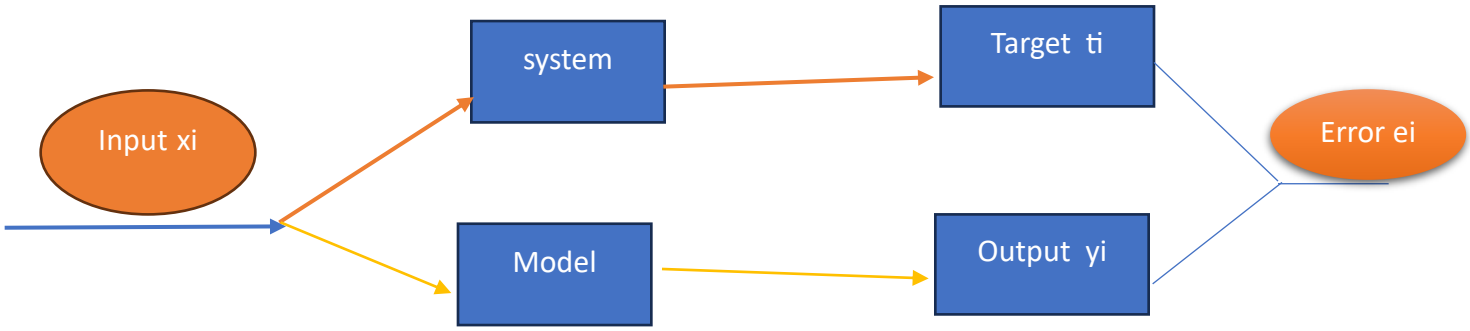


خب در حالت کلی سیستم یک ورودی میگرید که بالایی می باشد و خروجی آن هدف ما است حال احتمال زیاد هدف خروجی امکان دارد با ورودی که ما داده ایم زیاد تطابق نداشته باشد برای این منظور به صورت موازی یک مدل را طراحی میکنیم حال منطق فازی یا یک مدل باشد و یک خروجی میدهد و این خروجی امکان دارد از هدف ما دور باشد در ابتدای کار اما با مرور زمان امکان دار د به هم نزدیک شوند که این بهترین حالت ممکنه برای ماست .

خب حال خروجی از مدل و هدف اگر صفر بشه ایده ال ترین حالت ممکن میشود که امکان رخ دادن این حالت زیاد نیست چون مدل باسیستم یکی شده و دیگر فرقی ندارد در حالت نرمال یک تفاوت مابین این دو وجود دارد .

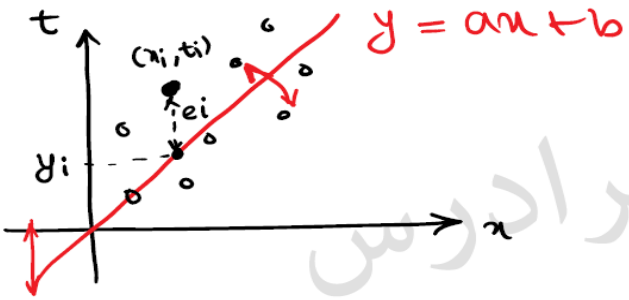
مثلا یکی از روش هایی که وجود دارد تابع هزینه/تابع مربعات خطا یا تابع شاخص عملکرد که همگی یکسان هستند ولی با بیان های متفاوت در علوم های مختلف که به فرم زیر بیان می‌شود :

$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^2$$

خب در حالت کلی ما نمی توانیم رفتار Plant یا system را تغیر دهیم مثلا فرض بر این باشد که یک روالی خاص را یک سیستم دارد اما ما میتوانیم بر روی مدل خودمان تغیرات اعمال کنیم تا خروجی کار درست بشود تا جایی که بتوانیم خطایی که بین سیستم و خروجی مدل است به حداقل ممکن برسد .

خب اگر در حوزه سیستمم های عصبی و شبکه های عصبی صحبت بشه به این مرحله و فاز Traning میگوییم .

یکی از ساده ترین مثال هایی که در این حوزه میتوان زد همان رگرسی.ن است که مابین دیتا ها که ورودی و خروجی های ما هستند یک خط میگذرانیم که حداقل فاصله را با دیتا ها داشته باشند به نحوی که همه ی آن ها را نیز پوشش دهد به شکل زیر توجه کنید :



$$e_i = t_i - y_i = t_i - ax_i - b$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(t_i - ax_i - b)^2$$

این اسلاید از کلاس در س استاد بزرگ بنده دکتر سید مصطفی کلامی هریس است .

در این اسلاید به خوبی نمایش داده میشود تابع هزینه و خطا چگونه رسم شده برای یک سری از اطلاعات آزمایشگاهی ما .

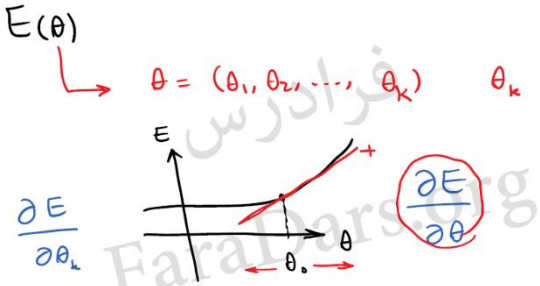
خب هدف چیه باید در این مسئله a,b طوری پیدا کنیم که مسئله کمینه شود . خب برای این روش متد های زیادی هست یکی از روش ها مشتق گیری نسبت به a , b می باشد .

خب در اینجا a| شیب خط ما استن که باهاش کار میکنیم و b هم عرض مبدا می‌باشد که بالا پایین می‌شود .

خب گاهی اوقات مثلا برای این مسئله از یک سهمی درجه 2 استفاده بشه برای رگرسیون گیری که بهش Curve fitting نیز میگویند .

خب همان طور در تصویر مشاهده میکنید یک سیستم داریم و مدل ما یک سری متغیر ها داریم که با تتا نشان داده شده در تصویر ما برای این کار ما باید یک جوری این پارامتر های خودمون تنظیم کنیم ، که مسئله ما حداقل شود برای کار ما که برای این هدف یک سری قید داریم که براساس آن باید تنظیمات تابع را به پیش بریم .

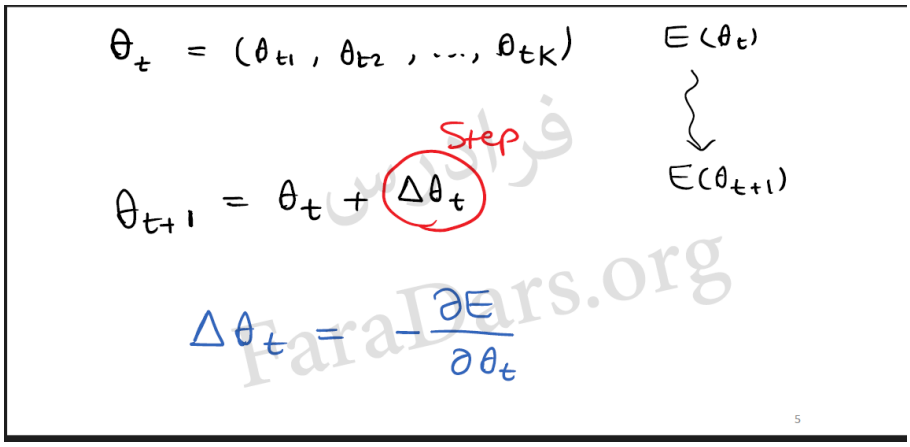
خب امروز کاری که ما میخواهیم پیش ببرریم خیلی کلی است و میخواهیم یک مسئله بهینه سازی برای تابع خطا با کمک منطق فازی پیش ببریم که در ادامه یک اسلاید آورد ه شده و توضیحات تکمیلی نیز خودم میدهم :



خب همان طور در تصویر مشاهده میکنید تابع خطا را با پارامتر های برداری نشان داده ایم که از تتا 1 تا تتا K که برای این که بخواهیم اتین مسئله را بهینه سازی کنیم از فرم گرادیان میرویم ، که گرادیان فاصله یا دوری یا نزدیکی را به ما نشان میدهد ، که ما برای که ببینیم متغیر های خود در چه مسیری حرکت دهیم بیشترین افزایش یا کاهش خواهد داشت .

خب مثلا در مثال شکل بالا میبنید مشتق در جایی که گرفتیم مقداری مثبت در میاید که اگر به سمت چپ حرکت کنیم مقادیر تتا های ما بیشترین E را به خودشون میگیرند و اگر رو به سمت راست حرکت کنیم E مقدارش کم ترین می‌شود .

