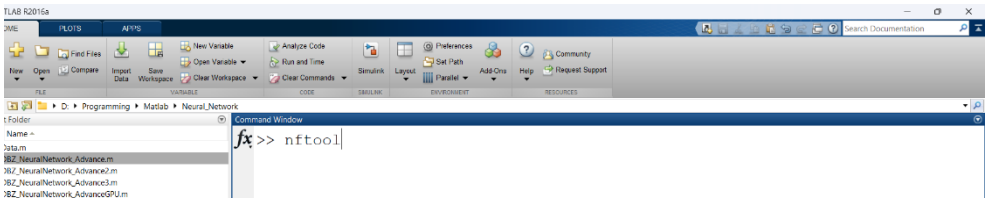
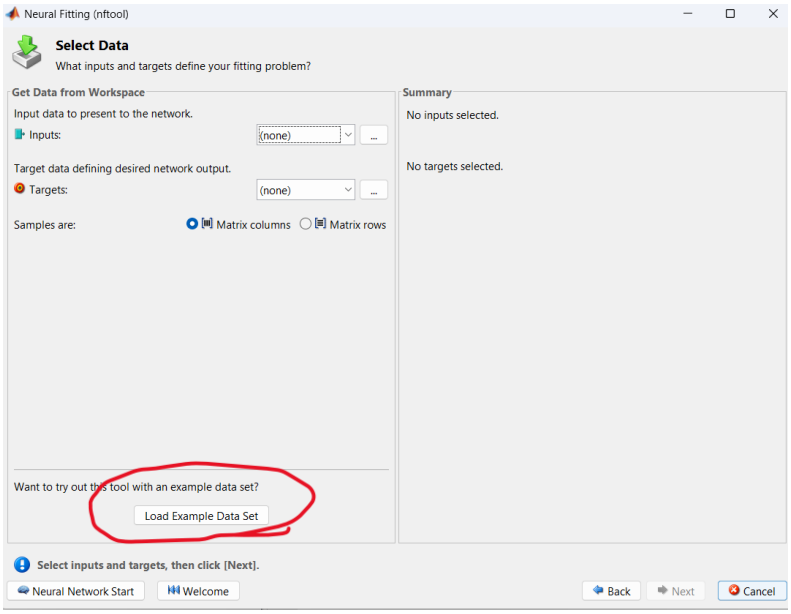
جلسه سوم شبکه های عصبی (MLP چند لایه) :

خب در این جلسه همانند جلسات قبلی ما ابتدای کار باید یک سری دیتابرای کار خود فراخوانی کنیم که برای این امر ما مستلزم این هستیم که دیتاهای مناسب و واقعی داشته باشیم و آن ها را Train کارهای دیگه بر روی آن ها انجام دهیم .

برای این که ما ازخود متلب استفاده میکنیم که در شبکه عصبی متلب وقتی که nftool را در قسمت ترمینال متلب فراخوانی کنیم پنجره ای به شکل زیر باز میشود که در دادامه برای شما توضیح میدهم :

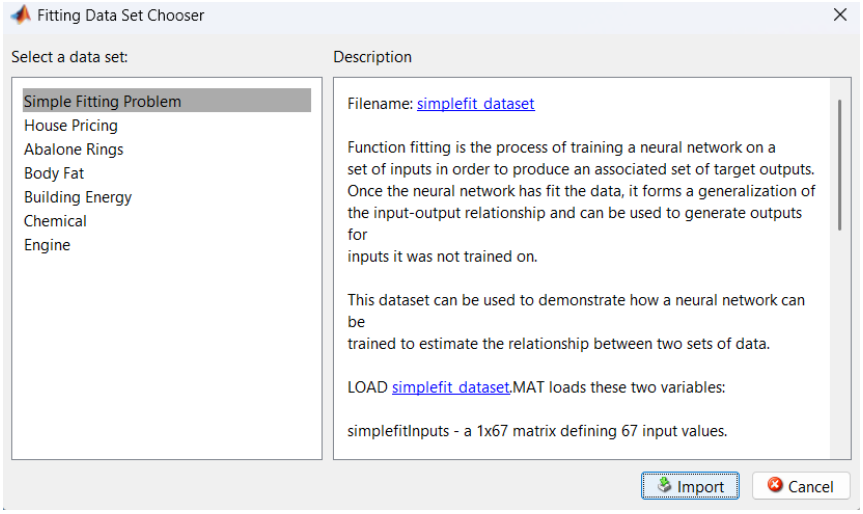


این دستور را مینزنیم برای فراخوانی جعبه ابزار خود سپس از قسمت بعد دیتا میخواهیم آماده در خود متلب به صورت پیش فرض آماده سازی کرده است استفاده کنیم :



خب این قسمت روی آن کلیک کنید یک سری اطلاعات برای کار با شبکه عصبی به ما میدهد که در این قسمت تصویر برای این قسمت همم برای شما میگذارم :

اینم اطلاعاغتی که در خود متلاب وجود دارد البته بگم این نسخه متلب من 2016 است و قطعا در نسخه های جدید تفاوت های دارد اما در کل و د ستورات و امکانات خاصی در نسخه های جدید من ندیده ام .



