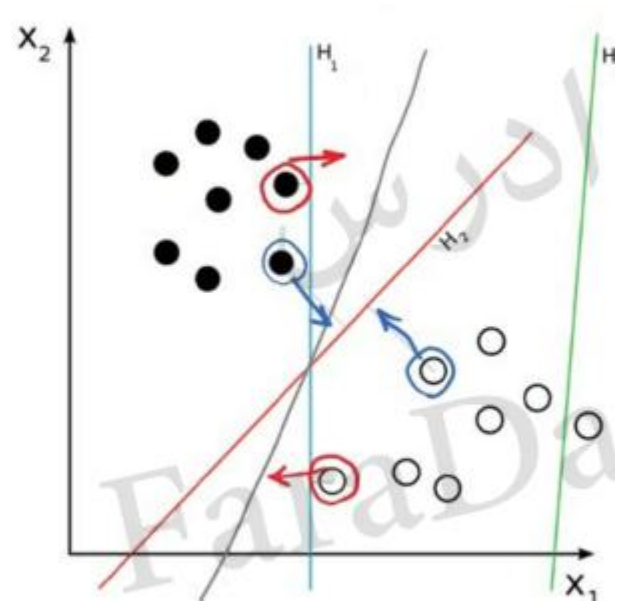
جلسه اول شبکه های عصبی بردار پشتیبان (Support vector Mechine or SVM)

در این نوع شبکه عصبی که در گذشته ما با ANFIS,MLP,RBF کار کردیم در نهایت هدف نهایی ما این بود که مقدار خطای خروجی را که با سیستم داریم کمینه کنیم .

اما در این نوع شبکه عصبی دیگر هدف این نیست ما در اینجا هدف کاهش ریسک عملیاتی است ، در یک شکل من برای شما یک سری توضیحات را میدهم :

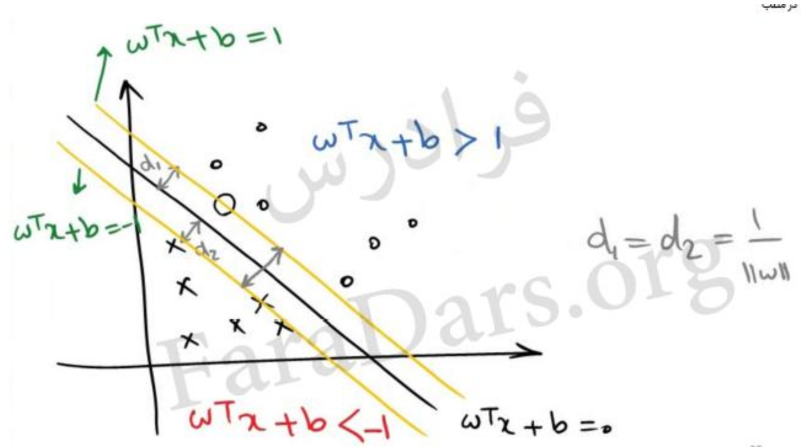


خب در این تصویر سه خط برای تفکیک د اده های ما وجود دارد میبینید که خط سبز رنگ اصلا برای کار عملیاتی ما اصلا مناسب نیست ، در نتیجه ما بین خط آبی و قرمز یکی حداقل ریسک عملیاتی را برای کار های ما ایجاد میکند ، خب در اینجا فرض بر این می‌شود خط آبی نزدیک به یک سری دیتا مشکی رنگ است این دیتا ها دوست دارند خط عملیاتی که داریم را به جهت مخالف خود پیش برانند و از طرف دیگر هم دایره های تو خالی که نزدیک به خط آبی هستند دوست دارند، همین روند را برای این اطلاعات شود، یعنی ناحیه خود را مانند توسعه طلبی گسترش دهند . حال این ممکن میشود که خط آبی از محل خود انحراف پیدا کند و به خوبی عمل نکند ،اما برای ما بهترین خط عملیاتی که حداقل ریسک را ایجاد میکند خط قرمز رنگ است .

