بنام خدا

گزارش تمرین کامپیوتری اول

هوش محاسباتی

آرشام لؤلؤهری

۹۹۱۰۲۱۵۶

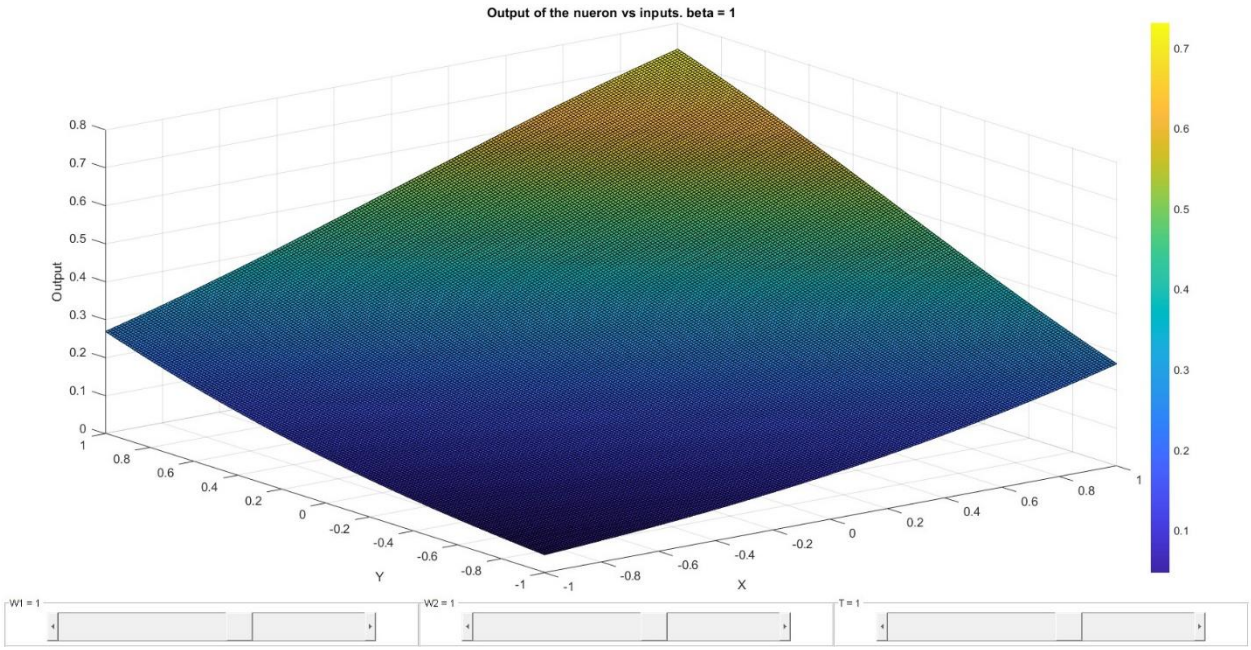
سوال ۱:

الف و ب)در section اول از کد مربوط به این سوال، پاسخ بخش های الف و ب قراردارد.

در این section، ابتدا بردارهای -1<X,Y<1 را برای نمایش در صفحه سه بعدی آماده کرده ایم. سپس پارامتر beta و مقادیر اولیه برای نمایش ابتدایی خروجی، برای w1,w2,T تنظیم شده اند (همگی مطابق با بخش ب، برابر ۱ قرار داده شده اند). سپس تابع Q1 با این ورودی ها صدا زده میشود تا نمودار interactive را اجرا کند.

داخل تابع Q1 از تابع activation (برای محاسبه خروجی شبکه)، تابع createPanels (برای ساختن پنل های slider برای تنظیم پارامترهای w1,w2,T)، و تابع selection برای آپدیت نمودار پس از آپدیت پارامترها، استفاده شده است. توضیحات هر تابع در غالب کامنت در کد آمده است.

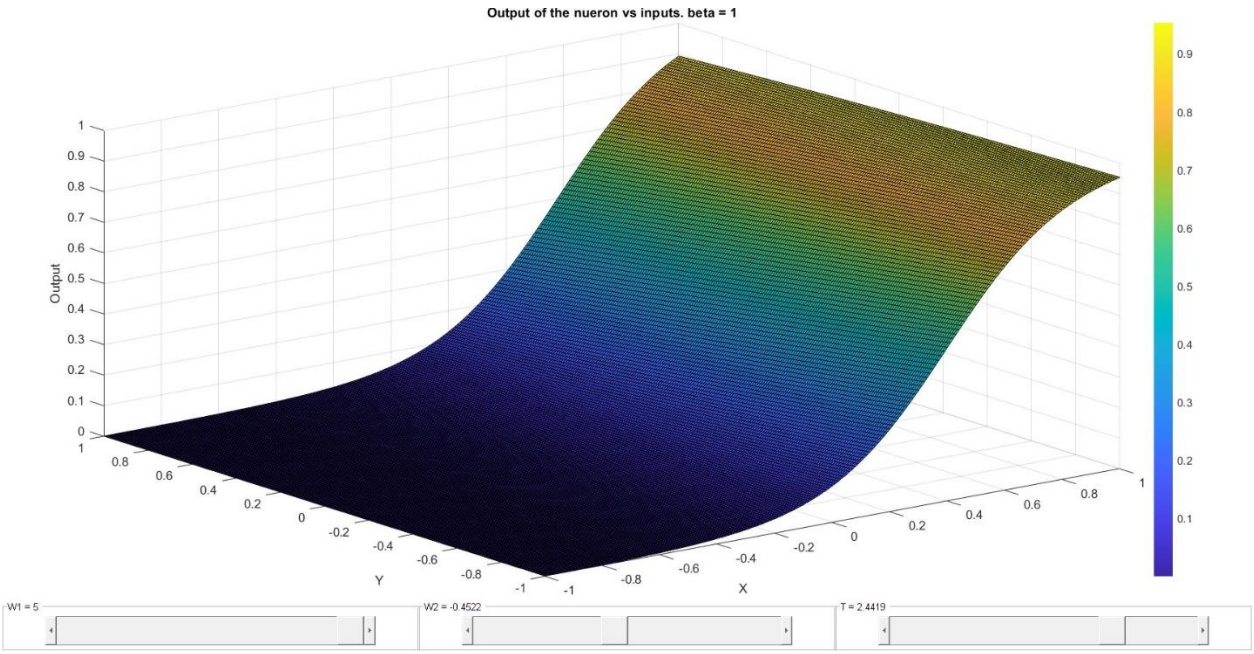
در نهایت به ازای مقدار ۱ برای تمام پارامتر ها (مطابق بخش ب سوال)، به چنین نموداری میرسیم:



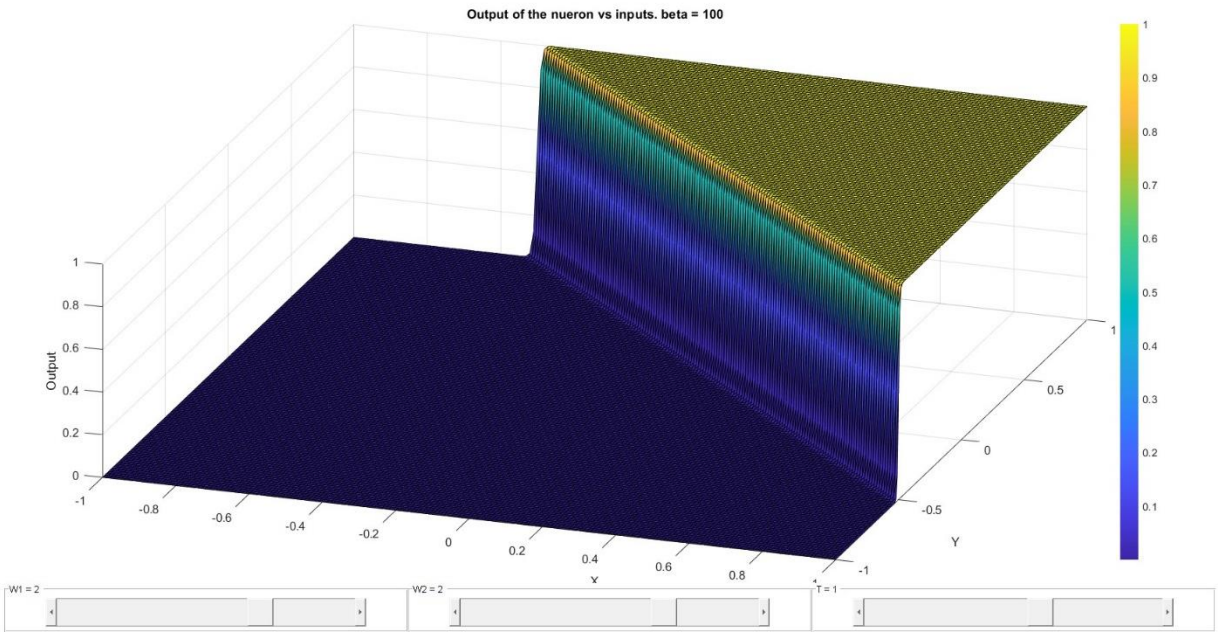
مقدار beta در عنوان نمودار آمده است. در زیر، سه پارامتر شبکه را میتوان از ۵- تا ۵+ تغییر داد و نمودار جدید را مشاهده کرد. مقدار کنونی هر پارامتر نیز در بالای هر slider نوشته میشود.