حال امروز با اطلاعات Engine میخواهیم کار کنیم که برای این کار این اطلاعات را Import میکنیم سپس در Workspace این اطلاعات حضور پیدا میکنند.

شما همیشه نمیتوانید یک شبکه عصبی را برای چندین مدل دیتا استفاده کنید اگر این کار کنید شبکه عصبی بسیار بزرگ میشود و خیلی محتمل است که دچار خطا شود و اطلاعات خروجی از آن به خوبی نباشد .

خب حال برای ادامه روند کار خود از فایل اسکریپی حاصل از خود nftool استفاده میکنیم که در جلسات قبلی هم آورده شده است .

خب در ادامه برای کار کردن بر روی این اطلاعات من چندین تا کد نوشته شده آماده در گیت هاب خودم میگذارم شما برای تحلیل به راحتی میتوانید از این کد برای سایر نوع داده نیز استفاده کنید .

DBZ_NeuralNetwork_Advance_Engine.m

برای رسم نمودار ها نیز یک فایل function نوشته شده است که در کنار این فایل باید باشد :

PlotResults.m

خب تمامی روال کاری دیگر در کد نویسی میتوانید با جزئیات بیشتر ببینند.

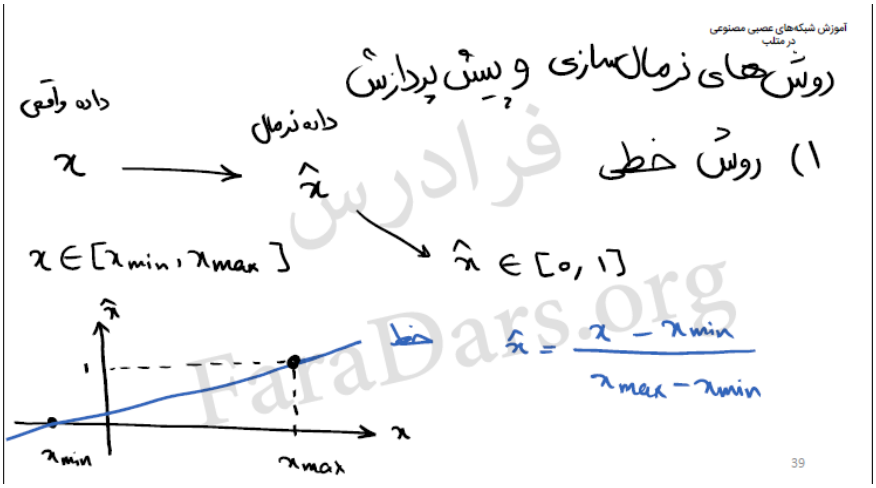
خب در حاضر برای یک مبحث میپردازیم تحت عنوان پیش پردازش که مثلا ما دوتا دیتای ورودی داریم اما از نظر اسکیل یا دامنه تغیراتی خیلی با یک دیگر تفاوت دارند مثلا برای بدنانسان فشار

خون بین بازه 8 تا 20 به عناون فشار کم تا خیلی زیاد است اما برای قند خون رنج تغیرات یک فرد بین 80 تا مثلا 120 به عنوان آستانه قند خون میگیریم میبینید که دامنه اعداد بشسیار متفاوت است که حال اگر این دو را بخواهیم به سیستم بدهیم باید با وزن های مناسب تنظیم شوند تا بتوانند دیتاها با یکدیگر رقابت کنند .

خب برای این که حال بتوانیم این اطلاعات را در کنار یکدیگر سازش کنند و استفاده کنیم راه حلی جز نرمال سازی نیست که داده ها را نرمال کنیم .

خب حال فرض بر این میگیریم یک منیمم داریم و ماکسیسمم خب برای این کار نرمال سازی کنیم یک بازه بین 0 تا 1 مثلا انتخاب میکنیم میتونیم این بازه را تغیر دهیم و رنج و دامنه ما فرق کند ، حال بر اساس این تغیرات نرمال سازی مثلا روش خطی کار میکنیم که در شکل زیر آورده شده است :

