

گزارش تمرین سری اول - شبکه عصبی

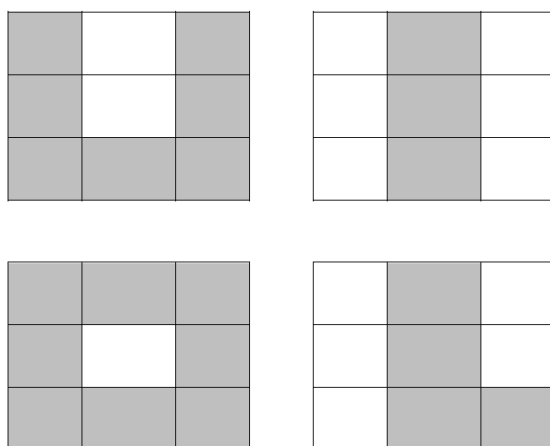
میثم پرویزی

دانشکده مهندسی برق و کامپیوتر

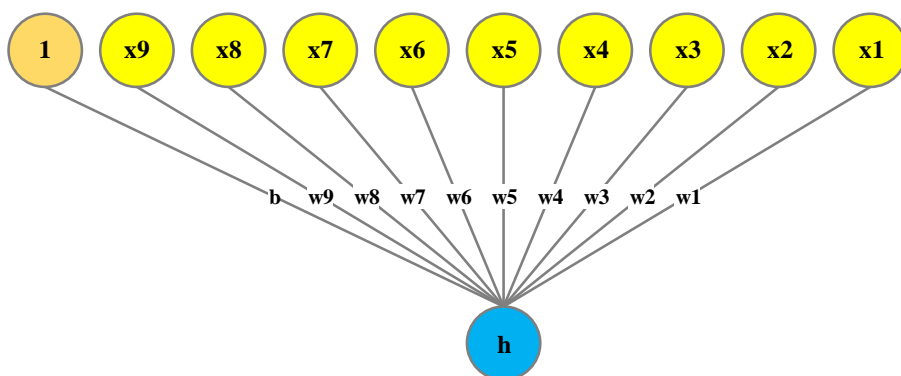
پردیس دانشکده‌های فنی

دانشگاه تهران

۱. طراحی شبکه عصبی برای جداسازی کاراکتر U از سه کاراکتر دیگر.



برای این کار یک شبکه عصبی تک لایه با ۹ ورودی (برابر با تعداد پیکسل‌ها) و یک خروجی (جهت جداسازی کاراکتر U از دیگر کاراکترها) مشابه شکل زیر طراحی می‌کنیم.



در این ساختار ارتباط بین ورودی‌ها و خروجی‌های شبکه بر اساس رابطه‌ی زیر تعریف می‌گردد.

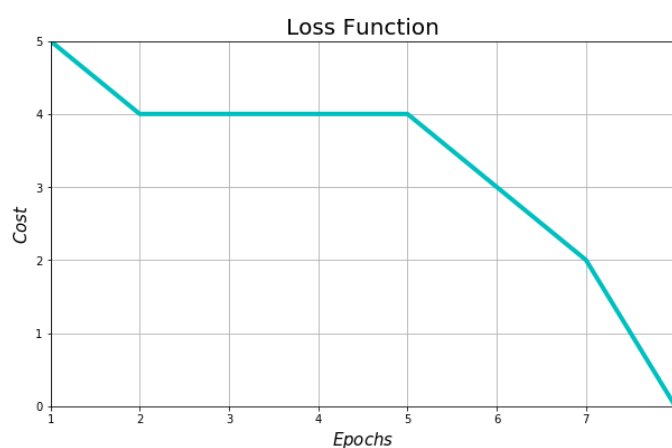
$$\text{net} = (w1 \ w2 \ w3 \ w4 \ w5 \ w6 \ w7 \ w8 \ w9) \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \\ x8 \\ x9 \end{pmatrix} + b$$

$$\mathbf{h} = f_{\text{activation}}(\text{net}) = \begin{cases} 1 & \text{net} - \theta > 0 \\ 0 & \text{net} - \theta = 0 \\ -1 & \text{net} - \theta < 0 \end{cases}$$

معمولاً در رابطه‌ی فوق به جای θ عدد ۰ قرار می‌دهیم که در عمل تابع فعالسازی دوقطبی تبدیل به تابع علامت می‌شود. در گام بعدی برای انجام فرایند یادگیری از روش پرسپترون استفاده می‌کنیم که در آن وزن‌ها و بایاس طبق رابطه‌ی زیر آپدیت می‌شوند:

$$\begin{aligned} \mathbf{W}_{\text{new}} &= \mathbf{W}_{\text{old}} + \alpha t \mathbf{x} \\ \mathbf{b}_{\text{new}} &= \mathbf{b}_{\text{old}} + \alpha t \end{aligned}$$

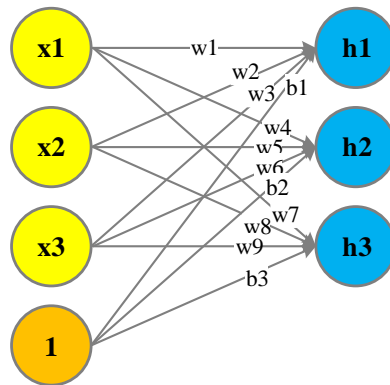
که در آن α ضریب یادگیری است که برابر با ۱ قرار خواهیم داد و t خروجی مطلوب به ازای ورودی x است. برای پیاده‌سازی این شبکه عصبی در این تمرین از زبان پایتون استفاده کرده‌ایم که کد مربوط به آن در فایل‌های پیوست شده با نام HW1-1 موجود است. همانگونه که در شکل زیر مشاهده می‌شود در فرایند یادگیری به روش پرسپترون در نهایت تابع هزینه به صفر می‌رسد و شبکه‌ی عصبی خروجی مورد نظر را تولید می‌کند.



۲. طراحی شبکه عصبی برای تفکیک سه میوه سیب، پرتقال و گلابی از روی مشخصات ظاهری آن‌ها.