خب شاید تا الان مفهوم کمی گنگ باشد حال برای ایجاد فهم بهتر یک شکل دیگر میاوریم :

خب در شکل زیر همان طور مشاهده میکنید دوتا خط به صورت موازی با رنگ های زرد رسم شده است و دارای دو معادله خط هستند کاملا یکسان اما با مقدار -1و+1 است و یک خط هم به رنگ

مشکی ما بین این دو خط حضور دارد حال دیتا های ما دیگر با خط مشکی از هم جداسازی نمیشود بلکه با دو خط زرد رنگی که ایجاد شده یک منطقه حائل درست کردیم که ناحیه ها کاملا از هم جدا و قابل تفکیک باشند ،تا بتوانیم حداقل ریسک عملیاتی را ایجاد کنیم .



خب برتری که اینجا داریم نسبت به روش های قبلی این است که به جای این که دیتا ها به صورت مرزی و کاملا مویی شکل از هم جداسازی شوند ناحیه ای را ایجاد میکنیم که امنیت بهتری را برای کار کردن با دیتا ها ایجاد میکنن د و خطای کار بشدت کاهش میابد. حال در یک اسلاید معادله خط جدید برای این دو خط آورده ام :

$$\{x_i, y_i\} \qquad i = 1, 2, \dots, n$$

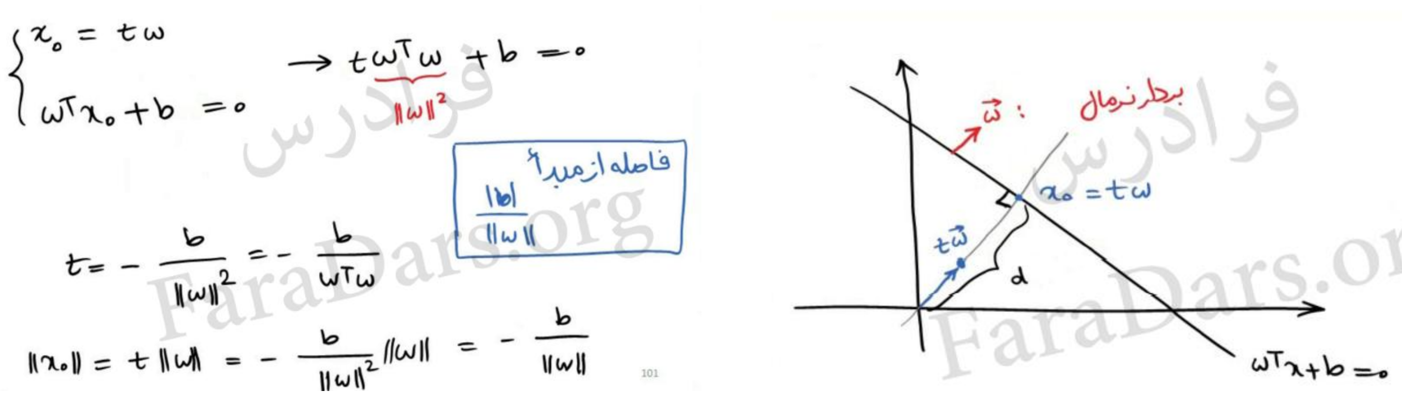
$$x_i \in \mathbb{R}^d \qquad y_i \in \{+1, -1\}$$