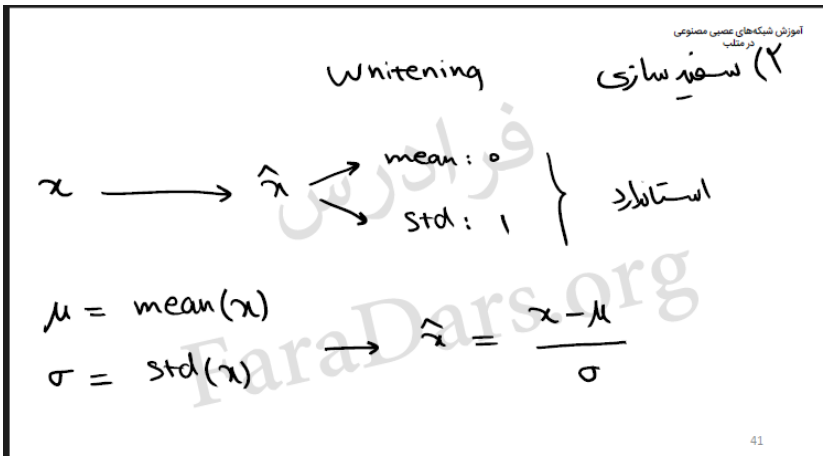
خب در این شکل به خوبی دیده می شود مه چگونه برای نرمال سازی کار میکنیمبه صورت خطی که دیتاهای ما چگونه باید این روند را پیش بگیرند.



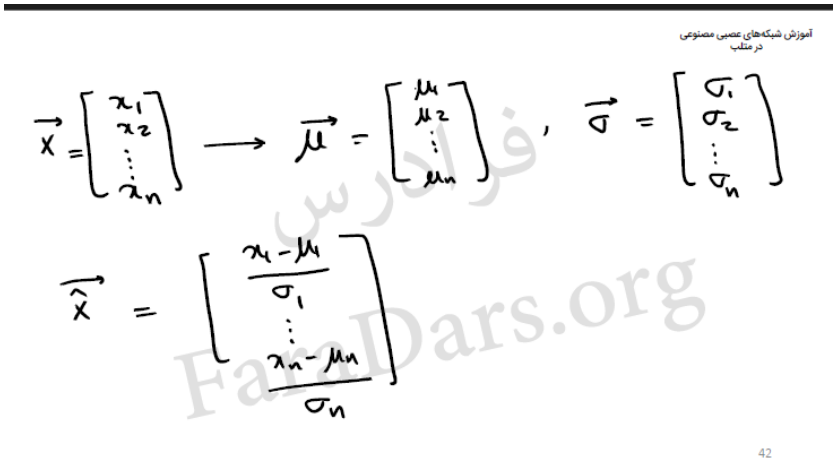
خب یکی از دیگر روش های که میتوانیم برای نرمال سازی اطلاعات خود استفاده کنیم روش سفید سازی است که اطلاعات نرمال دو بخش دارند یکی mean = 0 , std =1 که به این متد نرمال سازی سفید گوییم .

در شکل زیر میتوانید ببنید چگونه عمل میکنیم :

خب خلاصه سخن برای یک متغیر این روش سفید سازی به فرم زیر عمل میکنیم حال اگر چند ورودی داشتیم باید به صورت برداری عمل کنیم که برای این کار هم برای شما یک اسلاید تصویر بعد از این گویای همه چیز است .



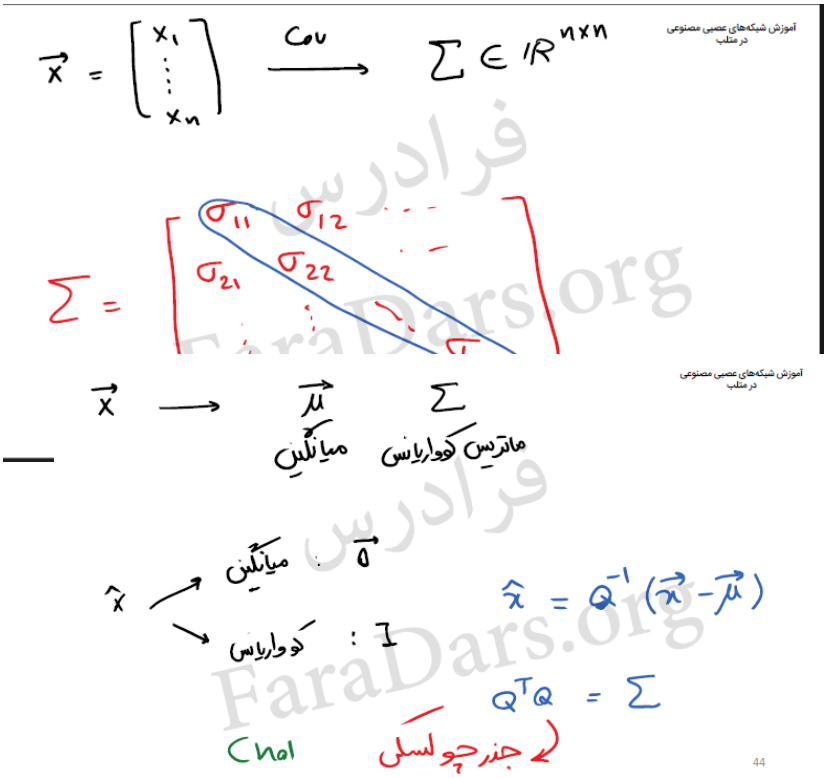
این فرم برداری بر ای چند ورودی کاربرد دارد.