نمودار بالا به ازای w1=w2=T=beta=1 است. طبق رابطه f_{act}، هرچه $net = w1.x + w2.y$ بزرگتر باشد، خروجی به سمت ۱، و هرچه کوچکتر باشد، خروجی به سمت صفر میل خواهد کرد (هرچند هرگز به این دو مقدار نخواهیم رسید). به همین دلیل در شکل بالا که w1,w2>0، به ازای X=Y=1، بزرگترین net و در نتیجه بزرگترین خروجی را داریم (نزدیک به ۱)، و برعکس همین روند برای X=Y=0. حالتی که X=1, Y=0 یا برعکس نیز حالتی بین این دو است که خروجی ۰.۵ خواهد شد (net = T).

در شکل زیر نیز نمونه دیگری را میبینیم که پارامترها با استفاده از slider ها عوض شده اند:



ج) کد این قسمت در section دوم قرار دارد. برای پیاده سازی OR، صرفا باید قرار دهیم w1=w2=2 و T=1 (در این حالت برای یک TLU، اگر هردو ورودی صفر باشند، خروجی صفر بوده و اگر حداقل یکی از آنها ۱ باشد، net>T شده و خروجی ۱ میشود). حال قرار میدهیم beta = 100. نمودار حاصله به صورت زیر خواهد بود:



وقتی beta خیلی بزرگ است، ترم نمایی در f_{act} ، به شدت به علامت net-T وابسته میشود. بطوریکه اگر net>T، این ترم به سمت صفر رفته و خروجی شبکه ۱ میشود. اگر net<T، این ترم به سمت بینهایت رفته و خروجی صفر میشود:

$$if\ 2x + 2y - 1 > 0: \qquad z = 1$$

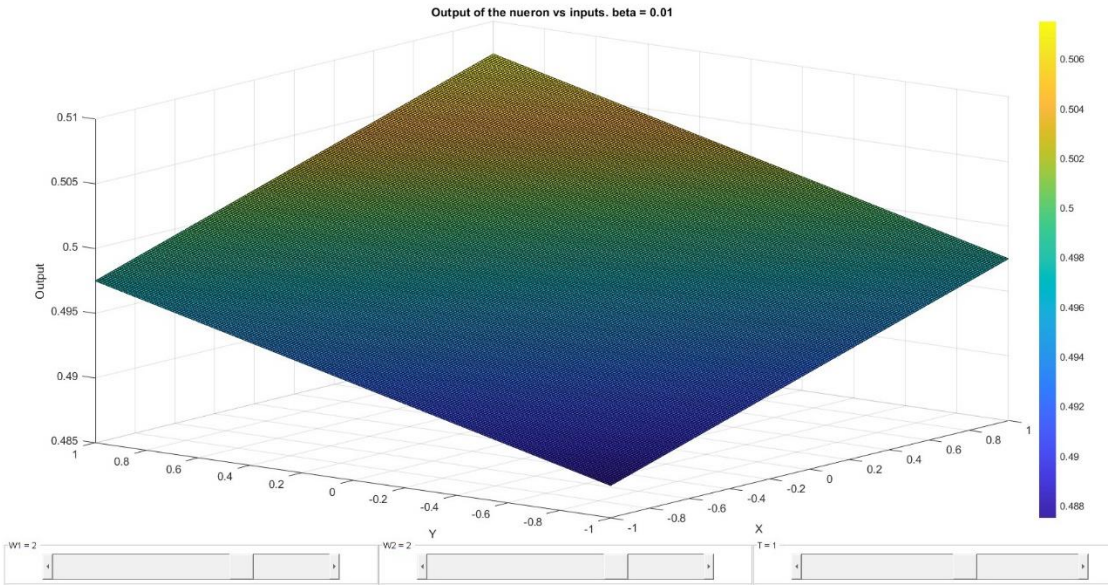
$$if\ 2x + 2y - 1 < 0: \qquad z = 0$$

خطی که مرز بین خروجی صفرو یک در شکل بالاست، همان $2x + 2y - 1 = 0$ است. بطور دقیق، خروجی روی خود این خط، برابر ۰.۵ خواهد بود.

بنابراین این شبکه، رفتاری مشابه یک TLU دارد که تابع فعالسازی آن پله است و وزن های 2,2 و آستانه ی ۱ دارد.

د) کد این قسمت در section سوم قرار دارد. حال قرار میدهیم beta = 0.01.

نمودار حاصله به صورت زیر خواهد بود:

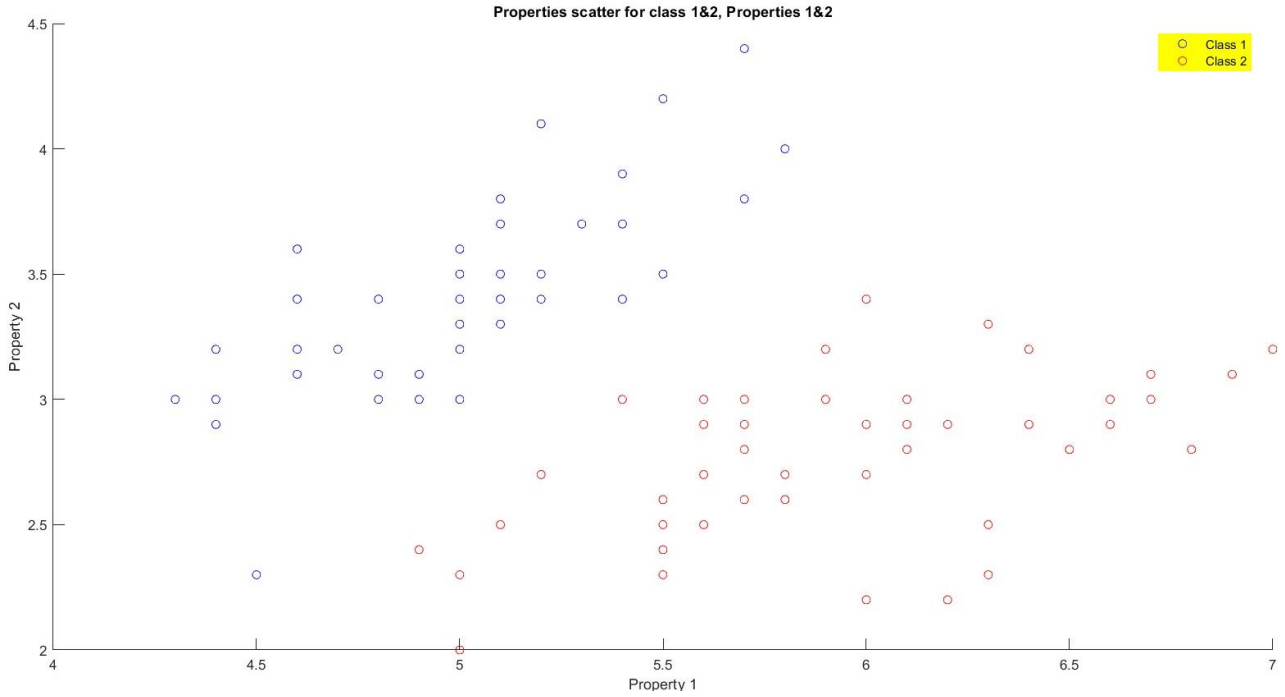


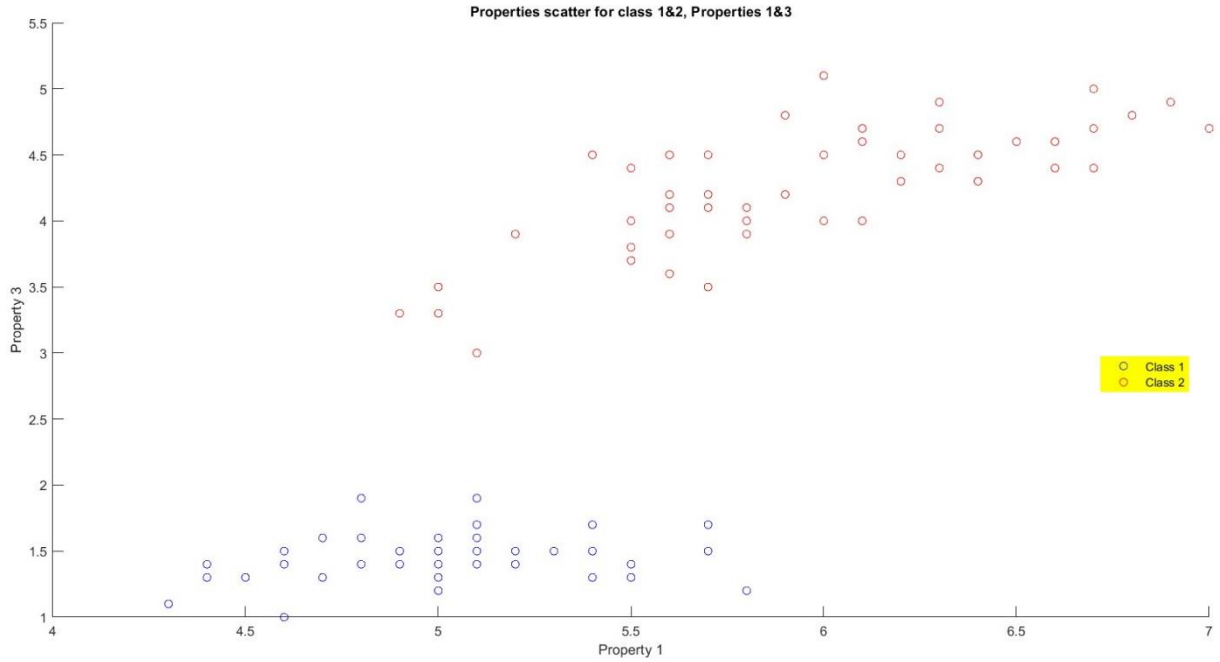
وقتی beta خیلی کوچک است، ترم نمایی در f_{act} ، صرف نظر از ورودی ها و وزن ها، مقداری بسیار نزدیک به ۱ دارد. بنابراین خروجی همواره در حدود ۰.۵ باقی میماند (به اسکیل محور z توجه شود) اما علامت و بزرگی ورودی ها و وزن ها، اندکی جهت رویه ی حاصله را تغییر میدهند (به همان ترتیبی که در بخش الف ذکر شد).

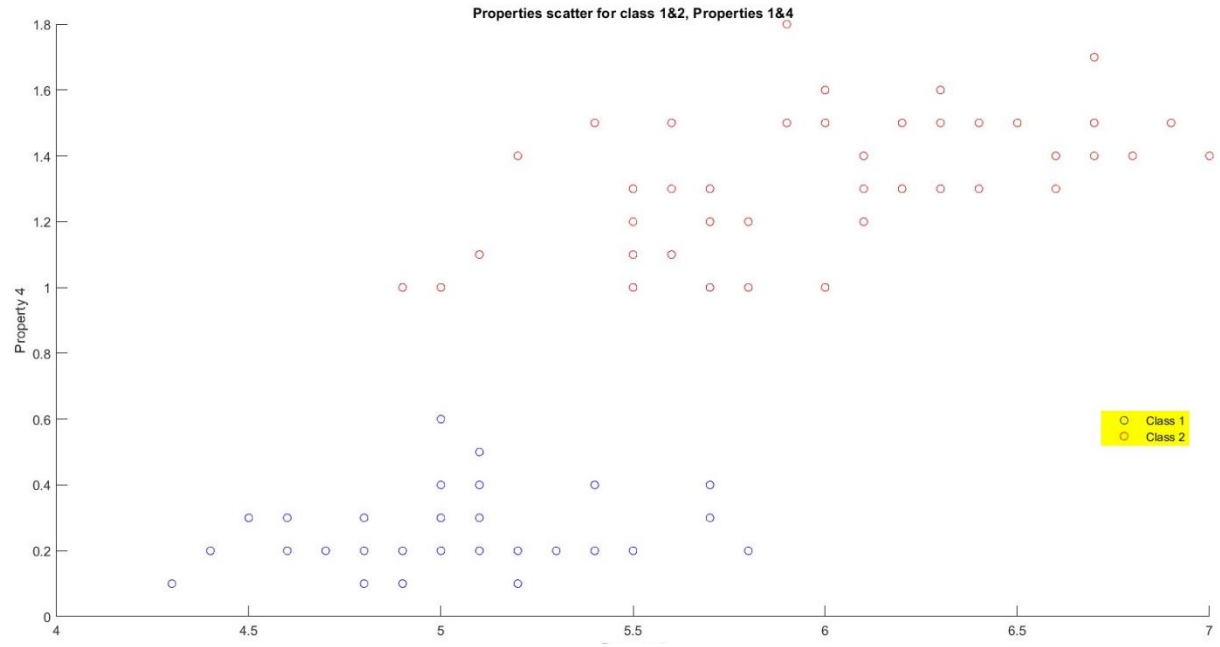
سوال ۲:

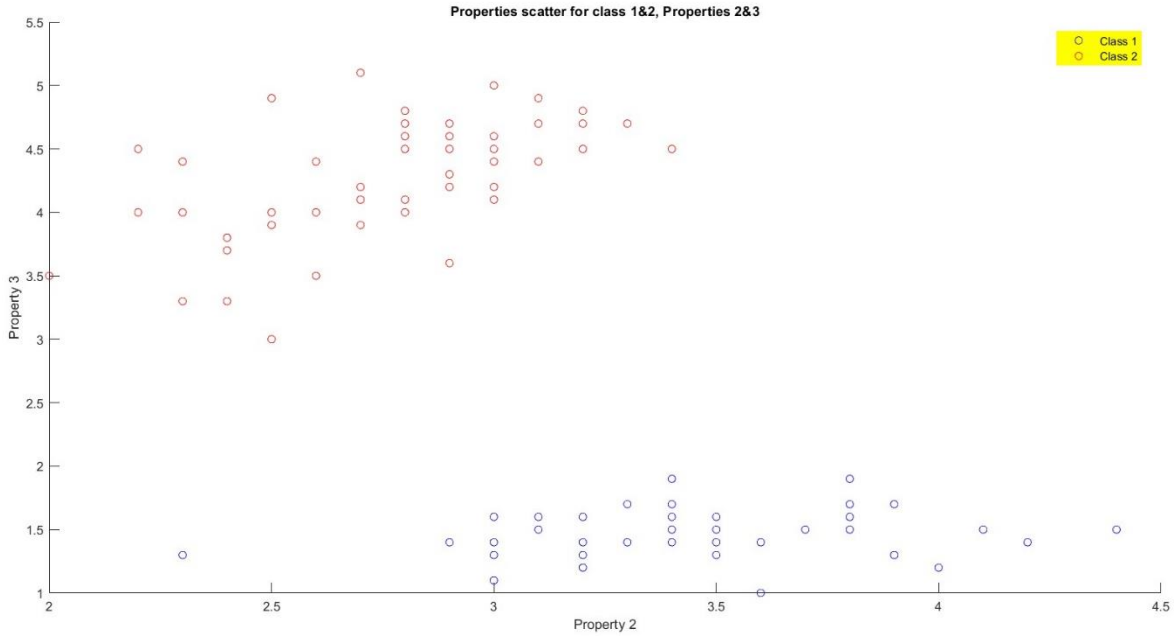
الف) کد این بخش در section اول از فایل سوال دوم (HW1_Q2_... .m) آمده است. توضیحات لازمه برای خود کد، در غالب کامنت نوشته شده است.

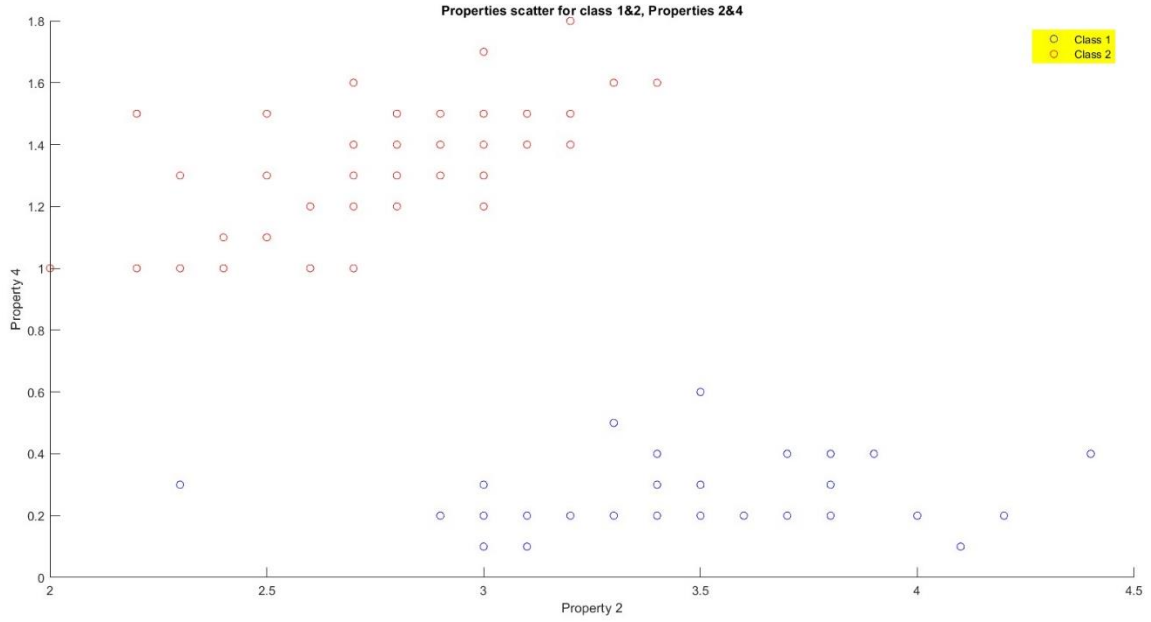
در section اول تمام حالات انتخاب دو ویژگی و رسم data point ها را انجام داده ایم. مجموعا به ۶ روش میتوان از بین چهار ویژگی موجود، دو تا را انتخاب و scatter رسم کرد. اگر کلاس گونه‌ی setosa را class1، و کلاس گونه‌ی versicolor را class2 بنامیم، ۶ نمودار حاصله به تفکیک ویژگی های انتخابی به صورت زیر خواهند بود:

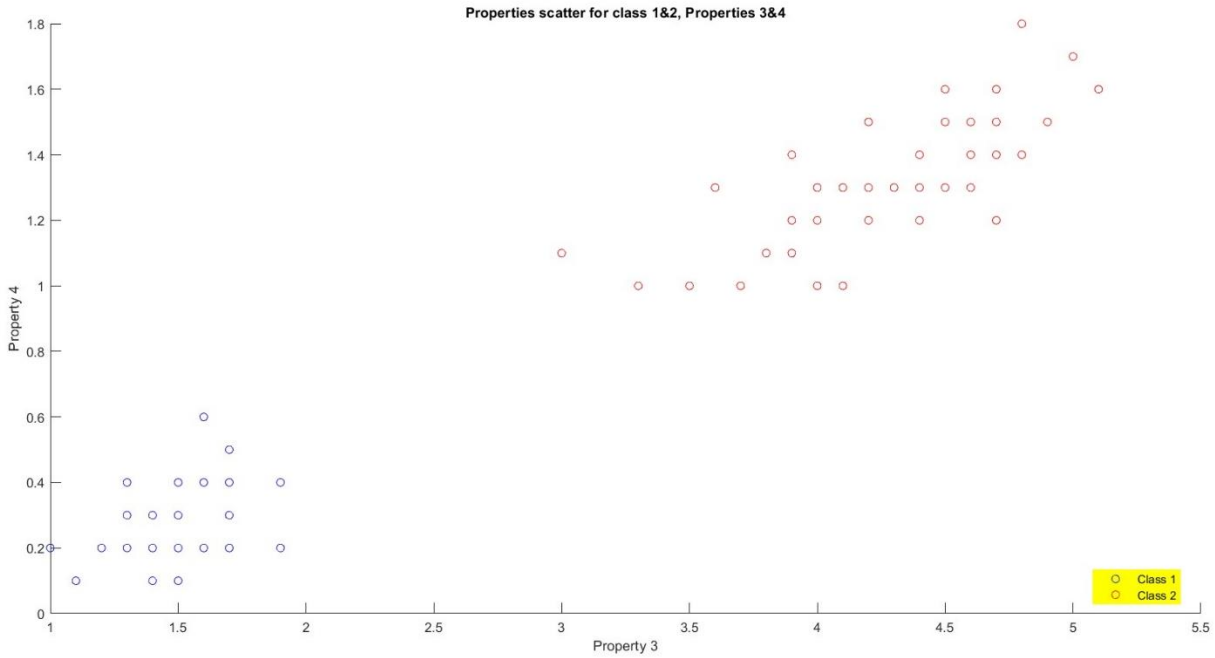



Properties scatter for class 1&2, Properties 1&3
Property 3
Property 1
Class 1
Class 2


Properties scatter for class 1&2, Properties 1&4
Property 4
Class 1
Class 2

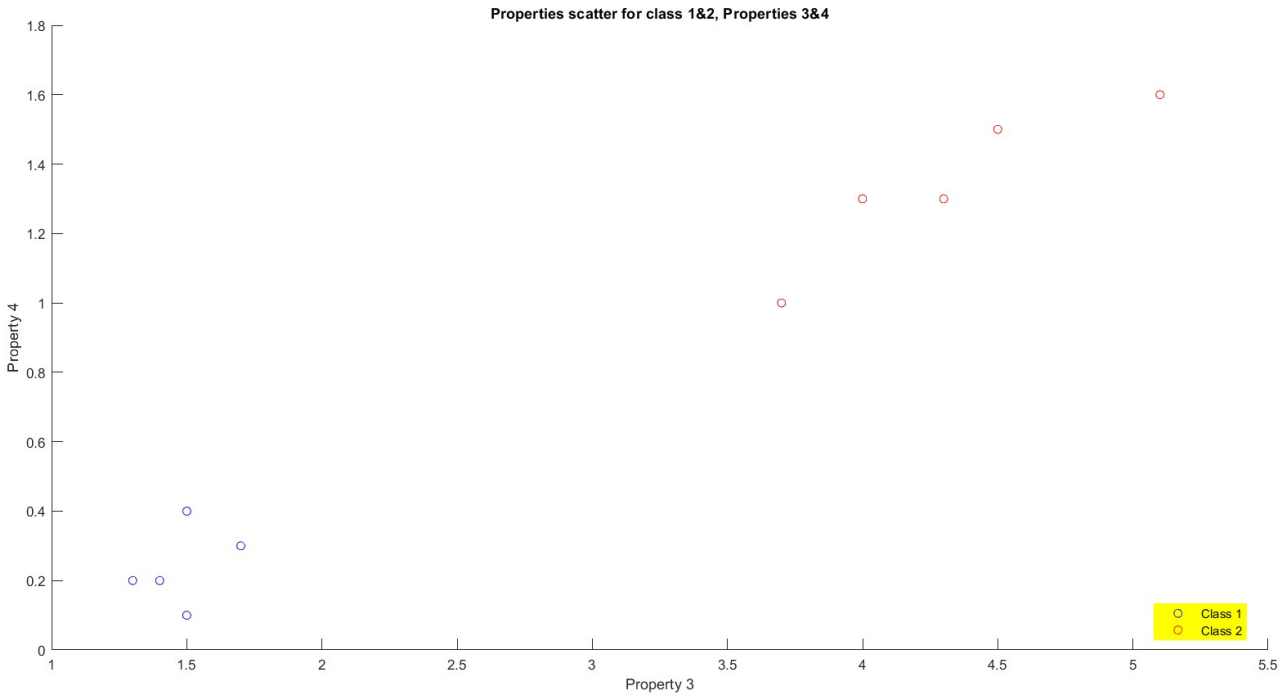

Properties scatter for class 1&2, Properties 2&3
Class 1
Class 2
Property 3
Property 2
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
2
2.5
3
3.5
4
4.5


Properties scatter for class 1&2, Properties 2&4
Class 1
Class 2
Property 4
Property 2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
2
2.5
3
3.5
4
4.5



به وضوح، دو کلاس در تمام این ۶ نمودار با یک خط قابل تفکیک اند (تفکیک پذیر خطی). از این میان، ما ویژگی های ۳ و ۴ را برای ادامه ی سوال انتخاب میکنیم (نمودار ششم در بالا).

ب) کد این بخش در section دوم آمده است. به صورت رندوم، از هریک از کلاس ها، (و از ویژگی های سوم و چهارم)، پنج نقطه انتخاب شده است. نمودار scatter به صورت زیر است که میخواهیم خطی را به آن فیت کنیم:



ویژگی سوم و چهارم را به ترتیب x1 و x2 مینامیم. وزن این دو را در TLU به ترتیب w1 و w2 مینامیم، و آستانه را θ مینامیم. در ابتدا از مقادیر اولیه ی w1 = 1 و w2 = 1 و $\theta = 1$ شروع میکنیم. نرخ یادگیری را نیز $\eta = 0.1$ میگیریم. فرض میکنیم خروجی y=0، متناظر با class1، و خروجی y=1 متناظر با class2 باشد:

$w_1 = w_2 = 1 \;,\; \theta = 1 \Longrightarrow \begin{cases} x_1 + x_2 \geq 1 : y = 1 \\ \text{“} \quad < 1 : y = 0 \end{cases}$

به روش online داریم:

Epoch 1: $(x_1, x_2) = (1.5, 0.4) \xrightarrow{\text{class 1}} 0 = 0 \,,\; w.x - \theta = 0.9 \geq 0 \rightarrow y = 1$

optimum value

$\eta = 0.1 \Longrightarrow \begin{cases} \Delta w_1 = \eta\,(0-y)\,x_1 = -0.15 \\ \Delta w_2 = \eta\,(0-y)\,x_2 = -0.04 \\ \Delta\theta = -\eta\,(0-y) = +0.1 \end{cases} \Longrightarrow \begin{cases} w_1 = 0.85 \\ w_2 = 0.96 \\ \theta = 1.1 \end{cases}$

$(x_1, x_2) = (1.7, 0.3) \xrightarrow{\text{class 1}} 0 = 0 \,,\; w.x - \theta = 0.633 \geq 0 \rightarrow y = 1$

