به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

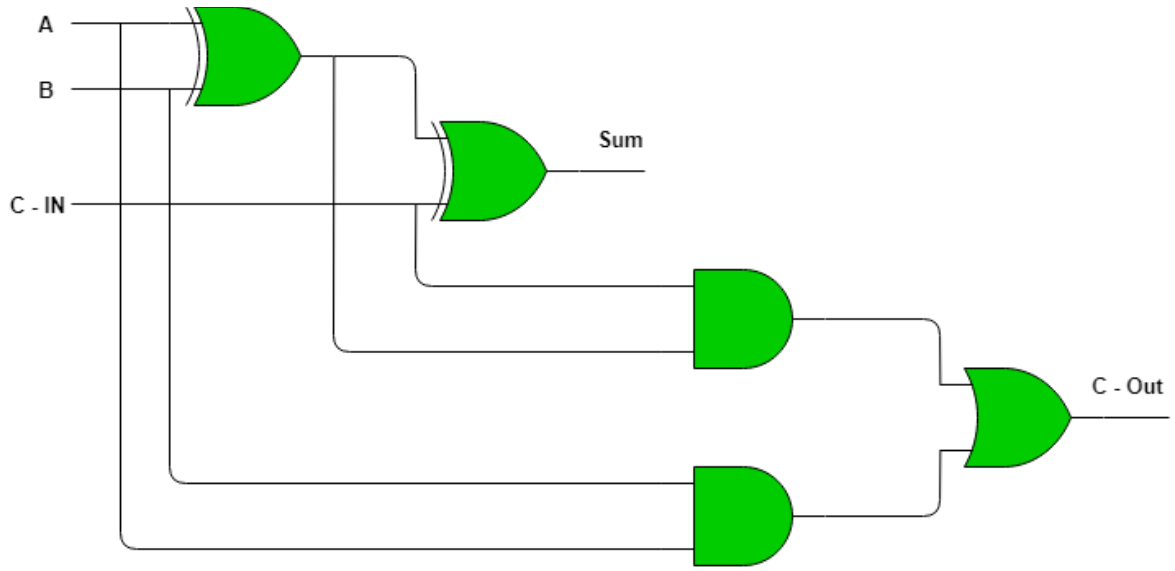
تمرین سری ۱

نام و نام خانوادگی	سیاوش شمس
شماره دانشجویی	810197644
تاریخ ارسال گزارش	۰۰/۱۲/۲۰

فهرست گزارش سوالات

سوال ۱ – Mcculloch Pitts

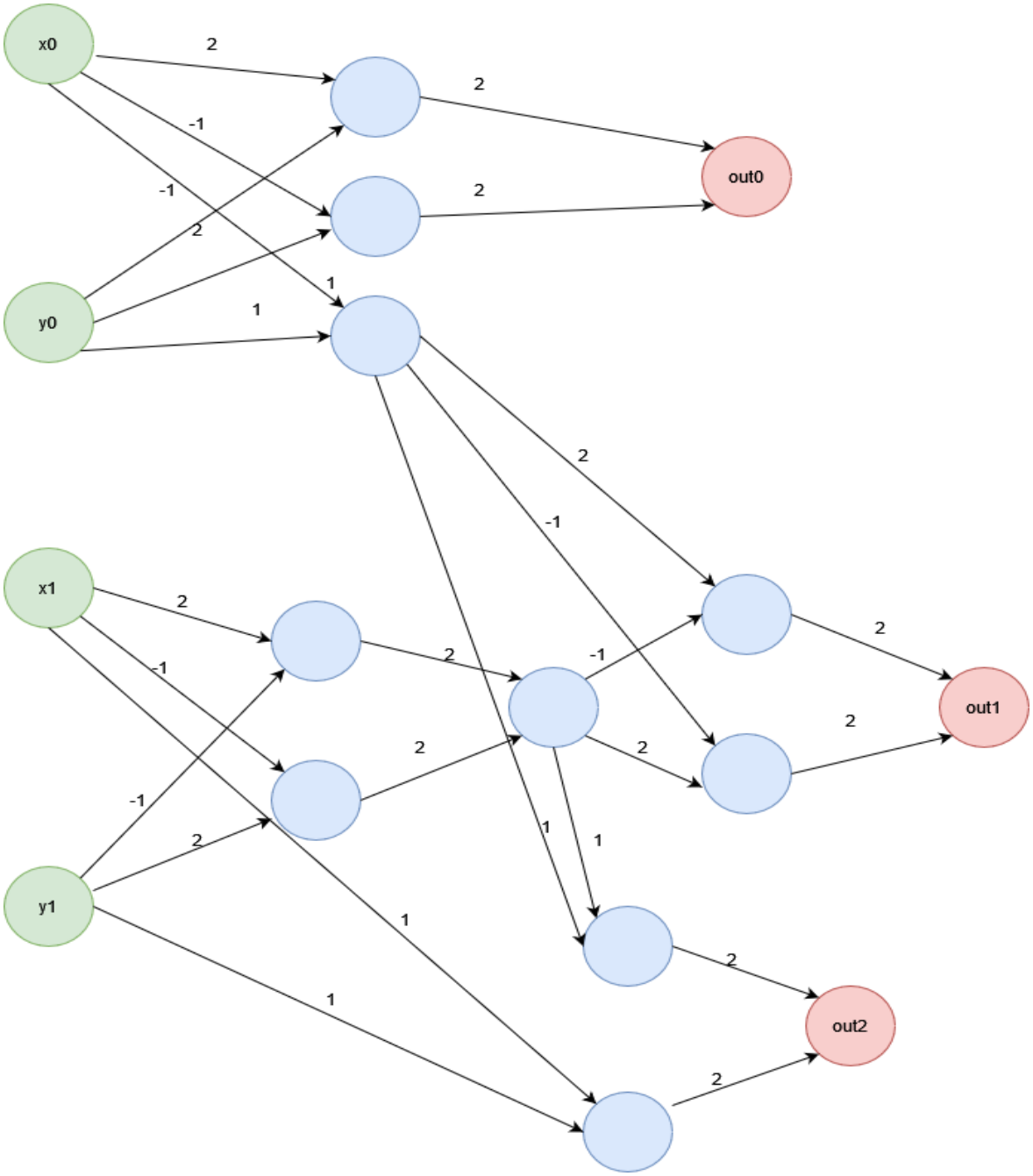
در این سوال قصد داریم به کمک نورون های Mcculloch Pitts یک Full Adder(FA) دو بیتی طراحی کنیم، مدل FA یک بیتی با استفاده از دروازه های منطقی در شکل ۱ نشان داده شده است.