خب برای این کار گاهی اوقات پیچیدگی ها زیاد میشود و تبدیل به یک ماتریس می‌شوند اطلاعات ما خب برای این کار ما باید یکسری تمهیداتی ببینیم که در اینجا برای این کار ما یک ماتریس میانگین اطلاعات داریم و یک ماتریس همانی و سپس برای نرمال سازی این روش به صورت زیر عمل میکنیم که در تصویر آمده است دستور متلب هم داریم برای نرمال سازی دیتا ها :

این همان ماتریسی است که برای چند ورودی حاصل شد .در ادامه برای نرمال کردن این اطلاعات یک سری کار های دیگه باید کرد که در تصویر میبینید :

اینم روند کاری برای نرمال سازی مجموعه ای از ورودی ها به سیستم خود



تحلیل مؤلفه‌های اصلی (PCA) و روش‌های دیگر کاهش ابعاد

مقدمه

روش **تحلیل مؤلفه‌های اصلی (PCA)** یکی از روش‌های پرکاربرد در پیش‌پردازش داده‌ها است که هدف آن کاهش ابعاد داده‌ها با حفظ بیشترین مقدار اطلاعات ممکن است. این کار با یافتن محورهایی انجام می‌شود که بیشترین واریانس داده‌ها را توضیح می‌دهند.

توضیح شهودی PCA

فرض کنید یک مجموعه داده از افراد سیگاری داریم. می‌توان گفت که بیشتر این افراد در طول زمان دچار **مشکلات تنفسی** می‌شوند. بنابراین، اگر بخواهیم این افراد را در یک فضای دو بعدی نمایش دهیم، دو ویژگی مهم می‌توانند **میزان مصرف سیگار** و **شدت مشکلات تنفسی** باشند. اما اگر بتوانیم یک **محور جدید** پیدا کنیم که ترکیبی از این دو ویژگی باشد و بیشترین تغییرات را توضیح

دهد، آنگاه می‌توانیم داده‌ها را روی این محور نمایش دهیم و به این ترتیب، تحلیل و پردازش آن‌ها را ساده‌تر کنیم.

این روش بسیار شبیه به **تجزیه مقدار منفرد (SVD)** در جبر خطی است، زیرا در هر دو روش به دنبال یافتن محورهای بهینه برای نمایش داده‌ها هستیم.

روش‌های دیگر کاهش ابعاد و پیش‌پردازش داده‌ها

علاوه برPCA ، روش‌های مختلفی برای کاهش ابعاد و پیش‌پردازش داده‌ها وجود دارد که برخی از آن‌ها خطی و برخی دیگر غیرخطی هستند:

۱ .روش‌های خطی

- **تحلیل تفکیک خطی :(LDA)** بر خلاف PCA که واریانس داده‌ها را در نظر می‌گیرد، LDA به دنبال یافتن محورهایی است که بیشترین جداسازی بین دسته‌های مختلف داده را ایجاد کنند. این روش معمولاً در مسائل طبقه‌بندی استفاده می‌شود.
- **تجزیه مقدار منفرد :(SVD)** مشابهPCA ، اما با تمرکز بر تجزیه یک ماتریس به فاکتورهای اساسی برای فشرده‌سازی و کاهش ابعاد.
- **تحلیل مؤلفه‌های مستقل :(ICA)** در این روش به جای یافتن محورهایی که بیشترین واریانس را توضیح دهند، محورهایی جستجو می‌شوند که **از نظر آماری مستقل‌ترین ویژگی‌ها** را ارائه دهند.

۲ .روش‌های غیرخطی

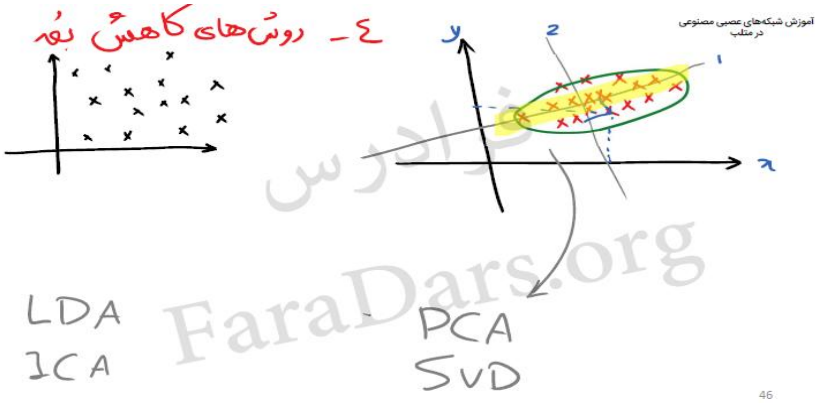
- **تحلیل مؤلفه‌های اصلی هسته‌ای :(Kernel PCA)** نسخه غیرخطی PCA که از نگاشت‌های کرنل برای یافتن ساختارهای غیرخطی در داده‌ها استفاده می‌کند.

