\right)^{\frac{2}{m-1}}} \quad (1)$$

$$V_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} X_k}{\sum_{k=1}^{N} u_{ik}^{m}} \quad (2)$$

ولی مشکل اینجوری شد که داستان مرغ و تخم و مرغ میشه. برای بدست آوردن u_{ik} به v_i نیاز داریم و بالعکس.
پس مثل c-means مراکز اولیه رو تصادفی بدست میاریم بعد u هارو بدست میاریم بعد v هارو بروز میکنیم و انقدر تکرار میکنیم که از یه حد آستانه کمتر بشه.
مراحل به گفته ی استاد:

FCM

۱- مراکز اولیه بصورت تصادفی یکنواخت از بین داده ها انتخاب می‌شوند

۲- با استفاده از فرمول (1) u_{ik} مقادیر تعلق جدید محاسبه می‌شوند

۳- با استفاده از فرمول (2) مراکز جدید محاسبه می‌شوند

۴- در صورتی که $\|U^{new} - U^{old}\| > \varepsilon$ به مرحله (2) برمی‌گردیم

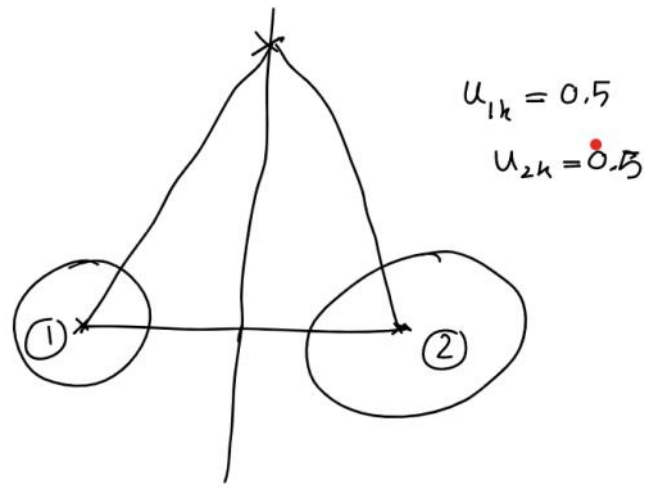
حالا بریم به معایبش پی ببریم:

1. هزینه محاسباتی
2. مراکز اولیه حساس ← همگرایی به مینیمم محلی
3. انتخاب تعداد خوشه ها
4. حساسیت به داده های پرت (Outlier)

مزایا:

- همگرایی

توضیح مورد 4 ام:



فاصله این داده به مراکز 1 و 2 یکسانه پس باعث میشه میزان تعلقش 1 بشه در صورتی که اصن تعلق نباید داشته باشه و این شرط که ما گفتیم مجموعش باید 1 باشه باعث شد که این مشکل به وجود بیاد.

برای حل این مشکل یک سری نسخه FCM اومدن محدودیت رو عوض کردن یا یک سری محدودیت جدید اضافه کردن ولی این کار هزینه محاسباتی رو زیاد میکنه. بهترین راه حل اینه که بیایم داده های پرت رو حذف کنیم و از FCM استفاده کنیم.

چجوری داده های پرت رو تشخیص بدیم!؟ 2تا کار میشه کرد:

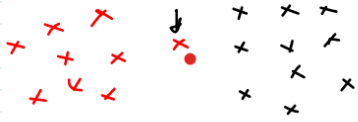
1. اگر یک داده فاصله اش از همه مراکز از یک حد آستانه بیشتر بود حذفش کنیم
2. واریانس 2 داده رو بدست بیاریم. میتونیم یک داده رو فاصله اش با نزدیک ترین داده اش حساب کنیم و اگر از یک حد آستانه بیشتر بود حذف کنیم.

روش اول بهتره چون هزینه محاسباتی کمتری داره.

چرا داده های پرت بد هستن!؟ مراکز رو میکشن به سمت خودشون و از مراکز واقعی دور میکنن.

ما خوشه بندی FCMرو انجام دادیم. چطوری بفهمیم هر داده متعلق به کدوم خوشه است؟ هرکدوم تعلق بیشتری داشت میگیم به اون خوشه داده تعلق داره

مزیت FCM نسبت به c-means چیه؟ روش FCM مراکز بهتری رو تعیین میکنه. در روش c-means در هر iteration ما یک سری اطلاعات رو حذف میکنیم



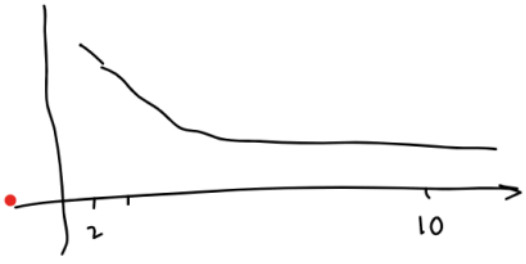
ما وقتی داده مرکزی رو به یکی میدیم به اون یکی نمیدیم تاثیرش بر هردو مرکز تاثیر میذاره و مرکز قرمز رو به سمت خودش میکشه و سیاه هم هیچ کششی به خودش نمیذاره و باعث میشه هیچ کدوم دقیق تعیین نشن. ولی وقتی مقدار فازی در نظر بگیریم برای هردو دقیق تر معلوم میشه.

حالا چجوری تعداد خوشه هارو تعیین کنیم؟ یک سری معیار های کارایی داریم که بستگی به کاربرد یکیش انتخاب میشه:

- آنتروپی: هرچه u_{ik} ها به 0 و 1 نزدیک تر بشن خوشه بندی بهتره و هرچی به 1/c نزدیک تر بشن بدتره.

$$-\sum_{k=1}^{N}\sum_{i=1}^{C} u_{ik}\log u_{ik}$$

این معیار به ازای c های مختلف رسم میشه و منحنی بدست میاد:



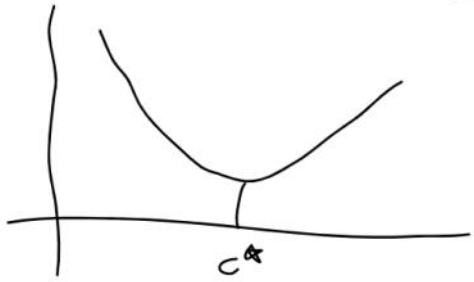
ما از طرفی نمیخوایم تعداد خوشه ها زیاد بشه چون هرچی بیشتر بشه و ما بیشتر تمایل به این پیدا کنیم که به ازای هر داده یک خوشه بگیریم و آنتروپی صفر میشه. پس میایم از جایی که شروع میشه شیب نمودار کم میشه رو به اندازه تعداد خوشه در نظر میگیریم.

- $\frac{1}{C} trac(U^T U)$

یه جورایی شبیه همون آنتروپی میشه فقط لگاریتم نداره.

- معیار خطای درون گروهی و برون گروهی:

$$E = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^{m} \|x_k - v_i\|^2 - \alpha \sum_{i=1}^{C} \|v_i - \bar{x}\|^2 - \sum_{i=1}^{C} \sum_{j=1}^{C} \|v_i - v_j\|^2$$



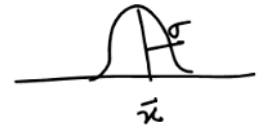
در این جا 2 به 2 فاصله مرکزارو مقایسه میکنیم و میخوایم فاصلشون ماکسیمم شه. نمودار رو استاد مطمئن نیست ولی گفت احتمالا همچین شکلی بدست میاد که مینیمم ش میشه اون تعداد خوشه ها.

جلسه بیست و یکم (انتهای فازی و ابتدای تکامل)

اگر ما با FCM خوشه بندی رو بکنیم یک سری مرکز v_i داریم و یک سری u_{ik} که همه u_{ik}هارو میشه به صورت یک ماتریس نوشت که سطر i و ستون k اش بشه u_{ik}.

$$V_i \qquad u_{ik} \qquad U = \begin{array}{c} \\ i \end{array}\begin{bmatrix} & \vdots & \\ \cdots & u_{ik} & \cdots \end{bmatrix}$$

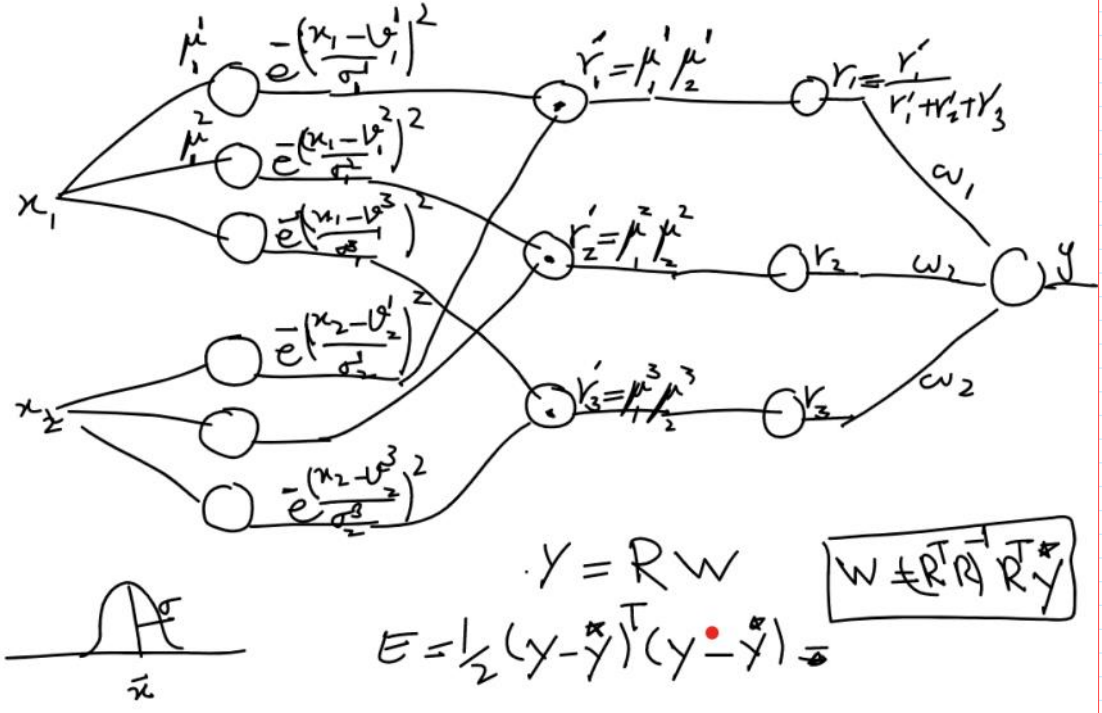
ابتدا برای شبکه عصبی فازی TSK_0: در شبکه عصبی فازی ما قانون هامون به این شکل هستش:
مجموعه های فازی را میخواهیم به صورت قوسی داشته باشیم. در نتیجه یک مرکز نیاز داریم و یک سیگما:



مرکز هارو که مرکز خوشه تعیین میکنه. در نتیجه خروجی پرسپترون ها رو در صورتی که مراکز برابر با

$$V_1 = \begin{bmatrix} v_1^1 \\ v_2^1 \end{bmatrix} \qquad V_2 = \begin{bmatrix} v_1^2 \\ v_2^2 \end{bmatrix} \qquad V_3 = \begin{bmatrix} v_1^3 \\ v_2^3 \end{bmatrix}$$

باشند داریم:



$$Y = RW \qquad \boxed{W = (R^T R)^{-1} R^T y^*}$$

$$E = \frac{1}{2}(y - y^*)^T (y - y^*)$$

حالا چیزی که میمونه اینه که سیگماها چطوری بدست میان.

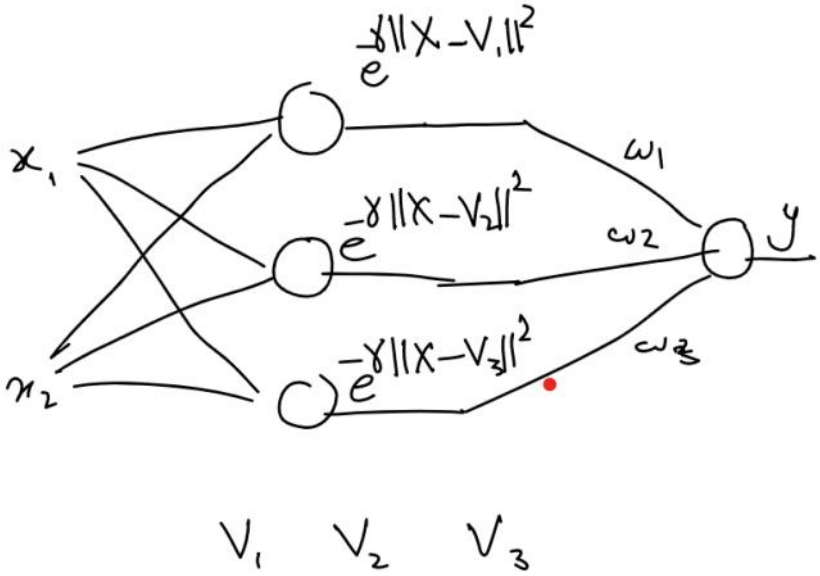
طبق فرمول داریم:

$$\sigma_j^2 = \frac{1}{N-1} \sum_{k=1}^{N} (x_{kj} - \bar{x}_j)^2$$

ولی ما برای یک خوشه ببینیم چی میشه. برای یک خوشه داریم:

$$(\sigma_j^i)^2 = \frac{\sum_{k=1}^{N} u_{ik} (x_{kj} - v_j^i)^2}{\sum_{k=1}^{N} u_{ik}}$$

حالا اگر بخوایم از همین روش خوشه بندی برای RBF بخوایم استفاده کنیم:



در حالت ساده اش ما مراکز که V_1 تا V_3 است رو بدست میاریم تموم میشه

تو حالت پیچیده ترش ما میتونیم Base function هارو اینطوری بنویسیم:

$$e^{-\gamma (X-V_i)^T \Sigma_i^{-1} (X-V_i)}$$

یا به طور کلی تر برای base function ـi ام میتونیم اینطوری بنویسیم:

$$e^{-\gamma (X-V_i)^T \Sigma_i^{-1} (X-V_i)}$$

در حالت ساده ما یک سری دایره، کره یا ابر کره بسته به ابعاد داریم. در حالت بهبود یافته ما ناحیه بیضی چرخیده و بیضی های مورب و در حالت n بعدی ابرکره های هم بسته و مورب داریم.

V_i که شد مرکز خوشه ها. سیگما i چطوری بدست میاد؟ :

$$\boxed{\Sigma_i = \frac{\sum_{k=1}^{N} u_{ik}(X_k - V_i)(X_k - V_i)^T}{\sum_{k=1}^{N} u_{ik}}}$$

ماتریس کواریانس خوشه i ام

ماتریس کواریانس ماتریسی هستش که قطر هاش انحراف معیاره از بُعد اول تا بُعد n ام.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & c_{ij} & c_{1n} \\ c_{j1} & \sigma_j^2 & c_{jn} \\ c_{n1} & c_{nj} & \sigma_n^2 \end{bmatrix}$$

همبستگی بعد i ام و بعد j ام

$$c_{ij} = \rho_{ij}\, \sigma_i\, \sigma_j$$

معنی همبستگی: وقتی بُعد i ام افزایش پیدا میکنه بُعد j ام چقدر افزایش پیدا کنه. اگر به همون نسبت افزایش پیدا کنه هم بستگیش 100 درصد میشه و اون ρ_{ij} میشه 1. اگر در جهت عکس به همون نسبت کم بشه میشه -1. بنابراین یک مقداری بین -1 و 1 میگیرد:

$$-1 \leq \rho_{ij} \leq 1$$

در برخی از نسخه های FCM هم به جای فاصله اقلیدسی به نحو زیر از فاصله ماهانوبیس استفاده میشه:

$$(X - V_i)^T (X - V_i) = \|X - V_i\|^2$$ فاصله اقلیدسی

$$\|X - V_i\|^2_{\Sigma_i} = (X - V_i)^T \Sigma_i^{-1} (X - V_i)$$ فاصله ماهانوبیس

تکامل

تکامل: تطابق با محیط
هدف نهایی تکامل: بقای ژن برتر
چرایی تکامل: محدودیت منابع

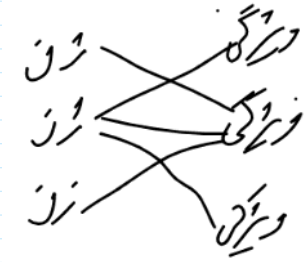
برای این که موجودات زنده بمونن و تولید مثل بکنن به یک سری منابع نیاز دارند که محدود است. در نتیجه یک رقابتی بین موجودات برای بدست آوردن منابع به وجود می آید و در این رقابت موجوداتی که بتوانند منابع را بدست بیاورند پیروز و باقی از بین میروند. در نتیجه موجودات برای پیروزی در این رقابت نیاز دارند خود را با محیط تطبیق بدهند. تطبیق با محیط خودش به 2 شکل صورت میگیره:

- به صورت جنگیدن با بقیه
- منبعی انتخاب شود که تقاضا برای آن کم باشد و به راحتی بدست آید. مثل داستان کرگدن و زرافه. زرافه با بلند کردن قد خودش با تکامل از برگ های بالاتر تغذیه میکند و کرگدن با برگ های پایین و دیگر دعوایی سر منبع نداریم.

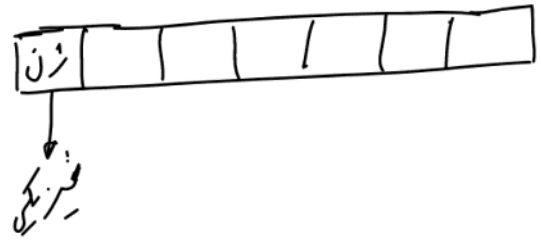
چگونه تکامل اتفاق میافته: یک بحث میکروسکوپی میشه کرد یک بحث ماکروسکوپی میشه کرد. بحث ماکروسکوپی اش به این صورته که یک موجود خود رو که نمیتونه تغییر بده. فرزندانش رو میتونه عوض کنه پس در تکامل تولید مثل خیلی مهمه. چون تولید مثل میتواند ویژگی هارو عوض کنه. اگر تولید مثل وجود نداشته باشه ویژگی ها عوض نمیشه.

چجوری این اتفاق میافته که فرزندان از والدین میتونن بهتر بشن؟ اینو باید از نظر میکروسکوپی بررسی کرد. همه موجودات ویژگی های فیزیکشون از ژن هاشون میاد. در نتیجه هر موجود زنده رو میشه با یک نقشه ژنتیکی معرفی کرد. ما در موجود زنده تعداد زیادی کروموزوم داریم که داخلشون ژن ها قرار میگیرن. ولی ما فرض میکنیم برای سادگی یک کرومزوم داریم که تمام ژن ها در آن باشند:

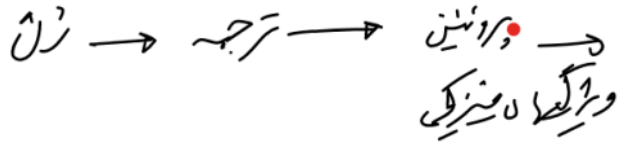
هر ژن یک یا چند فیزیکی رو تحت تاثیر قرار میده. در نتیجه رابطه ژن و رابطه فیزیکی چند به چنده:



ولی در حالت ساده و در الگوریتم تکاملی فرض میکنیم این رابطه یک به یکه. یعنی هر ژن یک ویژگی فیزیکی مثل قد و رنگ چشم رو در بر میگیرن.

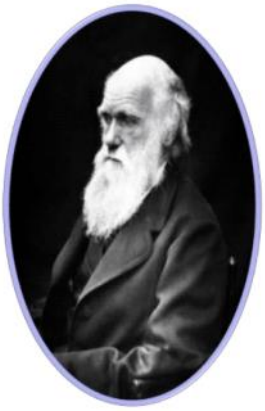


ژن ها کد هستند. این ژن ها در سلول ترجمه و Decode میشوند و یک پروتئین میشوند که این پروتئین ها ویژگی فیزیکی رو تعیین میکنند.



جلسه بیست و دوم

تئوری تکامل



تولد: 1809

وفات: 1882

منشا موجودات: 1859

تکامل توسط داروین معرفی شد.

تئوری تکامل

تکامل

شایستگی بالاتر فرزندان نسبت به والدین

ویژگیهای فیزیکی مناسبتر تطابق بیشتر با محیط

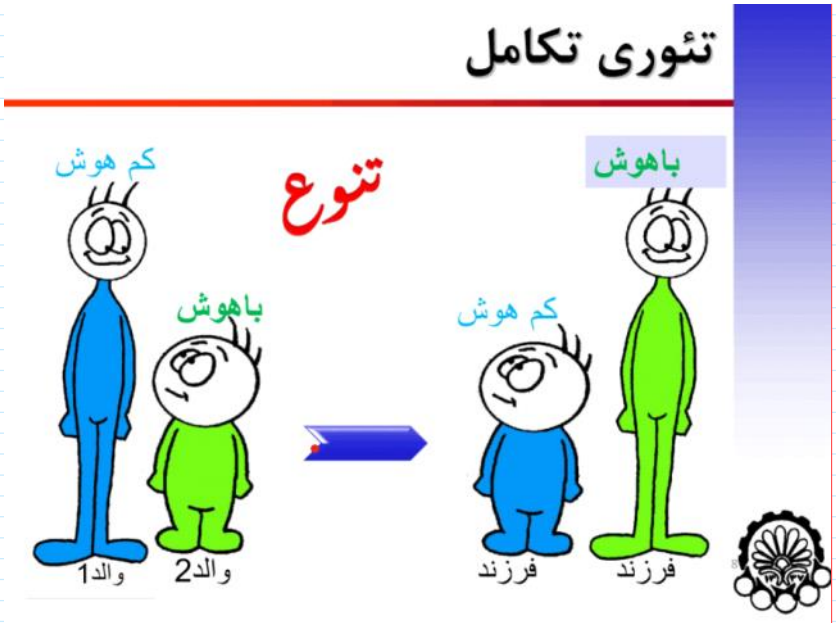
تولید فرزندان شایسته بیشتر

تکامل معانی مختلفی داره. که در اسلاید مشخص شده.

انتخاب طبیعت

قانون بقای شایسته ترینها

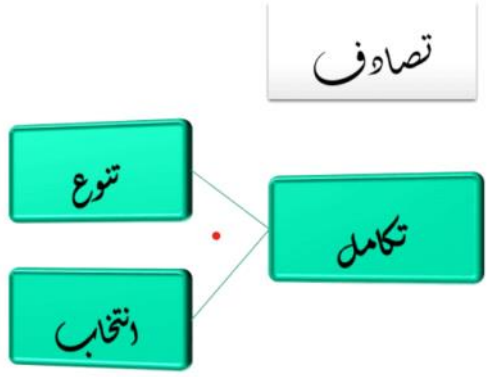
منظور از شایسته ترین ها کسایی هست که میتونن تولید مثل کنن و از منابع به خوبی استفاده کنن.