شکل ۱- مدل مدار منطقی Full Adder

یک FA دو بیتی از کنار هم قرار دادن دو تا FA تک بیتی حاصل می شود، در ادامه به کمک نورون های Mcculloch Pitts و تعیین وزن ها، FA دو بیتی را پیاده سازی می کنیم

شبکه متشکل از نورون های Mcculloch Pitts برای پیاده سازی یک Full Adder دو بیتی، وزن ها در شکل ۱ مشخص شده اند و ورودی های با رنگ سبز x0 و y0 به ترتیب بیانگر بیت اول و دو ورودی x1 و y1 بیانگر بیت دوم دو عدد ورودی هستند. همچنین خروجی ها با رنگ قرمز نشان داده شده و out0، out1، out2 به ترتیب بیت های اول، دوم و سوم خروجی هستند. سایر نورون ها با رنگ آبی نمایش داده شده اند و آستانه فعال شدن برای همه نورون ها برابر ۲ است



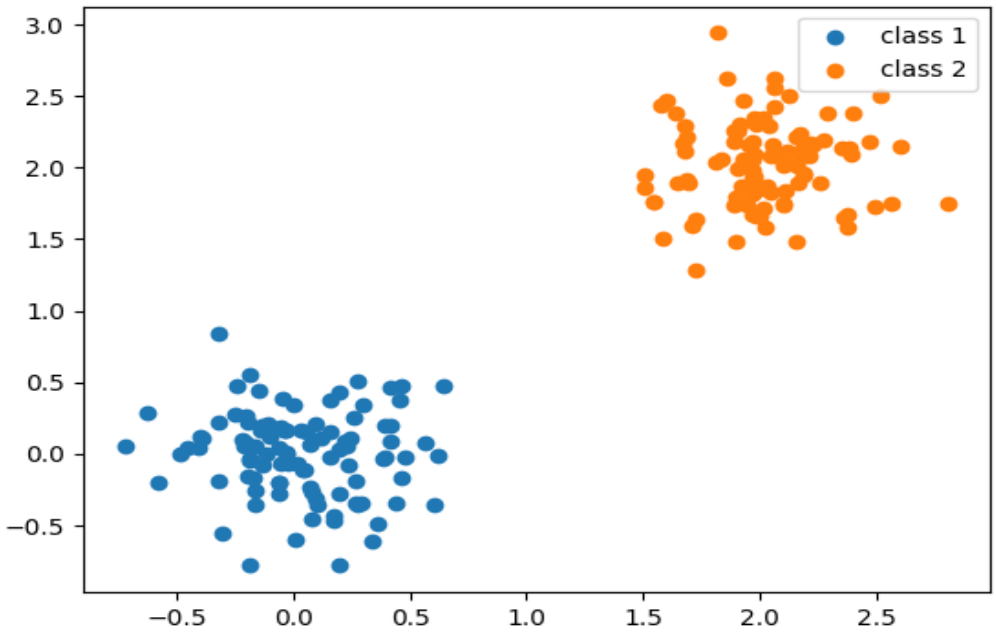
شکل ۲- دیاگرام شبکه Full Adder

```
The result of adding binary numbers 11 and 11 equals: 110
The result of adding binary numbers 11 and 10 equals: 101
The result of adding binary numbers 11 and 01 equals: 100
The result of adding binary numbers 11 and 00 equals: 011
The result of adding binary numbers 10 and 11 equals: 101
The result of adding binary numbers 10 and 10 equals: 100
The result of adding binary numbers 10 and 01 equals: 011
The result of adding binary numbers 10 and 00 equals: 010
The result of adding binary numbers 01 and 11 equals: 100
The result of adding binary numbers 01 and 10 equals: 011
The result of adding binary numbers 01 and 01 equals: 010
The result of adding binary numbers 01 and 00 equals: 001
The result of adding binary numbers 00 and 11 equals: 011
The result of adding binary numbers 00 and 10 equals: 010
The result of adding binary numbers 00 and 01 equals: 001
The result of adding binary numbers 00 and 00 equals: 000
```