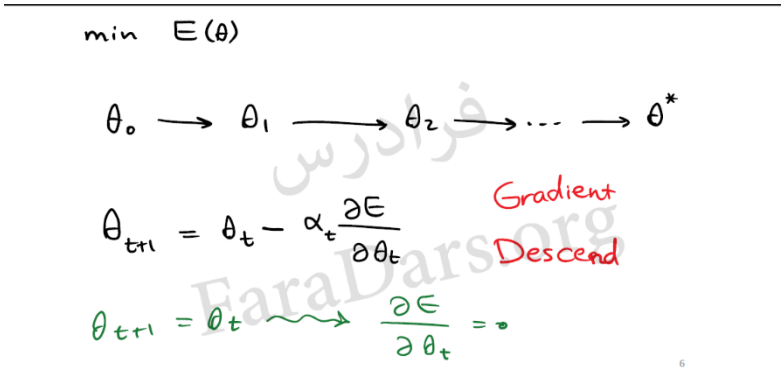
خب در اسلاید بعدی توجه شما را به یک سری تعاریف معطوف میکنم :



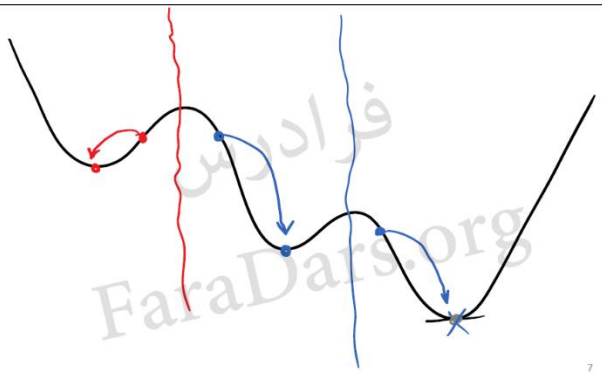
خب همانطور که ملاحظه میکنید ما پارامتر های تخمین خود را تحت عنوان تتا تعریف نمده از تتا زمان اول تا آخرین تخمینی که زدیم .

سپس برای اینکه مسئله ما بهترین شکل ممکنه را به خودش بگیرد باید خطای ما بین سیستم و مدل کمینه شود برای این کا خب ما هر تتایی که رو به جلو حرکت میکند با یک $\Delta\theta$ یک step رو به جلو حرکت میکند و θ_{t+1} را به وجود میاورد حال باید در این نقطه ای که ایجاد شده نسبت به پارامتر θ_t مشتق گرفت که با علامت منفی که جهت نزولی بودن یا کمینه شدن را به ما ارائه دهد .

خب در اسلاید بعدی که به شما نمایش میدهم فرمت کلی این الگوریتم را به نمایش میگذاریم که بهش الگوریتم گرادیان نزولی نیز گفته می‌شود :



معمولا برای این متد یک حدس اولیه از جواب را میزنند و این خیلی خیلی مهم است برای کار ما که میخو.اهیم روند بهینه سازی را پیش بگیریم . در یک اسلاید بعدی برای شما به نمایش میگذاریم این مفهوم که شرایط اولیه چقدر تاثیر گذار بر روی بهنیه سازی و دینامیک مسئله ما است :

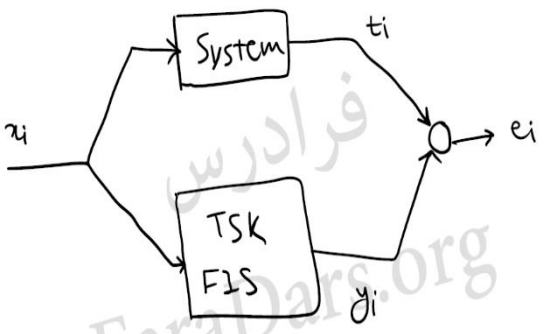


خب در تصویر میبینید که یک سری نقاط به رنگ های قرمز ،سبز و آبی وجود دارد خب اگر ما شرایط اولیه خود را در نقاط قرمز و سبز قرار درهیم دینامیک مسئله ما روندی پیش میگیرد که مینیمم آن محلی شود و بهینه کامل نشود اما اگر شما به نقطه آبی نگاهی کنید میبینید که یک مسیری را طی میکند که به بهینه ترین شکل ممکن در کل دینامیک مسئله ما برسد . نقاط مینیمم همان تعادل است که یک سری جاها را میتواند جذب کند که با خطوط جداسازی نموده ایم در تصویر به خوبی نشان داده شده است .

خب یک سوال پیش میاید که ما به اون نقطه آبی که بهینه ترین نقطه مسئله ماست آیا دسترس پذیری داریم یا خیر ؟ هیچ تضمینی برای این که به بهینه ترین نقطه همراه با همگرا شدن پارامتر های θ ما به دینامیک مسئله پس این یکم به خوش شانسی بودن تو برمیگردد رفیق .

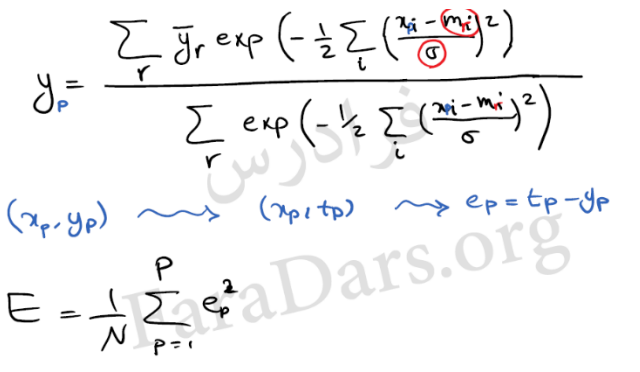
خب حال میخواهیم در حالتی پیش بریم که بتونیم این خوش شانسی را برای خودمان ایجاد کنیم که بتوانیم مسئله را بهینه ترین حالت برایش ایجاد کنیم.

برای این منظور مثلا از منطق فازی بهره گیری میکنیم که در اسلاید زیر آورده شده است :



خب در تصو.یر ورودی ها یکسان است سیستم درایم یک مدل منطق فازی TSK که قبلا ما در جلسانت قبلی هم قائده کار کردن با این و هم کد نویسی های آن و فرمول اصلی آن را هم کار کرده بودیم که داریم :

خب در خط اول این تصویر مشاهده میکنید که فرمت TSK میباشد یا فرمول عمومی TSK میباشد ، خب یک بار ورودی ها برای منطق فازی کار میکنند یک بار هم برای زمان ها خود سیستم



ایجاد خروجی میکند در نهایت $e_p = t_p - y_p$ این خطا ایجاد میشود حال با این خروجی خطا باید تابع هزینه خود بسازیم که با فرمت زیری آن یا این که اینجا تایپ شده

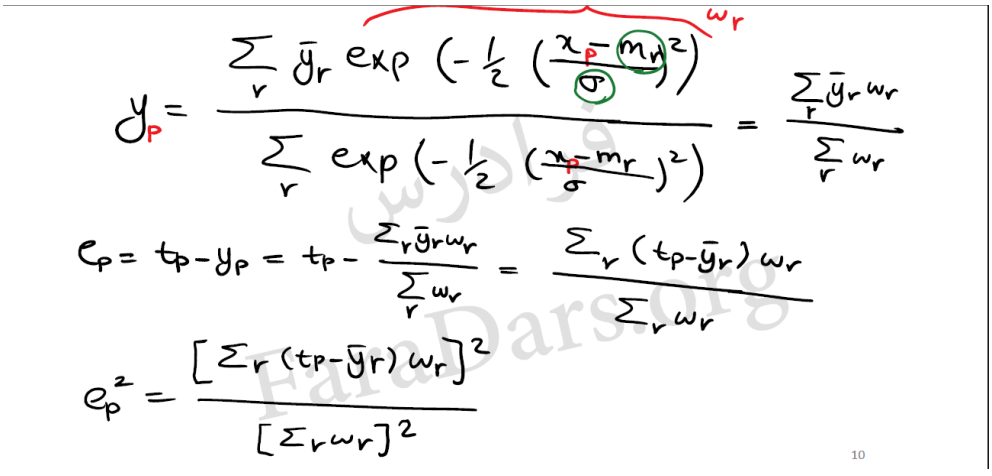
$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^2$$

تابع هزینه خود را مینویسم و در ادامه راه باید به بهینه سازی این بپر دازیم .

در ان مسئله پارامتر های ما شامل σ, m_i ها به حساب میایند .

خب در ادامه برای این کهن بخواهیم مسئله را کمی از پیچیدگی فرم بالا در این حالت کاهش دهیم و بتوانیم آن را بهتر بیان کنیم اسلاید زیر آورده ام که در ادامه توضیحات کافی برای آن میاورم :

خب همان طور که میبنید بخشی از تابع TSK ما که قسمت نمایی یا همان گووسی آن است را با ω_r نمایش میدهیم که ساده تر شو سپس خطای بین سیستم و



تابع مدل سازی شده خودمان را میاوریم در نهایت با یک سری عملیات جبری و ساده سازی به معادله خط ئدو میرسیم که مخرج مشترک گیری و... دارد .

در انتهای اسلاید هم که تابع خطا خود را معرفی نموده ایم که تمامی عبارت های خطا باید به توان دوکم رسد یا مربع شود بعد از این کار باید عملیات گرادیان گیری روی پارامتر های متغیر انجام داد تا مسئله بهینه شود.