با وجود 2 والد که یکی کم هوش و قد بلند و یکی باهوش و قد کوتاه هستند فرزندان 4 حالت میتوانند داشته باشند. انتخاب است که فرزند باهوش و قدبلند طرفدار بیشتری داره. پس میتونه زوج ها و فرزندان بیشتری داشته باشه

ولی فرزند کم هوش و قد کوتاه کسی سمتش نمیره اگر بره هم خواهان و فرزندان کمتری به لحاظ آماری داره.

تئوری تکامل



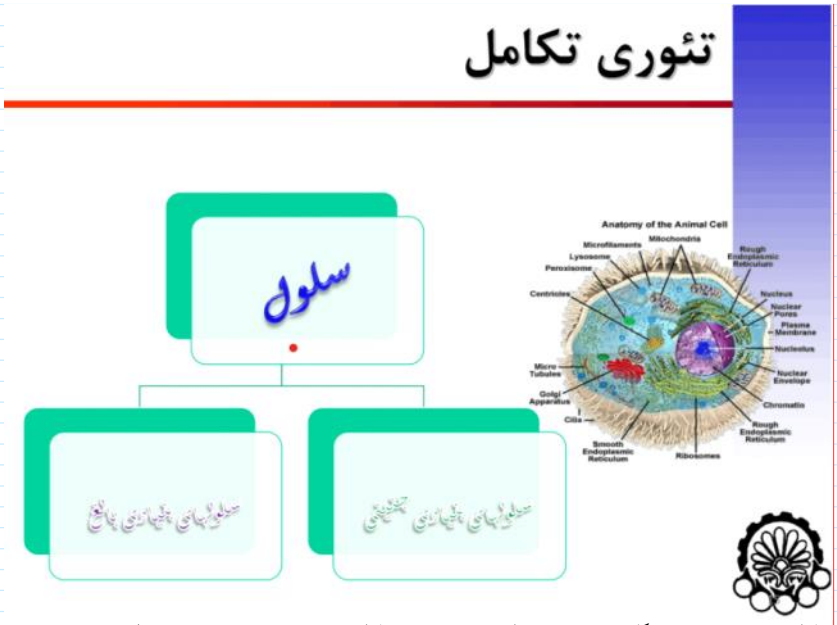
برای تنوع تصادف هم کمک میکنه تنوع داشته باشیم
انتخاب باعث میشه موجودات با شایسته انتخاب بشن و متوسط شایستگی رو میره بالا. ولی خود تنوع متوسط شایستگی رو میاره پایین. با این که این 2 ویژگی در تضاد هستن باید جفتشون باشن تا تکامل به وجود بیاد.
به دلیل تضاد باید بین این 2 یک تعادل باشه و به همین دلیل تکامل کُند هستش و باید در رنج میلیارد سال نگاه کرد بهش.

تنوع	انتخاب
واگرایی	همگرایی
تولید مثل	انتخاب طبیعت
افزایش کمیت	افزایش کیفیت

حالا میخوایم ببینیم

ویژگیهای فیزیکی والدین چگونه به فرزندان منتقل می شوند؟

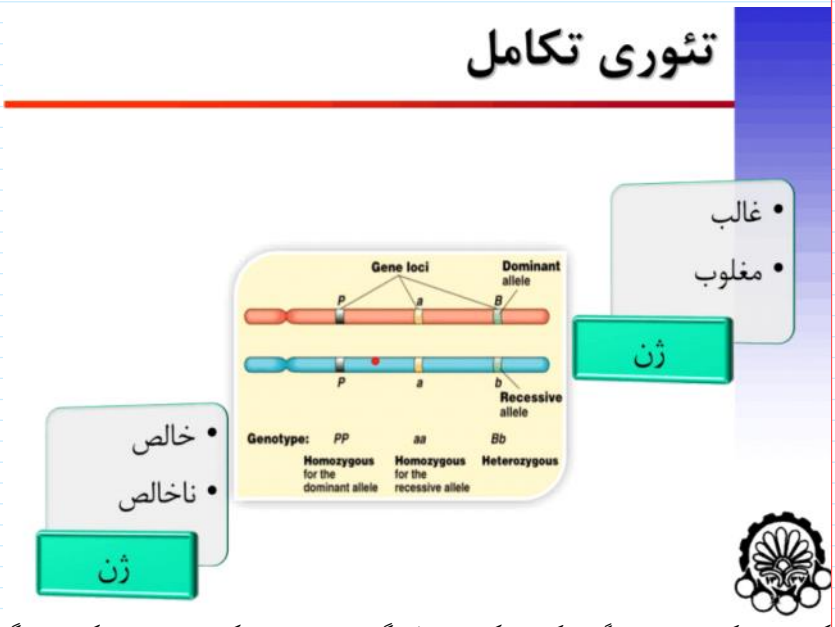
از دید میکروسکوپی هر موجود زنده یک سری تک سلول داره که به تک سلول اولی زیگوت میگن. زیگوت با تقسیم سلولی خود بسته به این که موجود تک سلولیه یا پر سلولی سلول های دیگر رو به وجود میاره. خود زیگوت از لقاح اسپرم و تخمک به وجود میاد که اسپرم از موجود نَر و تخمک از موجود ماده میاد. این اسپرم و تخمک با لقاح خود یک تک سلولی زیگوت رو درست میکنن و زیگوت با تقسیم سلولی موجودات پر سلولی مثل گیاهان و انسان ها و حیوانات رو به وجود میاره.



سلول خودش به دو دسته کلی بنیادی و معمولی تقسیم میشه. سلول بنیادی هم 2 تا داره جنینی و بالغ.
زیگوت اولین سلول بنیادی جنینی هستش که با تقسیم سلولی تبدیل میشه به یک سری سلول های بنیادی جنینی بعد سلول های بنیادی جنینی به یک تعدادی که رسیدن باز با تقسیم سلولی تبدیل میشن به سلول های بنیادی بالغ مثل مغز و استخوان یا تخصصی مثل پشم و گوشت و پوست و ... .
فرق بین سلول های بنیادی جنینی و بالغ: سلول های بنیادی جنینی به هر نوع سلولی مثل سلول های بنیادی بالغ و تخصصی میتونن تبدیل بشن ولی سلول های بنیادی بالغ فقط به یک دسته خاصی از سلول های بنیادی تخصصی میتونن تقسیم بشن.
بنابرین وقتی دستمون زخم میشه، به دلیل سلول های بنیادی بالغ که در مغز و استخوان هستن میان ترمیم میکنن و سلول ههای تخصصی پوست و گوشت رو درست میکنن. ولی اگر نخاع قطع بشه چون دیگه سلول های بنیادی جنینی وجود ندارن (بعد یک مدتی که نوزاد به دنیا میاد سلول های بنیادی جنینی تموم میشن) و فقط سلول های بنیادی بالغ و تخصصی داریم. سلول های بنیادی بالغ هم به هر نوع سلولی نمیتونن تبدیل بشن و سلول بنیادی بالغی نداریم ک بتونه نخاع تولید کنه و اگر قطع بشه دیگه ترمیم پذیر نیست.
به همین دلیل میان جدیدا سلول های بنیادی جنینی رو در یک بانک هایی نگهداری میکنن که اگر یک زمانی یک فردی مشکل قطع نخاع پیدا کرد با همین سلول های بنیادی جنینی مشکل قطع نخاع رو حل کنن (فقط یک فرد برا خودش رو میتونه).
البته برای حیوانات تونستن یک سری سلول های بنیادی جنینی تولید بکنن و در آینده برای انسان ها هم بتونن تولید کنن (ممکنه مخفیانه الان وجود داشته باشه). چون باید زیگوت درست بشه برای زیگوت هم اسپرم و تخمک مشابه نیاز هست که میشه همانند سازی. برای انسان که همانند سازی کردن زیگوت بعد چند هفته میمیره ولی برای حیوانات تونستن انجام بدن و گوسفند درست کنن. البته میتونن یک سری تغییرات بدن در DNA که مثلا گوسفند شیرش انسولین درست کنه و یک کارخونه انسولین سازی درست میکنن.

اون چیزی که ما در هسته کار داریم کرومزوم هستش که در خود کد هایی(ژن) ذخیره میکنن که ویژگی موجود زنده رو تعیین میکنن.

به مجموعه ژن ها ژنوم میگن.

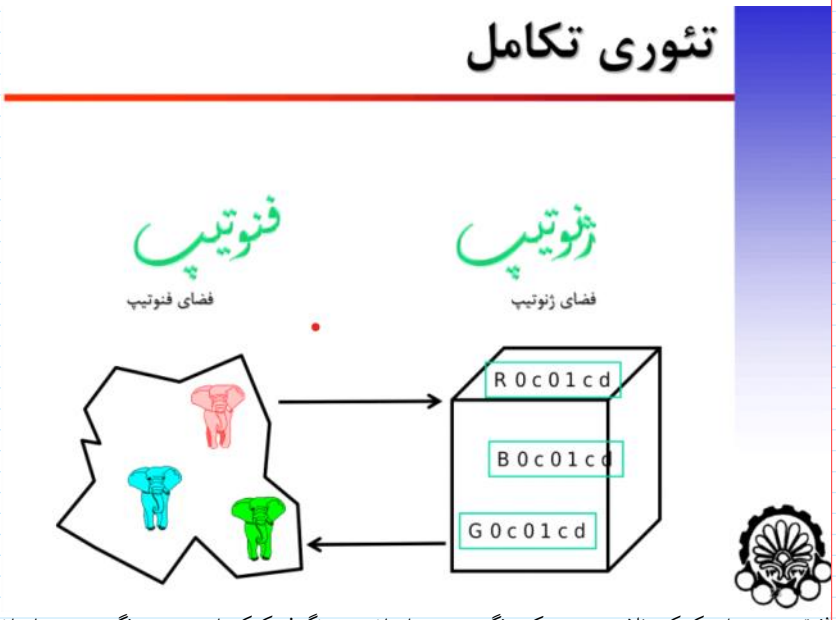


کرومزوم ها یک جفت هستن. اگر در یک جا از کروزموم مثلا رنگ چشم رو مشخص کنه در همون جا در کروزموم دیگه از اون جفت هم رنگ چشم رو مشخص میکنه. اگر هردو مثلا رنگ سیاه رو مشخص کنن بهش میگیم ژن خالص. اگر یکی سیاه باشه اون یکی مثلا قهوه ای باشه میگیم ژن ناخالص.

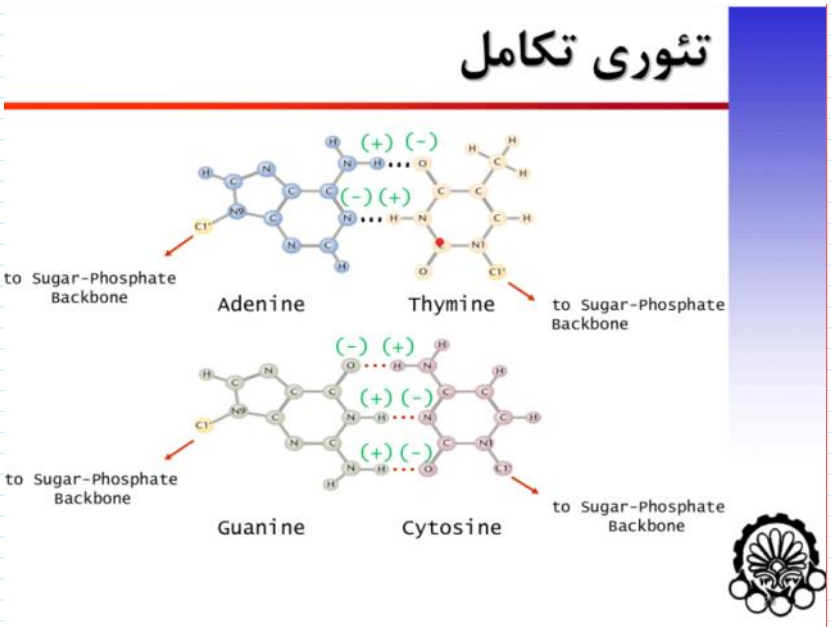
در الگوریتم ها ما کروموزوم ها رو جفتی در نظر نمیگیریم و داستان خالص و ناخلاصی و غالب و مغلوب رو نداریم.

وقتی ناخالصی داریم مثل رنگ سیاه و قهوه ای اونی که تو موجود ظاهر میشه میشه غالب اونی که نمیشه میگیم مغلوب. و این باعث میشه یک ژن مغلوب در نسل ها باقی بمونه بعد یهو ظاهر بشه و از انقراض نجات بده.

آزمایش مندل هم همین بحث غالب و مغلوب رو میگه این که چطوری یک ژن مغلوب بعد مدت ها ظاهر میشه که وارد بحثش نمیشیم.

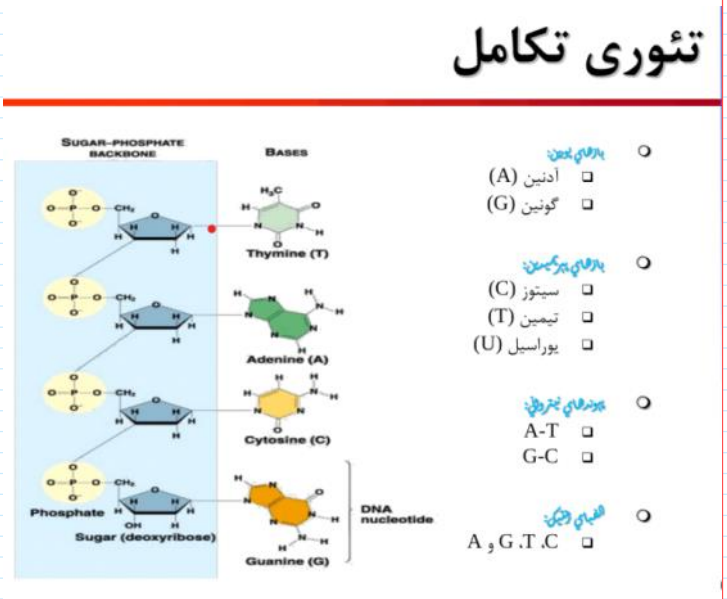


ژنوتیپ میشه اون کد که مثلا مشخص میکنه رنگ چشم سیاه باشه. و ویژگی فیزیکی که باعث میشه رنگ چشم سیاه باشه میشه فنوتیپ



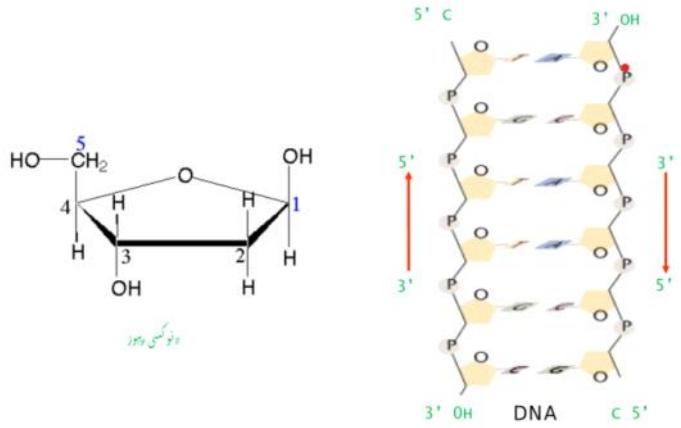
هر DNA مثل یک نرده بان هستش که پله هاش کد های ژنتیکی رو ذخیره میکنن و این کد های ژنتیکی یک سری باز های شیمیایی هستن که در اسلاید میبینیم. این ها 2 به 2 مکمل هستن و پیوند شیمیایی میتونن بقرار کنن و پله نرده بان ساخته بشه.

در نتیجه 4 تا کد ژنتیکی داریم که میگیم A و T و G و C و از اسم اون باز هایی اومد که پله های نرده بان را میسازه.



قسمت عمودی نرده بان هم یک گروه کربن و یک گروه فسفات هستش که باهاش کاری نداریم.

تئوری تکامل



چیزی که در این نرده بان اهمیت داره برا ما این کد هایی که در هر پله هستن هستش چون کد ها ژن هستن و ویژگی فیزیکی رو مشخص میکنن.

تئوری تکامل

Second letter

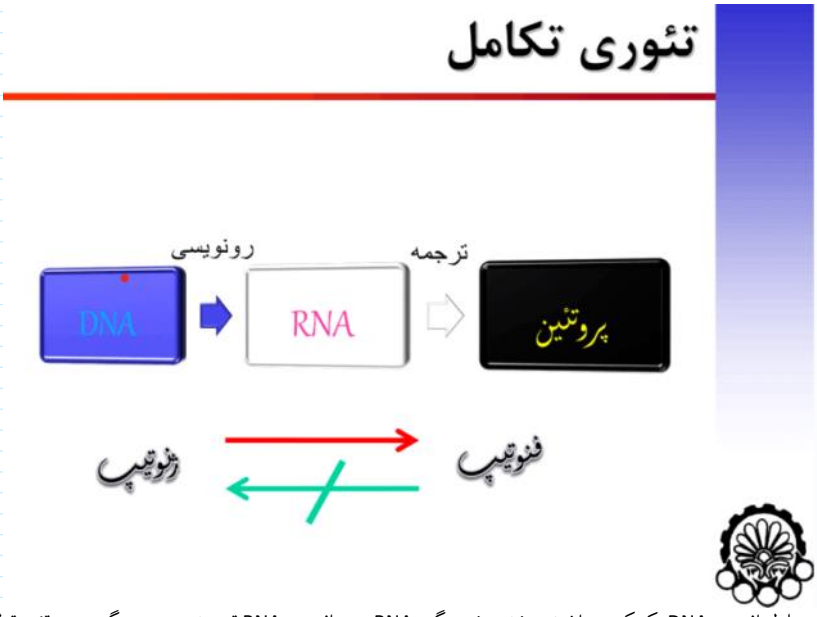
First letter	U	C	A	G	Third letter
U	UUU Phe UUC UUA Leu UUG	UCU UCC UCA Ser UCG	UAU Tyr UAC UAA Stop UAG Stop	UGU Cys UGC UGA Stop UGG Trp	U C A G
C	CUU CUC CUA Leu CUG	CCU CCC CCA Pro CCG	CAU His CAC CAA Gln CAG	CGU CGC CGA Arg CGG	U C A G
A	AUU AUC Ile AUA AUG Met	ACU ACC ACA Thr ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG	U C A G
G	GUU GUC GUA Val GUG	GCU GCC GCA Ala GCG	GAU Asp GAC GAA Glu GAG	GGU GGC GGA Gly GGG	U C A G

ویژگی های فیزیکی از پروتئین به وجود میان و پروتئین ها از نظر شیمیایی یک زنجیره ای از آمینو اسید ها هستن. و در نتیجه برای این که بفهمیم چه پروتئینی تولید میشه باید بفهمیم چه آمینو اسید هایی باید دنبال هم قرار بگیرن تا یک پروتئین خاص به وجود بیاد. و این که چه آمینواسید هایی کنار هم قرار بگیرن این کد های ژنتیکی مشخص میکنن. هر 3 تا کد که بهش میگیم کُدُن. یک کدی هستش که مشخص میکنه چه آمینو اسیدی داشته باشیم.

در DNA ما T A G C داریم و در RNA هم U A G C (فرق بین T و U رو کاری نداشته باشیم). در نتیجه هر 3 تا حرف میشه یک آمینو اسید. چند تا کد هم داریم که نشون میده ژن ها از کجا شروع میشن و چندتا داریم مثل UAA و UAG که نشون میده کجا تموم میشن.

یک پروتئین تعداد آمینو اسید هاش چندتاست؟ همه پروتئین ها از با یک کد یکسان شروع میشه و بسته به این که چه تعداد بعدش آمینو اسید بعدش هست و چه آمینو اسید هایی باشن پروتئین های مختلفی میشه ازش درست کرد خیلی میشه.

با توجه به تکراری بودن بعضی کد ها تعداد آمینو اسید هامون 21 تا هستش. حالا اگر به متوسط هر پروتئین 100 تا آمینو اسید داشته باشه. تعداد پروتئین ها میشه 21^{100} که باهاش میشه یک جهان هستی رو درست کرد.

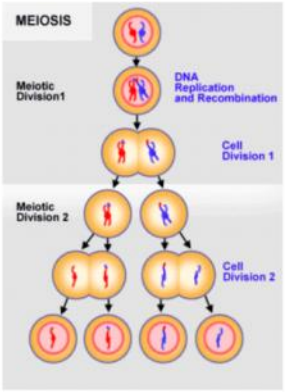


در سلول از روی DNA یک کپی برداشته میشه بهش میگیم RNA. بعد از روی RNA ترجمه صورت میگیره و پروتئین تولید میشه. و همه سلول ها یک کارخونه پروتئین سازی هستن.
ما مسیر از ژنوتیپ به فنوتیپ داریم ولی بالعکس رو نداریم. معنیش اینه وقتی یکی ورزش میکنه و ماهیچه قوی داره این ها به ژن و فرزندان نمیتونه انتقال پیدا کنه.

مسیر بالعکس رو لامارک اعتقاد داشت که بعدا این نظریه اش اثبات شد که اتفاق نمیافته.

ما 2 نوع تقسیم سلولی داریم:

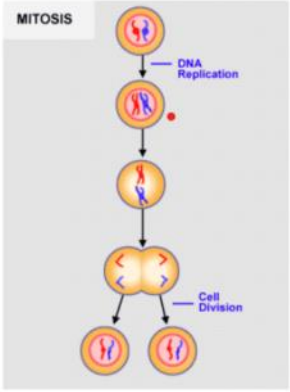
تئوری تکامل



تقسیم سلولی

میوز

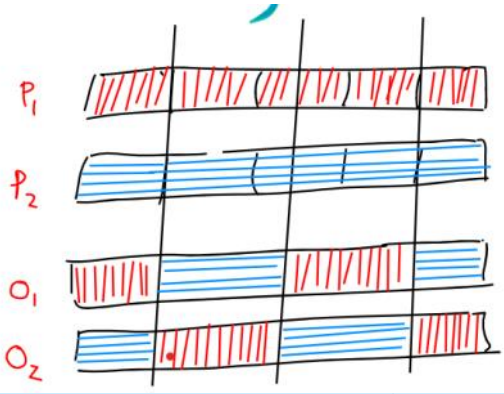
تئوری تکامل



میتوز

در میتوز وقتی زیگوت در میاد تقسیم میشه به 2 تا که این 2 تا میتونن اگر اولی بنیادی بود بالغ یا تخصصی باشن یا اگر اولی جنینی بود جنینی و بالغ بشه.
در میوز ویژگی والدین به فرزندان منتقل میشه. یک سلول جنسی با تقسیم سلولی تبدیل میشه به 4 تا اسپرم(نر) یا 4 تا تخمک(ماده) و اسپرم و تخمک لقاح پیدامیکنه و یک سلول فرزند رو تولید میکنه. که البته اینا دیپلوئیدی هستش که کاری باهاش نداریم.