فرم	سفتی پوست	رنگ	میوه
دایره	شل	زرد	سیب
دایره	سفت	نارنجی	پرتقال
غیر دایره	شل	زرد	گلابی

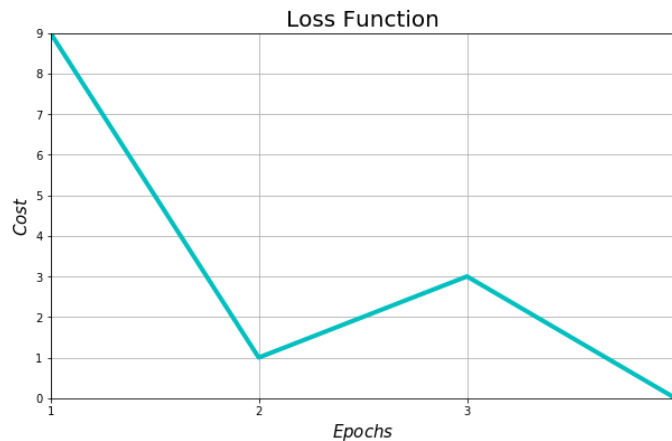
برای این تمرین یک شبکه عصبی با ۳ ورودی (مشخص کننده سه ویژگی رنگ، سفتی و فرم) و ۳ خروجی از نوع One-Hot مشخص کننده‌ی یکی از سه میوه‌ی سیب، پرتقال و گلابی مطابق شکل زیر طراحی می‌کنیم.



در این ساختار ارتباط بین ورودی‌ها و خروجی‌های شبکه بر اساس رابطه‌ی زیر تعریف می‌گردد:

$$\begin{pmatrix} h1 \\ h2 \\ h3 \end{pmatrix} = f_{\text{activation}} \left(\begin{pmatrix} w1 & w2 & w3 \\ w4 & w5 & w6 \\ w7 & w8 & w9 \end{pmatrix} \begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} + \begin{pmatrix} b1 \\ b2 \\ b3 \end{pmatrix} \right)$$

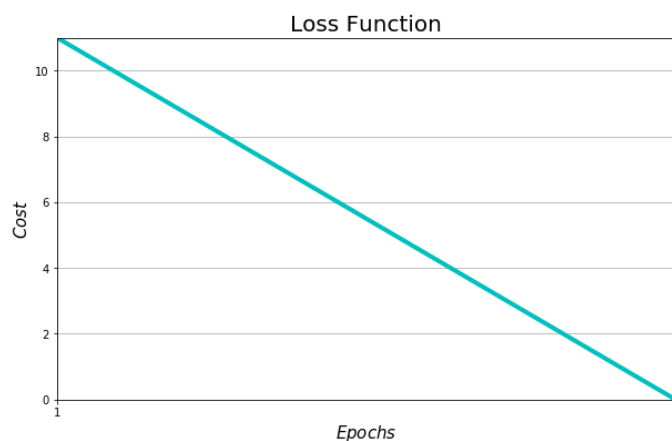
در این بخش نیز برای انجام فرایند یادگیری از روش پرسپترون با تابع فعالسازی دوقطبی استفاده می‌کنیم. برای پیاده‌سازی این شبکه عصبی نیز از زبان پایتون استفاده کرده‌ایم که کد مربوط به آن در فایل‌های پیوست شده با نام HW1-2-A موجود است. همانگونه که در شکل زیر مشاهده می‌شود در فرایند یادگیری به روش پرسپترون در نهایت تابع هزینه به صفر می‌رسد و شبکه‌ی عصبی خروجی مورد نظر را تولید می‌کند.



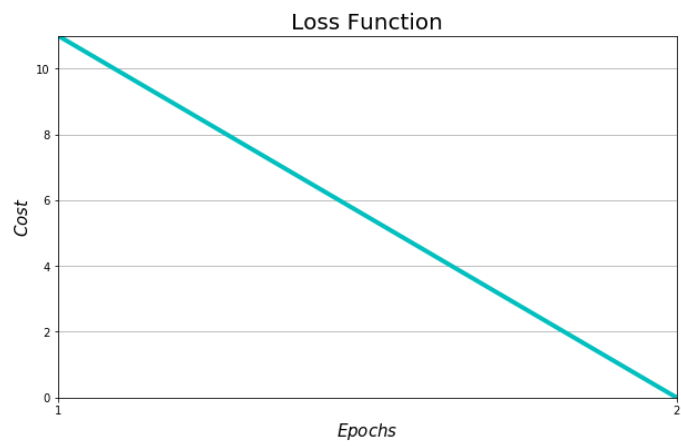
در صورت تمرین سوال شده است که «آیا می توان فقط از دو ورودی برای جداسازی داده ها استفاده کرد؟»

با بررسی داده های موجود در جدول ویژگی های میوه ها متوجه می شویم که دو ویژگی رنگ و سفتی پوست برای دو میوه سیب و گلابی کاملاً یکسان هستند و نمی توانند این دو میوه را از هم تفکیک کنند. اما زوج ویژگی های رنگ و فرم و همچنین سفتی پوست و فرم می توانند هر سه میوه را از میوه های دیگر تفکیک کنند.

بنابراین یک بار با حذف ویژگی سفتی پوست، یک شبکه عصبی ۲ ورودی و ۳ خروجی طراحی کردیم که کد آن در فایل HW1-2-B موجود است و نمودار تابع هزینه ی آن نیز در شکل زیر آورده شده است. این شبکه ی عصبی نیز به خوبی جداسازی را انجام می دهد.

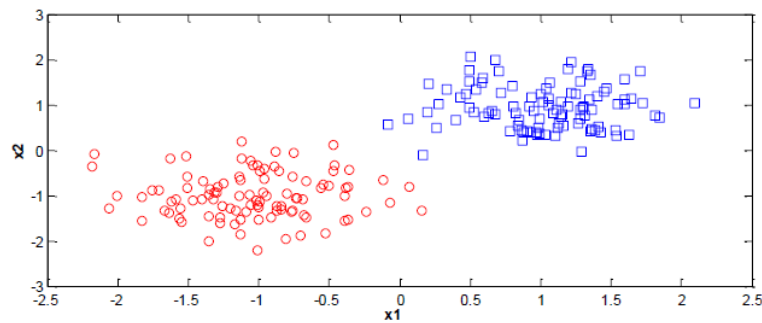


بار دیگر با حذف ویژگی رنگ، یک شبکه عصبی ۲ ورودی و ۳ خروجی طراحی کردیم که کد آن در فایل HW1-2-C موجود است و نمودار تابع هزینه ی آن نیز در شکل زیر آورده شده است. این شبکه ی عصبی نیز به خوبی جداسازی را انجام می دهد.