در ادامه روند کاری یک اسلاید آورده شده که عملیات گرادیان گیری روی پارامتر m_r, σ چگونه است :

$$\frac{\partial}{\partial m_r}\left[\sum_{r'}(t_p-\bar{y}_{r'})\,\omega_{r'}\right]^2 =$$

$$\frac{\partial}{\partial \omega_r}\left[\sum_{r'}(t_p-\bar{y}_{r'})\,\omega_{r'}\right]^2 \cdot \frac{\partial \omega_r}{\partial m_r}$$

$$2(t_p-\bar{y}_r)\sum_{r'}(t_p-\bar{y}_{r'})\,\omega_{r'}\,\frac{\partial \omega_r}{\partial m_r}$$

$$\frac{dy}{dt}=\frac{dy}{du}\,\frac{du}{dt}$$

روال کاری بدین صورت است که برای عملیات گرادیان گیری یا مشتق گرفتن ضمنی از پارامتر های خود باید از قائده زنجیره ای استفاده کنیم .که در انتهای شکل فرم روند مشتق زنجیره ای آورده شده است .

حال برای این که چگونه کار کرده میبینیم که ابتدای کار از عبارت نسبت به ω_r مشتق گرفته شده سپس برای این که گرادیان آن نسبت به m داشته باشیم در عبارت زنجیره ای $\frac{\partial \omega_r}{\partial m_r}$ ضرب شده است کلا عبارت اول نسبت به ω_r مشتق گیری شده است .

در نهایت برای پارامتر سیگمات هم همچنین روالی را داریم که باید ما سیستم خود را انقدر یواش یواش جوری جلوببریم که این تابع هزینه ما به سیستم مقدارش کم کم بشه به نحوی که سیستم ما خطای حداقلی خودش برسد .

$$m_{r,t} \rightsquigarrow m_{r,t+1}$$

$$\begin{cases} m_{r,t+1} = m_{r,t} - \alpha_{r,t}\,\dfrac{\partial E}{\partial m_{r,t}} & \forall r \\ \sigma_{t+1} = \sigma_t - \alpha_{\sigma,t}\,\dfrac{\partial E}{\partial \sigma_t} \end{cases}$$

خب شکل گویای صحبت های من است که کلا میخواهیم چیکار بکنیم در این روند کاری که در حال حاضر پیش گرفته ایم .

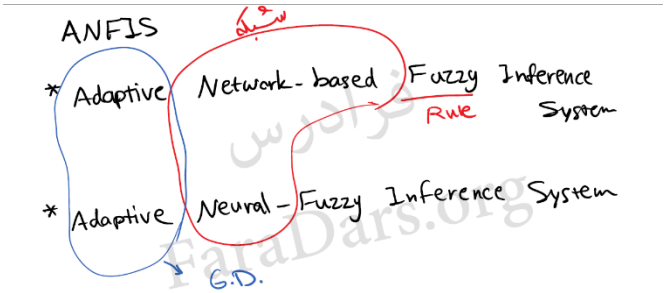
خب رسیدیم به یک جای زیبا در این قسمت براتون یک سخن زیبا دارم که اینه که ما الان شروع شبکه های عصبی را یادگرفته ایم که برای این روال کاری که از ابتدای این جزوه یا مقاله برای شما آورده ام نتا ایجا این بوده که بتونیم با کمک دانش های قبلی و پایه ای ریاضیات و منطق فازی یک چیزی الان درست بشه تحت عنوان

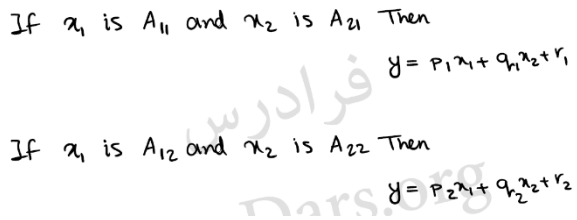
(ANFIS or Adaptive Nueral Fuzzy Inference System) گفته میشود حال به ادامه کار برای ما هیجان انگیز تر از قبل میشود اون قسمت تطبیقی یعنی ما یک سری اطلاعات به مدل میدهم که رفته رفته خودش را با مدل مرجع که سیستم مدنظر ما است تطبیق پیدا میکند .

خب به دلیل این که ما از متلب برای کار خود استفاده میکنیم باید با روال کاری ANFIS در متلب بلد باشیم مثل سیستم های FIS در جزوات قبلی به صورت کامل در محیط متلب کد نویسی و کار با ماژول فازی را یادگرفتیم .

این نکته در رابطه با این که این ANFIS در متلب چگونه کار میکند یک مدل خطی مرتبه اول است ، از مدل TSK پیروی میکند یعنی ورودی یک یا چند سری ورودی داریم اما خروجی به صورت یک تابع است حال قبلا در سیستم TSK در متلب به صورت یک ثابت در نظر گرفتیم ما در حال حاضر که در ادامه یک اسلاید از قوائد فازی و چگونگی کار کردن این سیستم در متلب است را بر ای شما آورده ام :

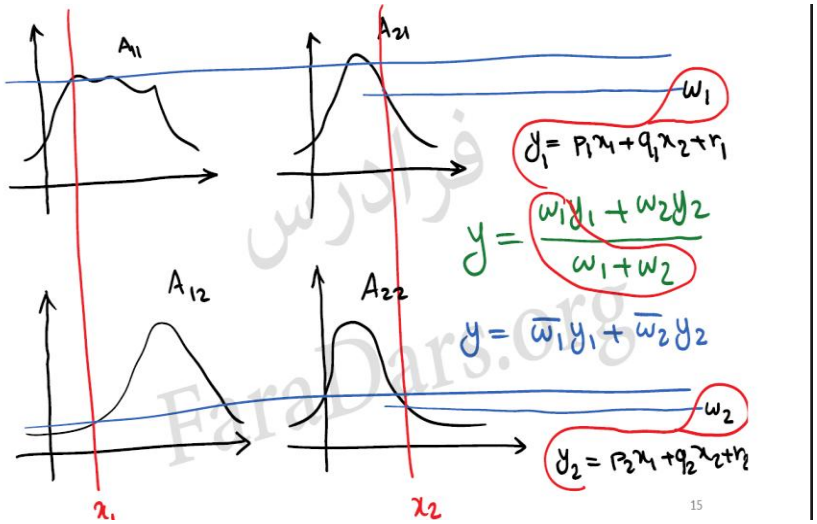
این تصویر شماتیک کلی سیستم را توضیح میدهد .





این تصویر بر ای ما مهم است از انجایی که یک سری ورودی ها داریم یعنی دوتا ورودی داریم و خروجی های ما در تصویر مشاهده میکنید که به صورت یک ترکیب خطی بیان شده است نه یک ثابت .

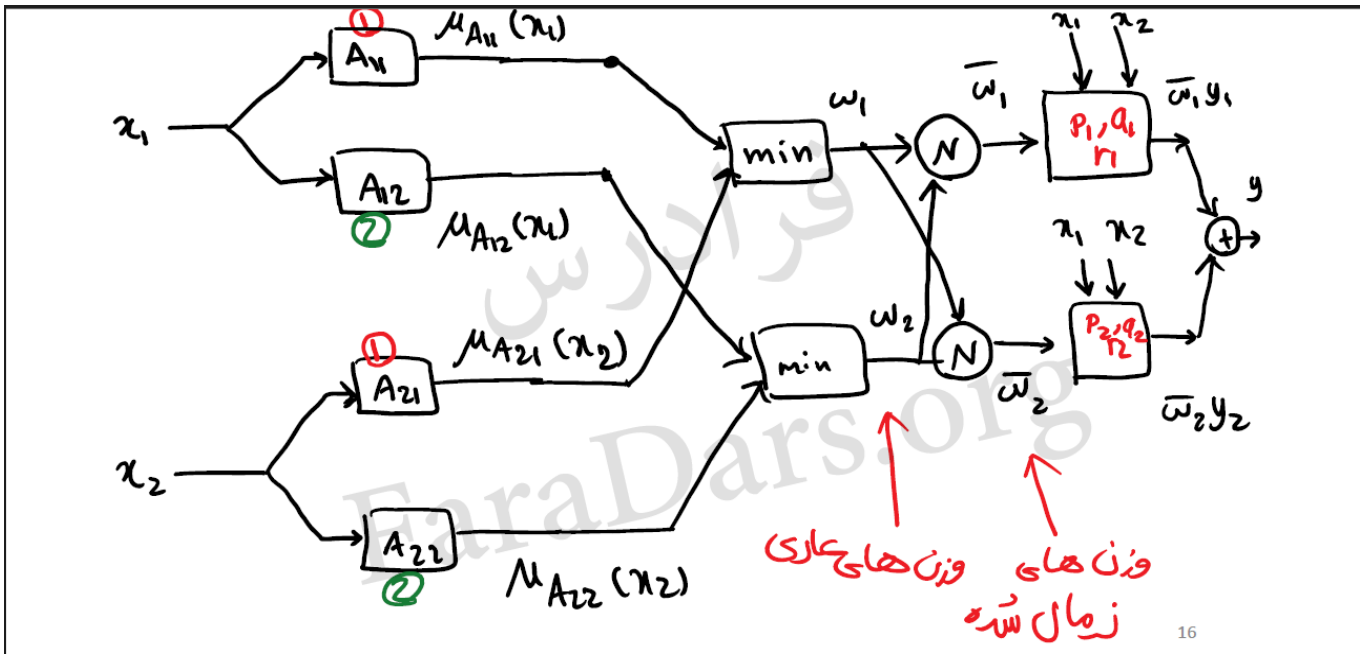
خب در ادامه برای این دو قائده فازی که نوشتیم یک اسلاید میاوریم توضیحات را روی آن میدهم :



خب در تصویر میبنید که 4 توزیع مختلف برای A در نظر گرفته شده است یک بار با ورودی x1 که با خط قرمز رنگ کشیده شده یک بار هم با ورودی x2 که با خط قرمز رنگ کشیده شده بر اساس قائده هایی که نوشتیم بر روی توزیع های خود اعمال میکنیم . (منظور از توزیع ها همان اطلاعات است دوستان عزیز خوبم) خب در ادامه خطوط آبی رنگ ما همان yهای ما هستند که بر حسب نوع ورودی که x1 یا x2 است رسم میکنیم که باید ما بین هر کدام از آنها اونی که کم تر است را انتخاب کنیم در شکل پیدا است که دوتا خط آبی رنگ

داریم به صورت افقی که بر حسب .ورودی های ما است حال باید مابین آنها کم ترین را انتخاب کنیم سپس اسمش ω_r است یا همان وزن که باید خروجی را در همان وزن کم تر ضرب کرد سپس مجموع آن ها برای یک سیستم تقسیم بر تمامی جمع وزن ها میکنیم که خروجی اصلی ما که بهینه ترین حالت ممکنه را به خود میگرد را به ما میدهد .