در حالت هاپلوئیدی که باهاش سر و کار داریم فقط در تک سلولی ها هستش. در این حالت کروموزوم نر یک سری ژن ها داره. کروموزوم ماده هم یک سری ژن ها داره. وقتی که فرزند میخواد به وجود بیاد به صورت تصادفی از یک سری نقاط که جاش یا وسط ژن هستش یا جایی که ژن ها از هم جدا میشن، جابه جا میشن.
اگر ژن های نَر رو عمودی و قرمز و ماده رو آبی و افقی در نظر بگیریم داریم:



که به این مکانیزم میگیم Crossover یا بازترکیب. اگر crossover انجام نشه همون P1 وP2 رو خواهیم داشت ولی اگر بشه O1 و O2 رو داریم.

به صورت هاپلوئیدی آیا میشه ویژگی جدید داشت یا نه؟ اگر نقاط Crossover بین ژن ها باشن نمیشه ولی اگر داخل ژن باشه نصف کد ژن از یک والد میاد نصف دیگش از یک والد دیگه اون موقع میشه ویژگی جدید هم داشته باشیم.

در تکامل واقعی جهش یک نوعی خطا هستش که سعی میشه ازش جلوگیری هم بشه. ولی در الگوریتم های تکاملی مصنوعی نمیتونیم انقدر کُند عمل کنیم. میایم جهش هم بهش اضافه میکنیم و اینطوریه که به صورت تصادفی کد هارو عوض کنیم.

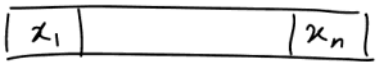
جلسه بیست و سوم

در جلسه قبل دیدیم اساس تکامل بر دو ویژگی هست. و این دو ویژگی:

- تنوع
- انتخاب

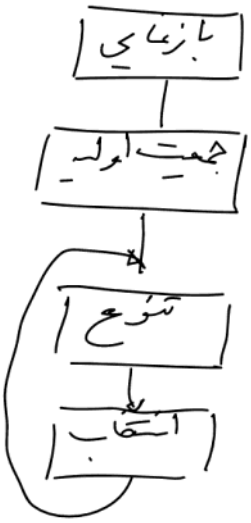
هستند. اگر از این ایده برای پیدا کردن مقدار بهینه یک تابع بخوایم استفاده کنیم چکار باید کرد؟
ابتدا باید یک تناظر بین تکامل و مساله خود به وجود بیاوریم. یعنی مساله ما به شکل ویژگی های تکامل در بیاد.

مثلا میخوایم یک تابع $f(x_1, ..., x_n)$ رو ماکسیمم کنیم، با توجه به این که ما در تکامل موجودات را با کرومزوم نشان میدهیم، ابتدا بایستی تعیین کنیم کروموزوم در مساله ما چه می باشد.
سپس بایستی شایستگی تعیین بشه. شایستگی میشه تابع هدف. که در مساله ما میشه همون f() ما.
ژن هامون میشه x_1 تا x_n.



ژن ها میشن متغیر هایی از مساله ما که تغییر آن متغیر ها باعث تغییر شایستگی بشه.

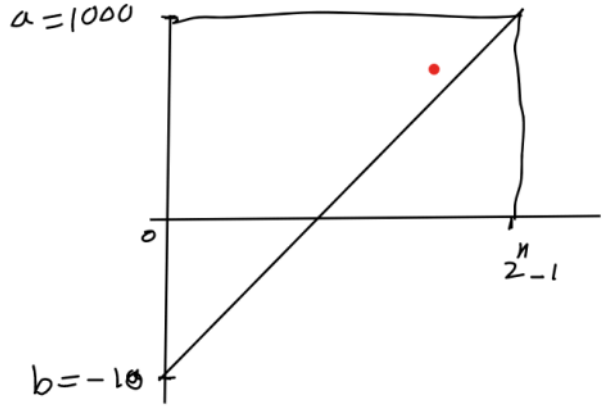
به گامی که انجام دادیم میگیم بازنمایی (کد کردن مساله). پس از بازنمایی ما نیاز به یک جمعیت اولیه داریم. سپس میتوان گفت:



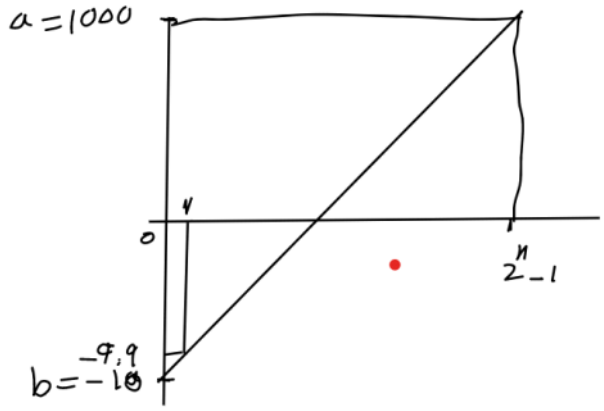
حالا جمعیت اولیه رو چطوری درست کنیم؟ جمعیت اولیه بایستی تصادفی یکنواخت باشه. **چرا؟** چون میخوایم تنوع زیاد بشه و با یکنواخت تولید کردن ما از تمام نقاط به صورت یکنواخت باعث میشه بیشترین تنوع رو در ابتدا داشته باشیم.

فرض شود بخوایم ماکسیمم تابع زیر را بدست آوریم:
$f(x) = x^{10} - 2x^5 - x^{100}$
در ابتدا بایستی بازنمایی کنیم و کروموزوم را بدست آوریم. برای این کار بستگی به دقتی که داریم تعداد ژن های داخل کروموزوم متغیره. مثلا اگر $-10 < x < 1000$ و دقت هم 0.1 باشه، چی میشه؟

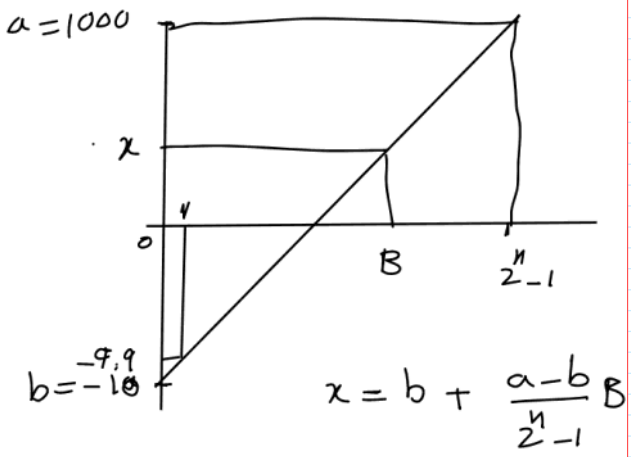
من با n بیت میتونم از 0 تا 2^n-1 رو نشون بدم. حالا ما از b=-10 تا a=1000 میخوایم عدد اعشاری داشته باشیم و با n بیت کمترین مقدار (-10) رو میخوایم با 0 نشون بدیم و بیشترین مقدار رو میخوایم با 2^n-1 نشون بدیم. در نتیجه یک رابطه خطی اگر بین اعداد اعشاری و باینری بدست بیاریم میشه این خط:



با دقتی که ما داریم عدد 1 باینری میشه -9.9:



حالا عدد اعشاری اگر x باشه عدد باینری چی میشه؟

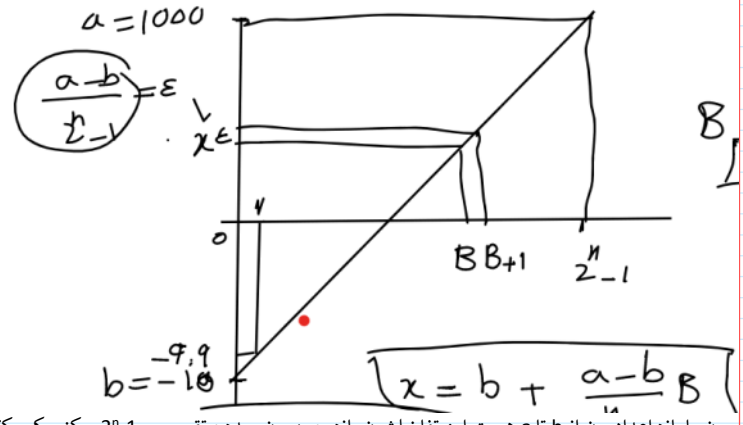


چطوری بدستش آورد؟ گفت من B رو صفر بذارم x میشه b اگرم B رو 2^n-1 بذارم میشه a و اینطوری بدست میاد.

حالا خود اون B هم یک عدد حقیقیه که از تبدیل باینری به دسیمال بدست اومده:

$$B = \sum_{i=0}^{n-1} b_i \cdot 2^i$$

پس ما یک سری 0 و 1 داریم اول اونارو تبدیل میکنیم به دسیمال که میشه بین 0 تا 2^n-1 بعد اونو مپ میکنیم بین -10 و 1000.
حالا اپسیلون که دقتمون هست چطوری بدست میاد؟



چون ما بازه اعدادمون از b تا a هست این تفاضلشون بازه رو بهمون میده و تقسیم بر 2^n-1 میکنیم کوچکترین واحدمون بدست میاد که میشه اپسیلون

$$\varepsilon = 0.1 = \frac{a-b}{2^n - 1} \Rightarrow 2^n - 1 = \frac{a-b}{\varepsilon}$$

$$2^n = 1 + \frac{a-b}{\varepsilon}$$

$$n \ln 2 = \ln\left(1 + \frac{a-b}{\varepsilon}\right)$$

$$\boxed{n = \frac{\ln\left(1 + \frac{a-b}{\varepsilon}\right)}{\ln 2}}$$

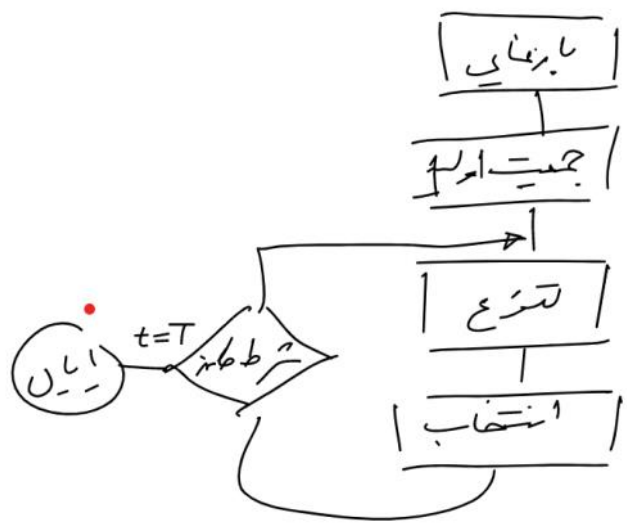
در مثال ما میشه:

$$\boxed{n = 14}$$

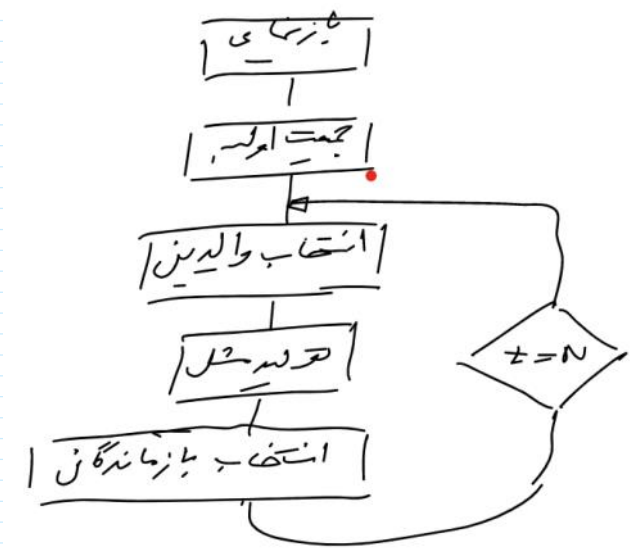
$$\frac{\ln\left(1 + \frac{10.0}{0.1}\right)}{\ln 2}$$

در واقع میشه 13.3 که چون تعداد بیت ها بایستی گسسته باشه سقفشو میگیریم که کم نیاریم.

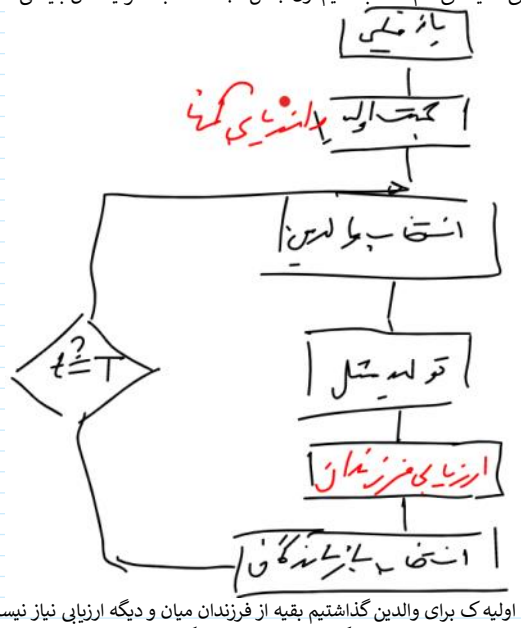
حالا اگر از a = 2.7 تا b = 274.5 داشته باشیم با دقت 0.05 اون موقع n چند میشه؟ میشه 12.4 یا به عبارتی 13 تا.
فرض کنیم میخوایم تعداد کروموزوم هامون 10 تا باشه. برای هر 13 ژن داخل این 10 کروموزوم مایم یک عدد بین 0 و 1 تولید میکنیم کمتر از 0.5 بود میگیم 0 بیشتر بود میگیم 1 (تصادفی یکنواخت)



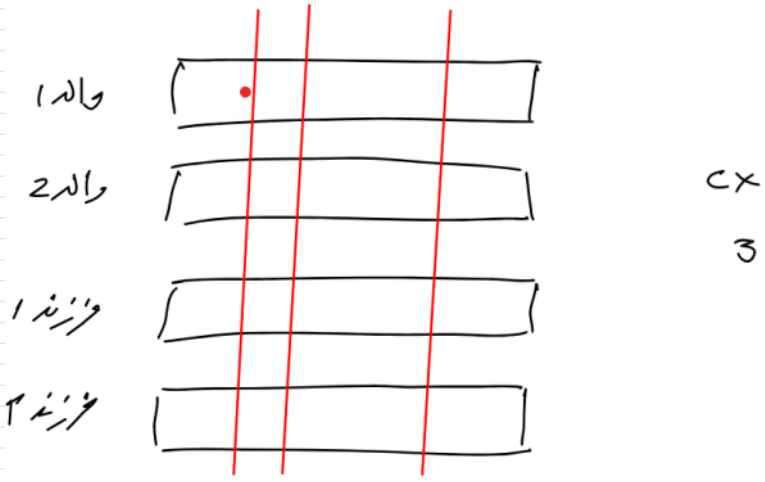
فلوچارت همچنان یه چی کم داره به طبیعت نزدیک بشه اونم انتخاب والدین و بازماندگانه، تنوع رو اگر تولید مثل در نظر بگیریم داریم:



منتهی بایستگی شایستگی هم حساب کنیم اون جاش کجاست؟ بعد تولید مثل بایستی شایستگی حساب بشه



غیر از ارزیابی اولیه ک برای والدین گذاشتیم بقیه از فرزندان میان و دیگه ارزیابی نیاز نیست. فقط مونده این که تولید مثل چجوری انجام میشه.
یکی از راهاش cross over هستش. یکی از پارامتر هاش اینه که از چند تا نقطه تقسیم بشه که حالا زیاد یا کمش چه مزیت و معایبی دارن جلوتر میگیم. فرض کنیم تعداد نقاط تقاطع 3 تا باشه. به صورت تصادفی یکنواخت سه نقطه روی کروموزوم انتخاب میشه:



پس از crossover داریم:

والد ۱	101	11	1011001	1001
والد 2	011	01	0001100	1100
فرزند ۱	101	01	1011001	1100
فرزند ۲	011	11	0001100	1001

CX

3

حالا میمونه جهش که

تعداد جمعیت

$$\frac{1}{n} < P_m < \frac{1}{\mu}$$

$$P_m = \frac{1}{10}$$

یک مقداریه بین تعداد بیت و تعداد جمعیت. یعنی برای هر کدوم یک عدد تصادفی بین 0 و 1 تولید میکنیم اگر از P_m کوچیکتر بود اون بیت رو عوض میکنیم.

فرزند ۱	101	0 ~~0~~	1011001	1100
فرزند ۲	011	11	0001101 ~~0~~	1001

پس الگوریتم:

۱- بازنمایی

۲- جمعیت اولیه و ارزیابی آن

۳- انتخاب والدین

۴- تقاطع

۵- جهش

تنوع یا تولید مثل

۶- ارزیابی فرزندان

۷- انتخاب بازماندگان

۸- شرط خاتمه یا برگشت به مرحله (۳)

$b \to B \to x \to f(x)$

جلسه بیست و چهارم

تو جلسه قبل دیدیم این مراحل رو داشتیم:

۱- بازنمایی کدکردن کروموزوم - شایستگی - تعیین روش ارائه ژنها

$b \to B \to x \to f(x)$

۲- جمعیت اولیه و ارزیابی

۳- انتخاب والدین

۴- تقاطع

۵- جهش

۶- ارزیابی فرزندان

۷- انتخاب بازماندگان

۸- شرط خاتمه برگشت (۳)

مساله n-وزیر رو با n=4 رو میخوایم با الگوریتم فوق حل کنیم.
در مرحله اول (بازنمایی) صفحه شطرنج رو تحت عنوان ژن ها در نظر میگیریم و هر ستون یک ژن باشه. هر ژن یدونه 1 داره بقیش 0 باشه.

۱- بازنمایی

0	0	1	0
1	0	0	1
0	1	0	0
0	0	0	0

ما هنگام بازنمایی میتونیم به هرنحوی بخوایم تعریف کنیم. جایگشت فوق میشه جایگشت باینری. یکی میاد میگه کروموزوم اینطوریه که هر ژن اش یک عدد باشه که داره نشون میده در ردیف چندم از اون ستون قرار داره. مثلا همین مثالی که زدیم در جایگشت عدد صحیح و باینری به این صورت میشه:

جایگشت صحیح

2	3	1	2

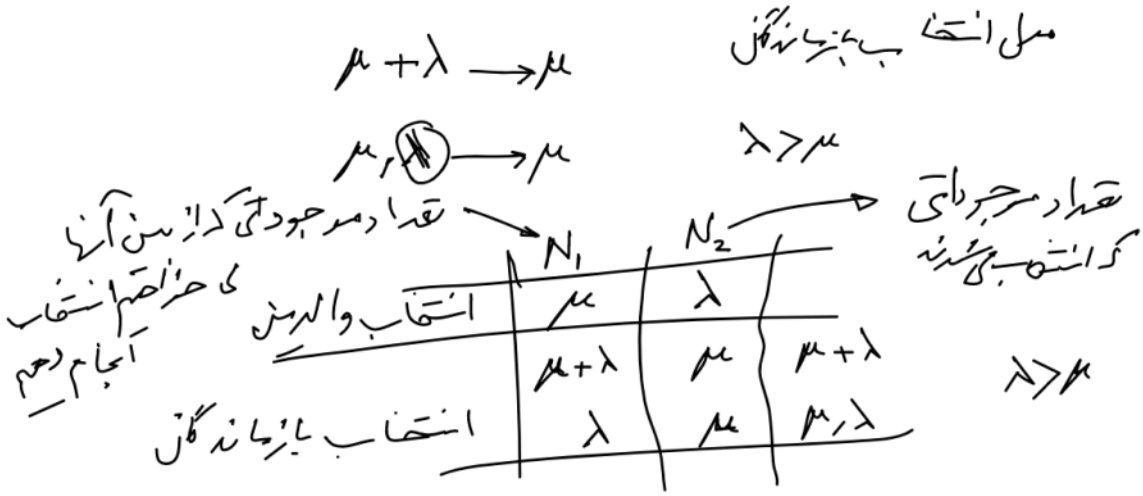
جایگشت باینری

0	0	1	0
1	0	0	1
0	1	0	0
0	0	0	0

اگر تعداد موجودات رو 4 فرض کنیم و تعداد فرزندان هم 4 باشه. μ میشه تعداد موجوداتی که به نسل بعد منتقل میشن λ هم میشه تعداد فرزندان.
--- نکات پرسیده شده حین حل سوال توسط بچه ها:
توی طبیعت والدا از بین میرن فرزندان میمونن ولی در الگوریتم های تکاملی لزوما اینطوری نیست. برخلاف طبیعت که جمعیت هی زیاد میشه در الگوریتم های تکاملی همیشه ثابت μ تا برای نسل بعدی انتخاب میشه.

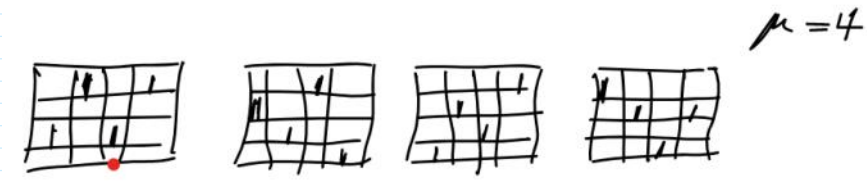
2 مدل انتخاب بازماندگان داریم. یکیش μ + λ هستش و دیگری μ, λ

- μ + λ: از بین موجوداتی که داریم و فرزندانی که تولید کردیم با یک روش انتخابی مثل متناسب با شایستگی میایم μ تا رو انتخاب میکنیم.
- μ, λ: فقط از بین فرزندان μ تا برای بعدی انتخای میشه. پس باید λ که تعداد فرزندان هست از μ بیشتر باشه که بتونیم μ تاشو انتخاب کنیم.

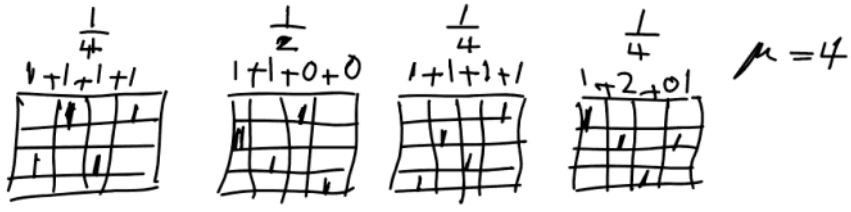


چراN_2 در انتخاب والدین میشه λ؟ چون تعداد فرزندان به اندازه تعداد والدین هستش. همیشه تعداد فرزندان برابر با تعداد والدینه. هر تعداد والدین انتخاب کنیم به همون تعداد فرزند تولید میشه.