- **نقشه‌های خودسازمان‌ده (SOM):** یک شبکه عصبی که داده‌ها را در ابعاد کمتر بازنمایی کرده و خوشه‌بندی انجام می‌دهد.
- **تحلیل چندبعدی مقیاس‌بندی شده (MDS):** از اطلاعات مربوط به فاصله بین نقاط استفاده می‌کند تا آن‌ها را در فضایی با ابعاد کمتر نمایش دهد.
- **تجزیه غیرمنفی ماتریس (NMF):** روشی که محدودیت غیرمنفی بودن را برای ماتریس‌های فاکتورگیری‌شده اعمال می‌کند و اغلب در پردازش تصویر و متن استفاده می‌شود.
- **شبکه‌های عصبی خودرمزگذار (Autoencoders):** از شبکه‌های عصبی برای یادگیری نمایش‌های فشرده از داده‌ها استفاده می‌کند.

انتخاب روش مناسب

انتخاب بهترین روش به **نوع داده‌ها، نیازهای مسئله و پیچیدگی محاسباتی** بستگی دارد. اگر داده‌ها دارای ساختار غیرخطی باشند، روش‌هایی مانند Kernel PCA یا Autoencoder عملکرد بهتری خواهند داشت، در حالی که در مسائل خطی، روش‌هایی مانند PCA و LDA کافی هستند.

شکل زیر برای توصیف بهتر آورده ام :



کلا این روش ها بهش کاهش بعد یا ساده سازی اطلاعات برای راحت تر شدن و عملکرد بهتر سیستم و شبکه عصبی ما است .

ویژگی‌های نگاشت غیرخطی و استخراج ویژگی‌ها

یکی از مهم‌ترین مزایای روش‌های نگاشت غیرخطی در کاهش ابعاد، استخراج ویژگی‌های پنهان و معنادار از داده‌ها است. در بسیاری از مسائل، داده‌ها در فضای اصلی خود دارای روابط پیچیده و غیرخطی هستند که با روش‌های خطی مانند PCA قابل تشخیص نیستند.

در این روش‌ها، داده‌ها به فضای ویژگی دیگری نگاشت (Mapping) می‌شوند که در آن، ساختارهای غیرخطی آشکار می‌شوند. این کار باعث می‌شود که ویژگی‌های جدیدی استخراج شوند که نمایش بهینه‌تری از داده‌ها ارائه می‌دهند .به عنوان مثال:

- Kernel PCAاز توابع کرنل برای نگاشت داده‌ها به فضایی با ابعاد بالاتر استفاده می‌کند تا مؤلفه‌های اصلی در آن فضا بهینه شوند.
- شبکه‌های عصبی خودرمزگذار (Autoencoders) با یادگیری نمایش فشرده‌ای از داده‌ها، ویژگی‌های جدیدی تولید می‌کنند که در فضای اصلی قابل مشاهده نبودند.
- نقشه‌های خودسازمان‌ده (SOM) داده‌ها را در یک شبکه عصبی مرتب می‌کنند و الگوهای پنهان را کشف می‌کنند.
-

مدیریت داده‌های بی‌مقدار (Missing Data) در شبکه‌های عصبی

در بسیاری از مواقع، هنگام جمع‌آوری داده‌ها ممکن است برخی از مقادیر مفقود یا ناقص باشند. این موضوع می‌تواند به دلایل مختلفی رخ دهد، مانند خطای حسگرها در یک آزمایش علمی، نقص تجهیزات، یا از دست رفتن بخشی از داده‌ها در حین پردازش .روش‌های مختلفی برای برخورد با این نوع داده‌ها وجود دارد.

رویکردهای مدیریت داده‌های بی‌مقدار

1. حذف داده‌های ناقص : اگر مقدار داده‌های از دست رفته کم باشد، می‌توان ردیف‌های دارای مقدار بی‌مقدار را حذف کرد.
2. جایگزینی با میانگین یا میانه : در برخی موارد می‌توان مقدار از دست رفته را با میانگین، میانه یا مد سایر داده‌های مشابه جایگزین کرد.
3. مدل‌سازی مقدار مفقود با استفاده از الگوریتم‌های یادگیری ماشین :روش‌هایی مانند رگرسیون، KNN، یا شبکه‌های عصبی می‌توانند برای پیش‌بینی مقادیر مفقود استفاده شوند.