$$\text{if } y_i = 1 \Rightarrow \omega^T x_i + b > 1$$

$$\text{if } y_i = -1 \Rightarrow \omega^T x_i + b < -1$$

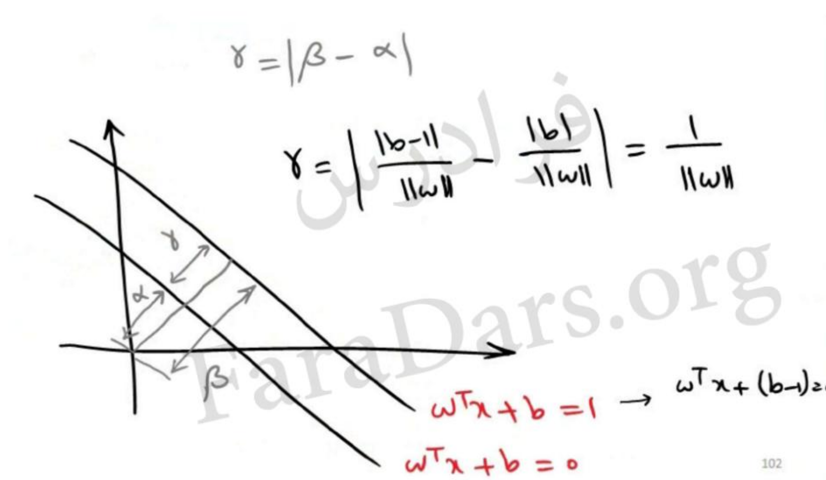
این هم دو معادله جدید برای کار کردن با دیتا ها.

حال باید به نحوی فاصله خط را محاسبه کنیم که در سیستم SVM کار کلا با وکتور ها پیش میرود ،پس برای این کار باید یک سیر عملیات های هند سی یا جبری هندسی انجام دهیم که به اسلاید زیر شما را دعوت میکنم :



خب در این دو تصویر کاری که برای خط مشکی رنگ میخواهیم انجام دهیم را میبینید.

خب حال برای این که بخواهیم به دو معادله خط زرد رنگ بالایی و پایین برسیم باید فاصله ها رو از یکدیگر کم کنیم در نهایت به یک معادله کلی برای این ها میرسیم که : $\frac{1}{\|W\|}$ است برای اثبات به شکل زیر توجه کنید :



این اون چیزی هست که ما انجام میدهیم برای کار های خودمان .

خب در نهایت هدف ما این است که میخواهیم این نرم برداری را بهینه کنیم که برای بهینه کردن آن باید مینیمم آ« را در نظر گرفت که به صورت زیر است :

کاری که میخواهیم انجام دهیم این میباشد که به صورت برداری آن عبارت قرمز رنگ را اجرایی کنیم در سیستم خود تا یک شبکه عصبی جدید بر حسب SVM تولید کنیم. حال برای بهینه سازی این دو قید داریم یکی خط بالایی زرد رنگ بیشتر 1 باشد و خط زرد رنگ پایینی کم تر از -1 باشد این دو شرط

$$\max \quad d_1 + d_2 = \frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$$

$$\min \quad \|w\|$$

$$\min \quad \frac{1}{2}\|w\|^2 = \frac{1}{2} w^T w$$

$$\boxed{\min \quad \frac{1}{2} w^T w}$$

یا قید داریم که به فرم ریاضی مینویسم :

$$\min \frac{1}{2}\, w^T w$$

$$st$$

$$if \;\; yi = 1 \;\rightarrow\; w^T x_i + b \; > 1 \;\; \forall\, i$$

$$if \;\; yi = -1 \;\rightarrow\; w^T x_i + b \; < -1 \;\; \forall\, i$$

فرم کاری ما برای مسئله بهینه سازی و کاری که میخواهیم کنیم این روند را داریم .

بعد این کار میتوانیم که مسئله بهینه سازی دیگر را از دل همین بیرون آورد که بهش میگن مسئله Primal گفته میشود که در شکل زیر است :

بهینه سازی این مسئله کاره ساده ای اصلا نیست که برای این کار ما باید مشقت زیادی بکشیم . خب راه حل چیست برای این کار خب ما میاییم این تابع جدید را از مقدار مینیمم سازی خود یعنی :

$$\min \quad \frac{1}{2} w^T w$$

$$s.t. \quad y_i (w^T x_i + b) - 1 \geq 0 \quad \forall i$$

Primal Problem

$$\min \frac{1}{2}\, w^T w$$

کم میکنیم و به صورت یا فرم زیر میرسیم :

خب میبینید که یک سری چیزا باید مینیمم شود و یک سری مقادیر هم ماکسیمم شود در این حالت نقطه زینی داریم .