حال صحبت هایی که کردیم را در قالب گرافیکی میاوریم یک سری توضیحات تکمیلی هم بر روی آن میدهیم :

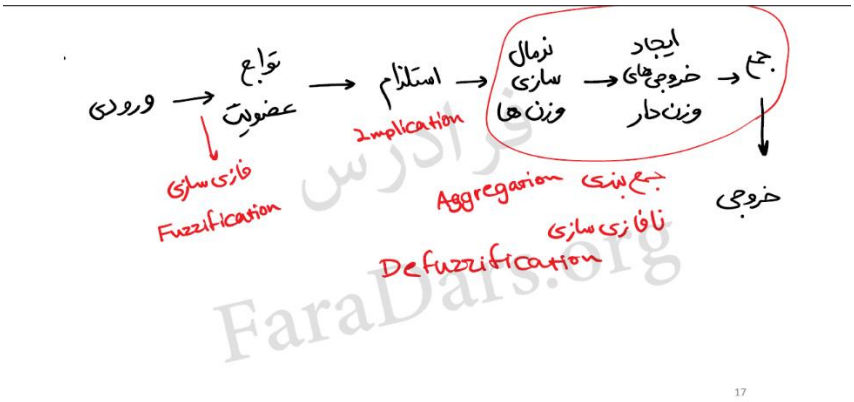


خب در تصویر یک شبکه عصبی را میبینید که ابتدای کار دوتا ورودی به سیستم داده ایم سپس در لایه اول که فازی کنندخه است قوانین بر روی ورودی ها پیاده سازی شده در کنار اطلاعاتی که داریم میبینید که یک بار مثلا x1 وارده قائده سبز یا اول شده خروجی آن یک Membership function از x1 ساخته میشود با اطلاعاتی که است یعنی همان تابع $\mu_{A_{11}}(x_1)$ به ترتیب همین روال برای ورودی دیگر و قائده های خود که بر حسب منطق فازی هستند را پیاده سازی میکنیم .

حال در ادامه باید در بین این توابعی که بدست آوردیم به عنوان خروجی منیمم را انتخاب کرده هر یک از قائده ها با خودش باید کمینه شود و قیاس شود که خروجی این کار همان وزن دهی است که این لایه را وزن دهی گویند سپس نرمال سازی میکنیم لایه بعدی که با N نمایش داده شده که در قسمت قبلی کامل توضیح داده شد، در نهایت خروجی های ما به عنوان بهترین عملکر هستند که

کمینه شدن و خطای حداقلی را با سیستم دارند .حال یک سری جزئیات در این بلوک دیاگرام است باید در ادامه کار که میرویم باهم جلو آن ها را هم یواش یواش یاد میگیریم در حال حاضر مهم این است که بفهمیم چه روالی طی میشود در حالت کلی .

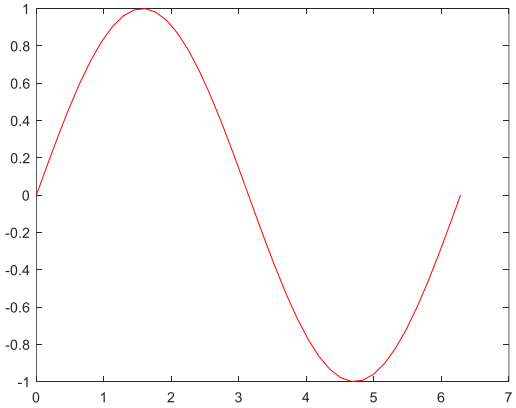
در این اسلاید که برای شما میاورم میبیند که ANFIS جز همان سیستم عصبی قبلی نیست فقط فقط ما یک خطایی را تعریف نموده ایم برای سیستم خود با مدل که ANFIS به صورت تطبیقی خودش را یا مدل را با آن یکسان میکند :



اگر یادتان باشد یک سیستم فازی از همین قسمت ها تشکیل شده بود اول فازی سازی میکردیم ورودی ها را سپس توابع ضویت ساخته میشد در ادامه میرفتیم بر حسب استلزام ها قائده های فازی را میساختیم و بعد از آن ترکیب میساختیم از آن ها و در نهایت نافازی سازی انجام میدادیم در نهایت خروجی میگرفتیم اینجا علاوه بر ترکیب کردن ما باید وزن دهی کنیم و کمینه سازی کنیم چون میخواهیم مقدار خطا را کم کنیم که اینجا همان بحث تطبیق پیدا شدن است سرو کله اش پیدا میشود. در کلام ساده همه اینا همون منطق فازی است که یک سری فرم های ریاضی در خودش تنیده شده که بتونیم یک سیستم را بهتر ازش بهر گیری کنیم که برای این امر که گفتیم شبکه عصبی میاد وسط ولی همان موضوع قبلی پابرجاست روال منطق فازی !!!!!!

خب در ادامه کار میخواهیم یک تابع را به صورت واضح برای ما آشنا است رسم کنیم مثل تابع sin(x) سپس در محیط (anfisedit) در متلب آن را تخمین بزنیم یا حدس بزنیم که سیستم ما مثلا یا Black box ما دارای یک رابطه sin(x) دارد که خروجی به ما میدهد یعنی پیشبینی کنیم در اون سیستمک رخداد های آن چگونه است .

خب بریم اصل مطلب که کد نویسی است خب برای شروع تابع sin(x) در محیط متلب بین بازه x = linspace (0,2*pi,40) که بین 0 تا 2 پی است که دستور linspace به 40 قسمت مساوری این بخش را برای ما تقسیم بندی میکند . سپس y = sin(x) بعد از آن با دستور plot مینویسم plot(x,y) نمودار سینوسی ما بین بازه 0 تا 2پی به ما میدهد که به شکل زیر است :



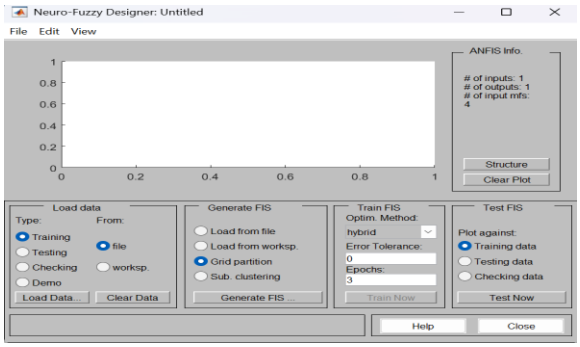
اینم خروجی ما برای تابع سینوسی ما بین بازه ای که گفتیم واضح بود که چه چیزی به ما میدهد متلب .

حال به محیط anfisedit میرویم که کافیه در قسمت commandwindow همین چیزی که نوشته ام را تایپ بشه که در تصویر زیر میاوریم :

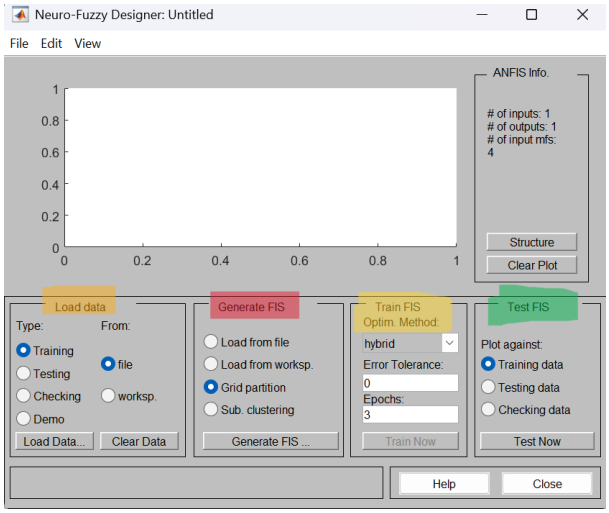
```
>> anfisedit
>>
```

این دستور را در ترمینال متلب هست که تایپ میکنیم سپس نمایی که میگیریم این پنجره برای ما باز میشود :

پنجره ای که باز میشود این است که ما در این محیط باید کار های خود را پیش ببریم .