۳. طبقه‌بندی دو دسته نقاط حول ۱ و -۱ به سه روش مختلف و با حجم‌های مختلف.

• داده‌های متقارن و هم حجم در فضای $x = [x_1 \ x_2]^T$

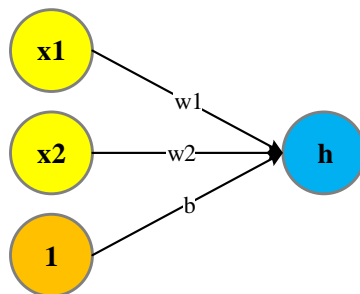


نحوه تولید داده در فضای دو بعدی به کمک MATLAB (و البته Python)

تولید ورودی ۱	تولید ورودی ۲	تعداد داده ها	کلاس
$X2=1+0.5 \text{ randn}(Q1,1)$	$X1=1+0.5 \text{ randn}(Q1,1)$	$Q1=100$	۱
$X2=-1+0.5 \text{ randn}(Q2,1)$	$X1=-1+0.5 \text{ randn}(Q2,1)$	$Q2=100$	۲
اجتماع دو کلاس در ورودی ۲	اجتماع دو کلاس در ورودی ۱	$Q=200$	در مجموع

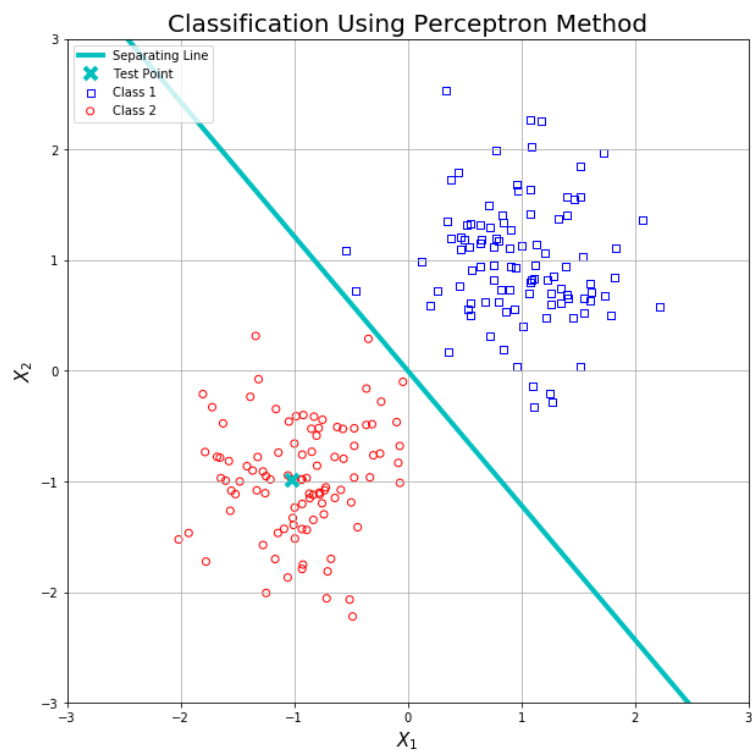
تابع $\text{randn}(a,b)$ یک آرایه a ردیفی و b ستونی شامل اعداد تصادفی با تابع توزیع گوسی میانگین صفر و واریانس ۱

در این تمرین در ابتدا یک شبکه‌ی عصبی تک لایه با ۲ ورودی و ۱ خروجی مطابق شکل زیر طراحی می‌کنیم که دو ورودی آن مختصات یک نقطه در فضای $[x_1 \ x_2]^T$ را مشخص می‌کند و خروجی آن مشخص‌کننده‌ی آن است که این نقطه به کدام یک از دو دسته داده‌ی مورد استفاده در مرحله‌ی آموزش اختصاص دارد.

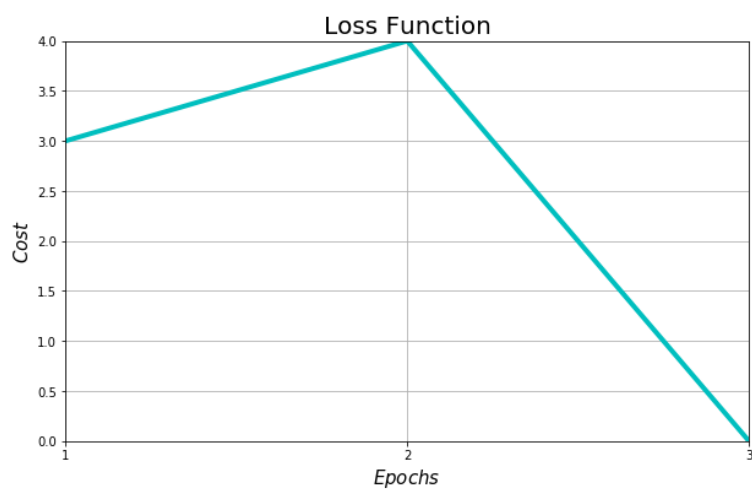


$$\mathbf{h} = f_{\text{activation}} \left((\mathbf{w1} \ \mathbf{w2}) \begin{pmatrix} \mathbf{x1} \\ \mathbf{x2} \end{pmatrix} + \mathbf{b} \right)$$

در قسمت (الف) این تمرین فرایند یادگیری را با روش پرسپترون انجام می‌دهیم که نتایج آن در دو نمودار زیر آورده شده است. شکل زیر نقاط ورودی شبکه و دسته‌بندی آن‌ها را نمایش می‌دهد. همچنین خط جداکننده به همراه یک نقطه‌ی نمونه که برای تست به شبکه داده شده است نمایش داده شده است. کد مربوط به این بخش در فایل HW1-3-A موجود است.



نمودار زیر نیز روند تغییرات تابع هزینه در تکرارهای مختلف فرایند آموزش را نمایش می‌دهد.