$$L_P = \frac{1}{2}\omega^T\omega - \sum_i \alpha_i \left[y_i(\omega^T x_i + b) - 1\right]$$

$$\alpha_i \geqslant 0$$

$$\min_{\omega, b} \longleftarrow L_P \longrightarrow \max_{\alpha}$$

Saddle Point نقطه زینی

خب برای حل کردن این موضوع باید عملیات مشتق گیری را انجام دهیم که برای این کوضع یک بار مشتق نسبت به w ,b باید بگیریم که به شکل زیر میرسیم :

اینم فرم مشتق گرفته شده و کار هایی که انجام داده ایم میتوانید ببینید که مشتق گیری نسبت به w,b چه چیزی را ایجاد میکند.

$$L_P = \frac{1}{2}\omega^T\omega - \sum_i \alpha_i \left[y_i(\omega^T x_i + b) - 1\right]$$

$$\frac{\partial L_P}{\partial \omega} = 0 \longrightarrow \omega - \sum_i \alpha_i y_i x_i = 0$$

$$\omega = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial L_P}{\partial b} = 0 \longrightarrow \sum_i \alpha_i y_i = 0 \longrightarrow \text{قید بایاس}$$

حال عبارت اصلی را که داریم با عبارت هایی که مشتق گرفته ایم جایگذاری میکنیم تا در نهایت :

که در نهایت میخواهیم این فرم معادله را ماکسیسمم کنیم .

$$L_D = \frac{1}{2}\omega^T\omega - \sum_i \alpha_i \left[y_i(\omega^T x_i + b) - 1\right]$$

$$= \frac{1}{2}\left(\sum_j \alpha_j y_j x_j\right)^T \left(\sum_i \alpha_i y_i x_i\right)$$

$$- \sum_i \alpha_i \left[y_i\left(\sum_j \alpha_j y_j x_j^T x_i + b\right) - 1\right]$$

$$= -\frac{1}{2}\sum_i\sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i$$

در نهایت برای ساده سازی این عبارت های داخل سیگما تبدیل به h میشود که در شکل زیر میتوانید ببینید:

$$\min \quad \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j \underbrace{y_iy_j\, x_i^T x_j}_{h_{ij}} - \sum_i \alpha_i$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \quad \forall i$$

این همان ساده سازی که بود گفتیم در ادامه باید آن قید هم که در زیر آورده ایم برای مسئله پیاده سازی شود :

$$\frac{1}{2}\sum_i\sum_j h_{ij}\,\alpha_i\alpha_j = \frac{1}{2}\alpha^T H \alpha$$

$$\alpha = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{bmatrix} \qquad H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n}\\ h_{21} & h_{22} & \cdots & h_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ h_{n1} & h_{n2} & \cdots & h_{nn}\end{bmatrix} = [h_{ij}]$$

$$H \in \mathbb{R}^{n\times n}$$

113

Dual Problem quadprog

$$\min \quad \frac{1}{2}\alpha^T H \alpha + f^T \alpha$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

Quadratic Programming

برنامه‌ریزی درجه ۲

در نهایت یک برنامه درجه دوم برای خودمان ساخته ایم که کار راحت تری داریم .

در نهایت مسئله خودمان را بخواهیم حل کنیم به بردار حالت پشتیبان میرسیم که شماتیک زیر برای این مسئله است :