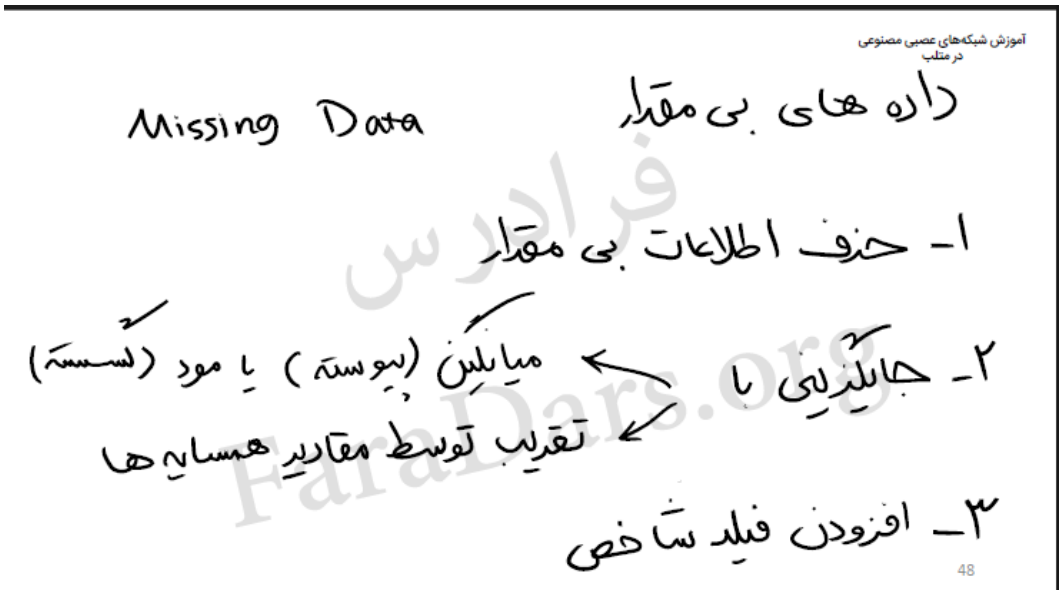
نقش شبکه‌های عصبی در مدیریت داده‌های ناقص

برخی از روش‌های یادگیری ماشین مانند MLP(شبکه پرسپترون چندلایه) و RBF تابع پایه‌ای شعاعی توانایی خوشه‌بندی داده‌ها را ندارند ، بلکه بیشتر بر یادگیری روابط بین داده‌ها متمرکز هستند. این در حالی است که روش‌هایی مانند SOM نقشه‌های خودسازمان می‌توانند داده‌ها را دسته‌بندی کرده و الگوهای پنهان را استخراج کنند که در مدیریت داده‌های ناقص مفید است. بنابراین، انتخاب مدل مناسب تأثیر زیادی در نحوه برخورد با داده‌های بی‌مقدار دارد.

در یک متد دیگر هم داریم ما اون دیتاهایی که پرت هستند یا اصلا وجودیت آن ها را نمیدانیم یک فیلد شاخص اضافه میکنیم و آن را محک میزنیم که بودن و یا نبودن اون دیتا چقدر اثر گذار بر روی کار ماست این هم یک متد دیگر برای برخورد با اطلاعات Missig Data است .

اینم شماتیکی از تمامی سخن هخایی که در رابطه با این بحث داشتیم

.



.1

مدیریت داده‌های پرت (Outliers) در شبکه‌های عصبی

داده‌های پرت مقادیری هستند که به دلیل خطاهای اندازه‌گیری، شرایط خاص آزمایش یا عوامل ناشناخته خارج از محدوده عادی سایر داده‌ها قرار می‌گیرند. در شبکه‌های عصبی، داده‌های پرت می‌توانند بر دقت و عملکرد مدل تأثیر منفی بگذارند. برای مدیریت این داده‌ها دو رویکرد اصلی وجود دارد:

1. حذف داده‌های پرت :در صورتی که داده‌های پرت تأثیر مخربی بر روی مدل داشته باشند، می‌توان آن‌ها را شناسایی و حذف کرد. روش‌هایی مانند فاصله‌یابی آماری، جعبه‌نگاری و الگوریتم‌های خوشه‌بندی برای تشخیص داده‌های پرت استفاده می‌شوند.
2. طراحی توابع عملکرد مقاوم :در این روش، به جای حذف داده‌های پرت، از توابعی مانند Huber Lossاستفاده می‌شود که اثر داده‌های پرت را کاهش می‌دهند. برای مثال، اگر داده‌های ما در یک محدوده دایره‌ای توزیع شده باشند اما چند مقدار دورتر از این محدوده

قرار بگیرند، تابع Huber به صورت مجازی این مقادیر را به مرز مجموعه داده نزدیک‌تر می‌کند تا از آن‌ها نیز در فرآیند یادگیری استفاده شود.