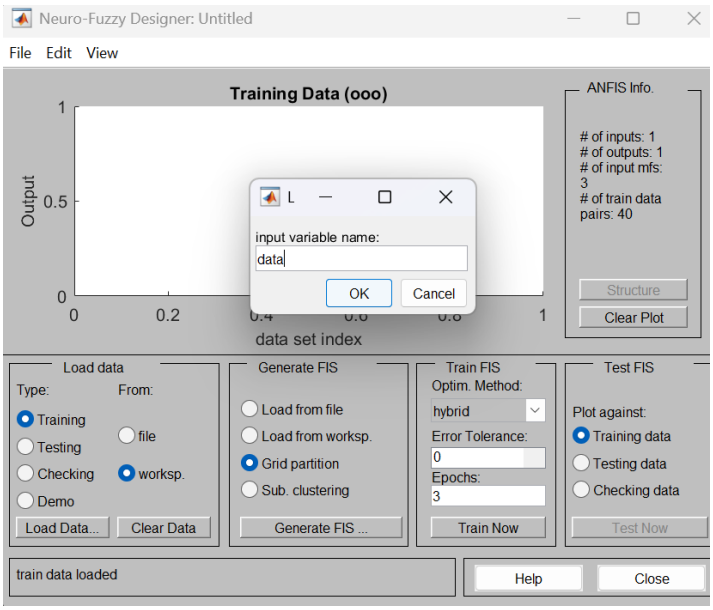
در ادامه میخواهیم این قسمت های این پنجره را کمی باز تر توضیح دهیم :



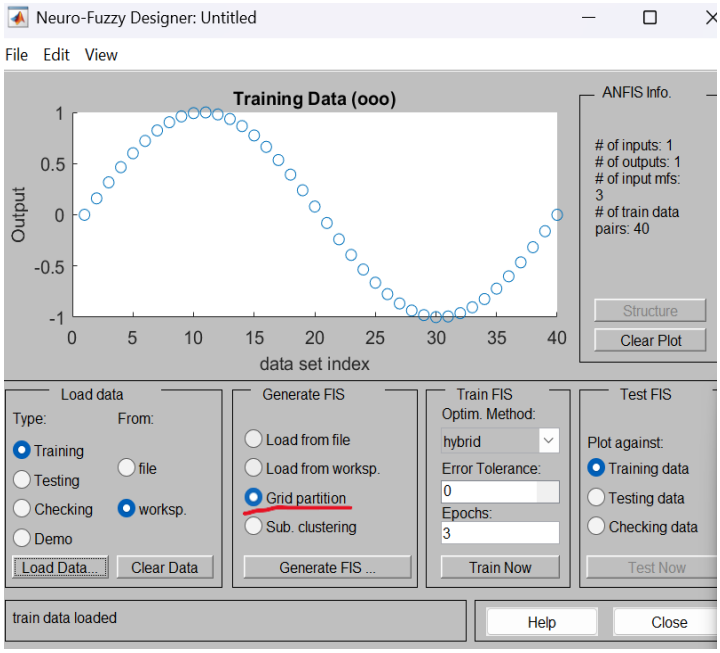
خب اینجا یک سری هایلایت شده بخش ها قسمت اول Load Data است از قسمت traning داریم که اطلاعات را از ما میگیرد به عنوان ورودی دومی testing برای زمانی که الگوریتم نوشته ایم حال میخواهیم تست کنیم cheaking هم برای چک کردن اطلاعات ورودی است و درنهایت Demo یک سری اطلاعات پایه ای را میدهد که ما بر روی آن کار میکنیم .

قسمت from در سمت راست همان این مشخص کننده این است که اطلاعات را از کجا فراخوانی کند به صورت فایل یا workspace که مثلا برای نمونه ما اطلاعات خودمون را در یک متغیر ریختیم تحت عنوان دیتا سپس از این قسمت workspace برای سیستم anfis خود فراخوانی کردیم که به شکل زیر است :

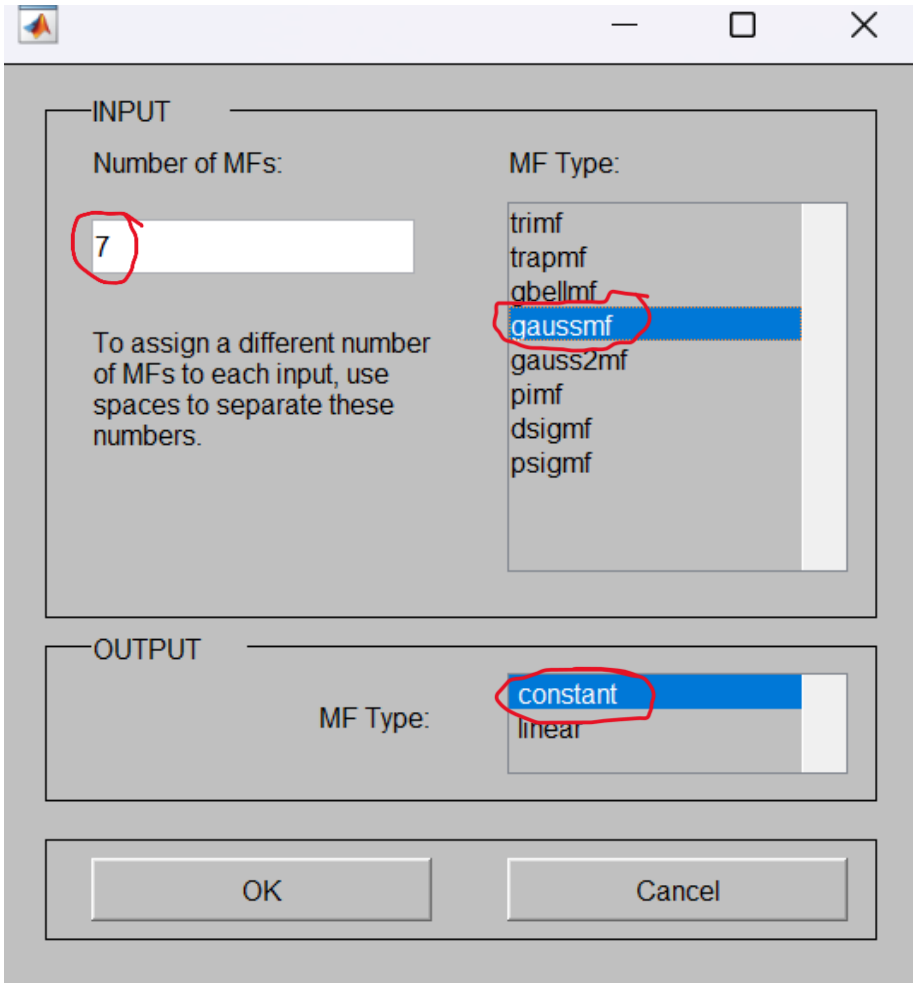
نحوه وارد کردن اطلاعات به سیستم anfis متلب در ادامه کار نمودار سینوسی ما را رسم میکند .



اینم خروجی کار پس از رسم نمودار ما در این محیط .

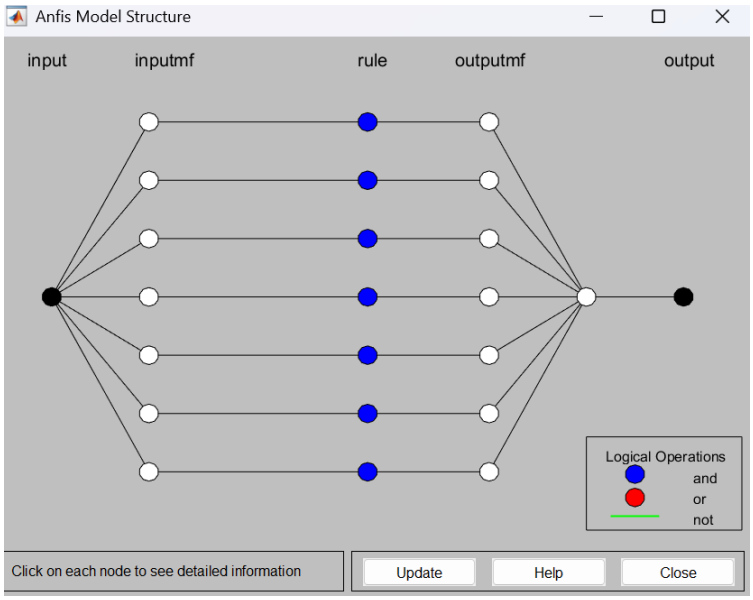


خب در ادامه میخواهیم با قسمت دوم هایلایت شده بحث کنیم که همان جایی است که ما قائده های خود را میدهیم که یعنی ساخت سیستم منطق فازی در این قسمت رخ میدهد که برای این کار دوتای اول فراخوانی یک فایل منطق فازی از یک جایی در سیستم یا workspace است سومی همان روشی است که ما دیتای ورودی خود را به تعداد هایی دلخواه تقسیم میکردیم سپس نوع تابعی member functions هم انتخاب میکنیم و در نهایت برای خروجی نوع آن را باید تعین کنیم که ثبات باشد یا مدل خطی که در اینجا مدل خط ی استفاده کرده ایم ، در ادامه چند شکل را برای روند کاری این بخش آورده شده است :