شکل ۳- خروجی به ازای همه حالت های ورودی

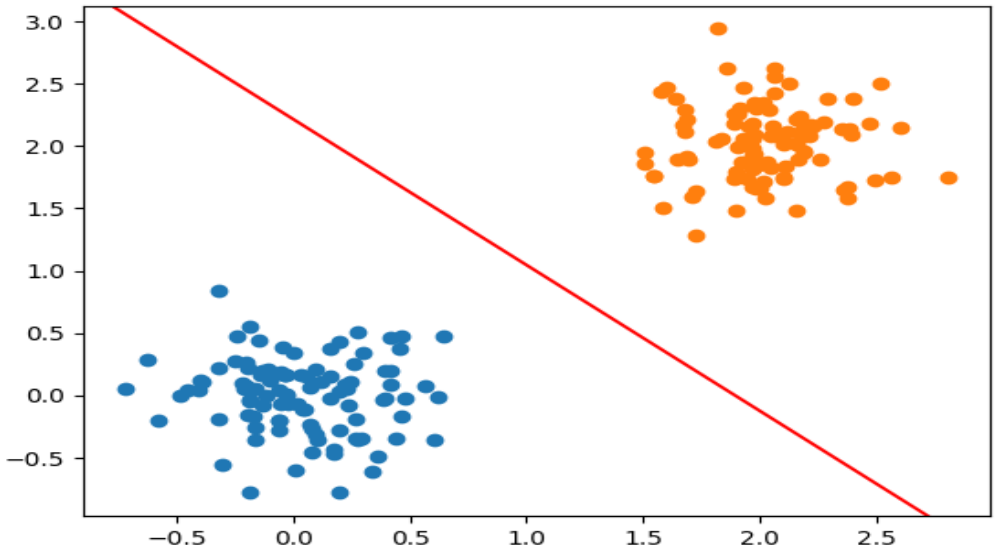
سوال ۲ – Adaline

الف)



شکل ۴- نمودار داده های دسته ۱ و ۲

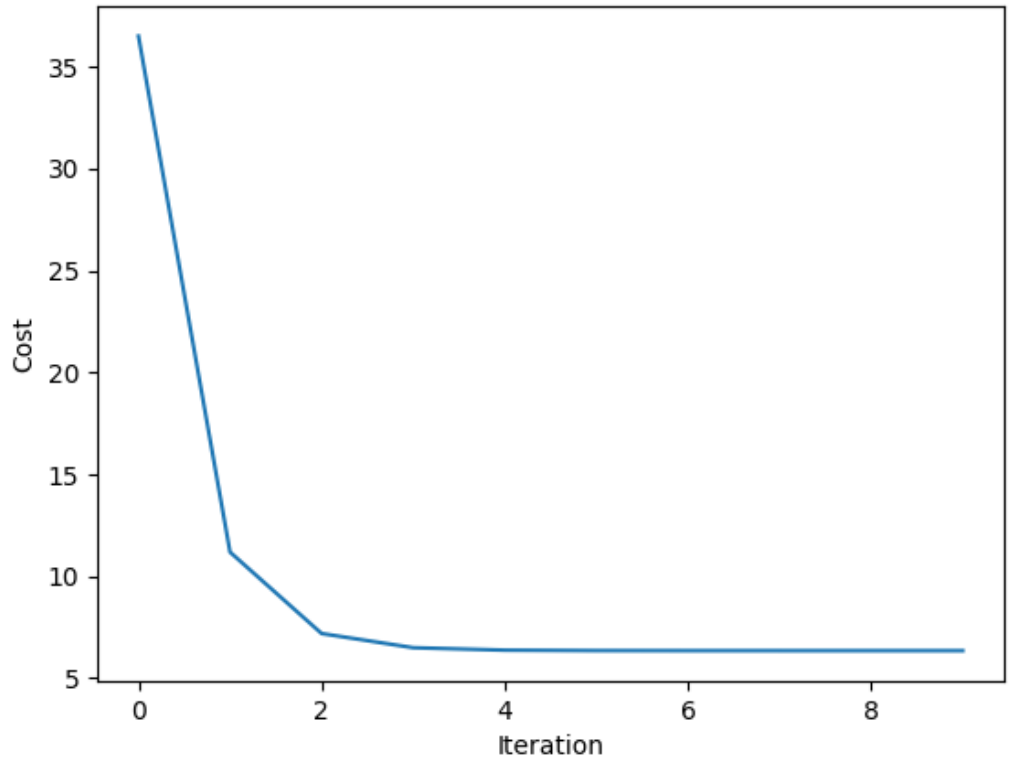
ب)