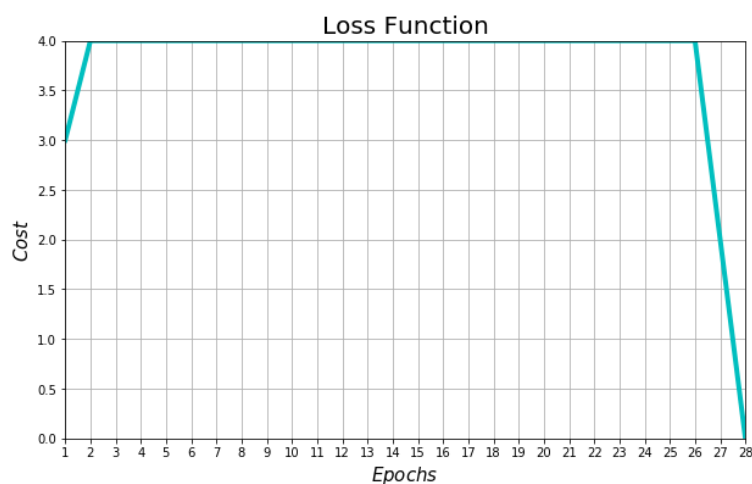
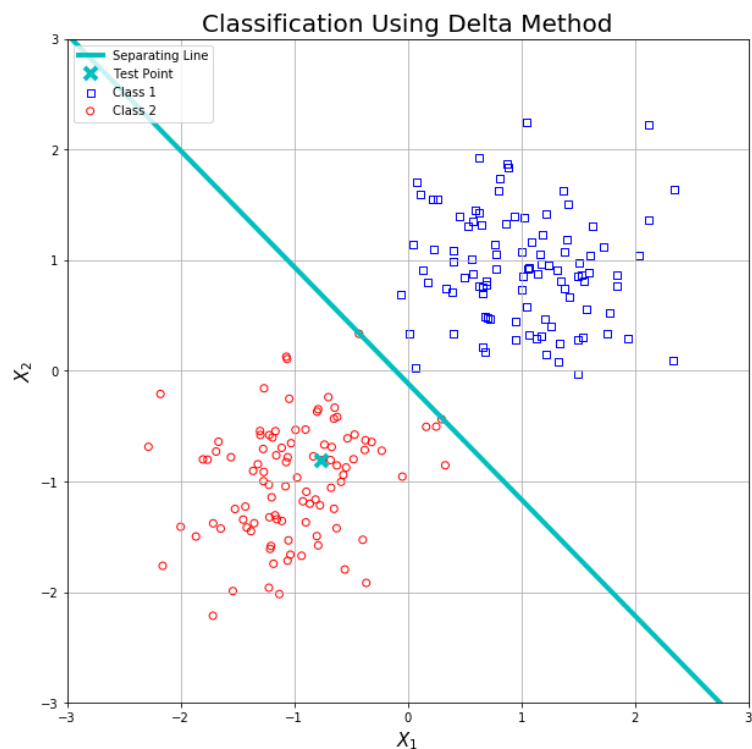


در قسمت (ب) فرایند آموزش را با روش دلتا انجام می‌دهیم. در این روش آپدیت کردن وزن‌ها و بایاس طبق رابطه‌ی زیر انجام می‌پذیرد.

$$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \alpha(t - h)\mathbf{x}$$

$$\mathbf{b}_{\text{new}} = \mathbf{b}_{\text{old}} + \alpha(t - h)$$

نتایج این روش نیز در نمودارهای زیر آورده شده است. کد مربوط به این بخش در فایل HW1-3-B موجود است.

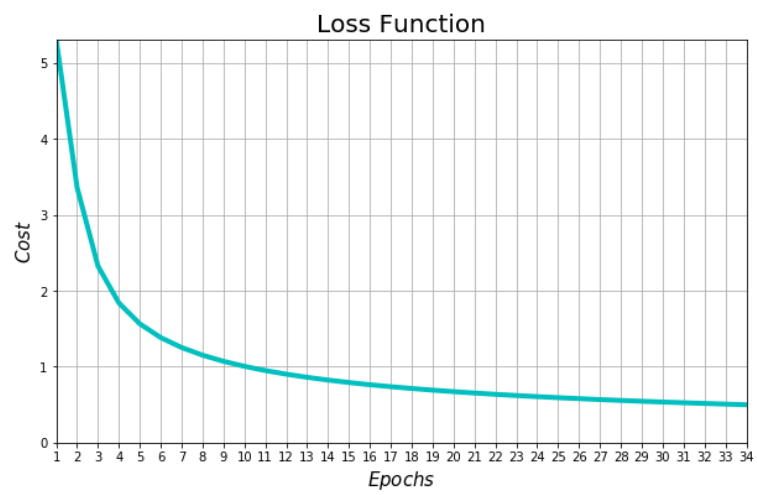
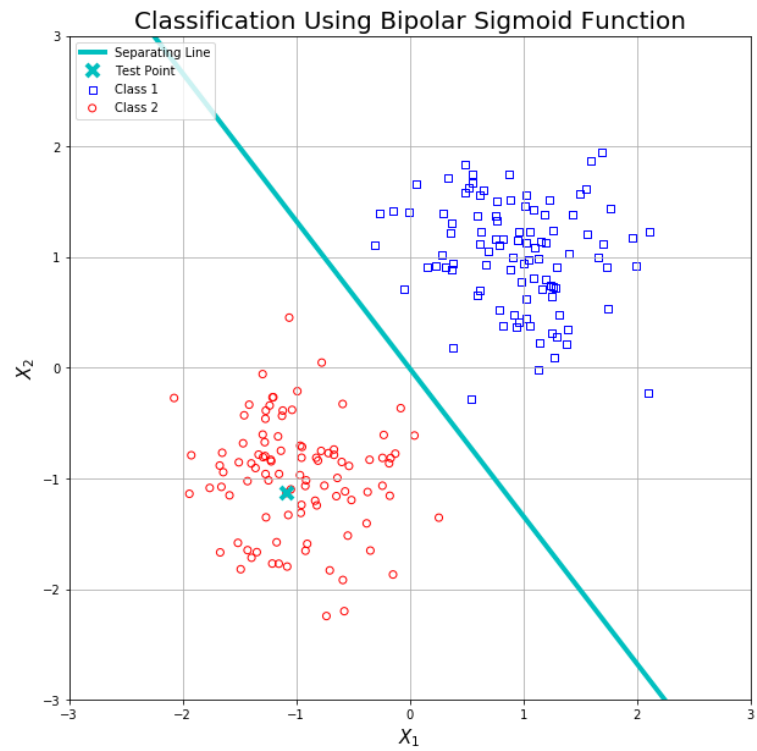


در قسمت (ج) فرایند آموزش را با استفاده از تابع فعالساز سیگموئید دوقطبی \tanh انجام می‌دهیم. در این روش آپدیت کردن وزن‌ها و بایاس طبق رابطه‌ی زیر انجام می‌پذیرد.