به روش مشابه

به ازای همه

$\begin{cases} w_1 \rightarrow 0.68 \\ w_2 \rightarrow 0.93 \\ \theta \rightarrow 1.2 \end{cases} \longrightarrow \begin{cases} 0.68\,x_1 + 0.93\,x_2 - 1.2 \geq 0 : y = 1 \\ \text{“} \quad < 0 : y = 0 \end{cases}$

به روش مشابه برای سایر دیتا ها، این کار را تکرار می‌کنیم:

epoch	x_1	x_2	0	$x.w$	y	e	θ	w_1	w_2
							1	1	1
1	1.5	0.4	0	0.9	1	-1	1.1	0.85	0.96
	1.7	0.3	0	0.63	1	-1	1.2	0.68	0.93
	1.3	0.2	0	-0.13	0	0	"	"	"
	1.5	0.1	0	-0.087	0	0	"	"	"
	1.4	0.2	0	-0.06	0	0	"	"	"
	⋮ for all other inputs	⋮ for all other inputs	1	≥ 0	1	0	"	"	"
2	1.5	0.4	0	0.19	1	-1	1.3	0.53	0.89
	1.7	0.3	0	-0.13	0	0	"	"	"
	1.3	0.2	0	< 0	0	0	"	"	"
	1.5	0.1	0	< 0	0	0	"	"	"
	1.4	0.2	0	< 0	0	0	"	"	"
	3.7	1	1	1.55	1	0	"	"	"
	4	1.3	1	1.98	1	0	"	"	"
	4.3	1.3	1	2.136	1	0	"	"	"
	4.5	1.5	1	2.42	1	0	"	"	"
	5.1	1.6	1	2.83	1	0	"	"	"
3	1.5	0.4	0	-0.15	0	0	"	"	"

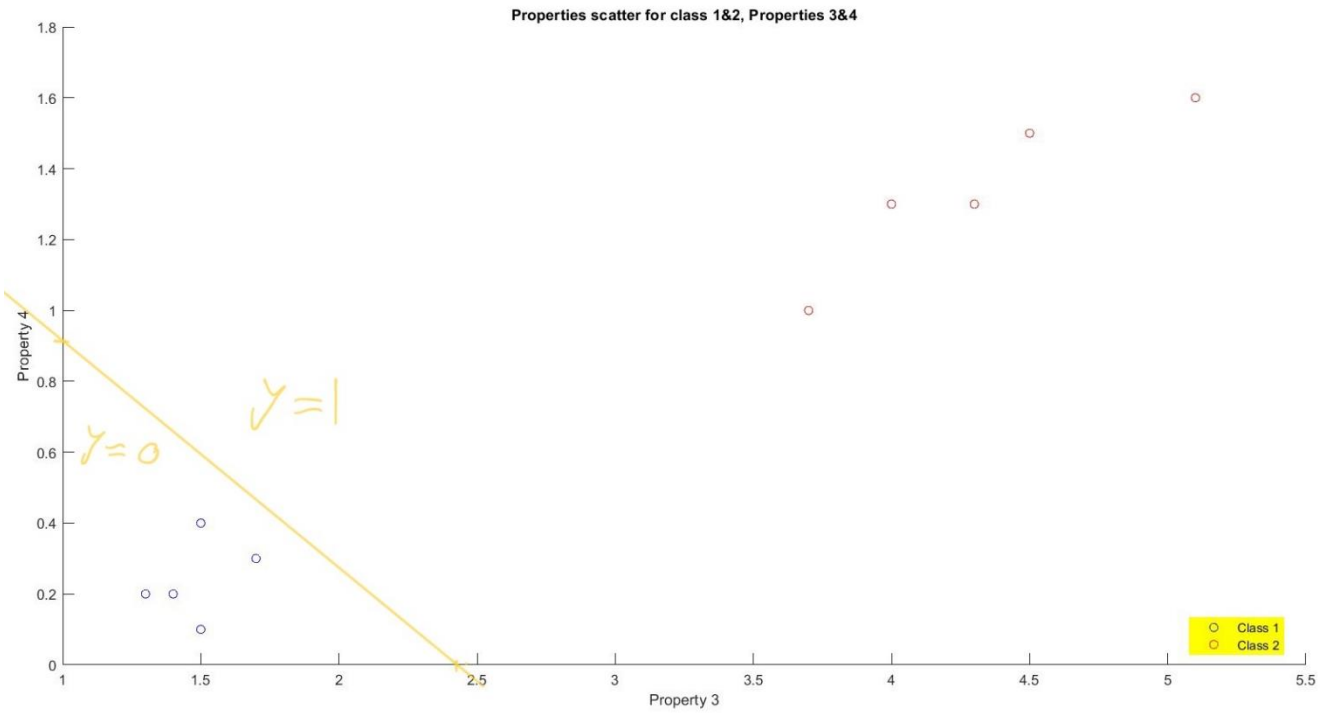
✓

در نهایت به ازای تمام ورودی ها، خطا صفر شد. به شکل زیر می‌رسیم:

$\begin{cases} 0.53\,x_1 + 0.89\,x_2 - 1.3 \geq 0 : y = 1 \quad (\text{class } 2) \\ \text{“} \quad < 0 : y = 0 \quad (\text{class } 1) \end{cases}$

(w_1: 0.53, w_2: 0.89, θ: 1.3)

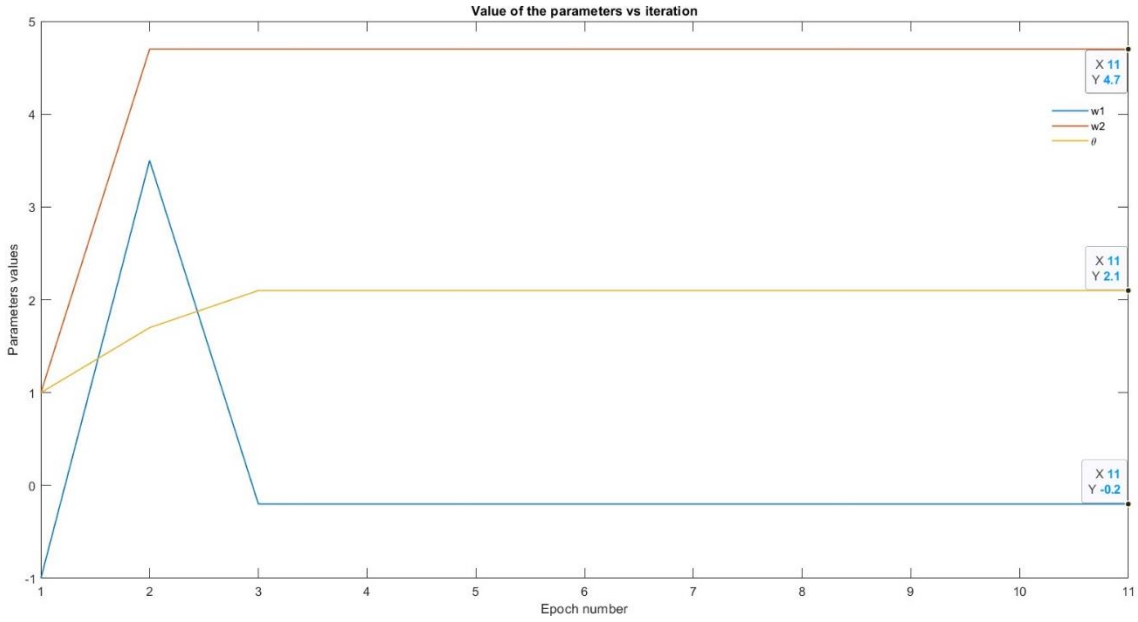
در شکل، این خط به صورت زیر است:

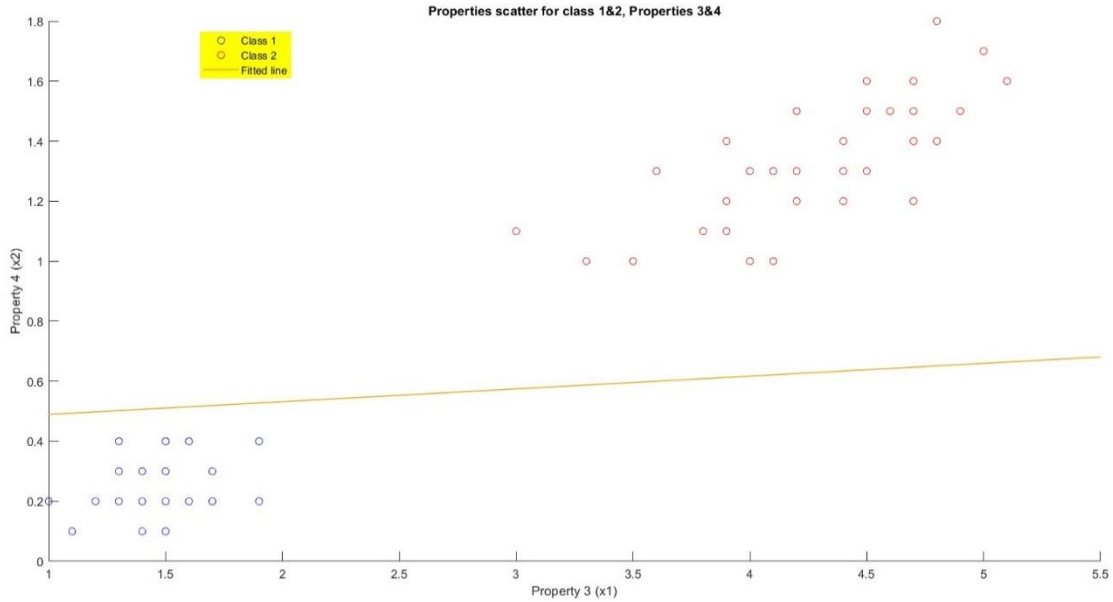


ج و د و ه) کد مربوط به این سه بخش، در دو section بعدی آمده است (اولی مربوط به روش online و دومی مربوط به روش batch است). توضیحات کد در غالب کامنت در فایل کد آمده است.