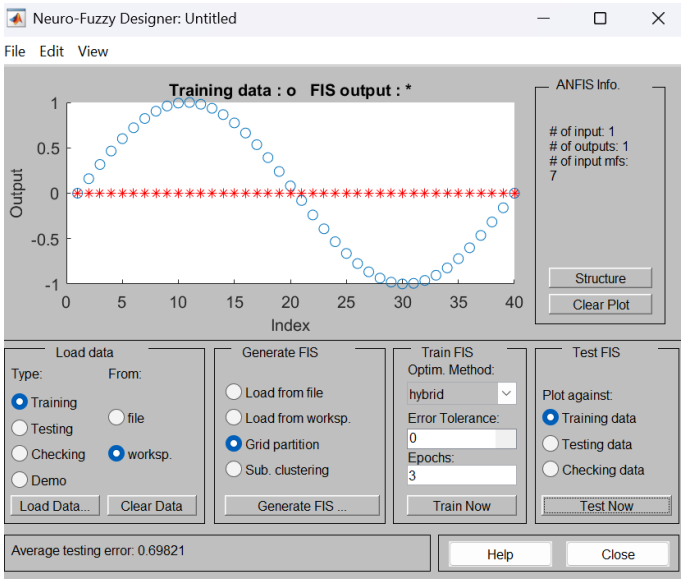
خب در تصویر به خوبی مشاهده مشیود روش grid position را انتخاب کرده ایم برای این کار ورودی خود را به 7 قسمت تقسیم بندی کردیم نوع تابع membership function از نوع guassmf استن و خروجی هم ثلابت گرفتیم .

حال اگر بیایم روی آیکون structure کنیم یک چنین تصویری مشاهده میشود که تعداد تقسیم ها لایه های ما را نشان میدهد :



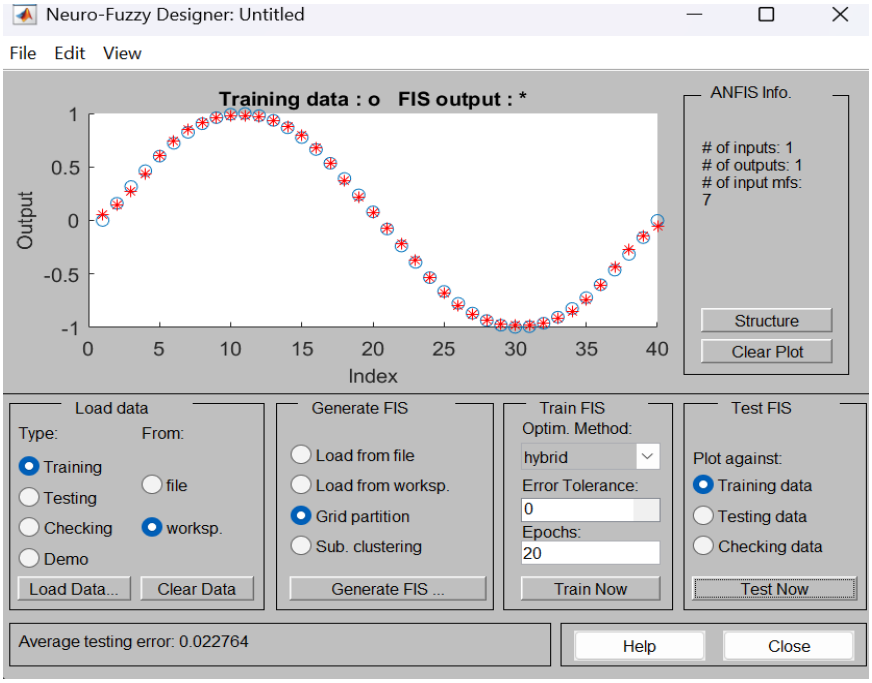
خب در حال حاضر مشاهده میکنید که سیستم ما شامل یک ورودی است 7 تا قسمت تقسیم بندی ورودی را انجام داده ایم سپس قوانین فازی که در اینجا ANDپیاده سازی شده است ، خروجی های ما هم به صورت ثابت عمل کرده و در نهایت در یک جمع کننده می رسد به خروجی .

حال یک بار این را بدون آموزش تست بگیریم میبنید که الگورتیم فازی ما به چه نحوی کار میکند :



خب همان طورز که مشاهده میشود به صورت یک خط راست تست شده اغین الگوریتم فازی که نشان دهنده این است که این الگوریتم فازی خام و بدون آموزش می‌باشد ، حال در قمست TrainFis که برای آموزش دهی به الگورریتم است استفاده میکنیم که از منوی کشویی از نوع hybrid که برای اطلاعات بیشتر به منابع متلب مراجعه کنید که این چیست ، و در قسمت Error یا همان خطایی که صحبتش بود 0 می گذاریم و Epochs

را روی عدد 20 قرار میدهیم و ببینیم چه چیزی رخ میدهد .

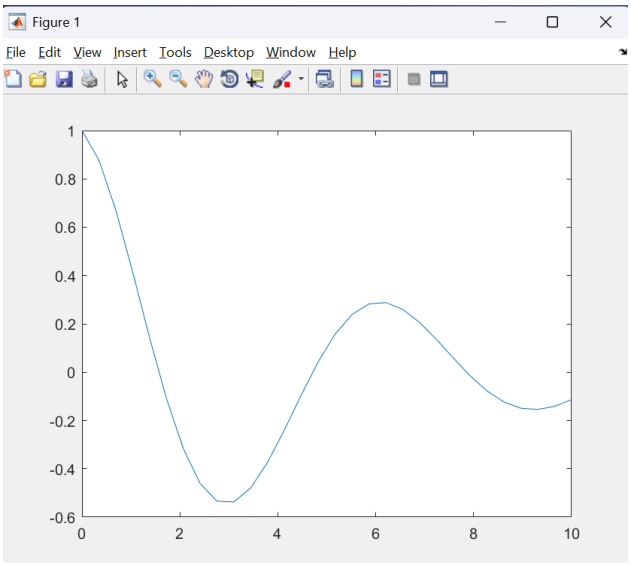


خب میبینید که الگورتیم آموزش دید و تقریبا با نمودار سینوسی که در ابتدای که ما رسم کرده بودیم که فرض میگریم Black Box ما باشد تقریبا یکسان شده حال این عدد را بیشترش کنیم نتایج به همگرایی بیشتری میرسیم. یادتان باشد به ساختار الگورتیم با تغیر دادن این عدد دستخوش تغیر نمی‌شود .

حال یک سری تغیرات بالای آیکون این تولباکس داریم که بزنیم همان روال مثل منطق فازی هست که نمودار میدهد ،قوانین ها را برای ما به نمایش میگذارد و یک سری اطلاعاتدیگر مثل اکسپرت گرفتن از فایل ایینجور چیزا رو میتوان در منو های بالای کادر انجام داد که دقیقا مثل منطق فازی است .

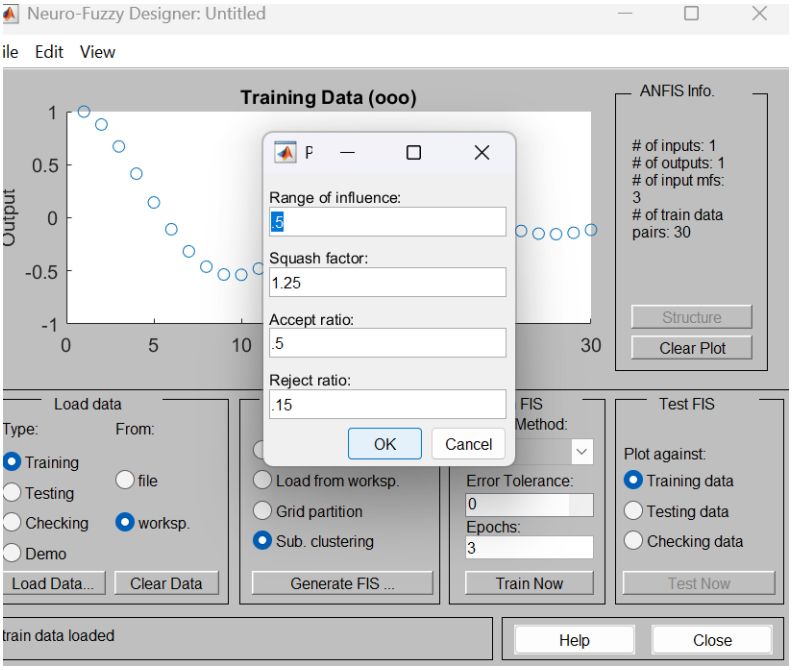
حال دوباره یک سری x , y را تولید کرده با این فرمول که x =linspace (0,10,30) و

y= exp(-0.2*x).*cos(x) که این هم دیتاهای y ما را تشکیل میدهد حل پلات کنیم به شکل زیر در میاید :



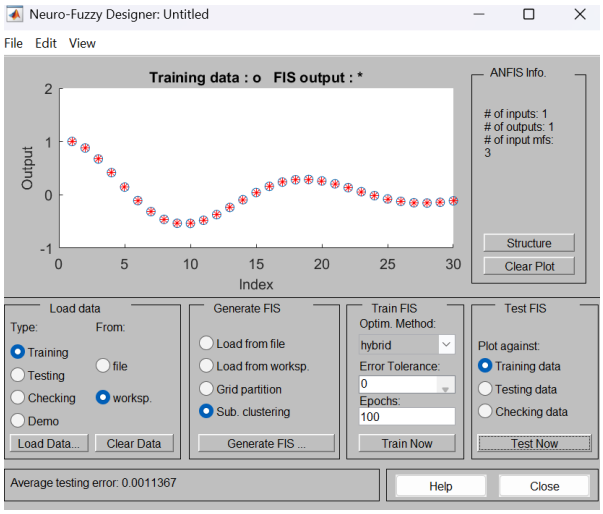
خب این خروجی کار ما است برای شروع این اطلاعات را به عنوان دیتا های اولیه به سیستم anfis خود میدهیم حال میخواهیم در قسمت دوم کار یعنی قوائد کاری کا در قسمت قبلی با Grid paratiation کار کردیم حال با گزینه آخر میخواهیم برای کار شروع به فعالیت کنیم.