شکل ۵- نمودار داده های دسته ۱ و ۲ به همراه خط جداکننده به دست آمده از Adaline

```
[ 0.9826567  -0.51845879 -0.44342787]
```

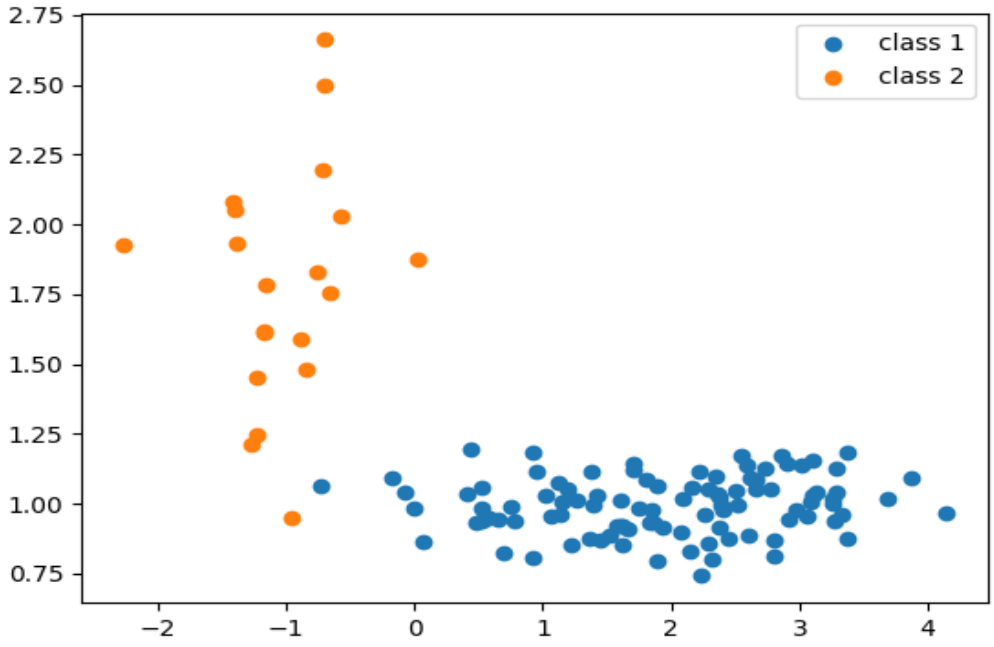
شکل ۶- وزن های به دست آمده از Adaline



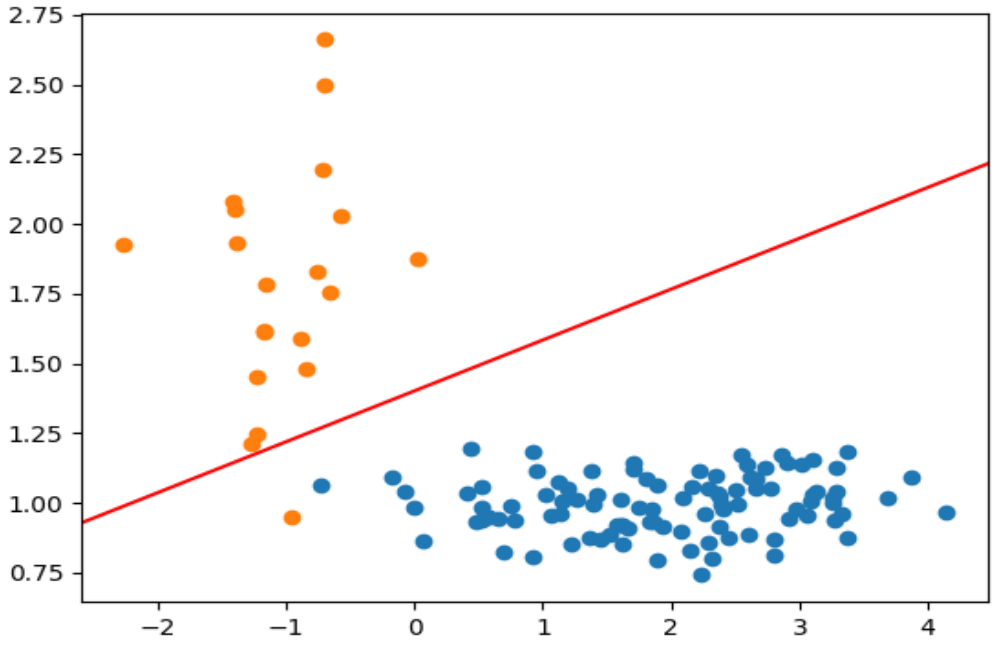
شکل ۷- خطای مدل به ازای هر تکرار

می بینیم که خط به دست آمده از الگوریتم Adaline به خوبی داده های دو کلاس را از هم جدا می کند. دلیل آن این است داده های دو به وسیله یک خط جدایی پذیر هستند و فاصله کافی دارند.