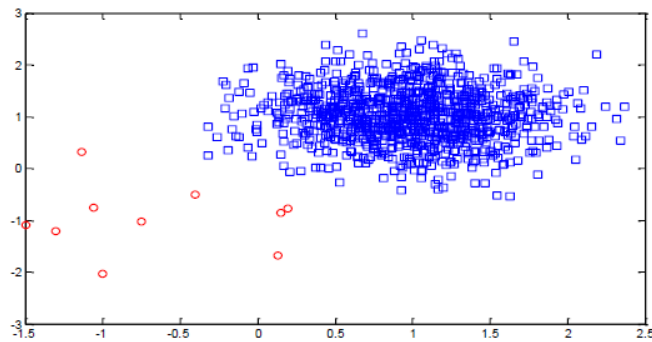
$$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \alpha \gamma (t - \tanh(\gamma * \text{net})) (1 - \tanh^2(\gamma * \text{net})) \mathbf{x}$$

$$\mathbf{b}_{\text{new}} = \mathbf{b}_{\text{old}} + \alpha \gamma (t - \tanh(\gamma * \text{net})) (1 - \tanh^2(\gamma * \text{net}))$$

نتایج این روش با قرار دادن $\alpha = 0.01$ و $\gamma = 10$ در نمودارهای زیر آورده شده است. کد مربوط به این بخش در فایل HW1-3-C موجود است.



- داده‌ها با حجم‌های متفاوت در فضای $x = [x_1 \ x_2]^T$



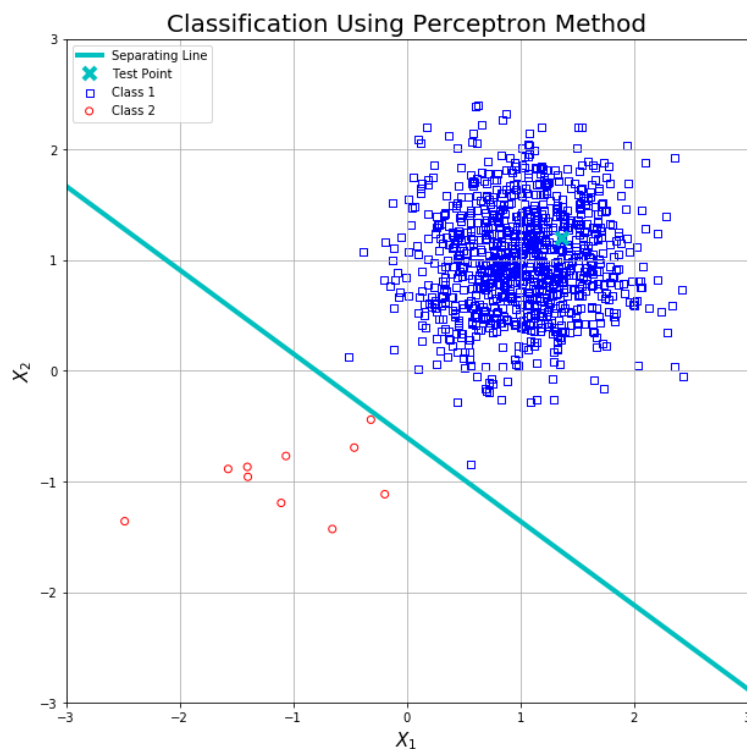
نحوه تولید داده در فضای دو بعدی

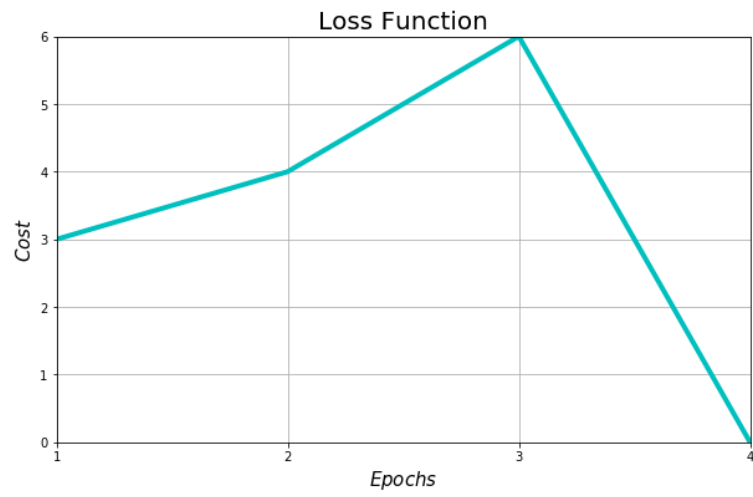
تولید ورودی ۱ در MATLAB	تولید ورودی ۱ در (و البته Python)	تعداد داده‌ها	کلاس
$X2=1+0.5 \text{ randn}(Q1,1)$	$X1=1+0.5 \text{ randn}(Q1,1)$	$Q1=1000$	۱
$X2=-1+0.5 \text{ randn}(Q2,1)$	$X1=-1+0.5 \text{ randn}(Q2,1)$	$Q2=10$	۲
اجتماع دو کلاس در ورودی ۲	اجتماع دو کلاس در ورودی ۱	$Q=1010$	در مجموع

تابع $\text{randn}(a,b)$ یک آرایه a ردیفی و b ستونی شامل اعداد تصادفی با تابع توزیع گوسی میانگین صفر و واریانس ۱