3. روش تطبیقی است (حذف در حین آموزش شبکه عصبی)در این روش شبکه عصبی در حین آموزش که میبیند اطلاعات را دارد پردازش میکند و همزمان با آن یک سری از اطلاعات ما پرت میشود که در همین هنگام شبکه عصبی آنم ها را حذف یا سالم میگذارد و تطبیقی عمل میکند که این روش از همه روش ها بهتر است. این روش سخت است ولی عملکرد خیلی خوبی داردو پیچیدگی پیاده سازی هم بیشتر است .

خاصیت مارکوف چیست؟

خاصیت مارکوف بیان می‌کند که **وضعیت فعلی یک سیستم فقط به وضعیت قبلی آن بستگی دارد و نیازی به دانستن تمام تاریخچه سیستم نیست** .به عبارت دیگر، اگر وضعیت فعلی یک سیستم را بدانیم، اطلاعات بیشتری از گذشته سیستم به پیش‌بینی آینده کمکی نمی‌کند.

مثال شهودی از خاصیت مارکوف

فرض کنید در حال رانندگی در یک مسیر هستید. اگر بدانید که در لحظه‌ی فعلی در چه سرعتی حرکت می‌کنید و در چه جهتی هستید، می‌توانید حدس بزنید که در ثانیه‌ی بعدی چه وضعیتی خواهید داشت. اطلاعات مربوط به مسیر طی‌شده در چند دقیقه‌ی قبل چندان تأثیری بر پیش‌بینی گام بعدی ندارد، بلکه **همین وضعیت فعلی تعیین‌کننده آینده است.**

مدل‌سازی سیستم‌های مارکوفی

در بسیاری از مسائل، از خاصیت مارکوف برای طراحی مدل‌هایی استفاده می‌شود که بتوانند رفتار یک سیستم را در آینده تخمین بزنند. این کار معمولاً با استفاده از فرایندهای مارکوفی و زنجیره‌های مارکوف انجام می‌شود.

یعنی احتمال رخ دادن یک وضعیت جدید فقط به وضعیت قبلی بستگی دارد و وابستگی به کل گذشته سیستم حذف می‌شود.

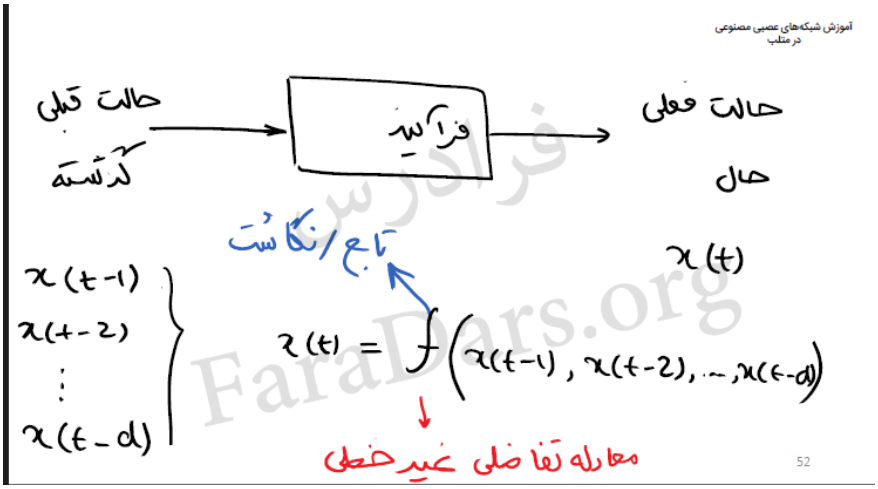
کاربردهای خاصیت مارکوف

- مدل‌سازی سیستم‌های دینامیکی :در بسیاری از سیستم‌های مهندسی، مانند مدل‌های پیش‌بینی آب‌وهوا، کنترل فرآیندهای صنعتی و رباتیک، خاصیت مارکوف به ما کمک می‌کند که بتوانیم با دانستن وضعیت فعلی، رفتار سیستم را در آینده تخمین بزنیم.
- پردازش زبان طبیعی :(NLP) در مدل‌های زبانی مانند مدل مارکوف پنهان(HMM) ، ترتیب کلمات در یک جمله را می‌توان بر اساس خاصیت مارکوف مدل‌سازی کرد.
- شبکه‌های عصبی بازگشتی :(RNN) در یادگیری عمیق، شبکه‌های بازگشتی مانند LSTMو GRUبرای مدل‌سازی دنباله‌های داده‌ای از خاصیت مارکوف بهره می‌برند.
- بازارهای مالی :در پیش‌بینی قیمت سهام و تحلیل روندهای اقتصادی، بسیاری از مدل‌ها بر اساس فرآیندهای مارکوفی ساخته می‌شوند.