ج)



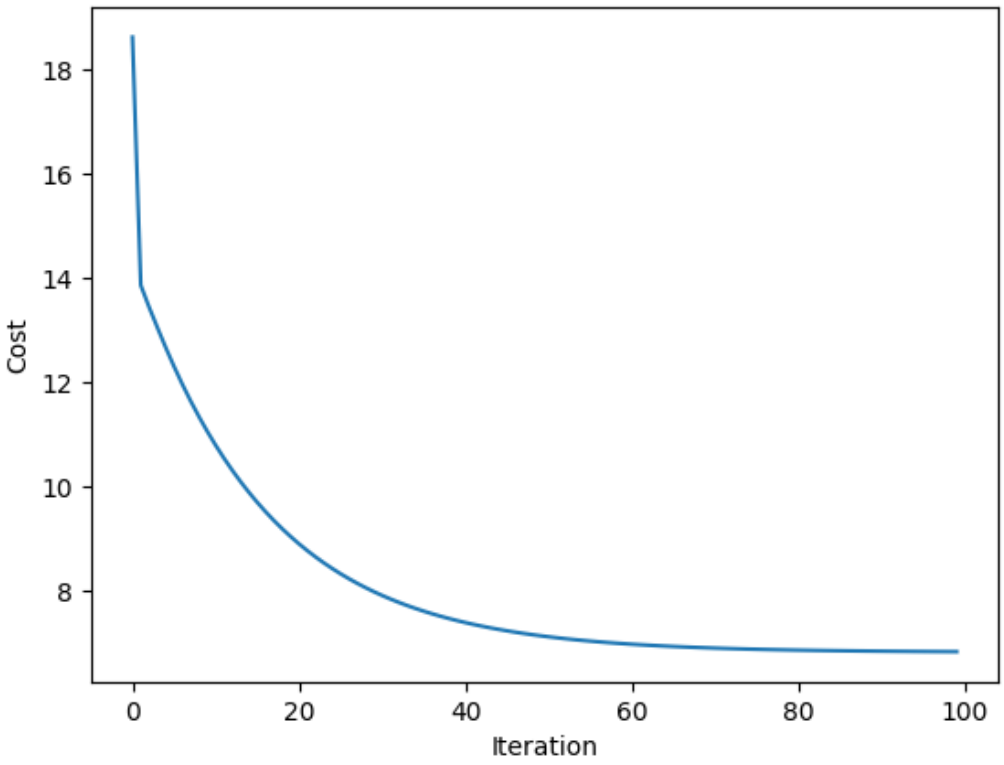
شکل ۸- نمودار داده های دسته ۱ و ۲



شکل ۹- نمودار داده های دسته ۱ و ۲ به همراه خط جداکننده به دست آمده از Adaline

```
[ 1.79123891  0.22633981 -1.30530315]
```

شکل ۱۰- وزن های به دست آمده از Adaline



شکل ۱۱- خطای مدل به ازای هر تکرار

می بینیم که خط به دست آمده از الگوریتم Adaline تا حد خوبی داده های دو کلاس را از هم جدا می کند، اما به دلیل اینکه داده های دو کلاس کاملا با یک خط جدایی پذیر نیستند دقت به دست آمده صد در صد نیست (یکی از داده ها اشتباه طبقه بندی می شود). و همچنین به دلیل نزدیک بودن داده ها ی دو کلاس به هم، در این حالت نمودار هزینه بر حسب تکرار با شیب کمتری کاهش می یابد.