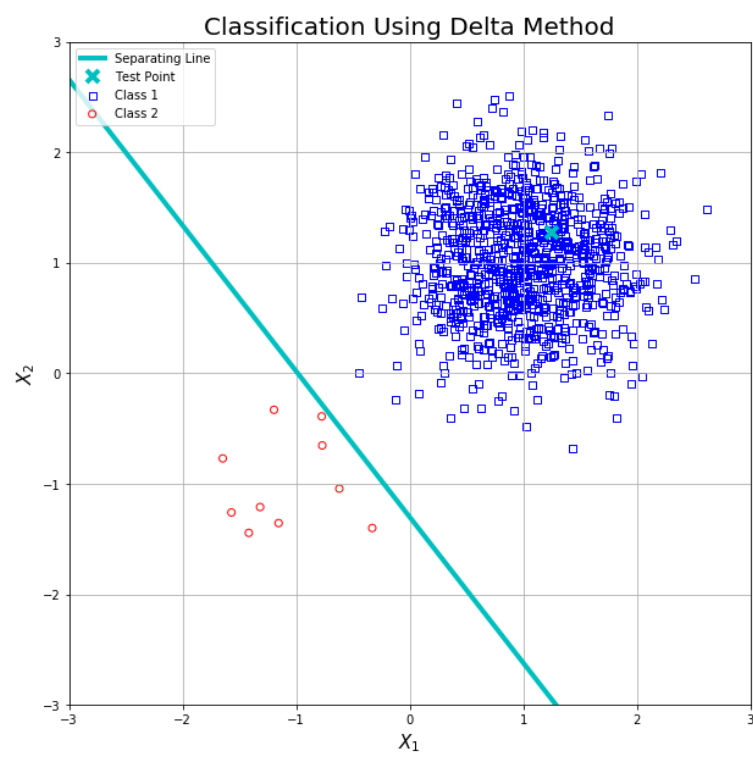
در این بخش حجم دو دسته داده‌ی مورد استفاده برای آموزش را طوری تغییر می‌دهیم تا یکی بسیار بزرگتر از دیگری باشد. پس از اعمال این تغییر مجدداً سه روش (الف)، (ب) و (ج) را برای آموزش مورد استفاده قرار می‌دهیم که نتایج آن در ادامه گزارش شده است.