در روش online، در داخل هر iteration و به ازای هریک از نقاط آموزشی، پارامتر ها آپدیت میشوند. اما در روش batch، این تغییرات داخل سه پارامتر به نام های w1_c, w2_c, theta_c با هم جمع میشوند تا در انتهای iteration به پارامتر های کنونی اضافه شوند.

نمودار های زیر، به ترتیب مربوط به بخش د و ه سوال، به روش online هستند. مقدار اولیه در این نمودارها برای w1 و w2 و theta به ترتیب 1- و 1 و 1 فرض شده است:





در این روش، از ۱۰ گام برای یادگیری استفاده شده است و نرخ یادگیری نیز ۰.۱ فرض شده است. توجه شود که با هربار ران کردن کد، ترتیب بررسی ورودی ها در حلقه ی for، عوض میشود و در نتیجه خط نهایی، ممکن است متفاوت شود. نمونه های بالا، یکی از این خط ها هستند.

در نمودار اول، میبینیم که با روش online، شبکه به راحتی با ۳ گام همگرا شده و به خطای صفر رسیده است. مقادیر نهایی پارامتر ها در این نمودار mark شده اند:

$$w1 = -0.2, \qquad w2 = 4.7, \qquad \theta = 2.1$$

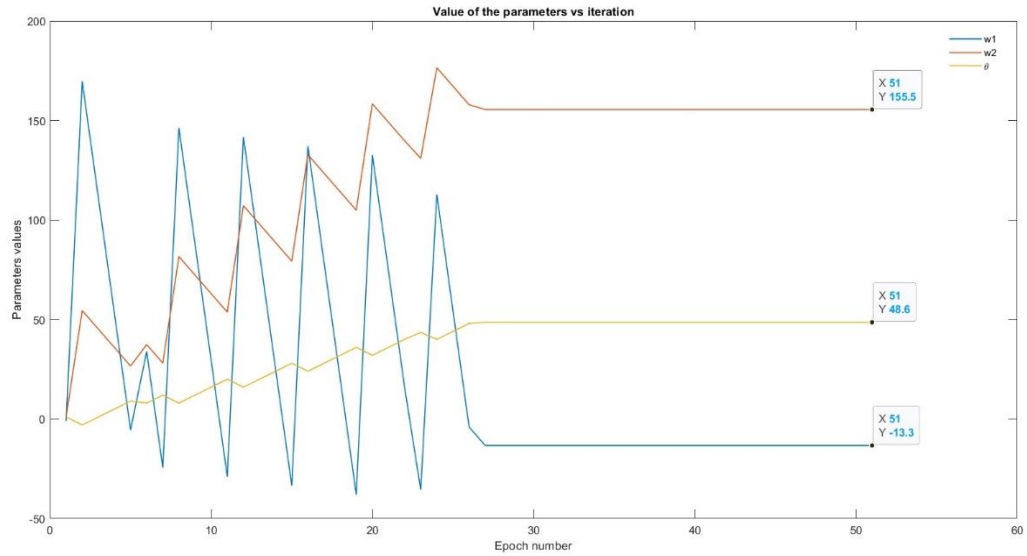
پس تصمیم گیری به صورت زیر خواهد بود:

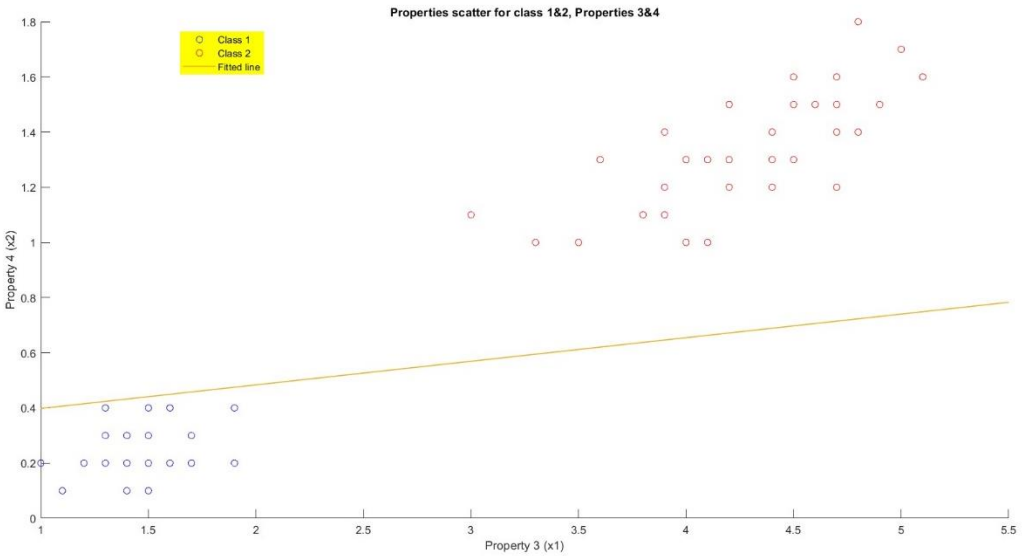
$$y = \begin{cases} 1 & , -0.2x1 + 4.7x2 - 2.1 \geq 0 \\ 0 & , -0.2x1 + 4.7x2 - 2.1 < 0 \end{cases}$$

پس خط فیت شده به صورت $-0.2x1 + 4.7x2 - 2.1 = 0$ است، که در نمودار دوم در بالا به همراه داده های یادگیری رسم شده است.

توجه داریم مثبت بودن w2 بدین معناست که بالای خط فیت شده، بعنوان y=1 (یا همان class2) و پایین آن، بعنوان y=0 (یا همان class1) در نظر گرفته شده است. شیب خط همان -w1/w2 و عرض از مبدا آن $\frac{\theta}{w2}$ است.

حال نمودار های روش batch را بررسی میکنیم:





نرخ یادگیری همچنان ۰.۱ است اما تعداد iteration ها در این روش، ۵۰ قرار داده شده است، زیرا این روش با تعداد گام های کم همگرا نخواهد شد. علت آن است که وقتی در یکی از گام ها، خط فیت شده زیر تمام دیتاها قرار بگیرد، یک class بطور کامل اشتباه بوده و خطاهای تمام آنها نسبت به این خط، روی هم جمع میشوند و در انتهای iteration، تغییرات بسیار بزرگی به پارامترها اعمال میشود. برعکس این روند را نیز برای حالتی داریم که خط فیت شده، بالای تمام داده ها باشد.

بدین ترتیب مشاهده میکنیم که قبل از همگرا شدن، پارامترها نوساناتی با دامنه های بزرگ دارند. اما در نهایت با کمتر از ۳۰ گام به همگرایی رسیده اند.

مقادیر نهایی پارامتر ها و نحوه تصمیم گیری به صورت زیر هستند:

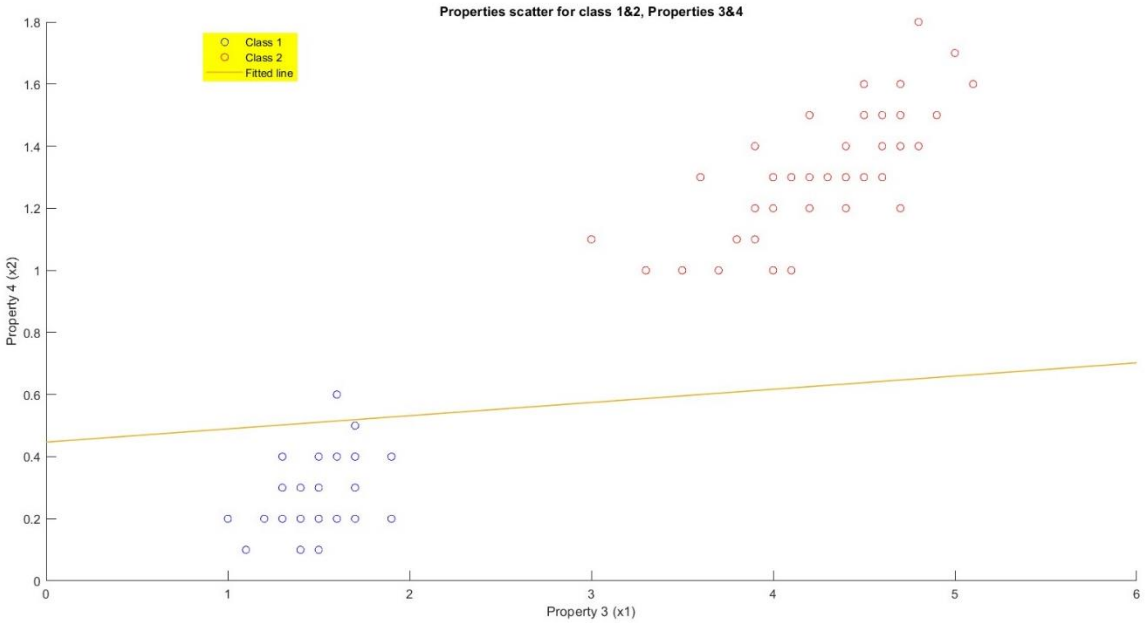
$$w1 = -13.3, \qquad w2 = 155.5, \qquad \theta = 48.6$$

$$y = \begin{cases} 1 & , -13.3x1 + 155.5x2 - 48.6 \geq 0 \\ 0 & , -13.3x1 + 155.5x2 - 48.6 < 0 \end{cases}$$

پس خط فیت شده به صورت $-13.3x1 + 155.5x2 - 48.6 = 0$ است، که در نمودار دوم در بالا به همراه داده های یادگیری رسم شده است. مثبت بودن w2 بدین معناست که بالای خط فیت شده، بعنوان y=1 (یا همان class2) و پایین آن، بعنوان y=0 (یا همان class1) در نظر گرفته شده است.

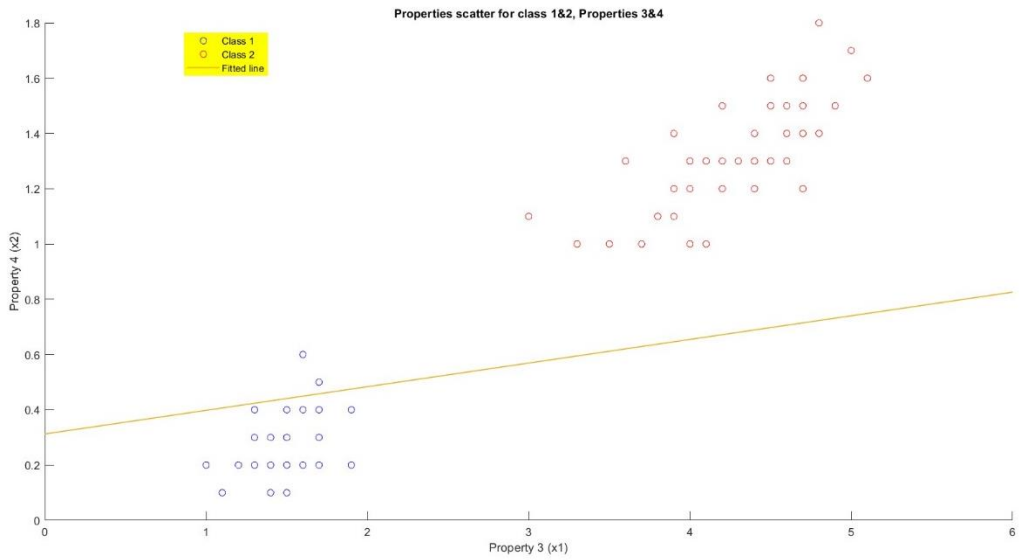
و) حال از ۲۰ ورودی باقی مانده برای تست کردن شبکه استفاده میکنیم. کد این بخش در section بعدی با نام مربوطه آمده است. مقدار w1_final و w2_final و theta_final، بعنوان پارامترهای نهایی شبکه، باید در ابتدای section داده شود. در نمونه ی بررسی شده، ما مقادیر مذکور را قرار میدهیم و نمودار نهایی کل داده ها، همراه با خط فیت شده نمایش داده خواهد شد. ضمنا مقدار پارامتر e_test نیز در command line چاپ میشود که در واقع تعداد داده های تستی است که به اشتباه کلاس بندی شده اند.

برای روش online، نمودار نهایی به صورت زیر است:



مشاهده میشود که یکی از داده های تست از کلاس ۱، در کلاس ۲ قرار گرفته است. پس e_test=1 است.

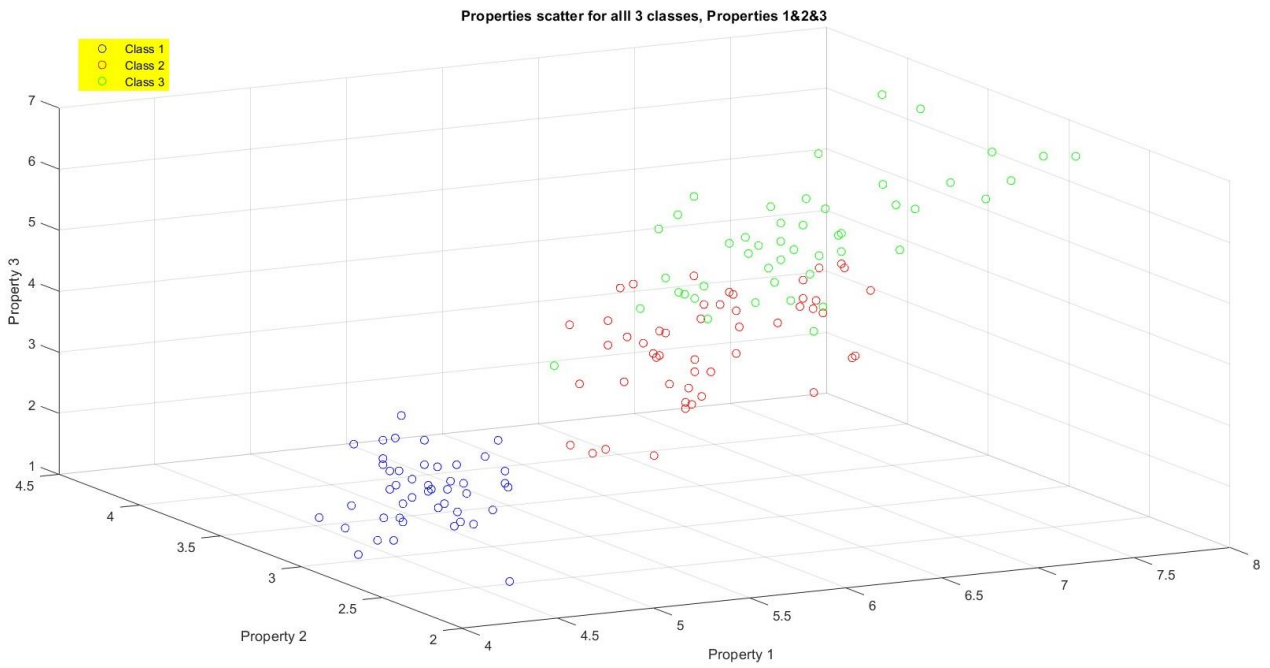
در روش batch و به ازای پارامتر هایی که در بالا به آنها رسیدیم، به نمودار زیر میرسیم:

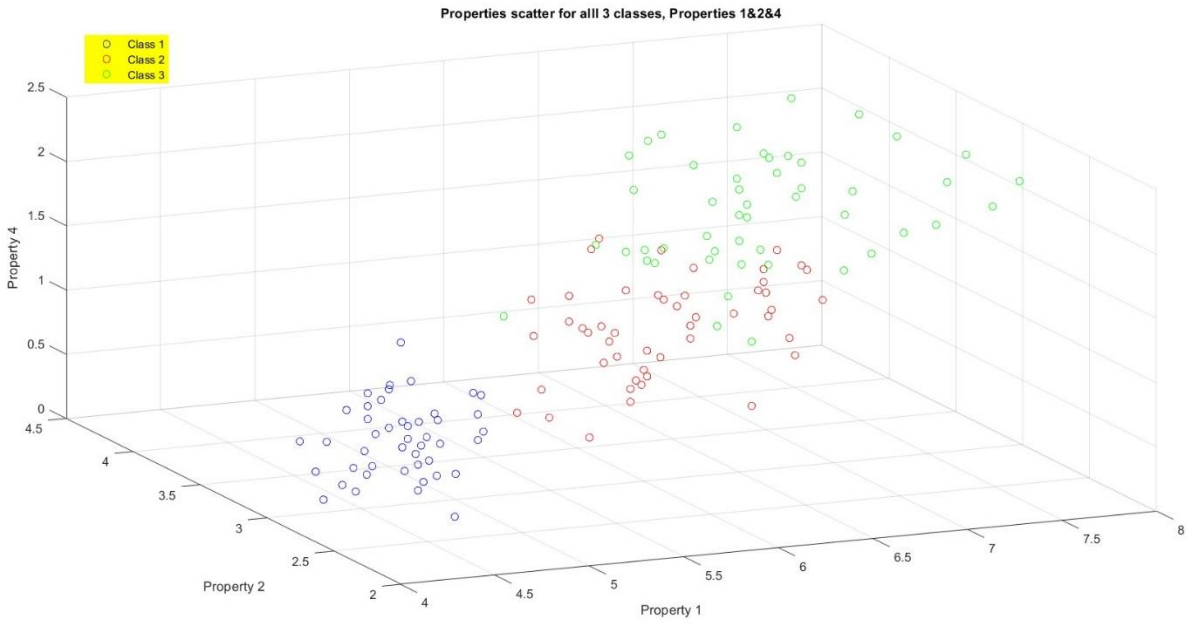


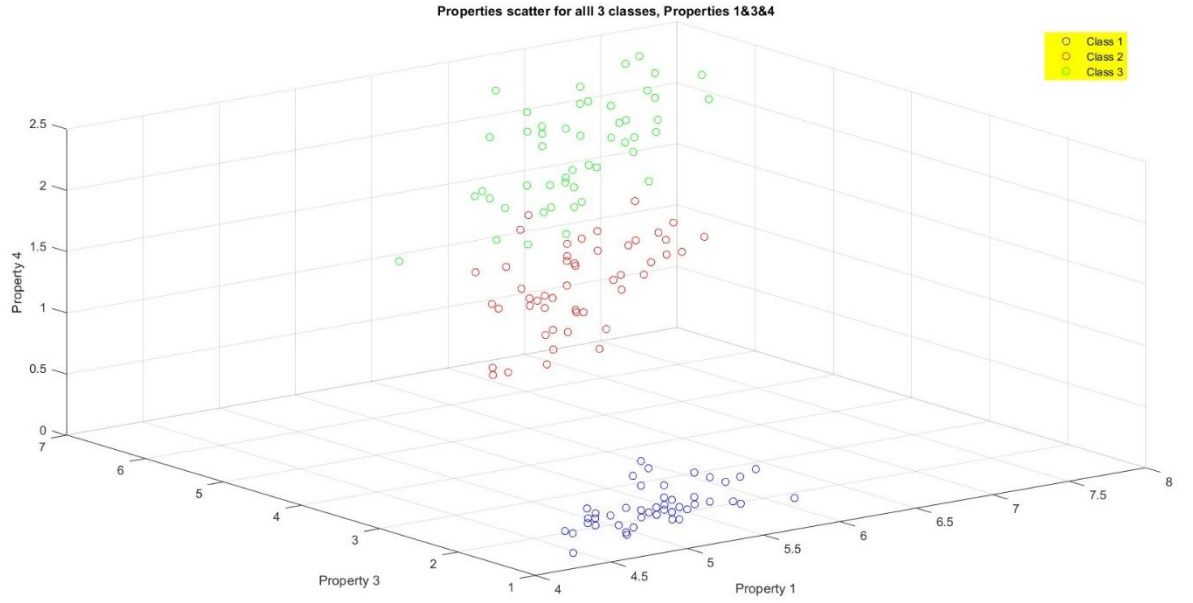
که در آن دوتا از داده های کلاس ۱ به اشتباه در کلاس ۲ طبقه بندی شده اند. پس e_test برابر با ۲ خواهد بود.

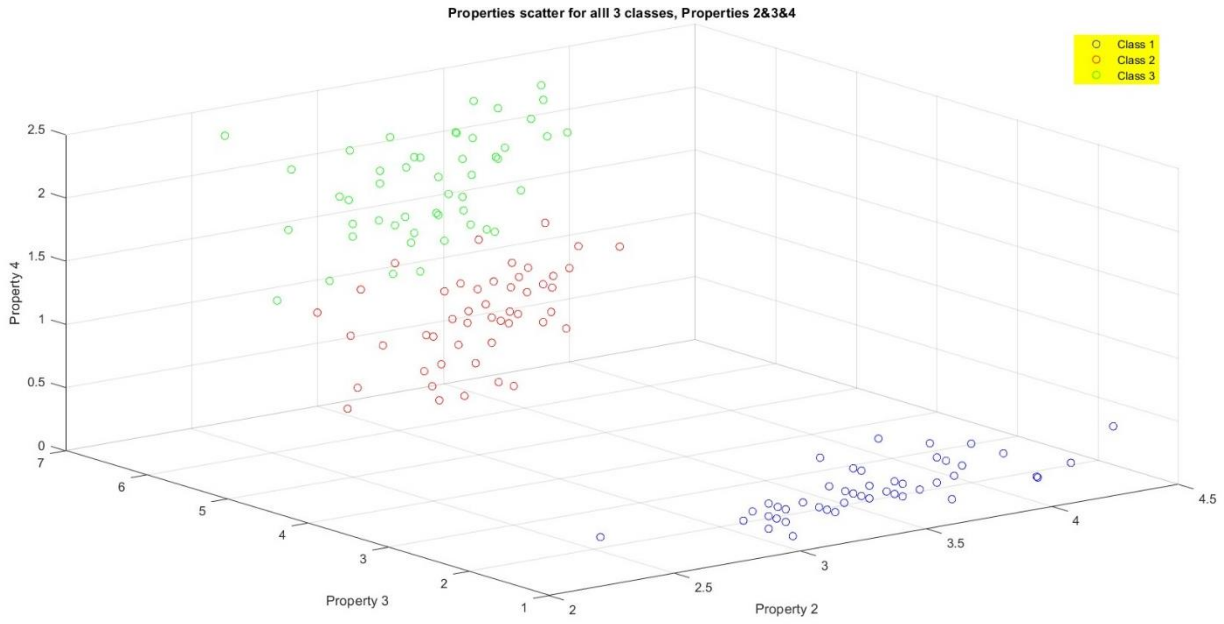
میتوان مقدار خطا را به صورت مجموع مربعات تفاضل خروجی و مقدار مطلوب نیز محاسبه کرد که در این حالت نیز به همان مقادیر e_test خواهیم رسید.

ز) این بخش در section آخر کد سوال ۲ آمده است. مشابه بخش الف عمل میکنیم اما این بار، سه تا حلقه ی for داریم که سومی، ویژگی سوم را انتخاب میکند. به ازای هر سه ویژگی انتخاب شده، یک نمودار سه بعدی رسم میشود که دیتای سه کلاس در آن قرار دارند. نمودارهای زیر، همین scatter ها به ازای ترکیب های مختلف ویژگی هستند. کلاس اول داده ها با رنگ آبی، کلاس دوم با رنگ قرمز و کلاس سوم با رنگ سبز نمایش داده شده است:




Properties scatter for alll 3 classes, Properties 1&2&4
Class 1
Class 2
Class 3
Property 4
Property 2
Property 1


Properties scatter for alll 3 classes, Properties 1&3&4
Class 1
Class 2
Class 3
Property 4
Property 3
Property 1



داده ها زمانی با یک TLU تفکیک پذیر هستند که (در فضای ۳ بعدی) بتوان آنها را با یک صفحه (ابر صفحه ۲ بعدی) از یکدیگر جدا کرد. طبق نمودارهای بالا، داده های کلاس ۱ و ۲ به راحتی و در هر ۴ نمودار، با یک صفحه قابل تفکیک اند و یک TLU برای این کار کافیست. داده های کلاس ۱ و ۳ نیز همینطور. اما داده های کلاس ۲ و ۳ در هم آمیخته شده اند (در هر ۴ حالت) و با یک TLU (یک صفحه) قابل تفکیک نیستند. اما میتوان از چند صفحه برای تفکیک آنها استفاده کرد. بدین ترتیب که هر یک یا چند صفحه، ناحیه ای مطلوب از یک کلاس را مشخص کنند، و خروجی این شبکه ها با یکدیگر OR شود. البته اینکار برای دو نمودار اول بسیار سخت تر است (چون تداخل داده ها بیشتر است)، اما در نمودار آخر، تقریبا با ۴ عدد TLU میتوان داده ها را جدا کرد. بدین صورت که (با فرض اینکه کلاس ۲، متناظر با خروجی ۱ باشد) ابتدا یک صفحه برای جداسازی بخش های بدون تداخل قرار میدهیم. سپس بخش دارای تداخل را با دو صفحه میتوان جدا کرد (این دو صفحه طوری باشند که ناحیه ی میانشان، فقط داده های کلاس ۲ که با کلاس ۳ تداخل دارند را شامل شود، و

این دو صفحه را AND کنیم). سپس با یک صفحه ی چهارم، نواحی مشخص شده توسط سه صفحه قبلی را OR کنیم.

اگر هم بخواهیم شبکه ای بسازیم که خروجی اش یکی از بین سه کلاس باشد (هر سه حالت را در خروجی داشته باشیم)، باز باید کلاس ۲ و ۳ را با روش مذکور در بالا جدا کنیم، و سپس با یک صفحه ی دیگر، این دو را از کلاس ۱ جدا کنیم تا در خروجی بتوان کلاس داده را (از بین ۱ و ۲ و ۳) مشخص کرد.