---- بازگشت به مساله تعریف شده



با توجه به این که μ = 4 ما در ابتدا 4 تا جمعیت اولیه داریم. برای ستون هرکدوم میایم یک عدد تصادفی بین 1 تا 4 تولید میکنیم و سطر متناسب با اون عدد رو 1 میکنیم.
حالا بایستی ارزیابی کنیم. **ارزیابی رو چطوری انجام میدیم؟** میبینیم هر وزیر چندتا وزیر رو گارد میکنه. اون میشه هزینه و عکسش کنیم میشه شایستگی



این روش ارزیابی روش خوبی نیست **چرا؟** چون حالت های مختلف رو نتونست شایستگی های متفاوتی بده. توی ارزیابی خوب باید تو حالت های مختلف شایستگی های متفاوتی داشته باشیم.

سپس میرسیم مرحله سوم: انتخاب والدین
یک راهش اینه که برای هر موجود یک احتمال میگیریم که متناسب با شایستگی در نظر میگیریم.
یه راه دیگش شایسته سالاریه. موجودات رو بر اساس شایستگی مرتب میکنیم از ابتدای لیست تا اونجا میخوایم رو بر میداریم.

در این مثال چون μ = λ = 4 هستش از روش اولی استفاده میکنیم. چون روش شایسته سالاری میشه همه رو برداریم و به دردمون نمیخوره. اگر متناسب با شایستگی بخوایم بریم احتمال هاش میشن:

۳ـ انتخاب والدین $\lambda = \mu = 4$

$P_i \propto f_i$

$N_1 = \mu = 4$

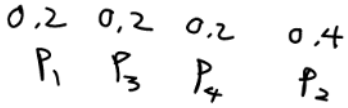
$$P_i = \frac{f_i}{\sum_{j=1}^{N_1} f_j} \rightarrow P_1 = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{2}+\frac{1}{4}+\frac{1}{4}} = \frac{1}{5}$$

$P_3 = P_4 = P_1 = 0.2 \quad P_2 = 0.4$

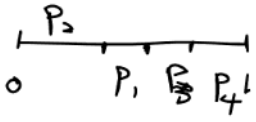
حالا ما میخوایم 4 تا موجود انتخاب کنیم.

-- در جواب سوال بچه ها که گفتن شایسته سالاری چطوری کار میکنه: ---

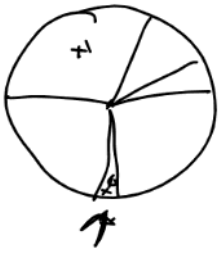
میایم اینطوری بر اساس شایستگی مرتب میکنیم و مثلا 4 تای اول رو بر میداریم
باید حتما $N_2 < N_1$ باشه.
-------- بازگشت به مساله:
احتمالامون اینطوری شد:



حالا میایم توی بازه بین 0 و 1 اینارو پخش میکنیم (ترتیبش مهم نیست):

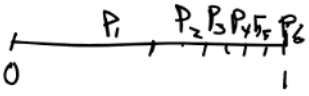


سپس یک عدد تصادفی یکنواخت بین 0 تا 1 تولید میکنیم. اگر بین 0 تا 0.4 بود P_2 انتخاب میشه. اگر بین 0.4 تا 0.6 بود P_1 . اگر بین 0.6 تا 0.8 بود P_3 و در آخر بین 0.8 تا 1 بود هم P_4.
به این روش میگن Roulette wheel. **چرا؟** چون مثل همون دایره چرخون در کازینو هاست:

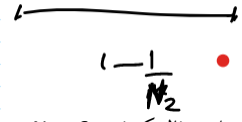


با توجه به این که N_2 حدود 10 الی 20 تا است و نمیشه اعداد کاملا تصادفی یکنواخت تولید کرد (تعدادشون چون کمه). یک روش تحت عنوان SUS (Stochastic Universal Sampling) که از نظر ریاضی مثل RW عه با این تفاوت که با تعداد کم هم Force میکنه یکنواختی رعایت بشه.

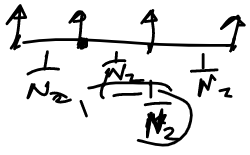
ما اگر بخوایم چهار تا موجود به صورت یک نواخت تولید کنیم یعنی این که اگر بین 0 تا 1 رو به 4 قسمت تقسیم کنیم باید از هر قسمت یدونه داشته باشیم.حالا میخوایم یه کاری کنیم این اتفاق بیافته.
برای این کار اون بازه بین 0 تا 1 رو در نظر بگیریم با این فرض که این شکلی باشه:



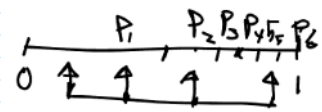
فرض کنیم میخوایم N_2 تا رو انتخاب کنیم. یک بازه به اندازه زیر تولید میکنیم:



الان در این مثالی که زدیم $N_1 = 6$ هستش ولی 4 تارو میخوایم انتخاب کنیم مثلا یعنی $N_2 = 4$ هستش. بعد میایم روی این بازه 3/4 تایی میایم $1/N_2$ تایی جدا میکنیم:



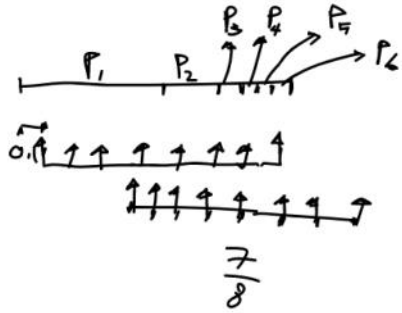
تو این مثال ما 3 قسمت میشه. حالا یک عدد تصادفی یکنواخت بین 0 تا $1/N_2$ انتخاب میکنیم. اون عدد میشه نقطه شروع بازه دوم که طولش $1 - 1/N_2$ بود:



با این کار انتخاب هامون با توجه به این فلش ها میشه:

P_1 P_1 P_2 P_5

یک مثال دیگه بزنیم بهتر جا بیافته. فرض کنیم 6 تا موجود داریم و 8 تارو میخوایم انتخاب کنیم. اینطوری میشه:



$N_1 = 6$

$N_2 = 8$

$1 - \frac{1}{N_2} = 1 - \frac{1}{8} = \frac{7}{8}$

عدد تصادفی نمیتونه $0 : \frac{1}{N_2}$

$0 : \frac{1}{8}$ $\quad$ 0.1

P_1 P_1 P_1 P_1 P_2 P_3 P_4 P_6

اگر n_i رو بگیم تعداد دفعاتی که موجود i ام انتخاب میشود. امید ریاضیش میشه انتگرال n_i در P_{ni} : که میشه:

$$E[n_i] = N_2 P_i$$

$P_i = \frac{1}{3}$ $\qquad$ $N_2 = 10$ $\qquad$ $E[n_i] = 10 \times \frac{1}{3}$

یعنی حدودا 3 و خورده ای و بین 3 تا 4 تا از n_i انتخاب میشه. (در مثالی که زدیم هم P_1حدود 0.5 بود 8 تا هم خواستیم انتخاب کنیم 8 * 0.5 = 4 تا انتخاب شد)

یک تعریف فشار انتخاب هم داریم که میشه امید ریاضی انتخاب بهترین موجود که برابره با:

فشار انتخاب $S_P = E[n_b] = N_2 P_b =$ تعداد موجوداتی که باید انتخاب شدند
ضرب در احتمال انتخاب بهترین موجود

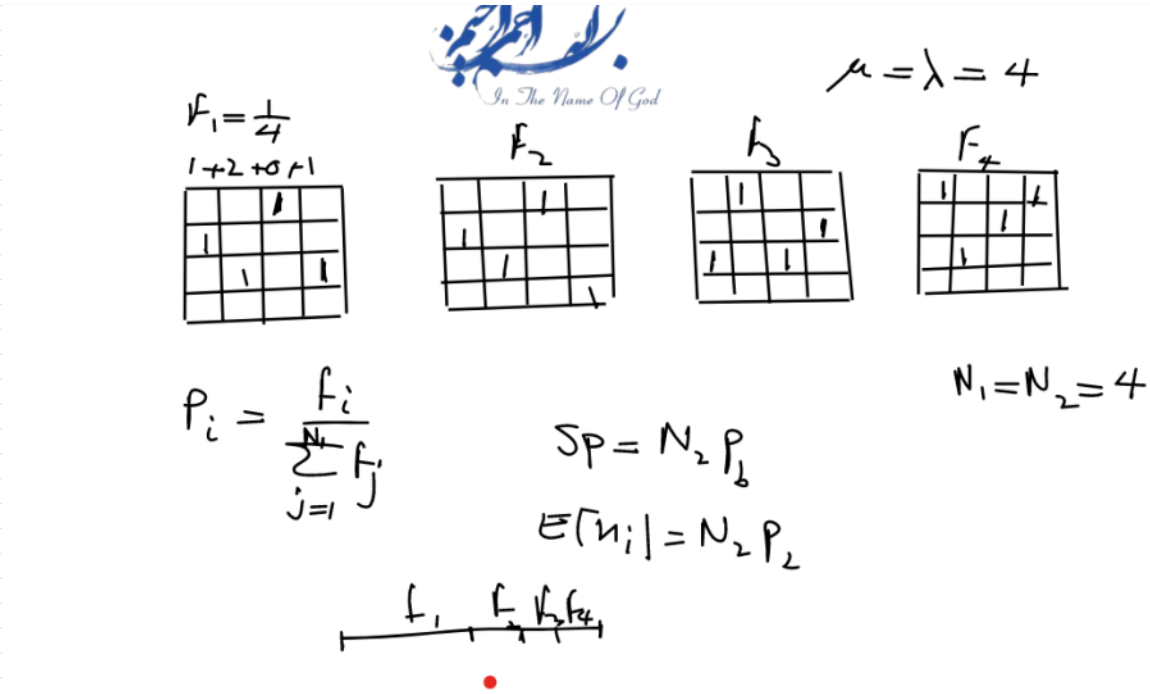
فشار انتخاب هرچی بیشتر باشه بدتره چون تکامل رو متوقف میکنه چون هرچی بیشتر باشه تنوع کمتر میشه.

جلوتر میبینیم روش انتخاب با شایستگی 2 تا مشکل داره:

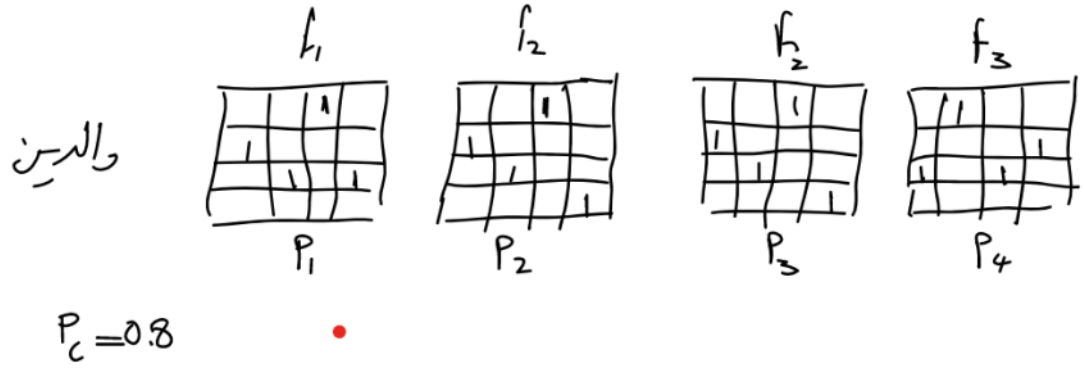
- ساکن شدن
- همگرایی زودرس

جلسه بیست و پنجم

12:08 PM Tuesday, June 8, 2021

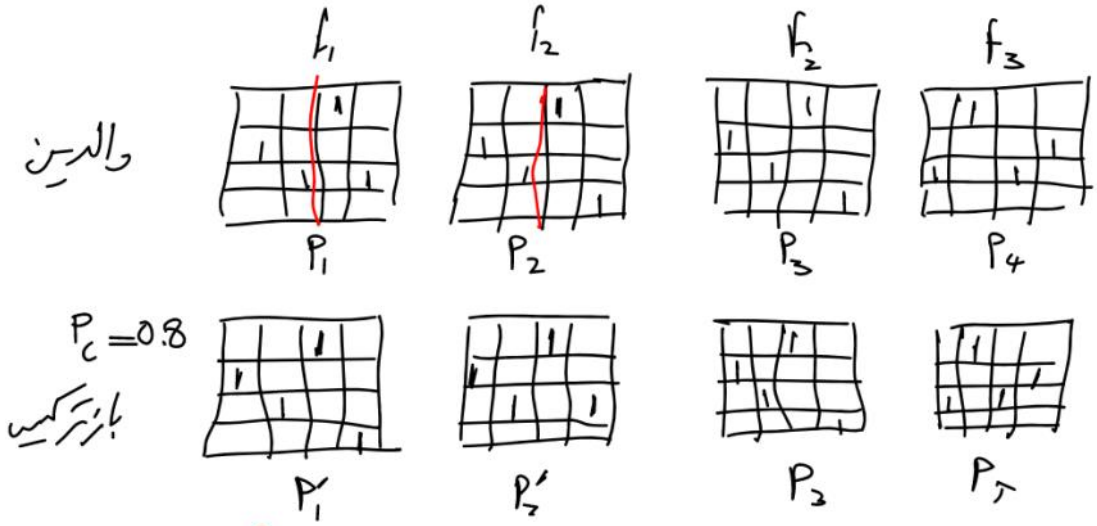


فرض کنیم انتخابا اینطوری بوده:

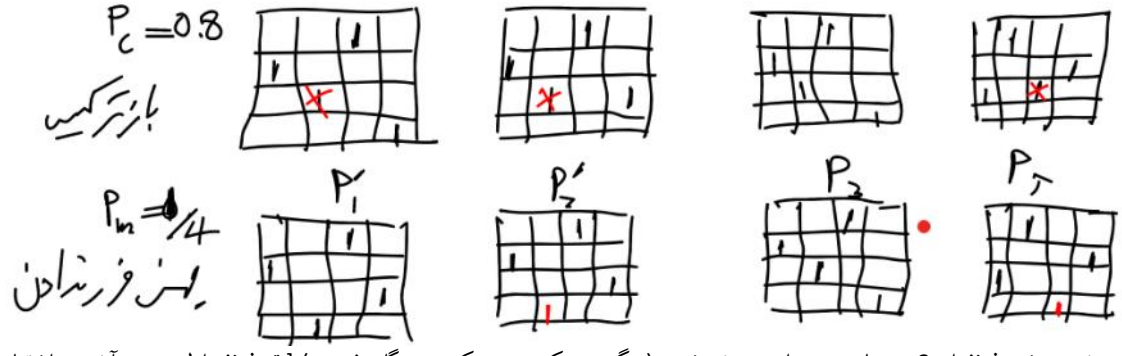


ما میایم $P_c = 0.8$ میزنیم. یعنی میایم بین هر جفت (جفت هامون P1, P2 هستند و P3, P4) یک عدد بین 0 و 1 تولید میکنیم. اگر کوچیکتر از 0.8 شد Crossover میکنیم بیشتر شد نمیکنیم.

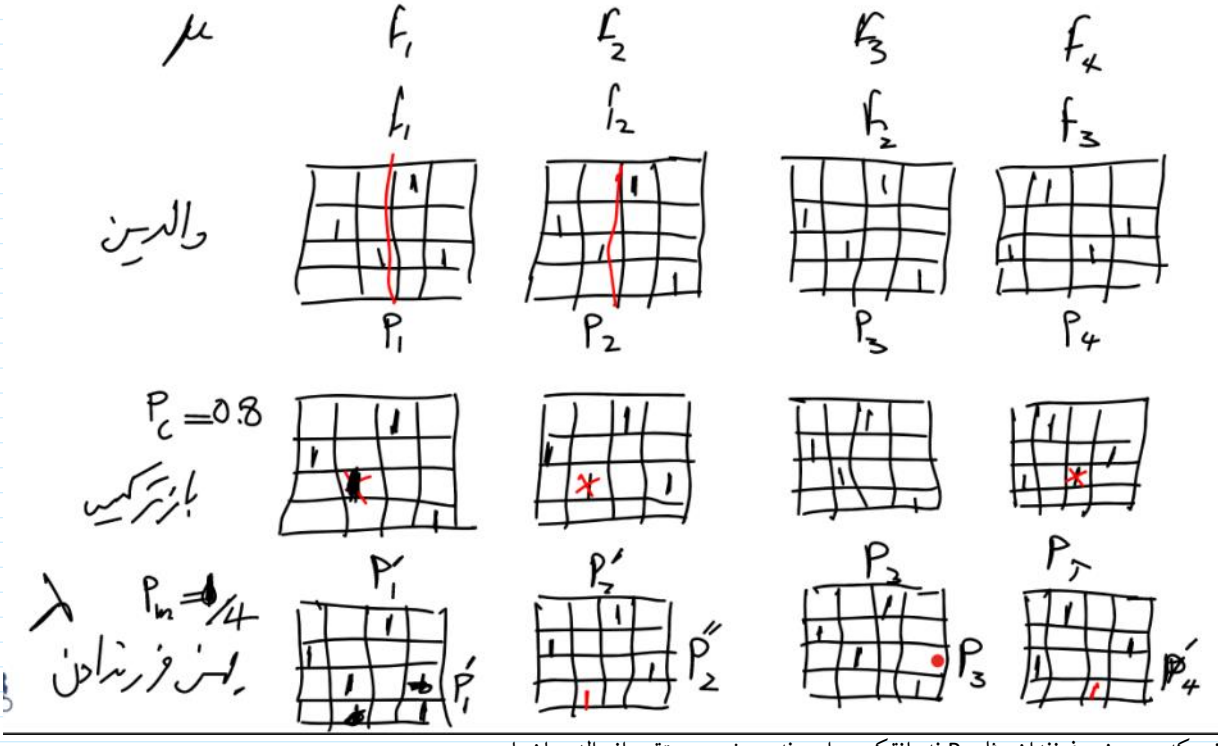
فرض کنیم برای P1 و P2 زیر 0.8 شه برای P3 و P4 بالای 0.8 بشه. فرزندان این شکلی میشه:



بعدش میایم با احتمال $P_m = 0.25$ جهش رو انجام میدیم. (جهش رو عددش رو کم میگیریم احتمالشو اینجا چون کم داریم نمونه هارو 0.25 گرفت)



بعد جهش میبینیم فرزندان 2 و چهارم به جواب منتهی شدن (دیگه هیچکدوم هیچکدوم رو گارد نمیدن) [تو فرزند اول ستون آخر رو اشتباهی یه خونه بالاتر گذاشت وگرنه یه خونه پایین تره].



میبینیم که در بعضی فرزندان مثل P_3 نه بازترکیبی داریم نه جهش و مستقیم از والد میان پایین.

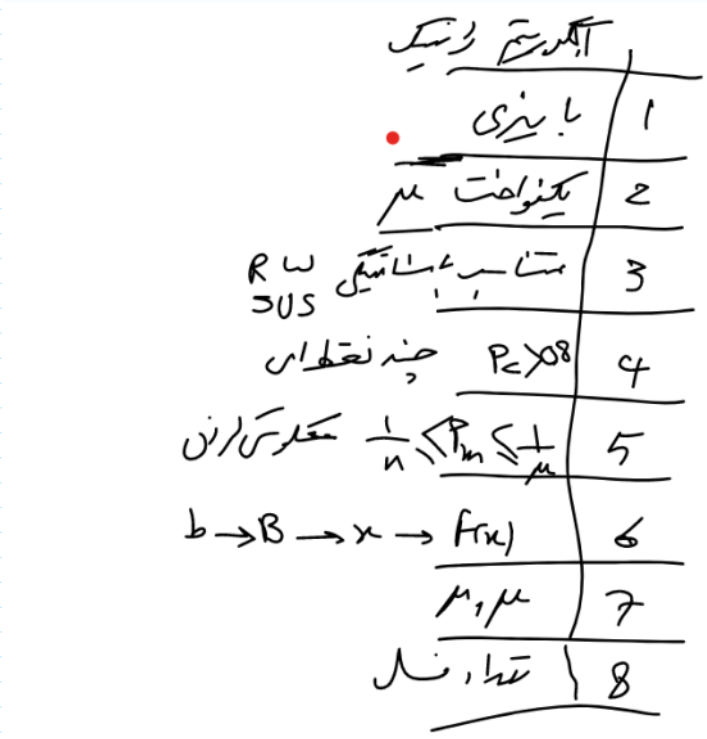
بعد انجام crossover و جهش، مرحله والدین و بازترکیبی رو حذف میکنیم. فقط μ و λ میمونه (فرق μ و والدین اینه که در μ از f1 تا f4 رو داریم ولی در والدین اول یه انتخابی بین اونا شده و f4 نیومده و f2 هم 2 بار اومده)

حالا میایم ارزیابی انجام میدیم بین p'_1و p''_2 و p_3 و p'_4. که حالا ارزیابی با μ,λ عه یا μ + λ.

در اینجا μ,λ هست و چون μ=λ میگیم μ,μ. یعنی فرزندان همشون میرن به نسل بعدی

به این روشی که استاد توضیح میده میگیم الگوریتم ژنتیک.

- بازنمایی به صورت باینری هستش.
- جمعیت اولیه تصادفی یکنواخت
- ارزیابی هم که bit باید تبدیل بشه int بعد تبدیل میکنیم به حقیقی و f(x) اشو حساب میکنیم
- انتخاب والدین متناسب با شایستگی RW - SUS
- در هنگام بازترکیبی یک P_c >= 08 داریم. به صورت چند نقطه ای میایم به صورت تصادفی یک نواخت انتخاب میکنیم.
- جهش بین $1/n < P_m < 1/\mu$ (البته اگه μ بزرگتر باشه جاشون عوض میشه). جهش هم به صورت bitwise میاد معکوس میکنه بیت رو
- بعد ارزیابی هم بیت رو میکنیم int، int رو میکنیم حقیقی و f(x) اشو حساب میکنیم.
- انتخاب بازماندگان به صورت μ,μ (چون μ=λ) و همه فرزندان به نسل بعد منتقل میشن.
- شرط خاتمه تعداد نسل



* اگر ما جهش نداشته باشیم هیچ موقع به جواب ممکنه نرسیم. چون مثلا تو جواب ما مثلا باید بیت دهم امون 1 باشه ولی تو جمعیت اولیه هیچ کدوم بیت دهم اش یک نباشه.