سوال ۳ – Madaline

الف)

توضیح الگوریتم MRI:

۱. مقدار دهی اولیه وزن ها (v, w, b) و تعیین نرخ یادگیری

۲. برای هر مجموعه داده به همراه برچسب آن نتیجه حاصل از اعمال فعالساز ورودی ها را به دست وی آوریم $x = s$

۳. به دست آوردن مجموع ضرب وزن ها در ورودی ها به علاوه بایاس (net)

$$z_{in} = wx + b$$

۴. به دست آوردن خروجی پس از اعمال تابع فعالساز زیر به net ها

$$z = f(z_{in}) = \begin{cases} 1 & if\ z \geq 0 \\ -1 & if\ z < 0 \end{cases}$$

۵. به دست آوردن مجموع ضرب وزن ها در نتایج مرحله ۴ به علاوه بایاس و خروجی آن بعد از اعمال فعالساز

$$y_{in} = vz + b$$
$$y = f(y_{in})$$

۶. اگر لیبل پیشبینی شده در مرحله ۵ (y) با لیبل اصلی (t) برابر نبود وزن ها را به صورت زیر آپدیت می کنیم

$$\begin{cases} w_k(new) = w_k(old) + \alpha(-1 - z_{in_k})x & if\ t = -1\ and\ z_{in_k} > 0 \\ b_k(new) = b_k(old) + \alpha(-1 - z_{in_k}) & if\ t = -1\ and\ z_{in_k} > 0 \end{cases}$$

$$\begin{cases} w_j(new) = w_j(old) + \alpha\left(1 - z_{in_j}\right)x & if\ t = 1\ and\ z_{in_j} = \min\left(z_{in}\right) \\ b_j(new) = b_j(old) + \alpha\left(1 - z_{in_j}\right) & if\ t = 1\ and\ z_{in_j} = \min\left(z_{in}\right) \end{cases}$$

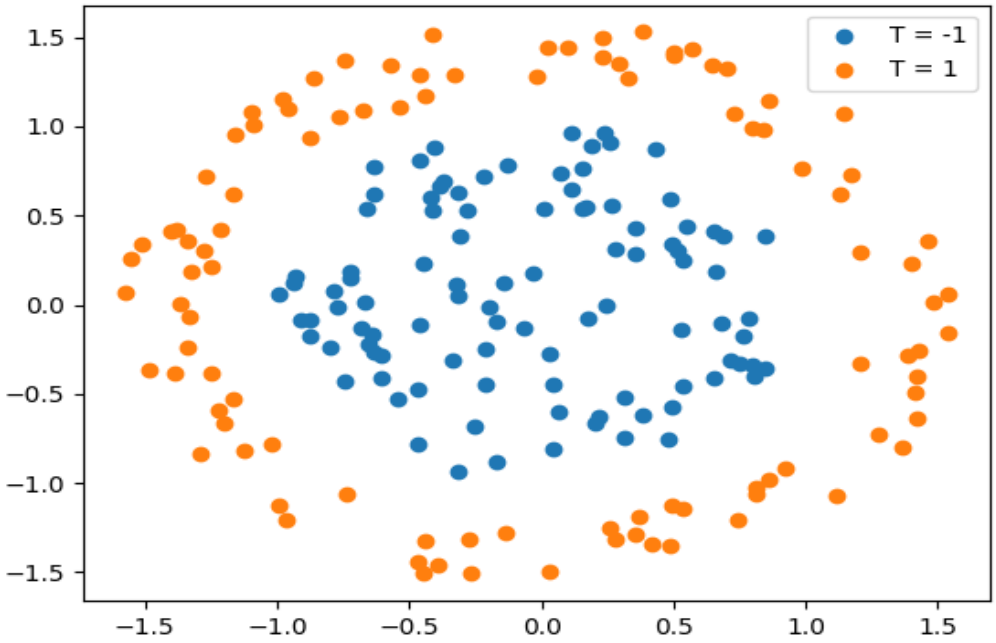
۷. اگر وزن ها تغییر نکردند یا تعداد تکرار ها از حد تعیین شده بیشتر شد متوقف می شویم در غیر این از مرحله ۲ به بعد را تکرار می کنیم.

نکته: در عبارت های بالا متغیر ها از جنس بردار و ماتریس هستند مثلا

$$w = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{33} \end{pmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

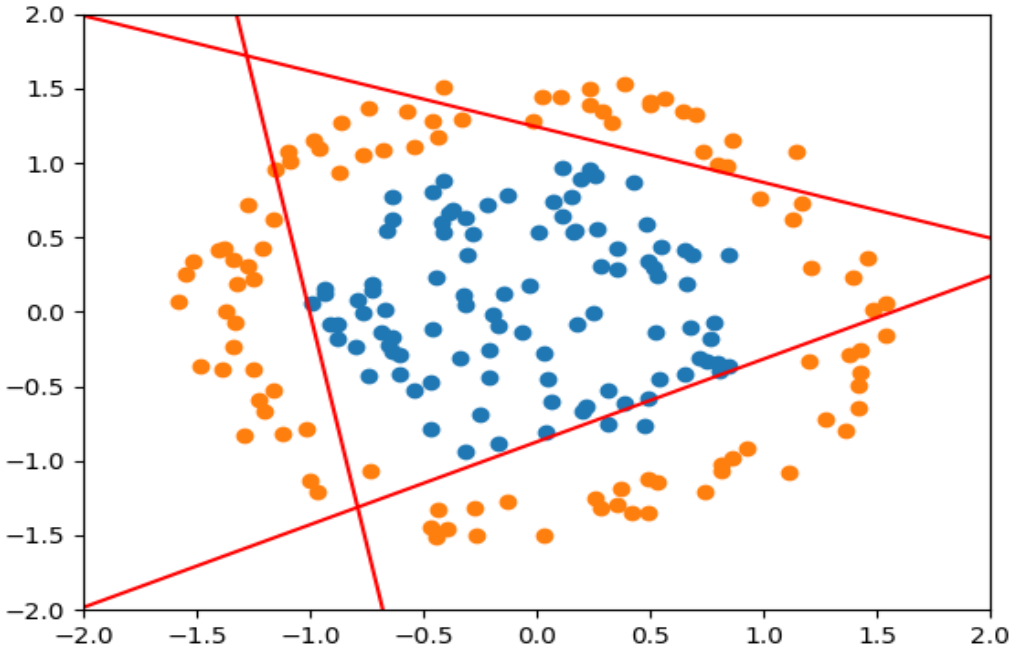
به طور کلی الگوریتم فوق سعی می کند وقتی طبقه بندی به درستی انجام نمی شود وزن ها را طوری آپدیت کند تا خطا کمتر شود.