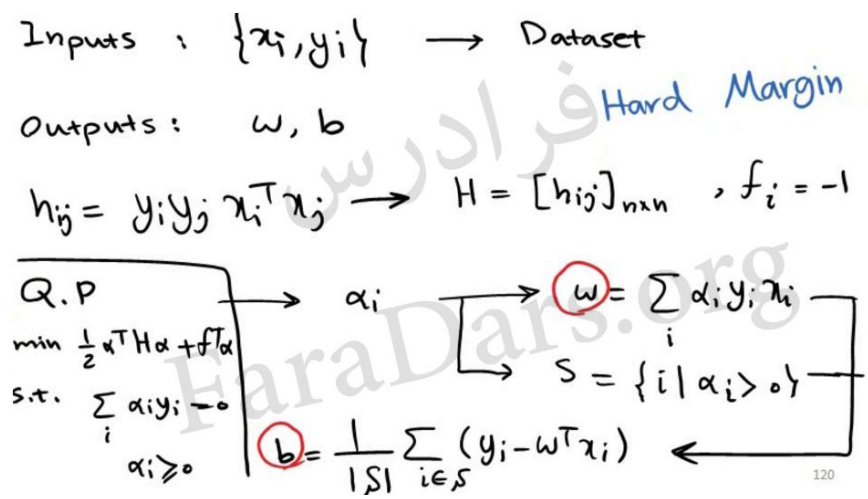
$$\alpha_i > 0 \longrightarrow y_i(w^T x_i + b) - 1 = 0$$

$$y_i(w^T x_i + b) = 1$$

$$\underbrace{\cancel{y_i^2}}_{=1}(w^T x_i + b) = y_i \Rightarrow w^T x_i + b = y_i$$

در ادامه کار میخواهیم روال کلی کار را برای شما به نمایش بگذاریم :

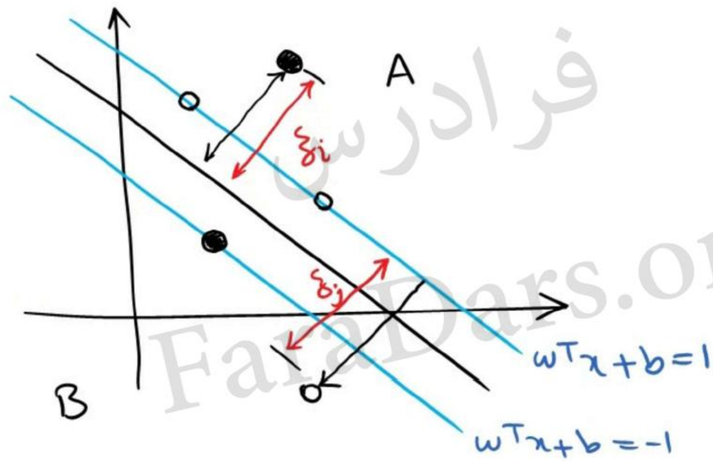
در تصویر زیر میبینید که دارای دوتا دیتای ورودی و خروجی هستیم ، ما کلیت کارمان دنبال w,b است که بعد از بدست آوردن ماتریس h باید مسئله درجه دوم را بهینه کنیم که quadratic programming نام د ارد بعد از بهینه سازی کردن این مسئله که در بالا هم گفته شد خروجی یک سری آلفا و S داریم که خروجی کارمان باید بتوانیم w,b را پیدا کنیم که کار اصلی ما است نقشه راه را در این تصویر میبینید.



خب گاهی اوقات ما با این دوتا مسئله یا خطی که داریم یک سری اطلاعات از مرز های ما خارج

شده و رفتن به آن سوی مرز دیگر که خط ما باید و حال باید چه کنیم ،در تصویر زیر برای شما یان حالت را به نمایش گذاشته ایم :

خب میبینید که برای میزان خارج شدن از خطوط یک پارامتری به عنوان ζ در نظر گرفته ایم حال این مقدار را باید به معادله خطوط خود اضافه کنیم تا بتوانیم میزان تخطی کردن اطلاعات از این مرز ها چقدر است که در اسلاید بعدی میتوان این موضع را به خوبی دید:



این همان توضیحاتی است که دادیم .

$$\text{if } y_i = +1 \Rightarrow w^T x_i + b + \xi_i \geq 1$$

$$w^T x_i + b \geq 1 - \xi_i \qquad \xi_i \geq 0$$

$$\text{if } y_i = -1 \Rightarrow w^T x_i + b - \xi_i \leq -1$$

$$w^T x_i + b \leq -1 + \xi_i$$

$$\xi_i \geq 0$$

خب در ادامه این پارامتری که ما در اینجا آورده ایم باید در شرایط بهینه سازی هم آورده شود که در مسئله بهینه سازی این پارامتر ما به عنوان یک جریمه به تابع هزینه باید اضافه شود که در تصویر زیر مشاهده میکنید :

$$\min \ \frac{1}{2} w^T w + \underbrace{C}_{\text{ضریب جریمه}} \sum_i \xi_i$$

s.t.