عیب های الگوریتم ژنتیک:

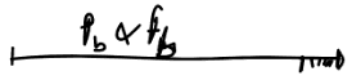
- متناسب با شایستگی بودن:

$$P_i \propto f_i \qquad P_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

  - همگرایی زودرس

$$f_0 \gg f_{b_1} \gg \bar{f}$$

فرض کنیم یک موجود داریم با شایستگیf_b که از شایستگی میانگین خیلی بیشتره و از شایستگی optimum هم کمتره. مثلا میخوایم به شایستگی 200 برسیم. موجودی داریم با شایستگی 90 و بقیه شایستگی اشون مثلا 1 و 1.1 و 0.9 و ایناس و روی هم شایستگیشون بشه 10.
چه با RW بریم چه با SUS اینطوری میشه که در اون خط کش ما 90 درصدش رو f_b میگیره 10 درصدشو بقیه:



فرض کنیم بخوایم 10 تا موجود رو بخوایم انتخاب کنیم. فشار انتخاب اینطوری میشه که :

$$S_p = 10 \times 0.9 = 9$$

یعنی 9 تا والد میشن f_b و دفعه بعد همگی میشن f_b و میشه که f_b جواب بهینه ما نیست. به این میگیم همگرایی زودرس
پس همرگایی زودرس چندتا شرط داره:

i. انتخاب ما متناسب با شایستگی
ii. فشار انتخاب بالا باشه. یعن f_b ما از شایستگی متوسط خیلی بیشتر باشه
iii. f_b با f_o فاصله داشته باشه.

- ساکن شدن

شرط هایی که موجب به ساکن شدن میشن:

i. انتخاب متناسب با شایستگی
ii. فشار انتخاب نداشته باشیم. یعنی f_b تقریبا با f متوسط یکی باشه و همه موجودات شانس انتخاب برابر داشته باشن

$$f_b \approx \bar{f} \qquad E[n_i] = 1$$

در این حالت اگر مثلا 8 تا موجود داشته باشیم احتمال همشون میشه 1/8 و شانس همه برابره. در انتخاب والدین همونا که قبلا بودن دوباره میان و چون جمعیت تنوع نداره کاری نمیتونه انجام بده. تو اینجا مثل همگرایی زودرس نیست که همشون یکی بشه. همشون نزدیک به هم میشن و همش تکرار میشن.

- **بازنمایی باینری:** هم مزایا داره هم معایب

منظور از بازنمایی نمایی اینه که مثلا میومدیم تابع $f(x) = x^{100} + 2x^{50} - x^{10}$ رو ماکسیمم کنیم میومدیم یه تعداد بیت براش در نظر میگرفتیم.

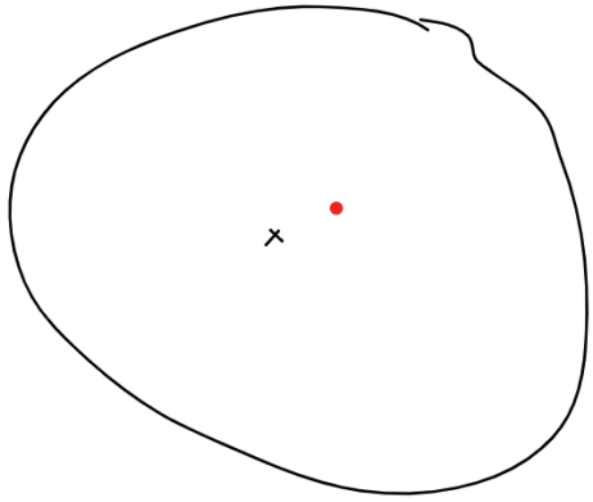
- معایب:
  - وقتی مثلا میایم یک تابع داریم 10 تا متغیر داشته باشه، برا هر کدوم 10 تا بیت بگیریم میشه 100 تا بیت و تعداد بیت زیادی رو باید مصرف کنیم.
  - فرض کنیم یک شرکتی میخواد طلا استخراج کنه. باید اول معدن رو پیدا کنه و کشف کنه. پیدا کردن این معدن هارو میگیم **اکتشاف** یا **جستجو عمومی.** بعد این که معدن رو کشف کردن باید طلا رو **استخراج** کنه که بهش میگیم **جستجو محلی**. این که چقدر دنبال اکتشاف باشیم چقدر دنبال استخراج یکی از مشکلات جستجو هستش.

اول باید محدوده جواب رو پیدا کنیم (جستجو عمومی) بعد بریم محدوده رو دقیق تر استخراج کنیم.

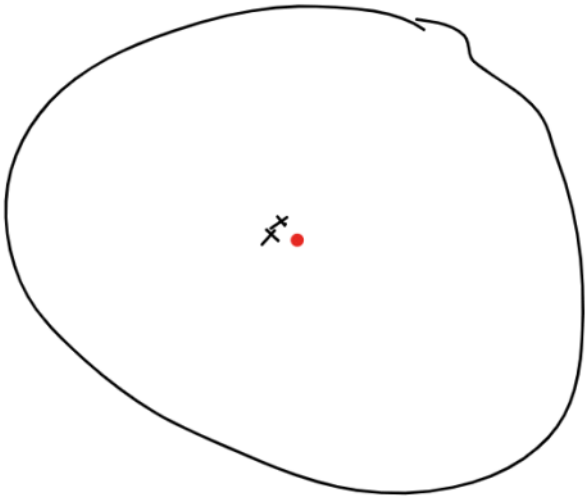
- **اگر اون شرکت اولین معدنی رو که پیدا کرد بره سمت استخراج چه اشکالی داری؟** ممکنه بدتر ضرر کنه و اون عیار داخل معدن هزینه استخراجشم حتی در نیاره.
- **اگر اون شرکت وسواس داشته باشه همش دنبال معدن بگرده چی؟** همش هزینه داره میکنه و هیچ موقع منفعت بدست نمیاره. استخراج رو بعضیا انتفاع هم میگن چون نفع میبرن. ولی اکتشاف رو فقط انجام بدیم هیچ موقع نفعی نداریم.

ما باید هزینه جستجو عمومی و نفع رو متعادل سازی کنیم.
در الگوریتم های جستجو معمولی اول اکتشاف انجام میدن بعد میرن سراغ استخراج.
در الگوریتم های تکاملی میشه اینارو همزمان انجام داد. ما یکی از عملگر های crossover و جهش رو میذاریم اکتشاف انجام بده اون یکیشرو میذاریم جستجو محلی انجام بده.

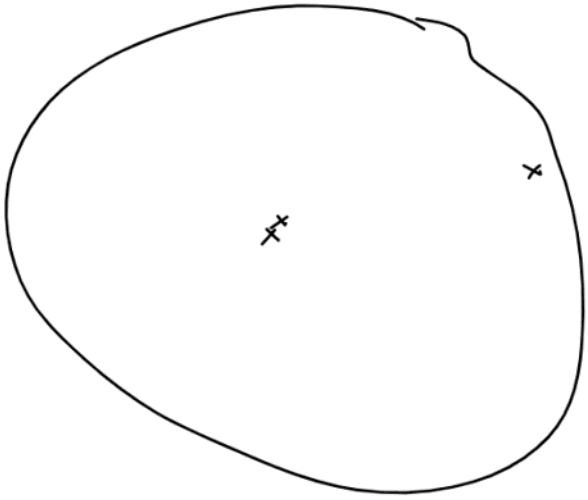
اگر ما بازنمایی باینری باشه جهش کدوم یکی میشه؟ بسته به این داره بیت با ارزش داره تغییر میکنه یا بیت کم ارزش



فرض کنیم این فضای جستجو ما باشه و یک موجودیتی اون وسط داریم. اگر بیت کم ارزش تغییر کنه میشه جستجو محلی چون تو همون حوالی فرق میکنه:



ولی وقتی بیت پر ارزش تغییر کنه میشه جستجو عمومی:



پس با بازنمایی نمایی ما نمیتونیم کنترل کنیم که جهش جستجو عمومی داره انجام میده یا محلی

Crossover چی؟ اونم باز بستگی داره کدوم قسمت داره جابه جا میشه ونمیشه تشخیص داد.

پس در بازنمایی باینری هیچ کنترلی نداریم جستجو عمومی داریم انجام میدیم یا محلی.

برای حل این مشکل حالا چیکار میشه کرد؟ یک راه حل سادش اینه که از gray code استفاده بشه. وقتی یک بیت رو در gray code تغییر بدیم مقدار عددیش تو همسایگی تغییر میکنه. مثلا اگر 4بیت در نظر بگیریم:

GROW

Decimal	Binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	01 11	0100
8	10 00	1100
9	1001	1101
11		

Two values differ in only one bit

LEARN

در این صورت وقتی یک بیت رو تغییر بدیم تغییر زیادی نداریم و وقتی gray code استفاده میکنیم جهش جستجو محلی انجام میده و crossover جستجو عمومی انجام میده.

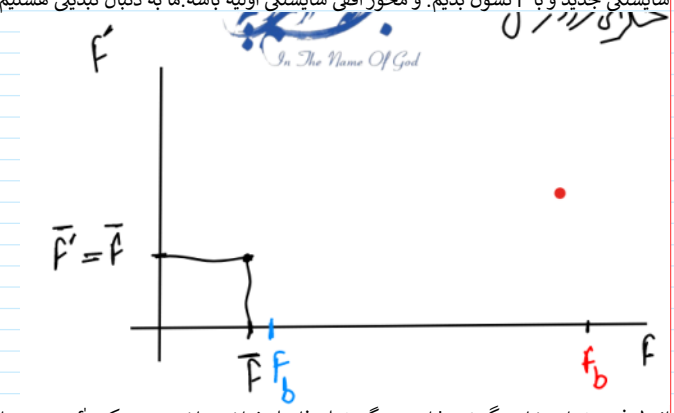
- مزیت: یک مساله بهینه سازی ترکیبی که یک سری متغیر ها int باشن یه سری float یه سری char همرو میشه با باینری کد کرد.

جلسه بیستو ششم

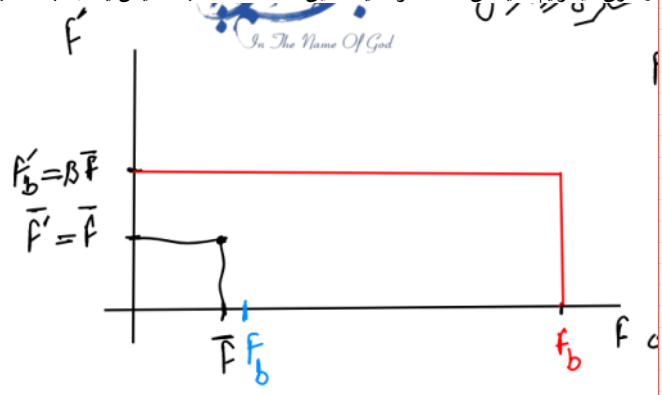
12:23 PM Sunday, June 13, 2021

در این جلسه میخوایم ببینیم برای مشکلاتی مثل همگرایی زودرس چه راه حل هایی داریم.

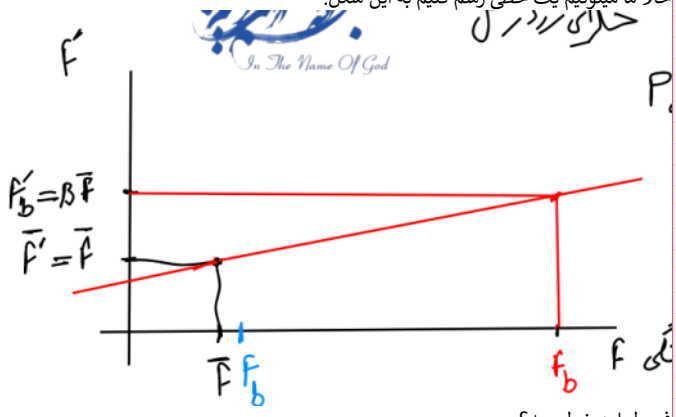
یک راه حل ساده که به ذهن میرسه اینه که روی شایستگی ها یک تبدیلی قرار بدیم و این تبدیل باعث بشه که اختلاف شایستگی خیلی زیاد یا خیلی کم نداشته باشیم. به این صورت که محور عمودی رو بگیریم شایستگی جدید و با f' نشون بدیم. و محور افقی شایستگی اولیه باشه.ما به دنبال تبدیلی هستیم که متوسط شایستگی رو تغییر نده:



از طرفی میخوایم شایستگی best امون دیگه خیلی فاصله نداشته باشه. یعنی یک f'_b بدست بیاد که بتا برابر متوسط باشه:



حالا ما میتونیم یک خطی رسم کنیم به این شکل:



فرمول این خط چیه؟

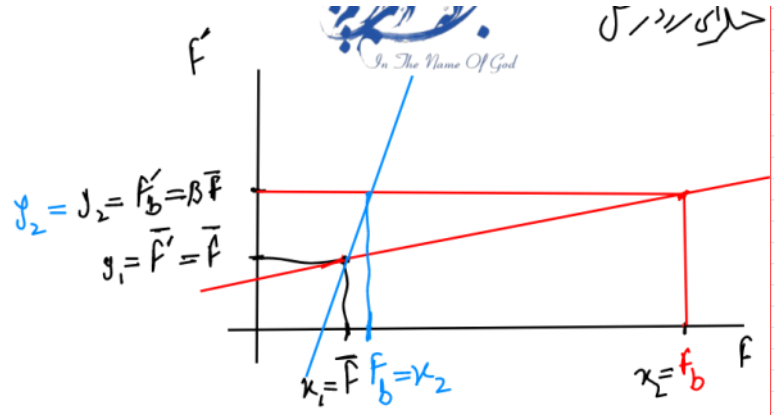
(x_1, y_1)

(x_2, y_2)

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$$

$$\boxed{f'_i = \frac{\beta \bar{f} - \bar{f}}{f_b - \bar{f}}(f_i - \bar{f}) + \bar{f}}$$

حالا اگر بخوایم برای ساکن شدن در نظر بگیریم:



میبینیم این رابطه که بدست آوردیم هم برای ساکن شدن هم همگرایی زودرس یکیه و جفتشو حل میکنه.
ولی این که بِتا رو چطوری بگیریم شاید حسی نداشته باشیم، میشه این فرمول رو به جای بتا در f متوسط بیایم بر حسب فشار انتخاب بگیریم و خودمون فشار انتخاب رو کنترل کنیم. اگر از فشار انتخاب استفاده کنیم داریم:
فشار انتخاب خودش خصوصیات جالبی داره. ما هرچی فشار انتخاب بالاتر باشه سرعت همگرایی بالاتره و هرچی کمتر باشه تنوع بیشتره. در نتیجه خودش یک پارامتر کنترلی برای تعادل بین تنوع و سرعت همگرایی به وجود میاره و اگر به جای بتا فشار انتخاب رو کنترل کنیم خیلی بهتر میتونیم الگوریتم تکاملی رو براساس این که ابتدا هستیم و تنوع نیاز داریم یا انتها هستیم و میخوایم به جواب برسیم کنترل کنیم. هرچه فشار انتخاب بیشتر باشه جستجو محلی تر و هرچی کمتر باشه عمومی تر میشه.

ما میدونیم جمع احتمالا باید برابر با 1 باشه

$$Sp = N_2 P_b \qquad \sum_{i=1}^{N_1} P_i = 1$$

توجه کنیم در احتمال N_1 عه ولی در فشار انتخاب N_2.
ما برای انتخاب لزومی نداره شایستگی رو حساب کنیم. در روش SUS (و اون یکیش) از روی شایستگی احتمال رو حساب میکردیم. توی اینجا P_i هارو باید با 'f های جدید بدست بیاریم:

$$P_i = \frac{f'_i}{\sum_{j=1}^{N_1} f'_j}$$

پس لزومی نداره ما شایستگی بدست بیاریم برای الگوریتم های تکاملی. میتونیم مستقیما احتمالات جدید بدست بیاریم. یعنی شایستگی رو همون شایستگی اولی داشته باشیم و احتمالای جدید بدست بیاریم. پس 2 تا معادله داریم و میخوایم P_i هارو خطی بر اساس شایستگی بیان کنیم و کنترل کنیم. پس P_i برابر میشه با:

$$P_i = a f_i + b$$

اگر بخوایمa و b در بیاریم میشه:

$$\underline{Sp = N_2 P_b} \qquad \sum_{i=1}^{N_1} P_i = 1 \qquad \underline{P_i = a f_i + b}$$

$$Sp = N_2 P_b = N_2 (a f_b + b) \rightarrow \boxed{a f_b + b = \frac{Sp}{N_2}} \quad (1)$$

$$\sum_{i=1}^{N_1} P_i = \sum_{i=1}^{N_1} (a f_i + b) = a \sum_{i=1}^{N_1} f_i + b N_1 = 1$$

$$a \frac{\sum_{i=1}^{N_1} f_i}{N_1} + b = \frac{1}{N_1} \rightarrow \boxed{a \bar{f} + b = \frac{1}{N_1}} \quad (2)$$

اینطوری a بدست میاد:

$$① - ② = a(f_b - \bar{f}) = \frac{S_p}{N_2} - \frac{1}{N_1} \rightarrow \boxed{a = \frac{\frac{S_p}{N_2} - \frac{1}{N_1}}{f_b - \bar{f}}}$$

حالا میخوایم b رو بدست بیاریم. اول میایم اون رابطه اصلی رو میذاریم 3:

$$\boxed{P_i = a f_i + b \quad ③}$$

و داریم:

$$P_i = a f_i + b \underset{②}{=} a f_i - a\bar{f} + \frac{1}{N_1} = a(f_i - \bar{f}) + \frac{1}{N_1}$$

$$\boxed{P_i = \frac{\frac{S_p}{N_2} - \frac{1}{N_1}}{f_b - \bar{f}}(f_i - \bar{f}) + \frac{1}{N_1}}$$

حالا تونستیم احتمال رو بر حسب فشار انتخاب بدست بیاریم.

حالا میتونیم فشار انتخاب رو طوری تعیین کنیم که همگرایی زودرس و ساکن شدن نداشته باشیم و همچنین میتونیم اونجا ها که تنوع میخوایم کوچیک فرض کنیم اونجاها که میخوایم سرعت همگرایی بره بالا ببریمش بالا.

فقط نکته ای که باید در نظر بگیریم اینه ک فشار انتخاب در بازه [1,2] هستش. خود 1 رو نمیشه ($1 < s_p <= 2$)

با یک روش ساده ترم میشه این کارو کرد. با این هدف موجوداتی که شایستگی هاشون خیلی فاصله داره رو کم کنیم و بالعکس اونا که کمه رو زیاد کنیم. برای این کار به جای شایستگی از رتبه استفاده میکنیم.

فرض کنید یک سری موجود داریم با یک سری شایستگی:

70	80	50	5	10	1	15

میایم اونی که کمترین شایستگی داره کمترین رتبه بدیم و بر اساس شایستگی به همه رتبه بدیم:

70	80	50	5	10	1	15
6	7	5	2	3	1	4

اگر sortکنیمشون داریم:

1	5	10	15	50	70	80
1	2	3	4	5	6	7

حالا اگر بخوایم احتمال حساب کنیم، داریم:

$$f_i = i \qquad P_i = \frac{f_i}{\sum_{j=1}^{N_1} f_j} = \frac{i}{\sum_{j=1}^{N_1} j} = \frac{2i}{N_1(N_1+1)}$$

این دفعه این شکلی میشه:

$$\boxed{P_i = ai + b \quad (3)} \qquad Sp = N_2 P_b \qquad \sum_{i=1}^{N_1} P_i = 1$$

$$Sp = N_2 P_b = N_2 (aN_1 + b) \rightarrow \boxed{aN_1 + b = \frac{Sp}{N_2} \quad (1)}$$

$$\sum_{i=1}^{N_1} P_i = \sum_{i=1}^{N_1} (ai + b) = a \sum_{i=1}^{N_1} i + N_1 b = 1$$

$$a \frac{N_1(N_1+1)}{2} + N_1 b = 1 \rightarrow \boxed{a \frac{N_1+1}{2} + b = \frac{1}{N_1} \quad (2)}$$

$$(1) - (2) = a\Big(\underbrace{N_1 - \frac{N_1+1}{2}}_{\frac{N_1-1}{2}}\Big) = \frac{Sp}{N_2} - \frac{1}{N_1} \qquad \boxed{a = \frac{\frac{Sp}{N_2} - \frac{1}{N_1}}{\frac{N_1-1}{2}}}$$

$$P_i = ai + \underset{\substack{\uparrow \\ (2)}}{b} = ai - a\frac{N_1+1}{2} + \frac{1}{N_1}$$

$$P_i = a\left(i - \frac{N_1+1}{2}\right) + \frac{1}{N_1}$$

$$\boxed{P_i = \left(\frac{Sp}{N_2} - \frac{1}{N_1}\right) \frac{i - \frac{N_1+1}{2}}{\frac{N_1-1}{2}} + \frac{1}{N_1}}$$

جمع بندی: روش های انتخاب

روش‌های انتخاب

۱- تصادفی یکنواخت

۲- بدون انتخاب

۳- شایستگی سالاری

۴- متناسب با شایستگی $P_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$ SUS RW

۵- متناسب با شایستگی مقیاس شده خطی $f'_i = af_i + b$ $P_i = \frac{f'_i}{\sum_{j=1}^{N} f'_j}$ SUS RW

۶- متناسب با شایستگی مقیاس شده خطی $P_i = af_i + b$ SUS RW
با کنترل فشار انتخاب $SP = N_2 P_1$ $\sum P = 1$

۷- متناسب با رتبه $f_i = i$ $P_i = \frac{2i}{N_1(N_1+1)}$ RW SUS

۸- متناسب با رتبه مقیاس شده خطی $P_i = ai + b$ RW SUS
با کنترل فشار انتخاب $SP = N_2 P_1$ $\sum P = 1$

مزیت 7 و 8 نسبت به 5 و 6 چیه؟ توی متناسب با رتبه مشکل همگرایی زودرس و ساکن شدن حل میشه. چون شایستگی ها یکی یکی داره اضافه میشه و f_0 با f قبلیش یدونه بیشتر نیست و یک موجود نمیتونه فاصلش با بقیه زیاد بشه چون یکی یکی دارن زیاد میشن. شایستگی ها هم با هم برابر نمیتونن بشن چون یکی یکی فاصله دارن.
مزیت خیلی جالب دیگش اینه که شایستگی میتونه حساب نشه. خیلی وقتا محاسبه شایستگی خیلی سخته. مثلا بخوایم به یک صفحه شطرنج شایستگی بدیم ممکنه خیلی سخت باشه ولی 2 تا صفحه شطرنج بدن بگن کدوم شایستگی بهتری داره راحت تر میشه جواب داد. یعنی باید به صورت کیفی مقایسه کنیم نه کمی. یا مثلا 2تا تابلو نقاشی بدن بگن نمره بدیم سخته ولی بگن کدوم بهتره راحت تر میتونیم بگیم. در نتیجه مقدار دقیق شایستگی نیازی نیست و باید یک تفاوتی ایجاد کنیم و sortکنیم و تابع دقیق شایستگی نیاز نیست. کافیه یا تابع غیر دقیق داشته باشیم یا انسان و چیز دیگه ای بیاد sort رو انجام بده.