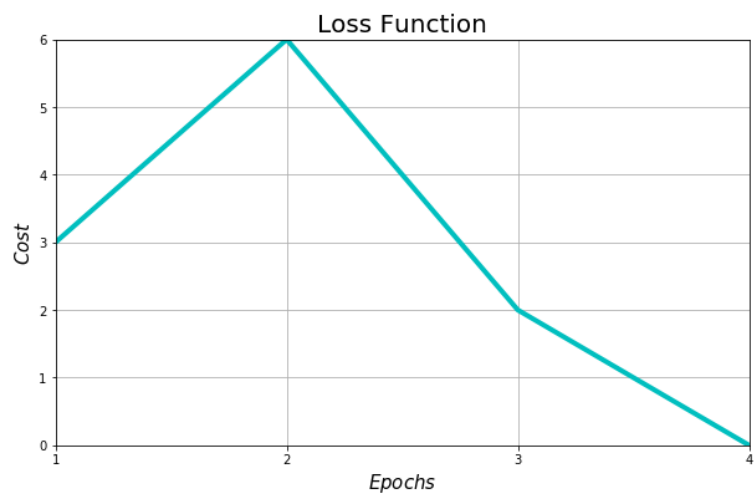
(الف) روش پرسپترون



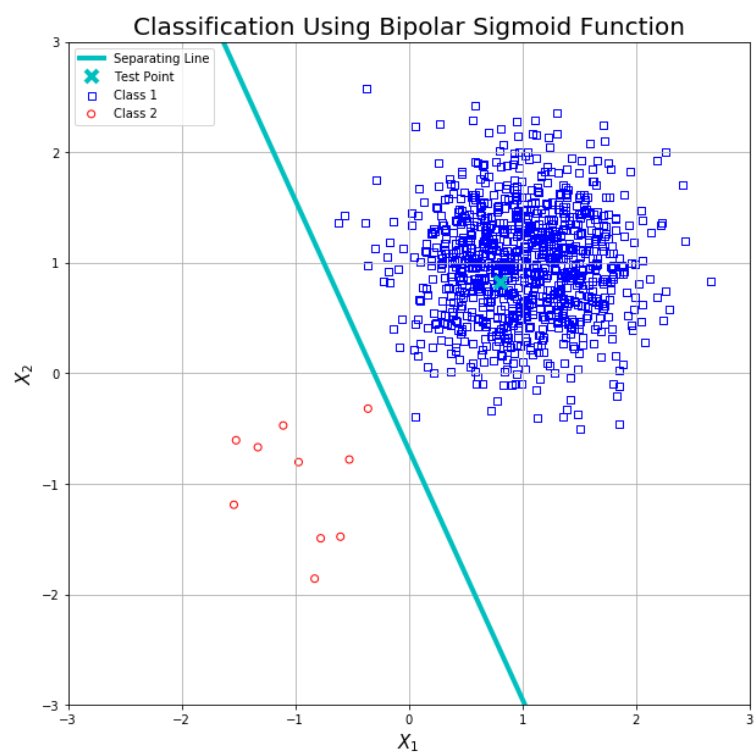


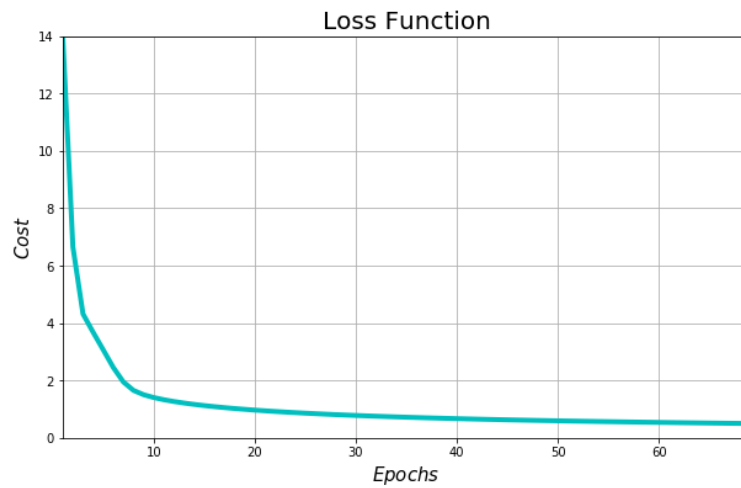
(ب) روش دلتا





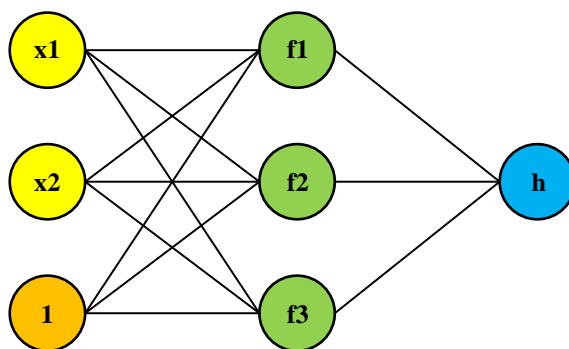
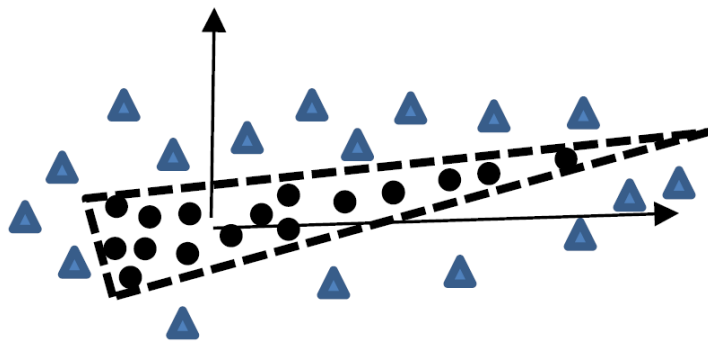
(ج) تابع سیگموئید دوقطبی



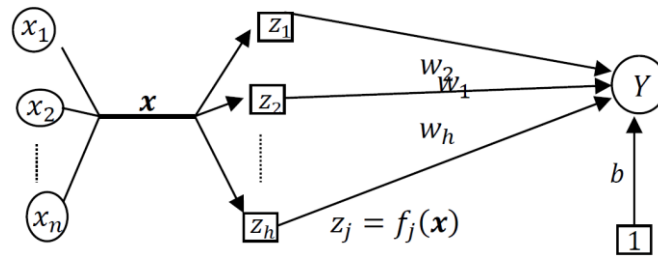


با توجه به نتایج به دست آمده متوجه می شویم که روش های (الف) و (ب) در مقایسه با روش (ج) به epoch های کمتری برای رسیدن به وزن های مطلوب برای شبکه عصبی نیاز دارد. همچنین در مقایسه ی دو روش (الف) و (ب) نیز می توان گفت که معمولاً روش (الف) سریعتر به جواب می رسد.

۴. الگوریتم آموزشی برای کشف سه خطی و عمل طبقه بندی

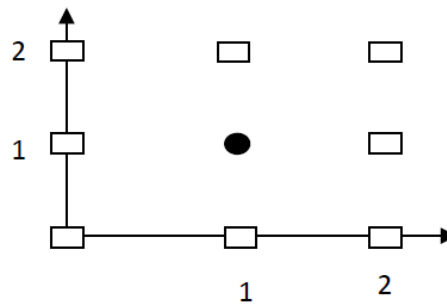


۵. استفاده از نگاشت برای ساده کردن فضای ویژگی ها



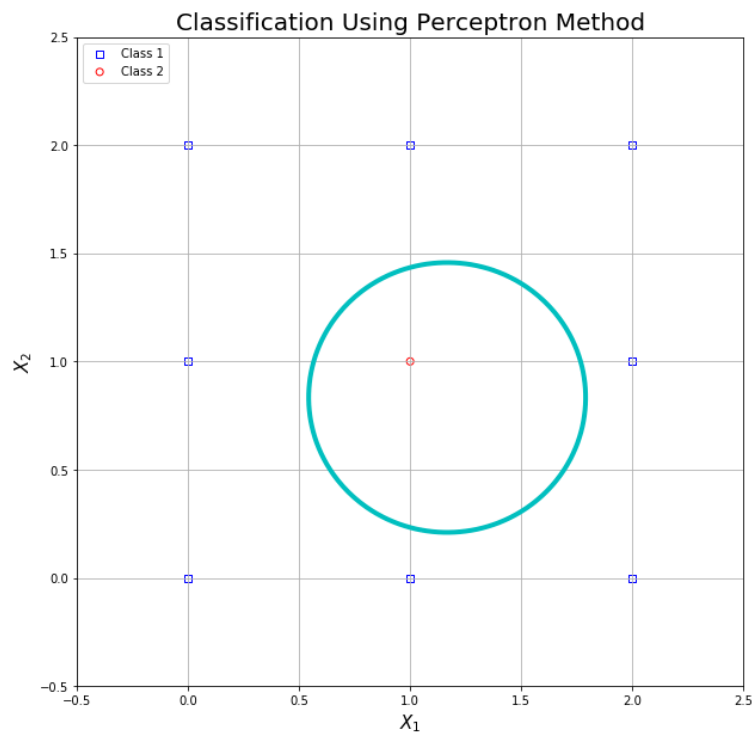
در این تمرین در ابتدا مطابق با محل قرارگیری داده‌های ورودی، آن‌ها را به فضای دیگری منتقل می‌کنیم تا بتوان ابرصفحه‌ی جداساز را به راحتی پیدا کرد. سپس با توجه به تعداد ورودی‌های جدید ایجاد شده، شبکه‌ی عصبی مناسب را طراحی می‌کنیم.