حال با گزینه آخر sub clustring میخواهیم کار کنیم که برای این منظور ما ابتدای کار بر روی آیکون آن زده و از چند قسمت دارد که برای توضیح بیشتر یک عکس آورده ام :



خب این قسمت شامل چند بخش است برای کار کردن که آیکون اول همان مقداری است که توابع عضویت ها در هم ادغام شوند که اینجا 50به 50 است دومین آیکون مربوط به شکل توابع عضویت است حال ما به این سیستم زیاد نمیپردازیم در ادا مه راه با تمامی این نکات آشنای کامل پیدا میکنیم .

کلید اوکی را میزنیم و Traning Data را زده سپس نتیجه روی دیتای اصلی چک میکنیم که میبینیم خروجی کار خیلی با خطای کمی به دیتای اصلی نزدیک شده است .



میزان خطا در در پایین کادر آورده شده که چیزی حدود 0.0011 است و میبینیم که دیتا ها با آن چیزی که ما با منطق فازی ساخته ایم یکسان شده است .

وقتی که دیتاها تعداد آن ها بالا میرود ما نیاز به تفکیک اطلاعات داریم و نظم بخشیدن درست حسابی به آن ها نیازمند هستیم که برای این کار از منوی sub clisturing میتوانیم بهترین بهره گیری را کرد .

مثلا برای تعداد رنگ ها یا دیتاهای مربوط به گیاهان و... که در دنیای ما است .

خب همان طور که مشاهد ه میکنید این روش بهتر عمل میکند نسبت به Look up Table چون روند الگورتیم به نحوی است که خودش را با مرور زمان با سیستم تطبیق دهد.

حال بر روی معادله Mackey glasses اگر یادتان بود ما Lookup Table را پیدا سازی کردیم در حاضر هم میخواهیم این کار را بر روی آن معادله با ماژول anfisedit متلب انجام دهیم که برای این کار من یک قایل به عنوان اطلاعات ساخته ام که با نام DBZ_Fuzzylogic_Modeling_project3 در گیت هاب این را آپلود میکنم که اطلاعات اولیه برای سیستم ما است که میخواهیم کار کنیم .

حال به محیط anfisedit میرویم فقط قبل از آن این فایل را باید ران کنیم که اطلاعات آن در workspace ریخته شود تا بتوانیم با آن کار کنیم .

فایل اطلاعات نوشته شدوه است حال در محیط anfis رفته :

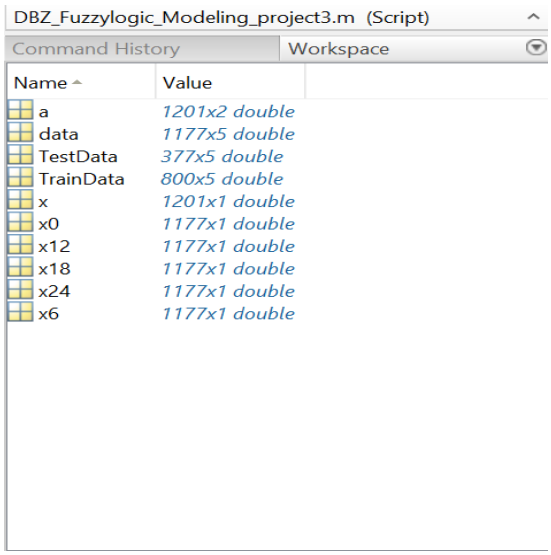
```
a = load('mgdata.dat');
x = a(:,2);
plot(x);

x0 = x(25:end);
x6 = x(19:end-6);
x12 = x(13:end-12);
x18 = x(7 :end-18);
x24 = x(1:end-24);
data = [x24 x18 x12 x6 x0];
TrainData = data(1:800,:);
TestData  = data (801:end ,:);
```

خب در ادامه راه میخواهیم دو رویکرد متفاوت از پیاده سازی این اطلاعات با کمکک منطق فازی در سیتم anfis متلب در نظر گرفت یکی با استفاده از Grid partion و یکی از sub clustering استفاده کنیم که ببنیم چقدر این دو روش برای این سیستم دینامیک آشوب ناک کدام یک نتیجه سریع تری به ما میدهد و وقت و زمان کم تر و بهنیه تر هم عمل میکنید .

خب ابتدای کار همان روال های قبلی برای اضافه کردن اطلاعات دادریم یکی اطلاعات TrainData است که به عنوان دیتای Train است یکی هم Tetst Data به عنوان دیتای تست از workspace میخوانیم که این کار را قبلات انجام د اده ایم حال در قسمت Genatfis یا ساخت سیستم فازی باید Grid را انتخاب میکنیم که در آن 4 ورودی برای ما در نظر گرفته است حال میتوانیم هر کدام از ورودی های خود را به چند قسمت تقسیم کنیم که من این پارتیشن بندی را به 5 قسمت برای 4 ورودی خود در نظر گرفته ام در ادامه سپس سیستم منطق فازی خود را ساخنته اتیم که خام است از قسمت Trian FIS آلگوریتم خود را آموزش دهیم که برای این کار خطا رو صفر و Epoches را روی 100 تنظیم مینکنیم که در ادامه تصاویر مربوط به اینم مراحل را میاوریم :

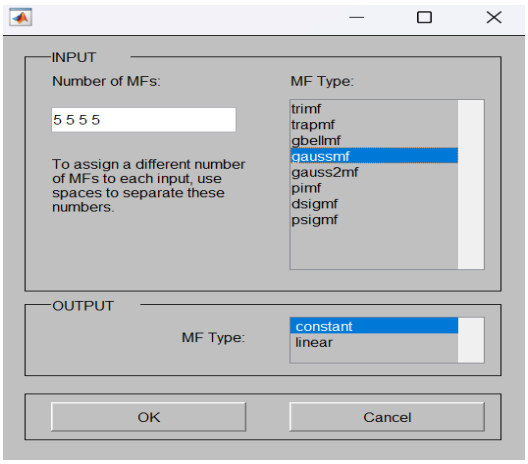
این همان اطلاعاتی است که در قسمت workspace ران کرده ایم آورده شده است .



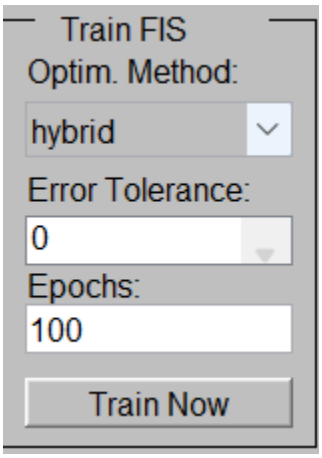
تصویر مقابل بعد اضافه کردن اطلاعات خود به محیط یا ماژول ANFIS Matlab میباشد.

این قسمت مربوط به مش بندی است که همانطور که گفته شد به 5 قسمت هر یک از ورودی ها تقسیم بندی کرده ایم .

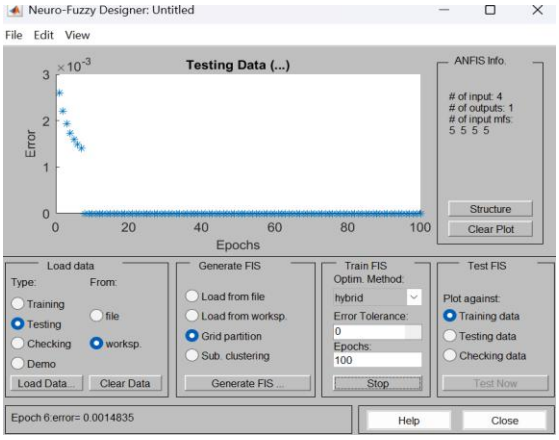
نوع توزیع داده ها را از تابع gaussmf استفاده کرده ایم و خروجی نیز به عنوان ثابت در نظر گرفته ایم .