جلسه بیست و هفتم

12:42 PM Tuesday, June 15, 2021

دو مرحله انتخاب داریم:

- والدین
- بازماندگان

گفتیم تکامل در دو صورت رخ میده:

- تنوع
- انتخاب درگیر با شایستگی

یک روش انتخاب در این جلسه یاد میگیریم تحت عنوان Q تورنومنت که مزیت همه روش های قبلی رو داره. Q یه جورایی شبیه فشار انتخابه. Q تا موجود رو به صورت تصادفی یک نواخت از جمعیت بر میداریم (اگر Q=2 باشه بهش میگیم باینری تورنومنت). در باینری تورنومنت 2 تا موجود به صورت تصادفی یک نواخت برداشته میشه، در بین این 2 موجود بهترین رو برمیداریم. این میتونه با جایگذاری باشه یا بدون جایگذاری

- در جایگذاری اون 2 موجود رو که برداشتیم میتونیم به جمعیت اضافه کنیم.
- در بدون جایگذاری دیگه به جمعیت اضافه نمیکنیم.

Q تورنومنت مزایای رتبه بندی رو داره یعنی مشکل همگرایی زودرس و ساکن شدن رو نداره. احتیاجی نیست مقدار شایستگی رو دقیق حساب کنیم.

مزیتی که Q تورنومنت نسبت به رتبه بندی داره: در رتبه بندی باید بر حسب شایستگی sort کنیم ولی در Q تورنومنت در بین Q تا ماکسیمم رو برمیداریم. در نتیجه هزینه محاسبه Sort کردن رو نداریم و در نتیجه از این جهت نسبت به رتبه بندی مزیت داره.
در رتبه بندی فشار انتخاب همیشه بین 1 و 2 بود، اگر بخوایم بیشتر از 2 باشه باید از Q تورنومنت بریم. وقتی Q=2 میذاریم انگار فشار انتخاب یه جورایی 2 هست (در صورتی که با جایگذاری باشه، بدون جایگذاری باشه یه موجود بیشتر از یه دفعه نمیتونه انتخاب بشه و در نتیجه فشار انتخاب حقیقی میشه 1. هرچند مشکل همگرایی زودرس و ساکن شدن رو نخواهد داشت)

* تو اسلایدا هست وقت نمیشه سرکلاس گفت ولی میشه نشون داد Q تورنومنت ادامه رتبه بندی خطی هست و تمام مزایا رو داره و sort کردنم توش نداره.

جمع بندی روش های انتخاب:

۱- تصادفی یکنواخت

۲- انتخاب همگی μ, μ

۳- شایستگی سالاری

۴- متناسب با شایستگی $P_i = \frac{f_i}{\sum_{j=1}^{N_1} f_j}$ SUS RW

۵- " " " متقارن شده خطی $f'_i = a f_i + b$ $\bar{f}' = \bar{f}$ $f'_b = \beta \bar{f}$ $P_i = \frac{f'_i}{\sum_{j=1}^{N_1} f'_j}$ RW SUS

۶- " " " " با کنترل فشار انتخاب $P_i = a f_i + b$ $SP = N_2 P_b$ $\sum P_i = 1$ RW SUS

۷- متناسب با رتبه $f_i = i$ $P_i = \frac{2i}{N_1(N_1+1)}$ RW SUS

۸- " " " متقارن شده خطی با کنترل فشار $P_i = a i + b$ $SP = N_2 P_b$ $\sum P_i = 1$ RW SUS

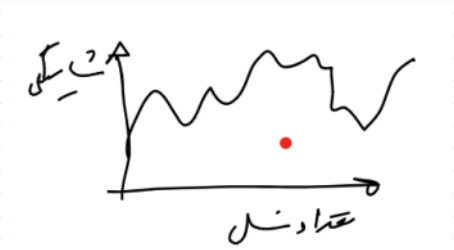
۹- Q تورنومنت

برای انتخاب بازماندگان 2 مدل داشتیم.

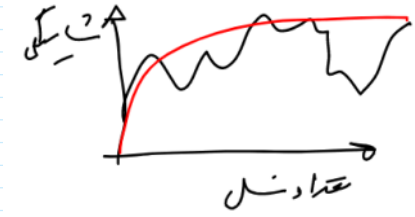
- μ+λ: از بین μو λ میایم μ تا بر میداریم.
- μ,λ: فقط از فرزندان μ تا برمیداریم.

هرکدوم از اینا یه ویژگی خاصی داره:

- μ,λ فراموش کاره (والدارو فراموش میکن) ولی μ+λ حافظه داره. در نتیجه چون μ,λ والدارو فراموش میکنه فرزندان جدیدن و **تنوع بیشتری داره.** ولی μ+λ سرعت همگرایی بیشتری داره.



با توجه به این که μ,λ فراموش کاره شایستگیش نوسان داره و هی زیاد و کم میشه. ولیμ+λ وقتی رو رسم میکنیم:



دیگه نوسان نداره و همگرا میشه.

- μ,λ چون فراموش کاره فرار میتونه بکنه از مینیمم محلی ولیμ+λ توش گیر میکنه
- μ,λ از پارامتر های بد میتونه فرار کنه (اگه بعدا وقت شه توضیح میده استاد که چیه)

ما قبلا هم دیدیم یه تعادلی بین جستجو عمومی و جستجو محلی باید برقرار کنیم. یا به طور دقیق تر یک تعادلی بین تنوع و همگرایی اینجاد کنیم (چون برای تکامل جفتشو نیاز داریم). اگر همش بریم سراغ تنوع خوب نیست، همشم بریم سراغ شایستگی و همگرایی خوب نیست. در نتیجه گفتیم اگر جهش رو جستجو محلی انتخاب کردیم crossover(بازترکیب) رو جستجو عمومی تعادل برقرار میشه.

- اگر انتخاب بازماندگان رو از μ,λ استفاده کنیم که تنوع زیادی داره. روش انتخاب رو نمیایم از Q تورنومنت بریم که تنوع بالایی داره. بلکه باید بیایم از روش متناسب با شایستگی بریم.که تعادل بین تنوع و همگرایی برقرار بشه.
- اگرم از μ+λ استفاده کردیم چون تنوع نداره بهتره کنارش از Q تورنومنت استفاده کنیم.

به همین دلیل نمیشه گفت روش متناسب با شایستگی خوبه یا بده.

یه نکته دیگه داریم اینه که معمولا میان بهترین موجود رو آرشیو میکنن. بهترین موجود حالا میتونه تو تکامل شرکت کنه و فرزند تولید کنه یا فقط همینطوری نگه داریم فراموش نکنیم و بهترین جواب رو از دست ندیم.

حالت های خاص:

فرض کنیم مدل های الگوریتم تکاملی هامون رو اینطوری نشون میدیم:

$EA(\mu + \lambda)$

$EA(\mu, \lambda)$

حالا در یک حالت خاص داریم EA(1 + 1). یعنی یه موجود داریم، یدونه فرزند تولید کنیم و از بینشون یدونه رو انتخاب کنیم برای نسل بعدی.
در این حالت ما بازترکیبی نمیتونیم داشته باشیم چون بازترکیبی 2 تا میخواد و ما فقط میتونیم جهش داشته باشیم. این الگوریتم میشه معادل با **تپه نوردی.**

ولی EA(1, 1) معادل با **Random walk** میشه. چون یه موجود داریم یه موجود رو انتخاب میکنیم همون رو میبریم نسل بعد. دیگه شایستگی مطرح نیست.

یه حالتی هم EA(1 + λ) یا EA(1, λ)باشه. یه موجود داریم لاندا تا فرزند با جهش تولید میکنیم. سپس از بین این لاندا تا فرزند یکی رو انتخاب میکنیم یا از بین λ + 1 تا موجود یکی رو انتخاب میکنیم.
در این روشم فقط جهش داره.

یه حالتی هم E(μ + 1) هستش. به این روش میگن **steady state**. به خاطر این که تغییراتش آرومه. و هردفعه یک فرزند تولید میکنه و بدترین موجود حذف میشه.
E(μ, 1) که نمیشه داشت چون از بین 1 فرزند باید μ موجود انتخاب کرد که نشدنیه.

حالت دیگه EA(μ , μ) که همیشه از μتا فرزند انتخاب میشن و والدین از بین میرن پس بهش میگن **Generational.**
EA(μ + μ) هم که فرزندان با تعداد والدین برابره ولی ما بازماندگان رو از بین فرزندان و والدین هردو انتخاب میکنیم.

حالت کلی:

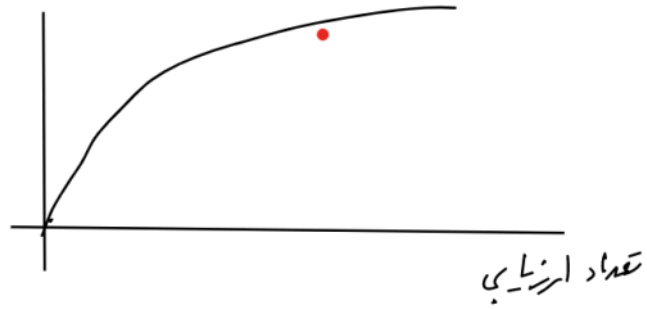
EA(μ+λ)

EA(μ, λ)

که در دومی توصیه شده λ≃7μ (هم توسط تجربی هم به کمک روابط ریاضی میشه اثبات کرد بهترین نتیجه رو میده)

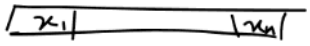
چرا بیشتر از 7 کنیم بد میشه؟ هزینه در الگوریتم تکاملی میشه تعداد دفعاتی که ما تابع شایستگی رو ارزیابی کردیم. الگوریتم تکاملی که N تا نسل داره، μ تا موجود و λ تا فرزند تعداد دفعاتی که تابع شایستگی رو محاسبه میکنه چندتا میشه؟ Nλ + μ (چون نسل اول هم بایستی شایستگی رو حساب کنیم). چون تعداد نسل معمولا زیاده اون μ اول رو صرف نظر میکنن میگن Nλ.

حالا اگر ما N=100 داشته باشیم و در یک حالت λ=10 و در حالت دیگه λ=20 باشه، در حالت اول 1000 تا هزینه محاسباتی داریم در دومی 2000 تا. در نتیجه 100 نسل حالت اول برابر نیست با 100 نسل حالت دوم.
به همین دلیل برای مقایسه الگوریتم های تکاملی محور x هارو تعداد نسل در نظر نمیگیریم و تعداد ارزیابی در نظر میگیریم. محور y هم متوسط شایستگی یا بهترین موجود در نسل میگیرن.



یک منحنی دیگه هم اینطوریه که محور x میشه ارزیابی محور y هم تنوع. تنوع هم 2 نوع داریم:

- ژنوتیپی: در یک نسل واریانس ژن ها حساب بشه. میایم 2 تا سیگما میذاریم یکی رو موجودات یکی در ژن ها و میشه:

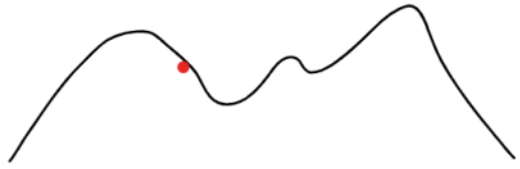


$$\frac{1}{n\mu}\sum_{l}\sum_{i}(x_i^l-\bar{x}_l)^2$$

- فنوتیپی

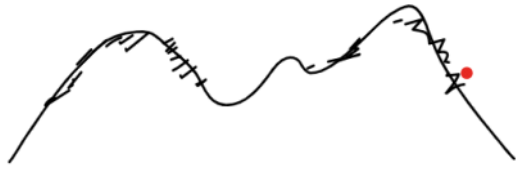
$$\frac{1}{\mu}\sum_{l}(f_i-\bar{f})^2$$

حالا کدوم رو باید بذاریم رو محور؟ تنوع ژنوتیپی درسته ولی خیلی جاها این 2 با هم یکسان میشن. اگر یه همچین شکلی داشته باشیم:



اگر موجودات اینطوری پخش بشن:

تنوع ژنوتیپی و فنوتیپی تقریبا برابر میشه.
ولی اگر همه جا پخش بشن:



این موقع دیگه تنوع ژنوتیپی و فنوتیپی برابر نیستن.
ولی چون در الگوریتم های تکاملی موجودا بعد چند نسل تمایل دارن برن رو یه قله در نتیجه خیلی جاها برای ساده سازی از تنوع فنوتیپی میریم بجای ژنوتیپی.

ولی همیشه اینطوری نیست. یک سری الگوریتم تکاملی داریم که چند جوابی هستن، یعنی میخوایم یه کاری کنیم تمام مینیمم ها و تمام ماکسیمم های محلی رو بدست بیارن اونجا دیگه ژنوتیپ و فنوتیپ یکی نمیشه تنوعش.

شرط خاتمه چیا میتونه باشه:

1. تعداد ارزیابی
2. حد آستانه
3. همگرایی
4. عدم تنوع

حالا ما همگرایی رو چطوری حساب کنیم؟ اگر در چند نسل متوالی تغییرات متوسط شایستگی از یه حدی کمتر بشه، میگیم همگرا شدیم:

حالا این میتونه همگرایی زودرس باشه یا همگرایی اصلی باشه. این که کدومشه نمیشه متوجه شد.

حالا مزیت و عیب هرکدوم:

- **تعداد ارزیابی:**
  - **مزیت:** در Loop بینهایت نمیافتیم. میتونیم بر اساس تعداد ارزیابی و زمان هر ارزیابی بیایم زمان الگوریتم رو محاسبه کنیم. پس تضمین خروج از الگوریتم داریم که **فقط در این شرط خاتمه هست.**
  - **عیب:** تعیین تعداد ارزیابی مناسب. اگر زیاد بگیریم ممکنه زود به جواب برسیم و زمان رو هی تلف کنیم تا به جواب برسیم. ممکنه کم هم بگیریم و بعد به جواب میرسیم در واقع ولی چون کم گرفتیم به نتیجه نمیرسیم. تضمین جواب رو نداریم
- **حد آستانه:**
  - **مزیت:** مزیت اینه اگه از الگوریتم اومدیم بیرون حتما مطمئن میشیم که جواب رو پیدا کردیم. تنها شرط خاتمه ایه که وقتی میایم بیرون مطئنیم جواب پیدا شده برای بقیه نمیشه اینطوری گفت. ممکنه همگرایی زودرس بخورن
  - **عیب:** Loop بینهایت. برای بعضی مسائل نمیشه فهمید حد آستانه کجاس (مثل پیدا کردن مینیمم یک تابع که اصن نمیدونیم چند هست که بعد بخوایم وایستیم. ولی در مساله 8 وزیر میدونیم شایستگی 0 شد یعنی هیشکی هیشکی رو گارد نمیده و حد آستانه رو داریم)
- **همگرایی:**
  - **مزیت:** اگر ما همگرا شدیم دیگه ادامه دادنش فایده نداره (چه زودرس باشه چه اصلی).
  - **عیب:** Loop بینهایت. تضمین جواب رو نداریم.
- **عدم تنوع:**
  - **مزیت:** اگر تنوع از بین رفته میایم بیرون
  - **عیب:** Loop بی نهایت. تضمین جواب رو نداریم.

جلسه بیست و هشتم

9:12 PM Saturday, June 19, 2021

مسئله 100 وزیر را میخواهیم حل کنیم.

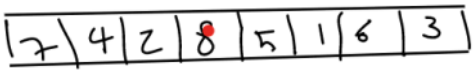
اولین مرحله رو گفتیم بازنمایی هستش. **بازنمایی برای 100 وزیر چه شکلی باشه خوبه؟** روش های مختلفی هستش ولی برای کد کردن یک سری شرایط داریم.

- مهمترین شرطش اینه که تمام حالات جواب (فضا) رو بتونه پوشش بده. مثلا حالت باینری در نظر بگیریم و تعداد بیت ها کم باشه یک قسمت از فضای جواب رو پوشش نمیدیم.
- تا اونجا که میتونه فضا رو کوچیک کنه:
  - مساله 100 وزیر رو مثلا اگر یک صفحه 100 در 100 بگیریم (10 هزار تا خونه).
  - یا میتونیم 100 تا ژن در نظر بگیریم هرکدوم یک عددی بین یک تا ده هزار باشه (بگه تو کدوم یکی از اون 10 هزار تا خونس).
    - چون هر خونه 1 تا ده هزار میتونه باشه و ما هم 100 تا خونه داریم تعداد حالتامون میشه 10000^{100}.

      در الگوریتم های تکاملی هرچی فضا بزرگ تر باشه کارایی کمتر میشه و بالعکس.
  - اگر حالت اولی رو بگیریم (100 در 100) اون موقع تعداد حالت ها میشه 100^{100}.
  - ولی بازم میشه کوچیکتر کرد. در مساله 100 وزیر میدونیم اگر در سطری یک وزیر قرار بگیره در ستون های دیگه در اون سطر وزیری نمیتونه باشه.

    پس میتونیم جایگشت بریم که بشه !100

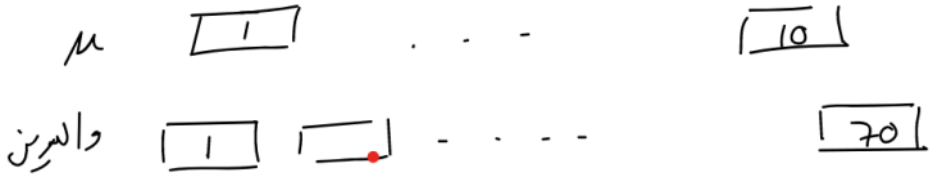
حالا چون 100 وزیر کشیدن سخته برامون استاد 8 تا رو کشید و با جایگشت مشخص کرد:



حالا اگر μ=10 داشته باشیم، در ابتدا 10 موجود به صورت **تصادفی یکنواخت** تعیین میکنیم. (مرحله دوم)

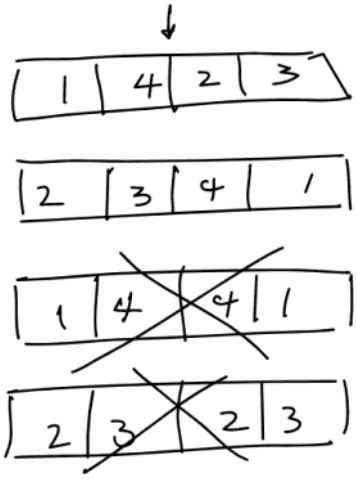
مرحله سوم - انتخاب والدین: اگر λ=7μ در نظر بگیریم، تعداد فرزندان میشه 70 تا. معمولا وقتی تعداد فرزندان رو از جمعیت خیلی بیشتر در نظر میگیریم، انتخاب والدین به صورت تصادفی یکنواخت میشه انتخاب کرد. (بدون شایستگی)

در اینجا 10 تا موجود داریم، اعداد تصادفی یکنواخت بین 1 تا 10 تولید میکنیم و هرکدوم رو اومد بر میداریم. 70 بار این کارو تکرار میکنیم. چون تصادفی یکنواخت هستش معنیش این میشه از هر موجود 7 تا کپی میاد پایین.



سپس میایم اینارو جفت جفت در نظر میگیریم. (چون تصادفی یکنواخته دیگه shuffle نمیکنیم).

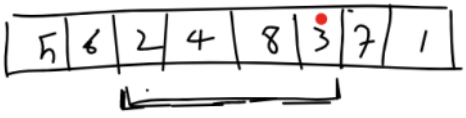
حالا میایم سراغ بازترکیب. اگر از بازترکیبی تک نقطه ای استفاده کنیم:



2 فرزند تولید شده هیچ کدوم جایگشت نیستن و شرط مساله مارو خراب میکنن. پس باید crossover ای تعریف کنیم که جایگشت رو خراب نکنه.

تعداد این بازترکیبا خیلی زیاده. 2 تاشو الان میگیم بقیش تو اسلایدا هست:

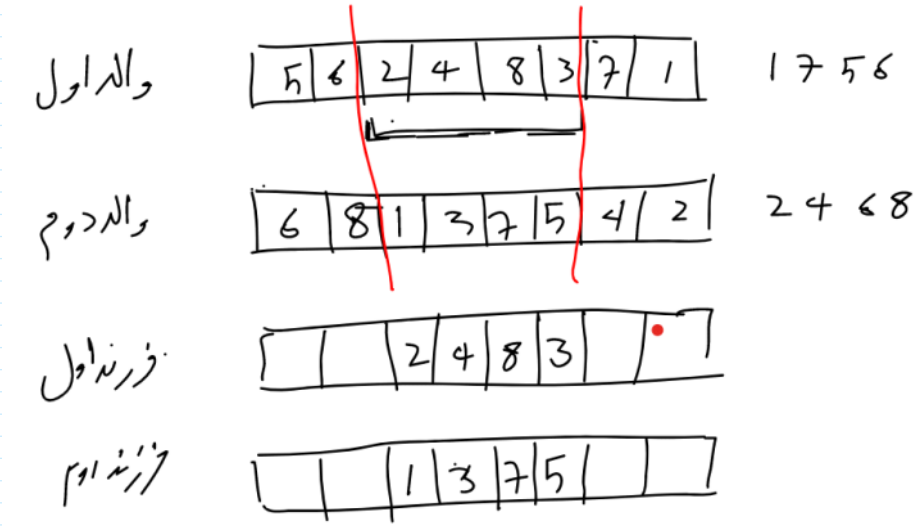
- امید crossover اینه که **به صورت کاملا تصادفی** بیاد ویژگی های خوب والد رو کنار هم بذاره و یک فرزند با شایستگی بیشتر رو تولید کنه.
- **ویژگی خوب در جایگشت:** یک ترتیب خاص باعث میشه یک موجود خوب باشه یا نه.



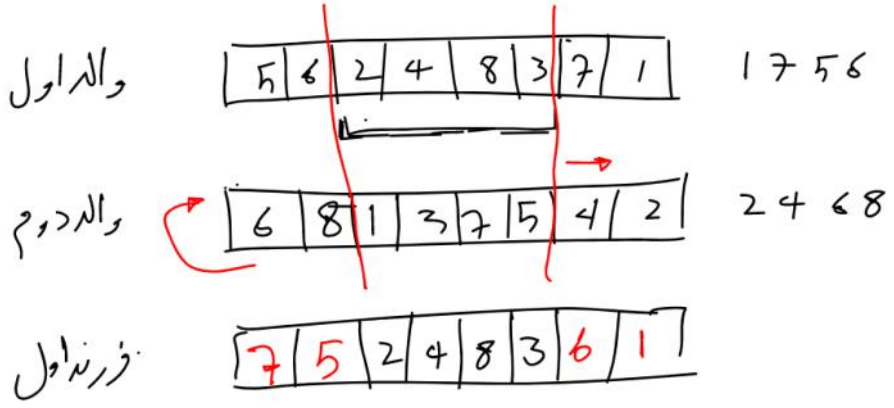
فرض کنیم این 3 8 4 2 یک ترتیب خوب باشه و بخوایم حفظ شه و بقیه عوض شن. **این ترتیب اگر بمونه و بقیه عوض بشن جستجوی عمومیه یا محلی؟** بستگی داره که این ترتیب چقدر بزرگ

باشه. در این جا چون 4 تا ثابت میمونن 4 تا عوض میشن به نوعی جستجو عمومیه. جستجوی محلی رو میشه همسایگی در نظر گرفت. این که 7 کنار 1 باشه میشه جستجوی محلی.