$$0 \leq \alpha_i \leftarrow y_i (w^T x_i + b) \geq 1 - \xi_i \qquad \forall i$$

$$0 \leq \mu_i \leftarrow \xi_i \geq 0$$

Primal Problem

حال در ادامه کار تمامی روال کاری که برای بهینه سازی مسئله خود داشتیم اینجا هم باید پیاده سازی کنیم از مشتق گیری نسبت به w, b, ζ و سپس تشکیل نقطه زینی که یک سری چیزا ماکسیمم است و یک سری چیزا مینیمم میشود که به صورت زیر است :

این اون روال تشکیل دادن نقطه زینی برای اکسترمم های خود سپس مشتق گیری های خود از عبارت تابع هزینه که

$$\underset{w,\, b,\, \xi_i}{\min} \longleftarrow L_p \longrightarrow \underset{\alpha_i,\, \mu_i}{\max}$$

به فرم زیر هستند :

اینم مشتق گیری های ما سپس بعد از این کار باید در عبارت اصلی یعنی تابع هزینه مشتق گیری های خودمان را بگذاریم سپس مسئله درجه دوم را ایجاد کنیم البته با یک تفاوت عمده که اینجا پارامتر زتا هم داریم .

$$\frac{\partial L_p}{\partial w} = 0 \longrightarrow w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial L_p}{\partial b} = 0 \longrightarrow \sum_i \alpha_i y_i = 0$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \longrightarrow C - \alpha_i - \mu_i = 0$$

$$\alpha_i + \mu_i = C$$

خب تفاوتی که ایجاد میشود حاشیه ما نرم و سخت میشود که بر اساس آلفا تغیر کرده است :

$$\min \quad \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \underbrace{x_i^T x_j}_{h_{ij}} - \sum_i \alpha_i$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0$$

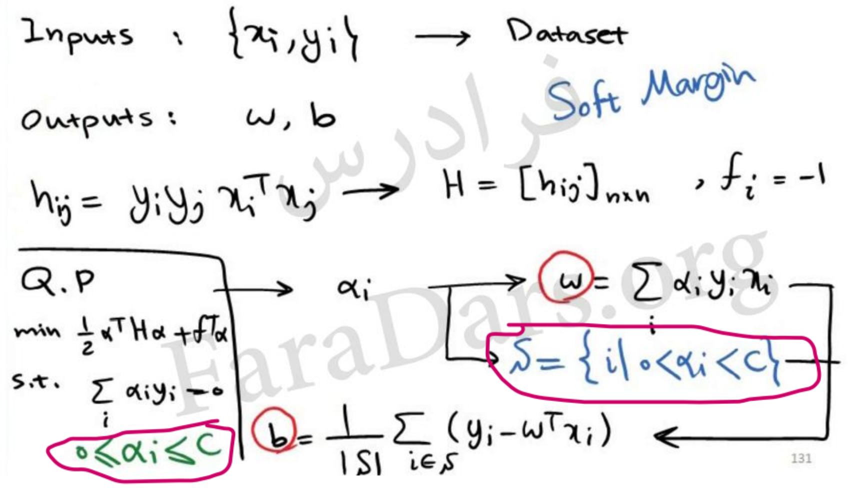
$$0 \leq \alpha_i \underbrace{\leq C}_{\text{تفاوت حاشیه نرم و سخت}} \quad \forall i$$

$$\left.\begin{aligned} \alpha_i &\geq 0 \\ \mu_i &\geq 0 \\ \alpha_i + \mu_i &= C \end{aligned}\right\} \longrightarrow \begin{aligned} &\boxed{0 \leq \alpha_i \leq C} \\ &0 \leq \mu_i \leq C \end{aligned}$$

Box Constraint

این دو تصویر کلیت کارمان را گفته است .