ب)



شکل ۱۲- نمودار پراکندگی نقاط

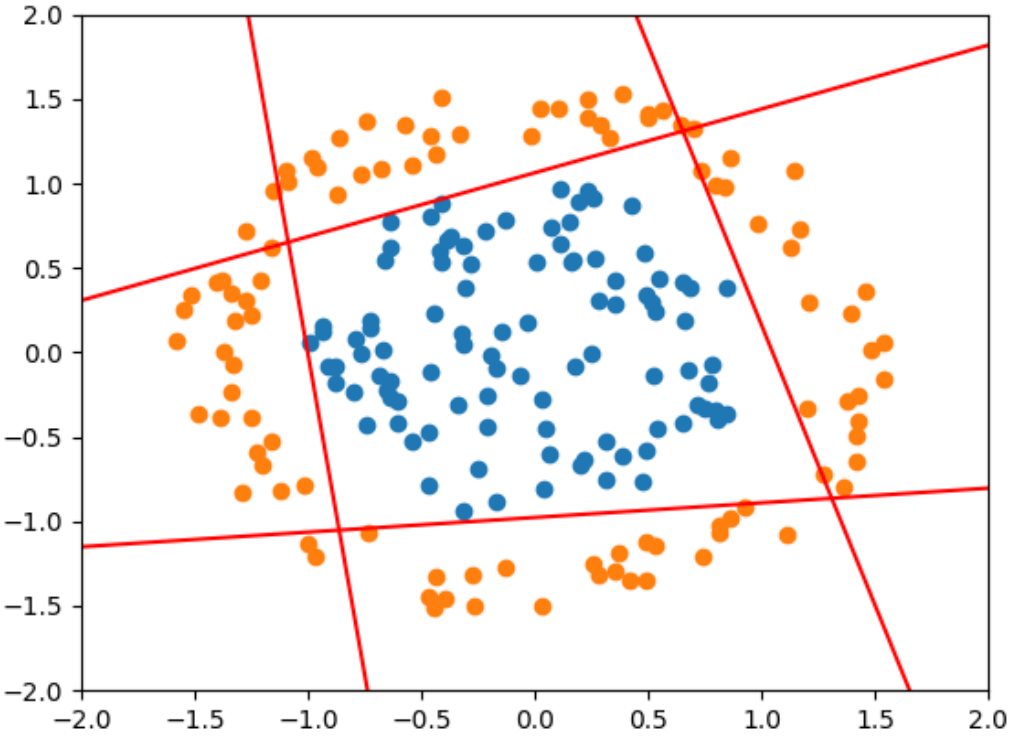
ج)



شکل ۱۳- نمودار پراکندگی نقاط به همراه ۳ خط جدا کننده تعلیم داده شده با الگوریتم Madaline

```
Total iterations with 3 lines: 161
Accuracy of prediction is: 0.8743718592964824
```