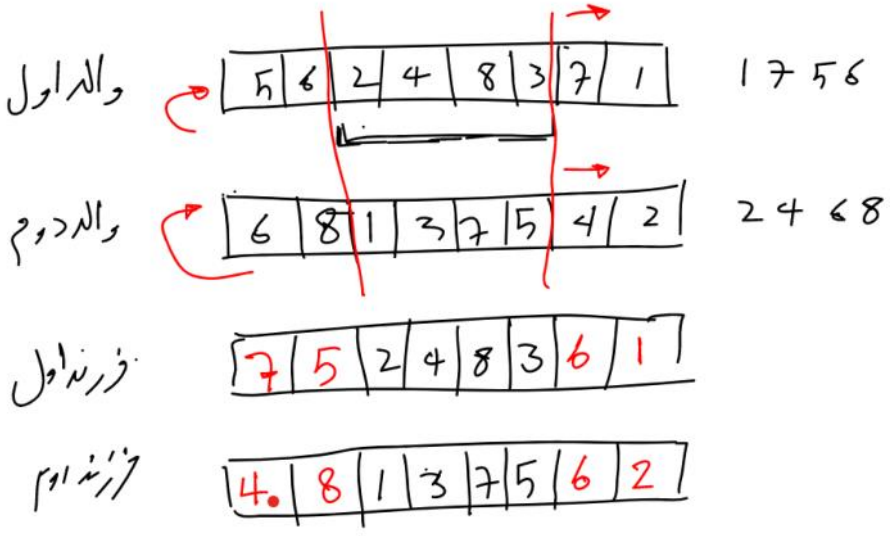
تو یه روش میایم اینطوری میکنیم:



حالا بقیه فرزند رو چطوری بچینیم؟ (اعدادی که توی 2 تا والد سمت راستش نوشته شده اعداد باقی موندس)

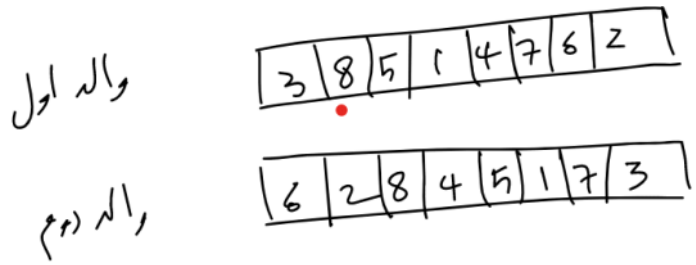


نحوه انتخاب اینطوریه که در فرزند اول اومدیم از نقطه بعد از crossover شروع میکنیم که عدد قرار بدیم (از سمت راست دومین خونه). چون فرزند اول هستش سعی میکنیم از اعداد والد دوم انتخاب کنیم. چطوری؟ میایم میبینیم در والد دوم اول 4 رو داریم. آیا 4 در بین 6 5 7 1 هست؟ نه. پس میریم سراغ 2. آیا 2 هست؟ نه. پس میریم سراغ 6. آیا 6 هست؟ اره پس میذاریمش. بعد میریم میبینیم 8 نیست ولی 1 هست میذاریمش. بعد به ترتیب 7 و 5 رو میبینیم و میذاریم.

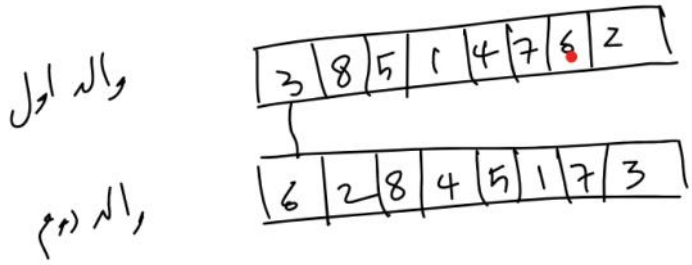


با این روش جایگشت خراب نمیشه و فرزندان تولید میشن.
** توی crossover ما همیشه تصادفی عمل میکنیم و شایستگی کمتر یا بیشتر میشه ولی امیدمون اینه بیشتر بشه.

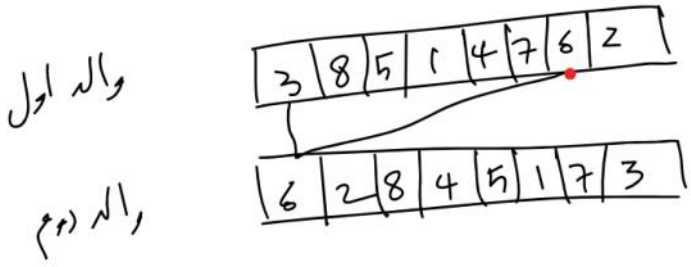
روش دیگه بازترکیبی:



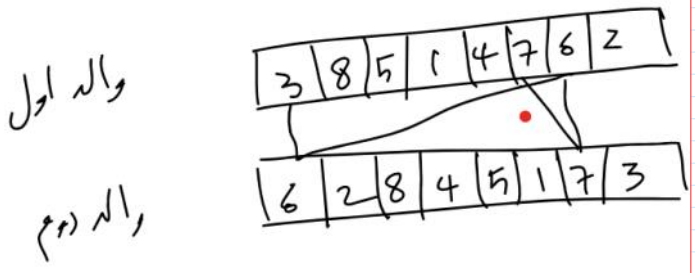
نحوه عملکرد اینطوریه که اول از 3 شروع میکنیم میبینیم روبروش 6 عه:

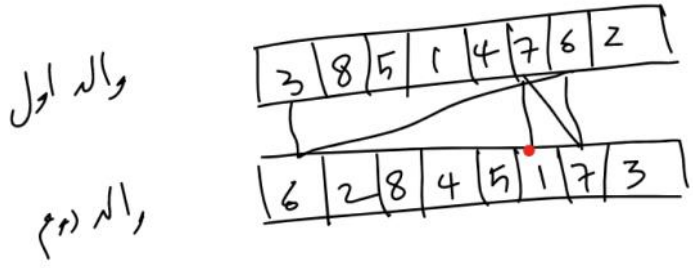


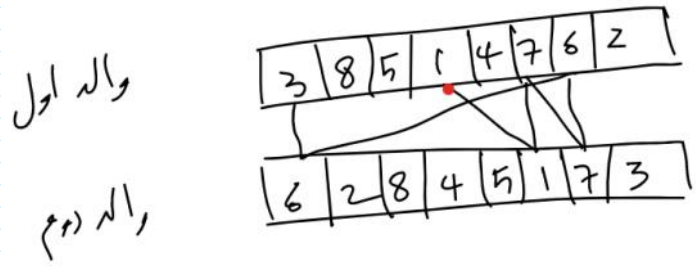
بعد میریم تو والد اول 6 رو پیدا میکنیم و وصل میکنیم:



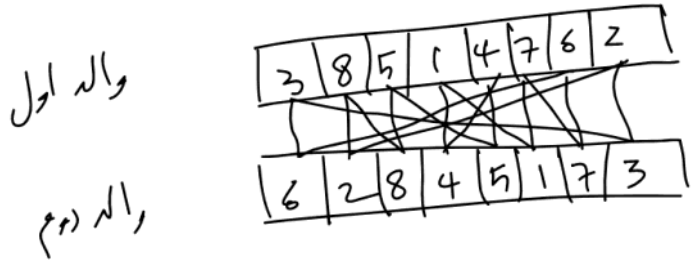
بعد روبروش 7 عه:





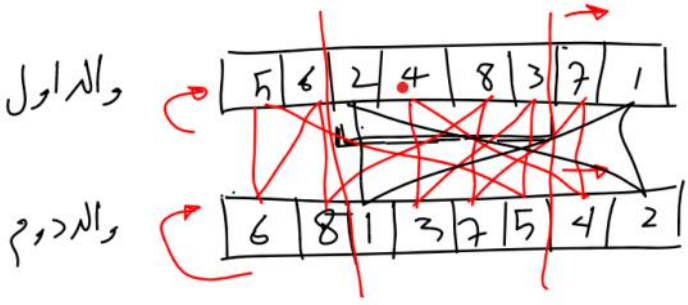


و همینطوری ادامه میدیم...



در این مثال یه حلقه بیشتر نداره. البته چون 8 تا بود اینطوریه 100 تا بکنیم چندتا حلقه به وجود میاد.

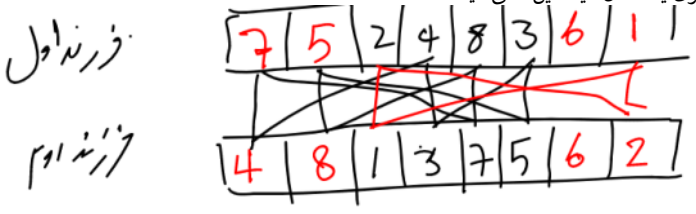
در یک مثال دیگه بررسی کنیم مثلا 2 تا حلقه میشه:



حلقه اول رو ثابت میذاریم و دومی رو جابه جا میکنیم.
یعنی مثلا 5 6 رو از والد اول مینویسم بعد 2 رو که میبینیم چون حلقه دومیه جاش 1 میذاریم میشه:
5 6 1 4 8 3 7
بعد حلقه دوم رو میبینیم جای 1 میاد 2 و داریم:
5 6 1 4 8 3 7 2
به همین ترتیب از والد دوم شروع کنیم داریم:
6 8 2 3 7 5 4 1

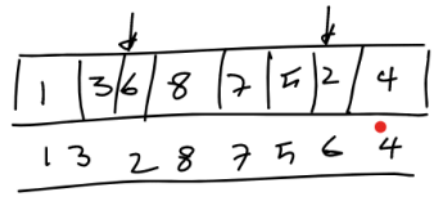
حالا این تعداد حلقه ها ممکنه بیشتر باشن یکی در میون جابه جا میکنیم.

توی یک مثال دیگه این مدلی میشه:

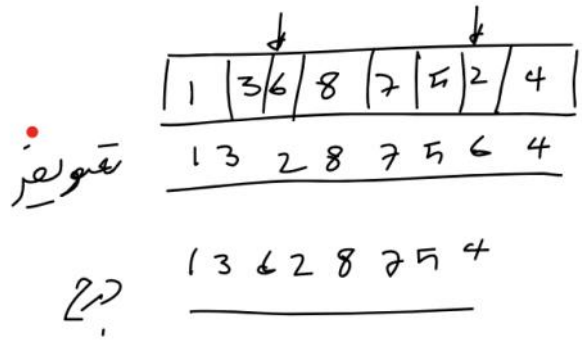


فرزندا:
7 5 1 4 8 3 6 2
4 8 2 3 7 5 6 1

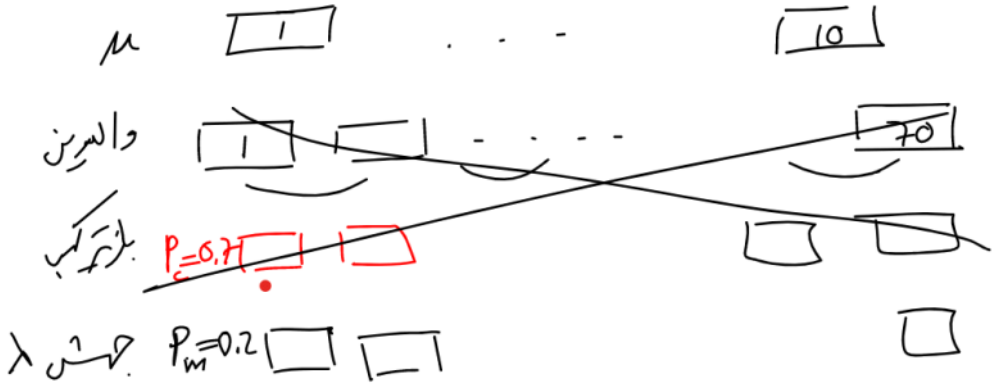
سپس میرسیم به جهش. یکی از جهش هایی که جستجو محلی هستش اینه که به طور تصادفی یکنواخت بیایم 2 تا بیت رو انتخاب کنیم تعویض کنیم:



چون 2 تا ترتیب زیادی رو عوض نمیکنه (مخصوصا وقتی ژن ها 100 تاس) میشه جستجو محلی.
یک راه دیگش درج کردنه. یعنی 2 رو بیاریم بعد 6:



در مساله ما با احتمال 20 درصد روی هر ژن جهش انجام میشود و قبلیا حذف میشن:



بعد میرسیم به انتخاب بازماندگان

- μ,λ با روش متناسب با شایستگی
- μ+λ با Q تورنومنت

۱- جایگشت

۲- $\mu=10$ تقاطع تکنیزامت

۳- $\lambda=70$ تقاطع تکنیزامت

۴- حلقه یا روش دیگر $P_c=0.7$

۵- $P_m=0.2$ تعویض یا درج

یکی جستجوی عمومی

یکی ~ محلی

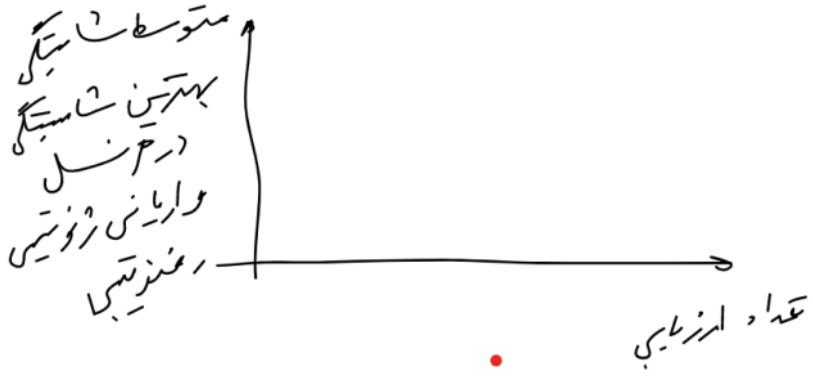
۶- $c \rightarrow F=\frac{1}{c}$

۷- $P_i \propto F_i$ μ, λ

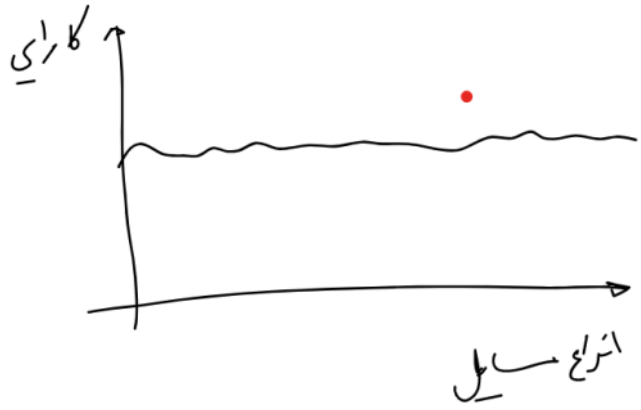
با برتری ترجیحی μ+λ

۸- ترکیب شرط های خاتمه

** بهترین موجود رو نگه میداریم
البته برای کارای تحقیقاتی معمولا شرط خاتمه رو تعداد نسل های زیاد در نظر میگیرن. منحنی که باید رسم کنیم برا این که ببینیم چطوری داره الگوریتم کار میکنه:

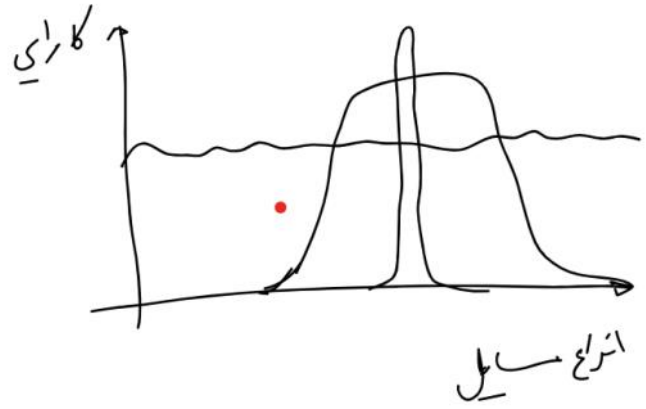


الگوریتم های تکاملی میتونن عمومی باشن:



برای همه مسائل یه کارایی قابل قبولی دارن.
اگر بازنمایی رو جایگشت کنیم همه رو که نمیشه حل کرد بلکه محدوده ای رو میشه با کارایی بهتر حل کرد:

میشه یه الگوریتمی رو خصوصی تر کرد به طوری که بازترکیبی و جهش رو از دانش اون مساله کمک بگیره. در این صورت یک الگوریتم داریم که برای یک مساله خاص یک جواب خوبی رو فقط داره:


کارایی
انواع مسایل

استراتژی های تکامل - Evolutionary Strategies (ES)

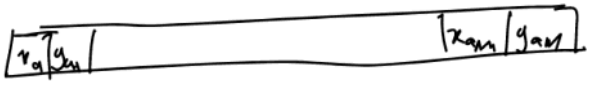
الگوریتم های تکاملی به طور کلی کند هستند. یک سری استراتژی میذاریم که این سرعت افزایش پیدا کنه. در ES استراتژی هایی در نظر گرفته میشه که سرعت رسیدن به جواب رو افزایش میده.

ساده ترین حالتش ES(1+1) هستش. که یک موجود داریم و یک فرزند تولید میشه. Crossover نداریم و فقط جهش اتفاق میافته.

1. **بازنمایی اعداد حقیقی**

فرض شود که میخواهیم در یک کشور یک تعداد فرودگاه بسازیم. N شهر و M فرودگاه داریم به طوری که M<N ولی میخواهیم اینارو طوری بسازیم که فاصله هر شهر تا نزدیک ترین فرودگاه مینیمم بشه.

اگر موقعیت شهر هارو با x_i و y_i نشان دهیم و موقعیت فرودگاه هارا با x_{aj} و y_{aj} نشان دهیم، تعداد ژن ها 2M تا می باشد به طوری که:



2. اگر مختصات شهر هارا از a تا b یعنی [a b] برای x ها و [c d] برای y ها در نظر بگیریم، **مقداردهی تصادفی اولیه** چطوری باید باشه؟

عدد تصادفی کنیم

$$x_{aj} = a + (b-a)U(0,1)$$

$$y_{aj} = c + (d-c)U(0,1)$$

با این فرمول اعداد تصادفی یک نواخت بین بازه مورد نظر رو تولید میکنیم

3. **تابع ارزیابی:**

تابع هزینه:

$$\boxed{C_g = \sum_{i=1}^{N} \min_{j=1}^{M} \sqrt{(x_i - x_{aj})^2 + (y_i - y_{aj})^2}}$$

شایستگی:

$$F = \frac{1}{C}$$

4. **جهش:**

$$X' = X + N(0, \sigma^2)$$

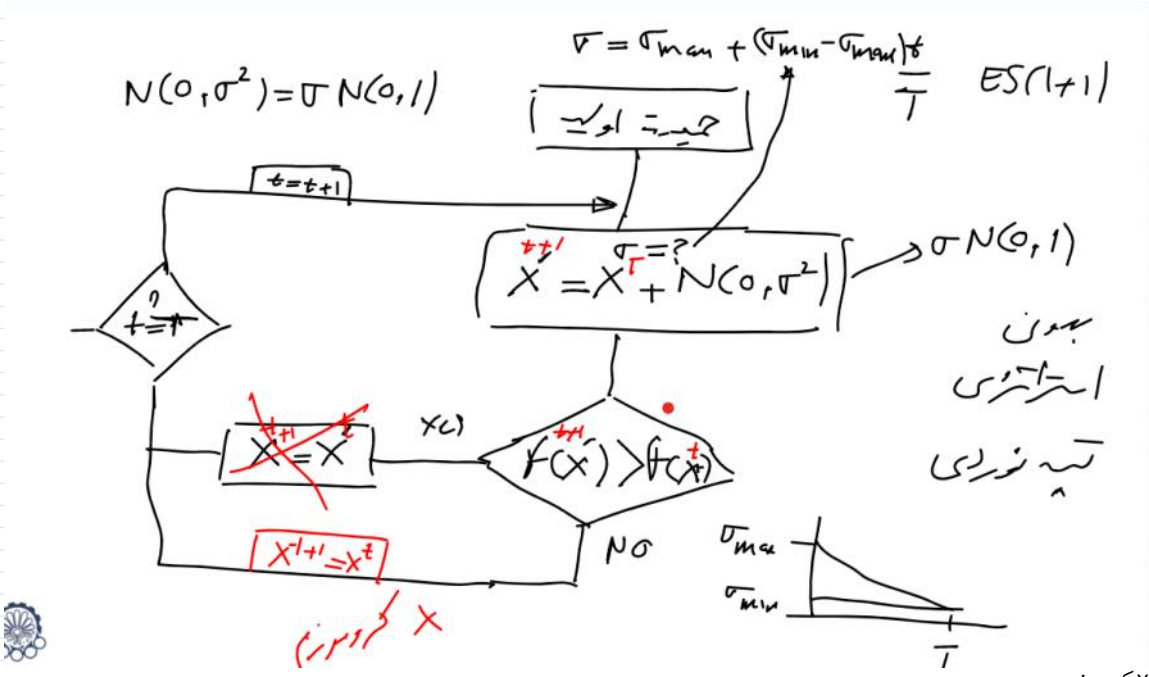
یعنی X قبلی را با یک نویز نرمال با انحراف معیار σ اضافه میکنیم.

از آمار احتمال باید بلد باشیم که:

$$N(0, \sigma^2) = \sigma N(0,1)$$

یعنی یک عدد تصادفی نرمال با میانگین 0 و واریانس 1 پیدا کنیم وضرب کنیم در σ

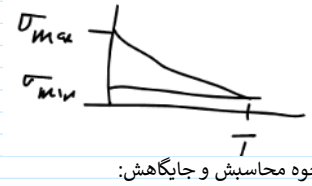
فلوچارت ES(1+1) (بدون استراتژی تکامل):



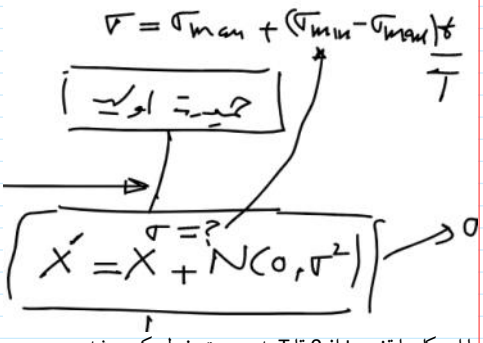
X کروموزوم

* مشابه تپه نوردی شد.

برای افزایش سرعت، استراتژی پیشنهادی به این صورته که وقتی نزدیک جواب شدیم قدم رو کوچیک و وقتی دور شدیم اندازه قدم بزرگ بشه. یعنی σ یکسان نیست و هرچی به T نزدیک میشیم کمتر میشه:



نحوه محاسبش و جایگاهش:



با این کار با تغییر t از 0 تا T به صورت خطی کم میشه.