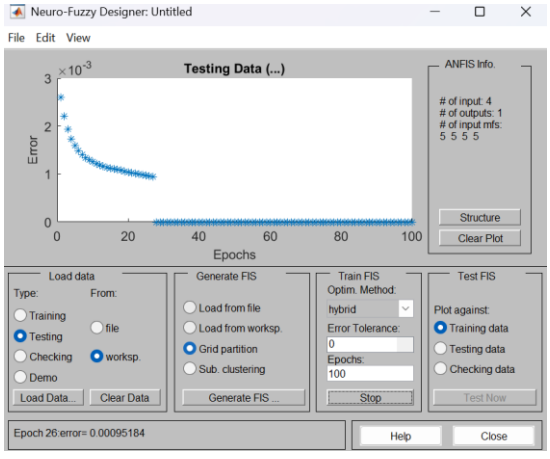
حال قسمت بعدی Train FIS است که تصویرش میگذاریم :



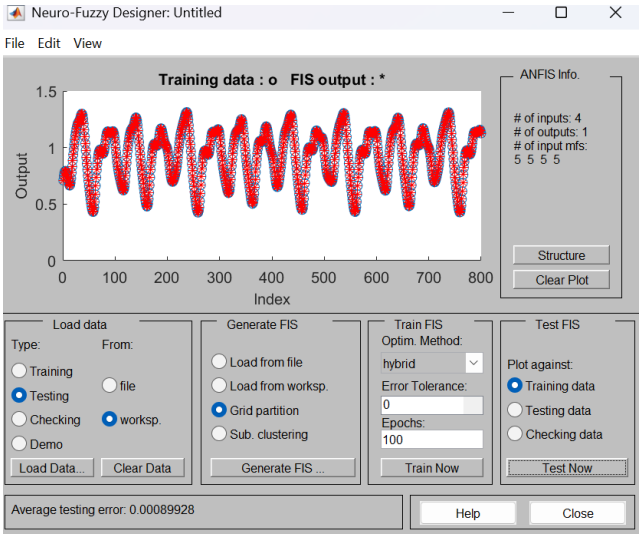
خب این تصویر گویای همچی است حال فقط دکمه Trian Now را بزنیم تا الگوریتم منطق فازی ما آموزش ببنید که سپس شروع به آموزش دیدن میکند و چند تصویر از هنگامی که این سیستم آموزش میبیند را برای شما میاورم که ببنید چه رفتار کندی را از خود نمایش میدهد وقتی که از این متد استنفاده میکنیم .



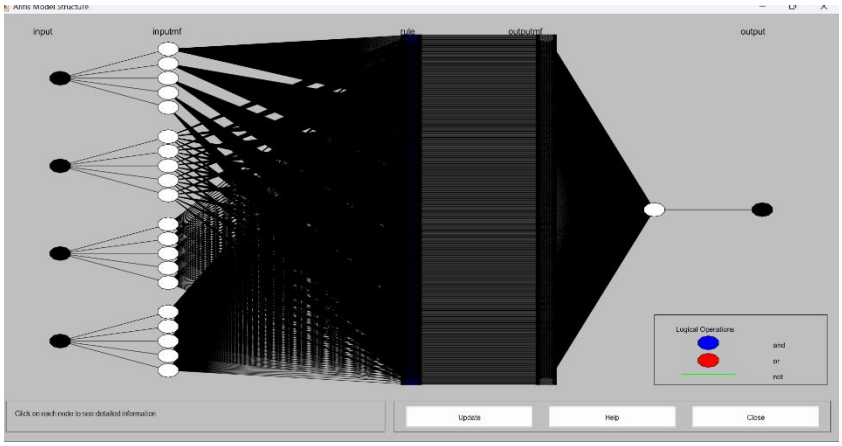
خب ما سیستم تا همین جایی که میرسد traningرا متوقف میکنیم چون زمان زیاد و سخت افزار خیلی درگیر میشود ولی خروجی کار خیلی خطایی کمی دار د با همین اوصاف ولی میخواهم یک نصویر برای شما نمایش بدم از پیچیدگی شبکه عبی که این سیستم از این روش تتشکیل میدهد چقدر سخت و پیچیدگی دارد .



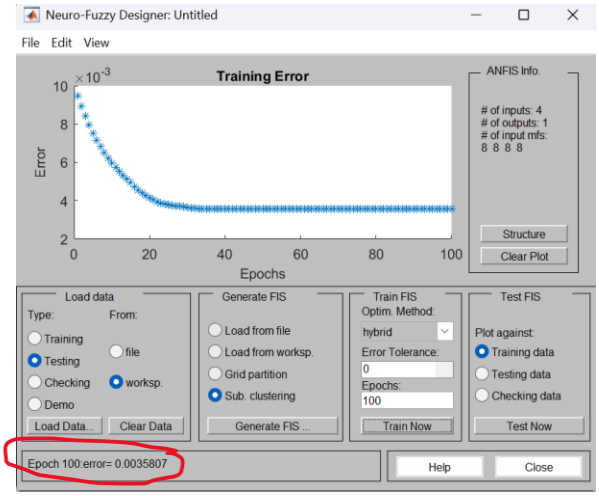
این خروجی کار است که میبینید بر روی دیتا های ما فیت شده است و با خطایی 0.0008 اما پیچیدگی سیستم عصبی فازی این روش را برای شما به نمایش میگذارم که این روش چقدر روش بدی است بر تقریب زدن این سیستم آشوب ناک :



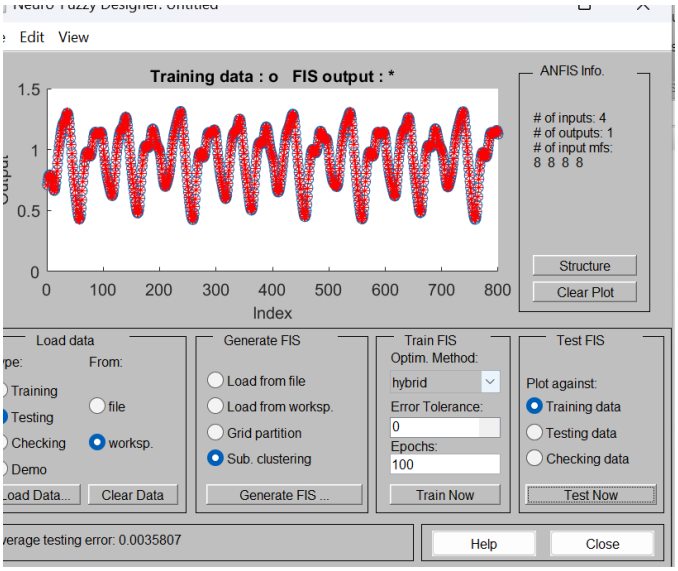
بله تصویر گویای همه چیز است حال این بار روش خود را تغیر داده و از قسمت نوع سیستم فازی نوع sub clustering استفاده میکنیم و نتایج را فقط برای شما میگذارم ببنید و مابقی روش همان روش های قبلی را در پیش میگیریم.

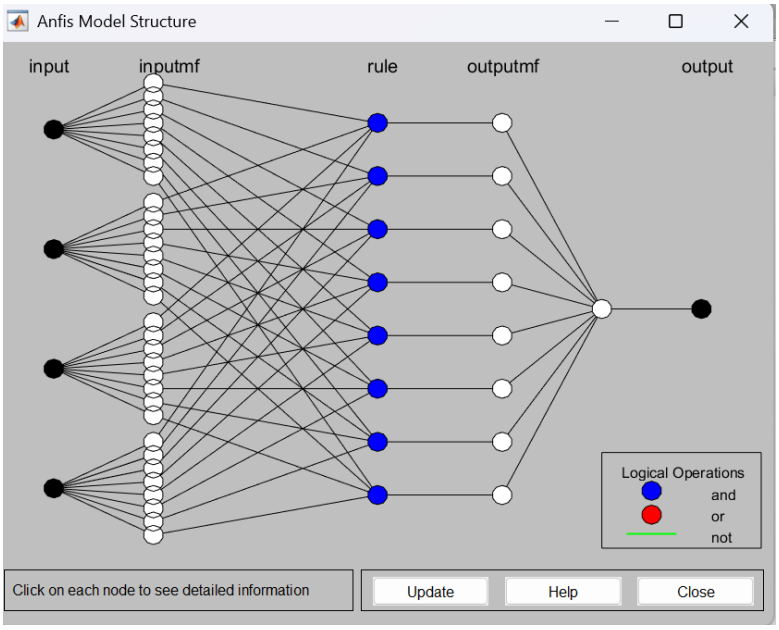


خب اینجا میبنید اولا که یادگیری سیستم ما کاملا انجام شده تا آخرین مرحله همچنینن خروجی کار هم برای شما به نمایش میگذارم که خروجی نیز به خوبی همگرا شده خطایی حدود 0.0035 دارد و نتایج خروجی برای شما میگذارم و در نهایت تصویر شبکه عصبی فازی ساده ای که از اینم روش میرویم برای شما میگذارم تا ببنید بسته به نوع مسئله ما باید کدام یک از روش های موجود در آموزش شبکه عصبی



استفاده کرد حتی گاهی اوقات ما باید روش هایی به کار ببریم فرابتکاری یا نوین برای آموزش دادن به سیستم های عصبی خودمان خب تصویر ها به شرح زیر است:





خودتان میتوانید قیاس کنید کدام روش منطقی تر است برای کار ما در دینامیک اینم مسئله ای که داریم حال برای این سیستم ها بنده دوتا فایل با نام های :

Meckey_Glasses_one.fis

Meckey_Glasses_two.fis

در گیت هاب آپلود میکنم فقط شما باید این دوتا فایل همراه با فایل :

DBZ_Fuzzylogic_Modeling_project3 در سیستم خود ران کنید و نتایج را ببنید .