در حالت کلی روال مسئله حاشیه نرم به شکل زیر است :



خب تنها تفاوتی که ایجاد شده در دو حالت سخت و نرم این قسمت هایی است که بنده با خط قرمز دورشان خط کشیده ام ،که سختی کار ما همین جاست دیگه که باید فاکتور C که وجود دارد تنظیم کنیم که این خودش داستان های متفاوتی را برای شبکه عصبی ایجاد میکند یعنی قائده ای نیست که بتونیم این را کنترل کنیم و میبایست با الگوریتم های تکاملی مثل ژنتیک یا PSO و... این کار را انجام داد.

دیگر از مقدمات تئوری تمام شده و مابقی کار به صورت کد نویسی در گیت هاب میگذارم .

در قسمت اول یک فایل گذاشته ام که دارای اطلاعات x,y ما است که برای کار نیازمند به این دیتا ها هستیم :

```
%% Load Data

load mydata;

n=numel(y);

ClassA=find(y==1);
ClassB=find(y==-1);
```

این دو کلاسی که تعریف کردیم همان خطوطی است که برای +1و-1 خودمان مرز تعین کردیم در این الگوریتم شبکه عصبی .

سپس باید به تولید ماتریس H بریم که n*n بود برای این کا دوتا حلقه for در هم داریم که برای عملیات تولید ماتریس به شکل زیر عمل میکنیم و همان فرمول خودمان که بر ای

تولید ماتریس H بود را میزنیم که به شکل زیر است فقط باید اندیس ها و سطر ها و ستون ها را دقت کنیم :

این همان روال تولید ماتریس H است و C همان پارامتر سخت و نرمی حاشیه ما است .

```
%% Design SVM

C=10;

H=zeros(n,n);
for i=1:n
    for j=i:n
        H(i,j)=y(i)*y(j)*x(:,i)'*x(:,j);
        H(j,i)=H(i,j);
    end
end
```

بعد از آن ماتریس f را باید تولید کنیم که به فرم زیر تعریف میشود که یک ماتریس n*1 بود یک سطر و تمای ستون ها منفی 1 بود:

```
f=-ones(n,1);
```

در ادمه راه باید به حل مسئله درجه دوم بپردازیم برای این کار شما میتوانید برای اطلاعات بیشتر به داکیومنت متلب مراجعه منید اما ما برای این کار روال زیر را در نظر گرفته ایم که سه الگوریتم بهینه سازی را برای کارمان استفاده نموده ایم :

اینم سه الگوریتمی که برای محاسبه کار خودمان در نظر گرفته ایم .

```
Aeq=y;
beq=0;

lb=zeros(n,1);
ub=C*ones(n,1);

Alg{1}='trust-region-reflective';
Alg{2}='interior-point-convex';
Alg{3}='active-set';
```

تا اینجای کار S را تولید کرده ایم و باید سراغ w هم برویم :

```
options=optimset('Algorithm',Alg{2},...
    'Display','off',...
    'MaxIter',20);

alpha=quadprog(H,f,[],[],Aeq,beq,lb,ub,[],options)';

AlmostZero=(abs(alpha)<max(abs(alpha))/1e5);

alpha(AlmostZero)=0;

S=find(alpha>0 & alpha<C);
```

این ادامه کار ما است بعد الگوریتم انتخاب کردن و یک سری چیزاهای اضافه در این آرگومان تابع کاری ما وجود دارد که شما میتوانید بر حسب نیاز از آنها استفاده کنید .

این تصویر زیرا هم برای تولید w,b امنجام میدهیم .

```
w=0;
for i=S
    w=w+alpha(i)*y(i)*x(:,i);
end

b=mean(y(S)-w'*x(:,S));
```

ادامه مسئله دیگر توابع رسم گرافیکی خروجی ماست که شما میتوانمید در فایل اصلی کد من ببینید.