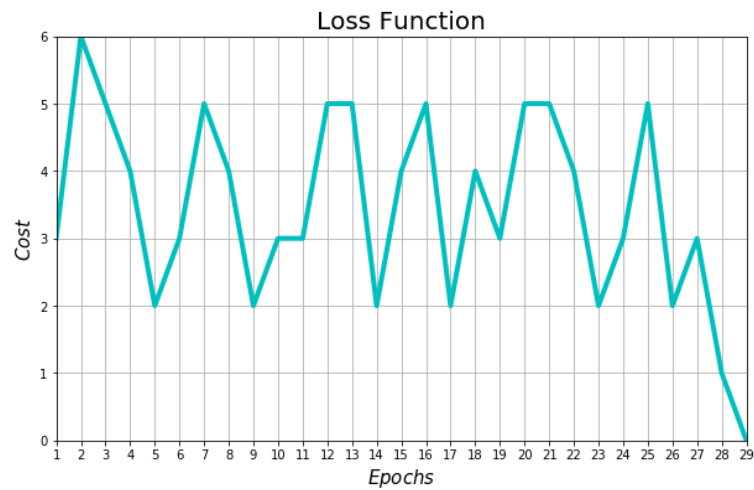
در قسمت (الف) با انجام تبدیل‌های موجود در شکل زیر ورودی‌های شبکه‌ی عصبی را به ۴ عدد می‌رسانیم.



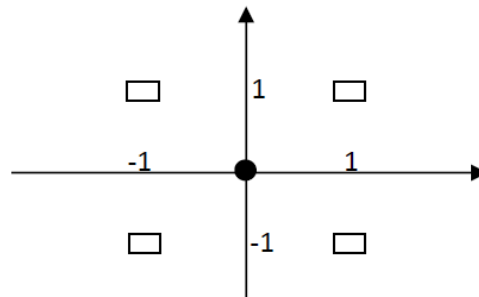
$$z_1 = x_1^2 \quad z_2 = x_2^2 \quad z_3 = x_1 \quad z_4 = x_2$$

نتایج مربوط به این قسمت در نمودارهای زیر آورده شده است. کد مربوط به این بخش در فایل HW1-5-A موجود است.



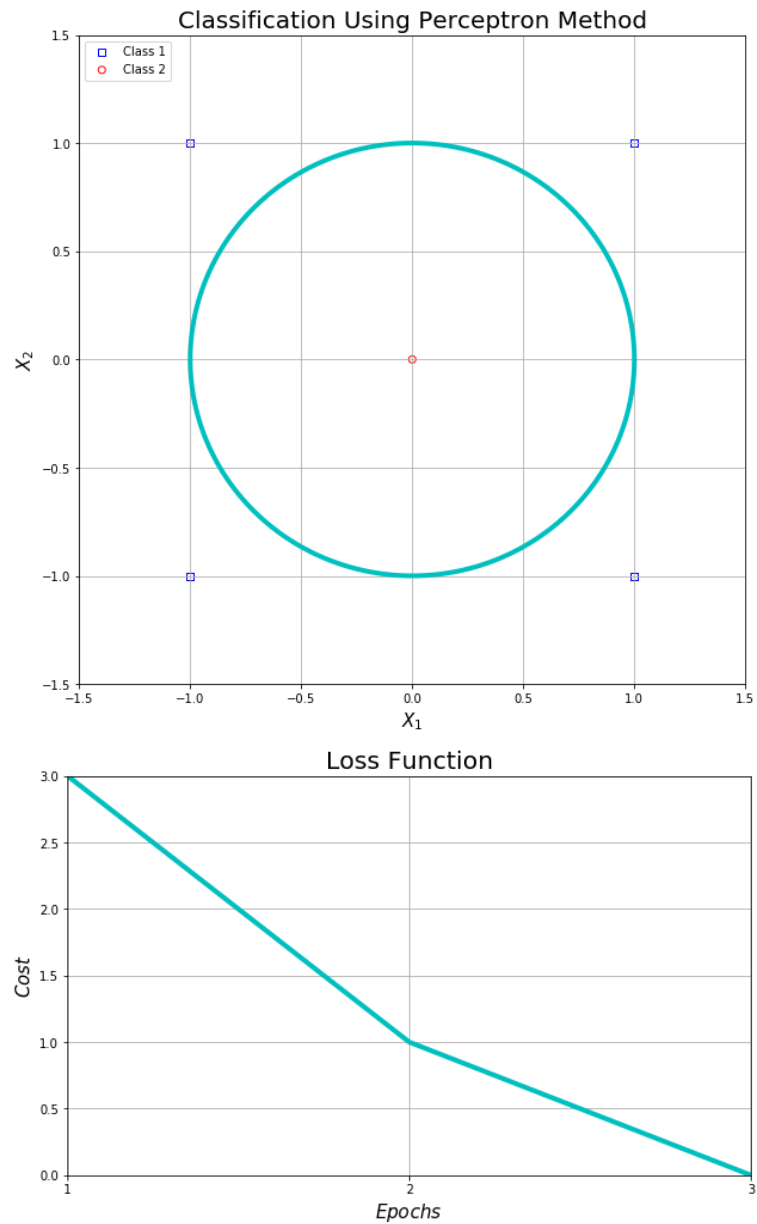


در قسمت (ب) با انجام تبدیل‌های موجود در شکل زیر ورودی‌های شبکه‌ی عصبی را به ۲ عدد می‌رسانیم.

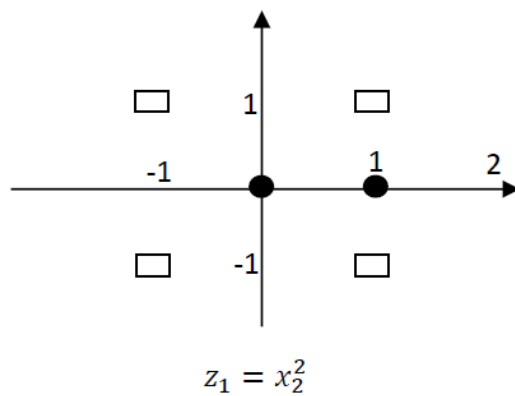


$$z_1 = x_1^2 \quad z_2 = x_2^2$$

نتایج مربوط به این قسمت در نمودارهای زیر آورده شده است. کد مربوط به این بخش در فایل HW1-5-B موجود است.



در قسمت (ج) با انجام تبدیل‌های موجود در شکل زیر ورودی‌های شبکه‌ی عصبی را به ۱ عدد می‌رسانیم.



نتایج مربوط به این قسمت در نمودارهای زیر آورده شده است. کد مربوط به این بخش در فایل HW1-5-C موجود است.