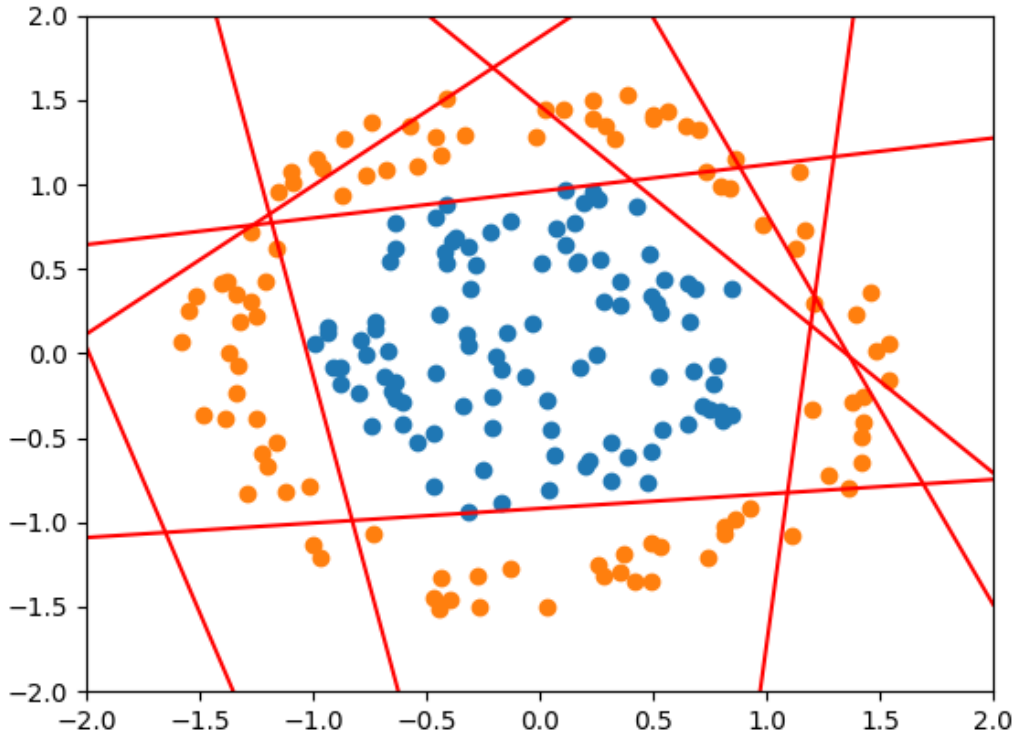
شکل ۱۴- تعداد تکرار و دقت مدل



شکل ۱۵- نمودار پراکندگی نقاط به همراه ۴ خط جدا کننده تعلیم داده شده با الگوریتم Madaline

```
Total iterations with 4 lines: 101
Accuracy of prediction is: 1.0
```

شکل ۱۶- تعداد تکرار و دقت مدل



شکل ۱۷- نمودار پراکندگی نقاط به همراه ۸ خط جدا کننده تعلیم داده شده با الگوریتم Madaline

```
Total iterations with 8 lines: 22
Accuracy of prediction is: 1.0
```

شکل ۱۸- تعداد تکرار و دقت مدل

د)

با مقایسه نتایج بالا می بینیم که افزایش تعداد نورون ها باعث کاهش تعداد تکرار لازم الگوریتم برای همگرایی و همچنین افزایش دقت می شود، دلیل آن این است که با تعداد نورون (خطوط) بیشتر آزادی عمل بیشتری برای مرز بندی بین دو کلاس داریم. همچنین در قسمت ۸ نورونه می بینیم که تنها با ۵ خط جداسازی انجام شده است و چون خطا بعد آن صفر می شود وزن سایر خط ها آپدیت نمی شود و به عبارتی به آن ها نیاز نیست.

شرط توقف الگوریتم این است که اگر در ۱۰ تکرار متوالی وزن ها کمتر از 0.001 تغییر کنند اجرای الگوریتم متوقف شود.

سوال ۴ – Perceptron

از الگوریتم مربوط به آموزش پرسپترون در صفحه ۶۱ کتاب مرجع استفاده می کنیم.

تکرار ۱

$$y\_in_1 = b^1 + w^1x = -0.7 + 0.2(0) + 0.7(1) + 0.9(1) = 0.9$$

$$y_1 = sign(0.9) = 1$$

$$w_1^2 = w_1^1 + \alpha tx = 0.2 + 0.3(-1)(0) = 0.2$$

$$w_2^2 = w_2^1 + \alpha tx = 0.7 + 0.3(-1)(1) = 0.4$$

$$w_3^2 = w_3^1 + \alpha tx = 0.9 + 0.3(-1)(1) = 0.6$$

$$b^2 = b^1 + \alpha t = -0.7 + 0.3(-1) = -1$$

تکرار ۲

$$y\_in_2 = b^2 + w^2x = -1 + 0.2(0) + 0.4(1) + 0.6(1) = 0$$

$$y_2 = sign(0) = 0$$

$$w_1^3 = w_1^2 + \alpha tx = 0.2 + 0.3(-1)(0) = 0.2$$

$$w_2^3 = w_2^2 + \alpha tx = 0.4 + 0.3(-1)(1) = 0.1$$

$$w_3^3 = w_3^2 + \alpha tx = 0.6 + 0.3(-1)(1) = 0.3$$

$$b_1^3 = b_1^2 + \alpha t = -1 + 0.3(-1) = -1.3$$

تکرار ۳

$$y\_in_3 = b^3 + w^3x = -1.3 + 0.2(0) + 0.1(1) + 0.3(1) = -0.9$$

$$y_3 = sign(-0.9) = -1$$

وزن ها تغییر نمی کند چون ورودی درست طبقه بندی شده است.