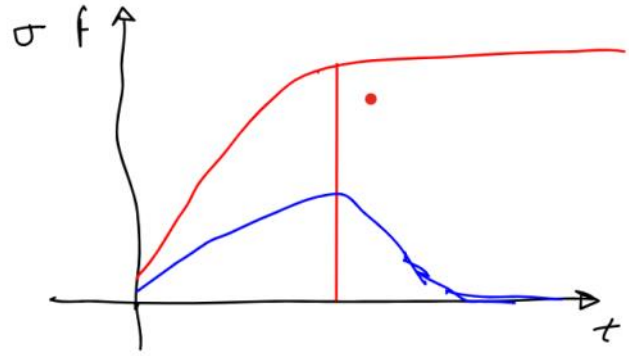
اما میشه بهترش کرد؟ یک تز دکترا در آلمان اثبات کرده که در ES(1+1) قانون یک پنجم موفقیت داریم. در این قانون میگه "سرعت وقتی ماکزیمم میشه که تعداد موفقیت ها برابر با 1/5 باشد".
← از هر پنج جهش ما یکیش بهتر از والد باشه.
اگر ما تعداد موفقیتمون بیشتر از 1/5 باشه دور از جوابیم یا نزدیکشیم؟ وقتی بیشتر از 1/5 باشه ما دور از جوابیم و هر قدمی که برمیداریم مارو به جواب نزدیک میکنه. در یک فضای سه بعدی یا n بعدی اگر ما بخواهیم به یک نقطه برسیم، وقتی فاصلمون زیاد باشه تعداد مسیر هایی که مارو به اون نقطه نزدیک کنن زیاده در نتیجه احتمال این که مسیری که انتخاب میکنیم مارو به جواب برسونه بالا میره ← مثلا جنوب تهرانیم میخوایم بریم تجریش. از این مسیر 360 درجه ای 180 درجه شمالیشو انتخاب کنیم حرکت کنیم به تجریش نزدیک میشیم (چون فاصله زیاده) ولی هرچی به تجریش نزدیک تر میشیم اون زاویه که ارو به تجریش هدایت میکنه کوچیکتر و کوچیکتر میشه.
یک دلیل دیگه که وقتی نزدیک جواب میشیم تعداد موفقیت کم میشه: اگر قدم ها زیاد باشه ما جواب رو رد میکنیم. هنگامی که نزدیک جوابیم اندازه قدم خیلی اثر داره ولی وقتی دوریم اهمیت نداره

اطلاعات فوق رو چطوری به سرعت ربطش بدیم؟ اگر احتمال موفقیت P_s باشه:

$0 < c < 1 \qquad c = 0.8$

$$\boxed{\begin{array}{ll} P_s > \frac{1}{5} & \sigma = \frac{\sigma}{c} \\ P_s = \frac{1}{5} & \sigma = \sigma \\ P_s < \frac{1}{5} & \sigma = c\sigma \end{array}} \qquad P_s$$

اگر قرمز رو با شایستگی رسم کنیم و آبی رو σ:



سیگما رو معمولا از یه حد آستانه هم میذارن کوچیکتر نشه، یعنی اگر σ>ε بشود آنگاه σ=ε میگذاریم.

حالا چطوری P_s رو میشه حساب کرد؟ یک راهش متوسط گیری پنجره ای هستش:
فرض کنید از x_1 تا x_k داریم (x متغیر دلخواه) و میخواهیم متوسط بگیریم:

$$x_1 \quad x_2 \quad \cdots \quad x_k$$

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_k}{K}$$

حالا اگر x_{k+1} بیاد تو کار و بخوایم از x_2 تا x_{k+1} متوسط بگیریم. به طوری که از x متوسط قبلی استفاده کنیم.

$$\bar{x}^{new} = \frac{K\bar{x}^{old} - x_1 + x_{K+1}}{K}$$

الان مشکل اینجاس که الان x_1 دفعه بعد x_2 و .. رو تو حافظه نگه داریم. یعنی اگر k تا اضاف بشه باید kتای قبل رو نگه داریم.
پس میایم یک تقریب میزنیم. به این صورت که:

$$\bar{x}^{new} = \frac{K\bar{x}^{old} - \underbrace{x_1}_{\approx \bar{x}^{old}} + x_{K+1}}{K} \approx \underbrace{\frac{K-1}{K}}_{\beta \,\left(1-\frac{1}{K}\right)} \bar{x}^{old} + \frac{1}{K} x^{new}$$

$$\bar{x}^{new} = (1-\beta)\bar{x}^{old} + \beta x^{new}$$

حالا اگر $x^{new} = S$ بشه. به طوری که S بشه موفق بودن. اگر موفق باشیم 1 و نباشیم 0. اون موقع x متوسط چی میشه؟ میشه احتمال موفقیت. چرا؟

x_1	x_2	x_3	x_4
1	0	1	0

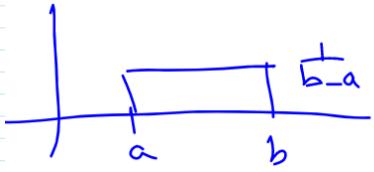
$$\frac{2}{k} = \frac{2}{4} = \frac{1}{2}$$

مثلا تو ای مثلا اگر موفق باشه 1 و نباشه 0 اون موقع 1/2 میشه احتمال موفقیت
پس میتوان گفت:

$$P_S = (1-\beta)P_S + \beta S$$

S هم برابره 1 عه اگ فرزند بهتره و برابر با 0 اگر فرزند بدتره
B هم میشه 1/k و k هم تعداد نسل هایی که احتمال در آن تعداد حساب میشود. اگر 1/100 = β یعنی متوسط گیری روی 100 تا نمونس ولی اگر 1/10 تا باشه روی 10 تا نمونه.

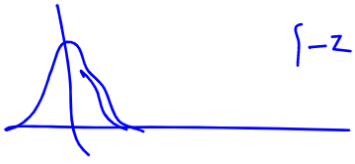
---- فرق بین تصادفی نرمال و یکنواخت
احتمال انتخاب شدن یک عدد در بازه a تا b برای یکنواخت:



برای نرمال:
بین[1 -1] باشه میشه 60 و خورده ای درصد. بین [2 -2] میشه نزدیک 100 درصد و برای بقیه نزدیک 0 درصد

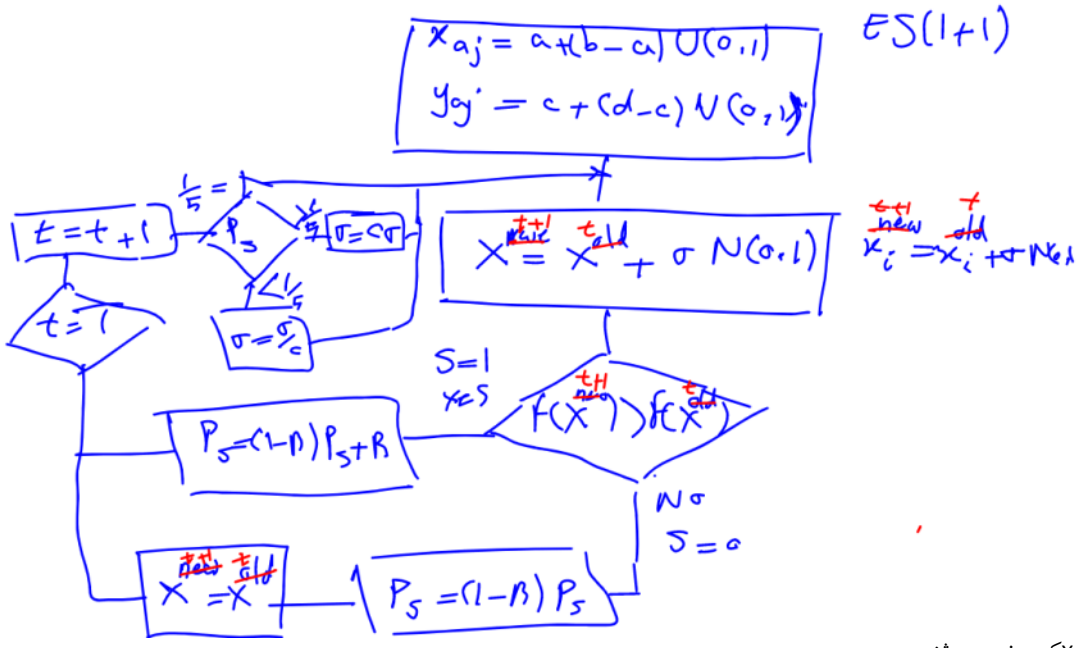
[-1 1] 63

[-2 2]



میانگین اعدادی که انتخاب میشه قدر مطلقشون نزدیک به 1 میشه. به همین دلیل به σ میگیم اندازه قدم. چون توی اندازه قدم که جهت مهم نیست و با یک توزیع نرمال با اندازه قدم سیگما قدم بر میداریم چون متوسط قدر مطلق تقریبا 1 هست در نتیجه ضریدر σ بشه میشه σ و به همین دلیل به σ میگیم اندازه قدم.

حالا میریم فلوچارت ES(1+1) رو میکشیم:



Xکروموزوم و x_i ژن

توی وقتی که شایستگی رو جک کردیم اگر بدتر بود $x^{t+1} = x^t$ میذاریم چون والد که شایستگی بیشتری داره باید بره نسل بعد.

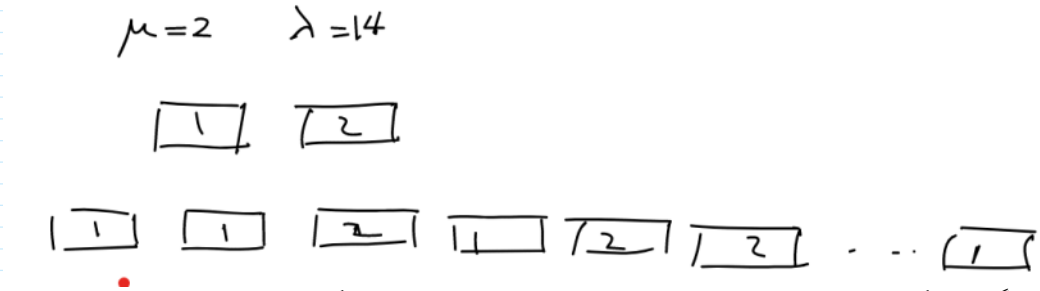
جلسه سی ام

ES(1+1) مثل تپه نوردی ممکنه توی مینیمم محلی گیر بکنه. در حالت کالی ولی ES(μ+λ) میتونه فرار کنه.
یک ورژن تصحیح شده از ES(1+1) با 1/5 موفقیت وجود داره. که میتونه از مینیمم محلی فرار کنه. به این صورت که یک احتمال1/20 هم میگیره . از 1/5 کمتر میگه اگر بین 1/5 و 1/20 باشه عین قبل عمل کنه ولی اگر از 1/20 کمتر باشه یک جهش بزرگ انجام میده و از مینیمم محلی فرار میکنه.

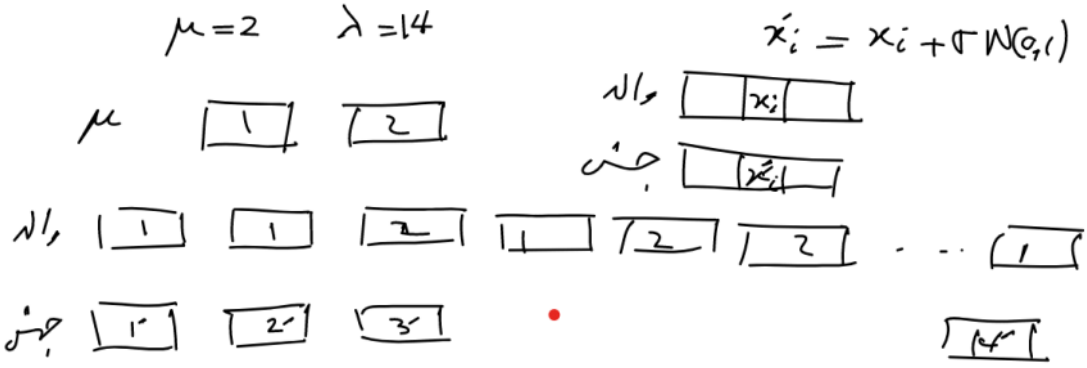
در ES(μ+λ) در حالت کلی:
بازنمایی اعداد حقیقی است.
جمعیت اولیه تصادفی یکنواخت است. (μ تا جمعیت)
انتخاب والدین تصادفی یکنواخت. $\lambda \simeq 7\mu$

در ES اول جهش انجام میشه بعد بازترکیبی.
- جهش: نویز نرمال با اندازه قدم σ ($x_i' = x_i + \sigma N(0,1)$)

چرا جهش قبل بازترکیبی؟



الان اگر اول بازترکیبی انجام بدیم چون خیلی شبیه والدین در میاد خوب نیست. اول جهش میدیم که با نویز نرمال اضافه شده با هم متفاوت شن بعد بازترکیبی انجام میشه.
*** احتمال جهش $P_n = 1$ در صورتی که $P_c < 0.4$ (Crossover)

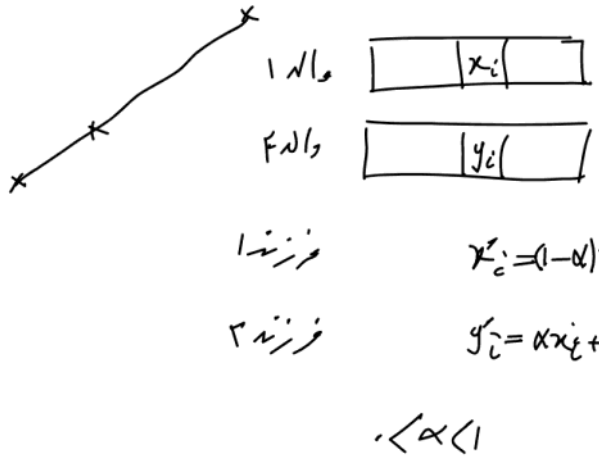


سپس موجوداتی که جهش داده شدن 2 تا 2 تا کنار هم قرار میگیرن و برای هر جفت یک عدد بین 0 تا 1 تولید میکنیم اگر از 0.4 کمتر شد بازترکیبی انجام میشه بیشتر شد نمیشه.
چرا بازترکیبی نقطه ای برای بازنمایی اعداد حقیقی مناسب نیست؟ چون از وسط ژن نمیشه جدا کرد و اعداد حقیقی وسط نداره و فقط میتونه ژن هارو جابه جا کنه و ژن جدیدی تولید نمیشه. بنابراین از بازترکیبی حسابی انجام میشه.

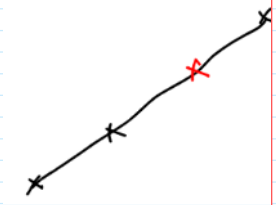
بازترکیبی حسابی:
سه نوع داره:
- کامل: برای هر دو ژن
- تکی: برای یک ژن
- ساده: از یک ژن به بعد

بازترکیبی حسابی کامل:

این 2 والد رو فرض کنیم داشته باشیم.
در بازترکیبی حسابی میگه برای این ک فرزند ویژگی اینارو داشته باشه باید در یک خطی که اینارو به هم وصل میکنه قرار بگیره. که فرمولشم میشه:

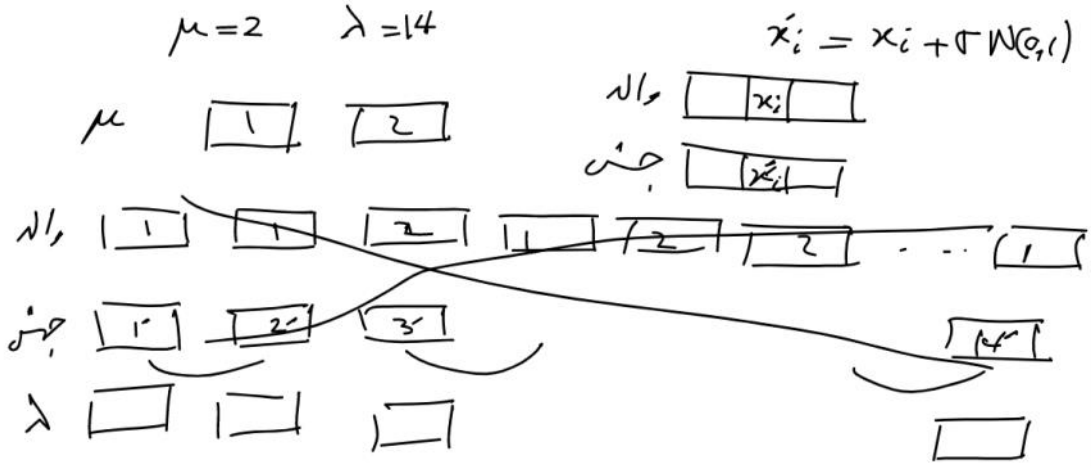


که بسته به این که آلفا چند باشه یه جا اون وسط میافتن:



اگر α=0.5 باشه دقیقا اون وسط میافتن جفتشون

پس از بازترکیبی حسابی داریم:



اگر مدل انتخابمون

- μ+λ: Q تورنومنت
- μ,λ: متناسب با شایستگی

پس مراحل:

۱- اعداد حقیقی

۲- μ تعداد والد یکنواخت

۳- $\lambda=7\mu$ تعداد فرزند یکنواخت

۴- جهش $P_m=1$ نویز نرمال با قدم σ

۵- بازترکیبی $P_c\leq 0.4$ حسابی

۶- $\mu+\lambda$ و ρ تورنمنت

μ,λ و تناسب با شایستگی

۷- شرط ترکیب

استراتژی تعیین اندازه قدم بین $\mu+\lambda$ و μ,λ:

1. **پارامتر ثابت:** روش خوبی نیست
2. **تابع مشخصی از زمان:** مثل تابع خطی یا تابع نمایی
3. **روش تطبیقی:** قانون 1/5 موفقیت
   اول این که باید یک فیدبک مناسب پیدا کرد (این که مثلا توی ES(1+1) موفقیت بود برای بقیه که نیست)
   این که 1/5 از کجا بدست میاد هم مورد نیازه (توی ES(1+1) رو میدونیم 1/5 عه برا بقیش که نمیدونیم)
   اگر 2 مورد فوق رو بشه پیدا کرد بهترین روشه
4. **خود تطبیقی:** اگر از 3 نتونیم استفاده کنیم میایم سراغ این روش که خودش خودشو تطبیق میده

در روش خود تطبیقی میایم یک ژن هم برای اندازه قدم میذاریم

و تکامل خودش باید تعیینش کنه.
در این ژن جدید فقط جهش انجام میشه و بازترکیبی رخ نمیده. جهشش هم با ژن های اصلی متفاوته.
اگر مقدار جهش یافته سیگما رو σ' بنامیم، داریم:

$$\sigma' = \sigma e^{\tau N(0,1)}$$

$$x_i' = x_i + \sigma N(0,1)$$

به دو دلیل جهشش فرق میکنه و نماییه به جای خطی:

- یک دلیل این که سیگما نمیخوایم منفی بشه و میخوایم یا کوچیکتر بشه یا بزرگتر.
- این کار تدریجی رخ بده و اگر فرکانس تغییراتش خیلی زیاد بشه تکامل نمیتونه بفهمه خوبه یا بده

به ضریب تاو که در نمایی میاد میگیم ضریب یادگیری و عدد کوچیکیه.

سوال: اول سیگما رو جهش بدیم بعد x یا بالعکس؟

$$x' = x + \sigma' N(0,1)$$
$$\sigma' = \sigma e^{-\tau N(0,1)}$$
حالت اول

$$\sigma' = \sigma e^{-\tau N(0,1)}$$
$$x' = x + \sigma' N(0,1)$$
حالت دوم

ما هیچ معیاری برا این که بفهمیم اندازه قدم خوب هست یا نه نداریم. در نتیجه بایستی بر اساس شایستگی که از x بدست میاریم بگیم خوب بوده. چون x خوب بوده پس اندازه قدمش هم خوب بوده (ولی 100 درصد هم اینطوری نیست چون اندازه قدم در یک نویزی داره ضرب میشه).
بنابراین برا این که بفهمیم خوب هست یا نه اول x رو جهش میدیم بعد شایستگی رو حساب میکنیم و احتمال میدیم که اندازه قدم خوب باشه. ← حالت دوم رخ میده و اول سیگما جهش یافته میشه و با سیگما جهش یافته شده x رو میریم میبینیم بهتر شد یا نه.
پس با خودتطبیقی:

x_1	x_n	σ

۱- اعداد حقیقی

۲- μ تعداد کروموزوم

۳- $\lambda = 7\mu$ تعداد فرزند

۴- جهش $P_m = 1$ نویز نرمال با قدم σ

$$\sigma' = \sigma e^{-\tau N(0,1)}$$
$$x' = x + \sigma' N(0,1)$$

۵- بازترکیبی $P_c \leq 0.7$ حسابی

۶- $\mu + \lambda$ و ρ تورنمنت

μ, λ متناسب با شایستگی

۷- شرط ترکیب

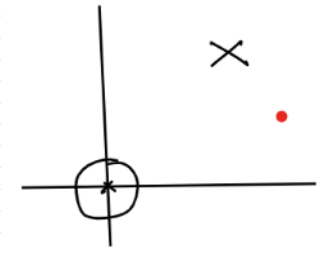
یک حالت دیگه هم خودتطبیقی داریم که صرفا بیان میشه سر کلاس:
در این روش برای هر ژن یک سیگما هم در نظر میگیریم:

x_1	x_n	σ_1	σ_n

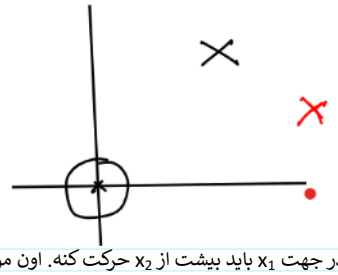
فرض کنیم یک ولد داریم:

جواب هم اینجاس:

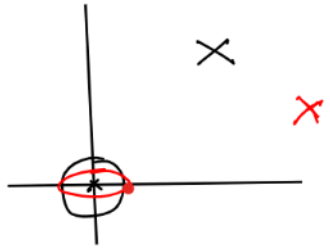
با جهش احتمال این که جواب در این دایره بیافته وجود داره:



ولی اگر یک سیگما 1 و سیگما 2 داشته باشیم. اگر الگوریتم بتونه سیگما هارو درست تشخیص بده و جواب اینجا باشه:



در جهت x_1 باید بیشت از x_2 حرکت کنه. اون موقع با چند سیگما به شکل بیضی در میشه آورد:



اینطوری شانس این که به جواب برسیم بیشتره. یه حالت دیگه هم هست بیضی رو بچرخونیم به سمت جواب که خیلی پیچیده میشه و در موردش حرف نمیزنیم.