نتیجه‌گیری

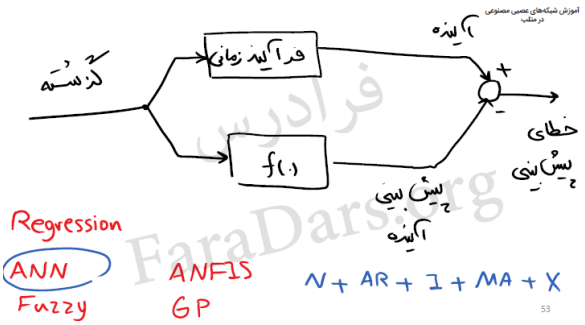
خاصیت مارکوف یکی از اصول اساسی در مدل‌سازی سیستم‌های پویا و پیش‌بینی رفتار آینده است. با استفاده از این خاصیت، می‌توان مدل‌هایی طراحی کرد که بدون نیاز به ثبت تاریخچه‌ی کامل سیستم، تنها با استفاده از وضعیت فعلی، آینده‌ی سیستم را تخمین بزنند.

اینم تصویری از کاری که میخواهیم انجام دهیم .

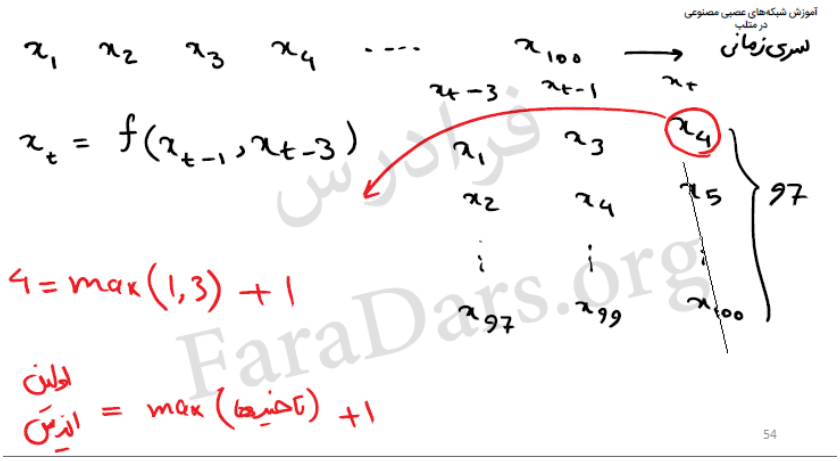


حال کار ما این است در کنار این مدل اصلی یک مدل خودمان با شبکه عصبی بسازیم و رفتار سیستم را پیش بینی کنیم با توجه با مدل اصلی سیستم .خب این کار در شکل زیر میتوان دید :

در تصویر هم مشاهده میکنید روش های بسیاری برای این کار وجود دارد که همگی این روش ها نوشته شده است مثلا ANFIS,Fuzzy,ANN که ما در این جا شبکه عصبی مصنوعی استفاده میکنیم .



خب در ادامه میخواهیم یک سری زمانی را تقریب بزنیم که دارای مثلا یک وردی است با تاخیر یک واحدی $x = f(x_{t-1}, x_{t-3})$ خب برای چنین تابعی یکی تاخیر یک داریم که از 2 شروع مشیه یکی هم تاخیر 3 زمانی داریم که از 4 شروع میشه و ما تقریب را میخواهیم تا 100 بزنیم این کار را ادامه در محیط متلب انجام میدهیم تصویر زیر هم برای روند کاری ماست :



برای اینکار یک کد متلب آورده ام و تابعی هم که میخواهیم کار کنیمک Mackey glass است که دارای دینامیک آشوبناک میباشد. تمامی روال کاری مثل فایل های قبلی برای شبکه های عصبی است .

فقط دوستان یک نکته بگوییم که تابع mgdata.dat تا نسخه 2016 که من نصب دارم جواب میدهد حال اگر نسخه بالاتری دارید نتیجه شاید ندهد و این مشکل من در نسخه 2023 دیدم که این فایل دیگر انگاری در متلب وجود ندارد . نکته ایکه باید بگوییم هم برای این فایل و قبلی یک function تعریف شده برای رسم پلات ها که د ر کنار کد باید باشد تا بتوانیم خروجی بهتری را برای کار خودتان ببینید.

DBZ_NeuralNetwork_Advance_mgdata.m این اسم فایل است و تابعی که برای رسمم نمودار ها استفاده کرده ام نیز :

PlotResults.m میباشد.