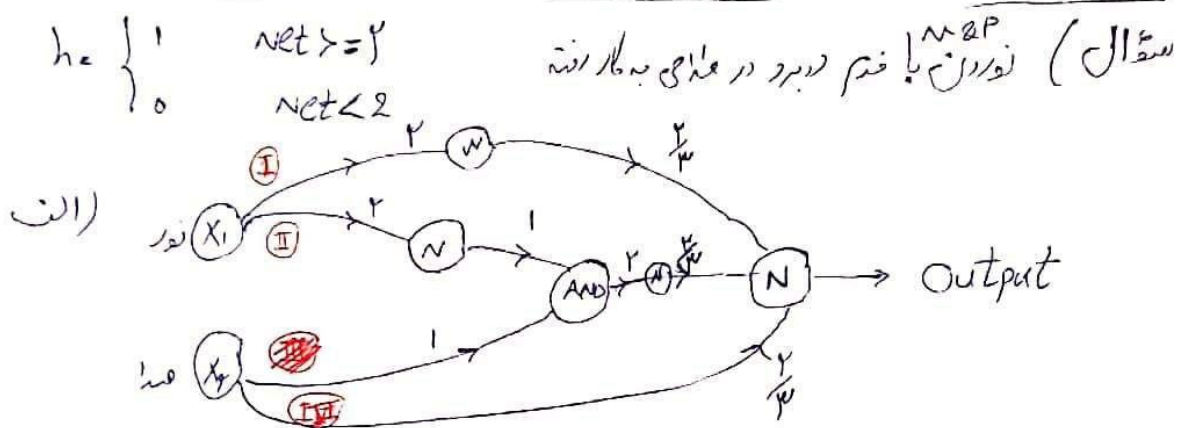


فهرست گزارش سوالات

2.....	سوال 1 – McCulloch_Pitts
3.....	سوال ۲ – Perceptron
7.....	سوال 3 – Adaline
10.....	سوال 4 – Madaline

سوال 1 - McCulloch_Pitts



ب) $X_1(N-4) \& X_2(N-3) \& X_1(N-2) \& X_2(N-1)$

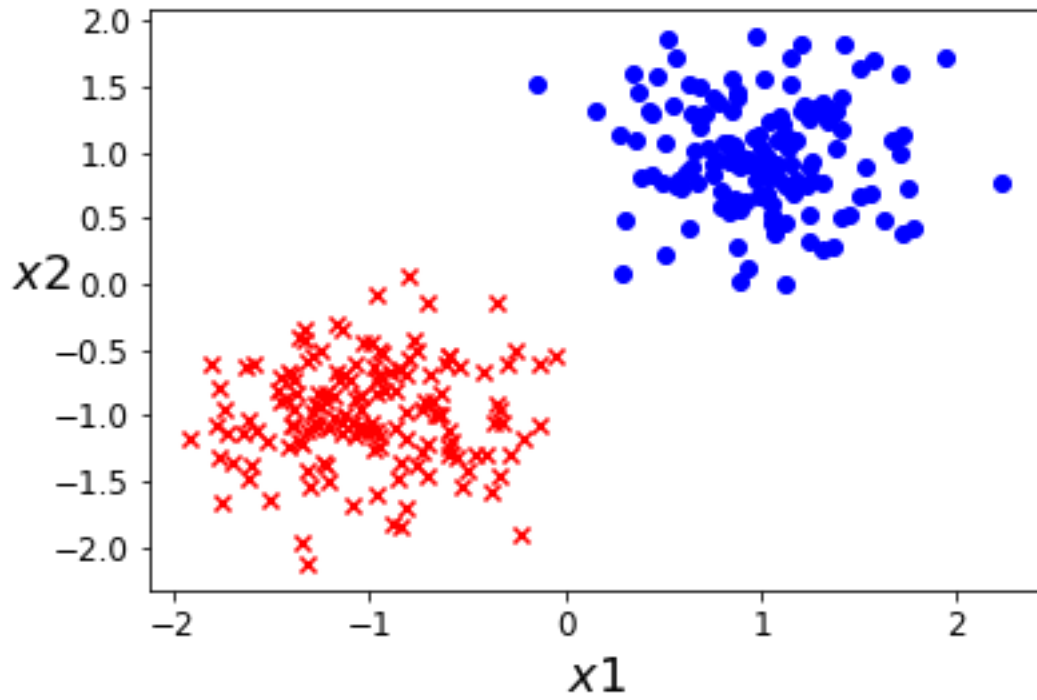
که به فتم ترکیب اول در ④ در ② در ④ در ④

$$Output = X_1(0) \& X_2(1) \& X_1(2) \& X_2(3)$$

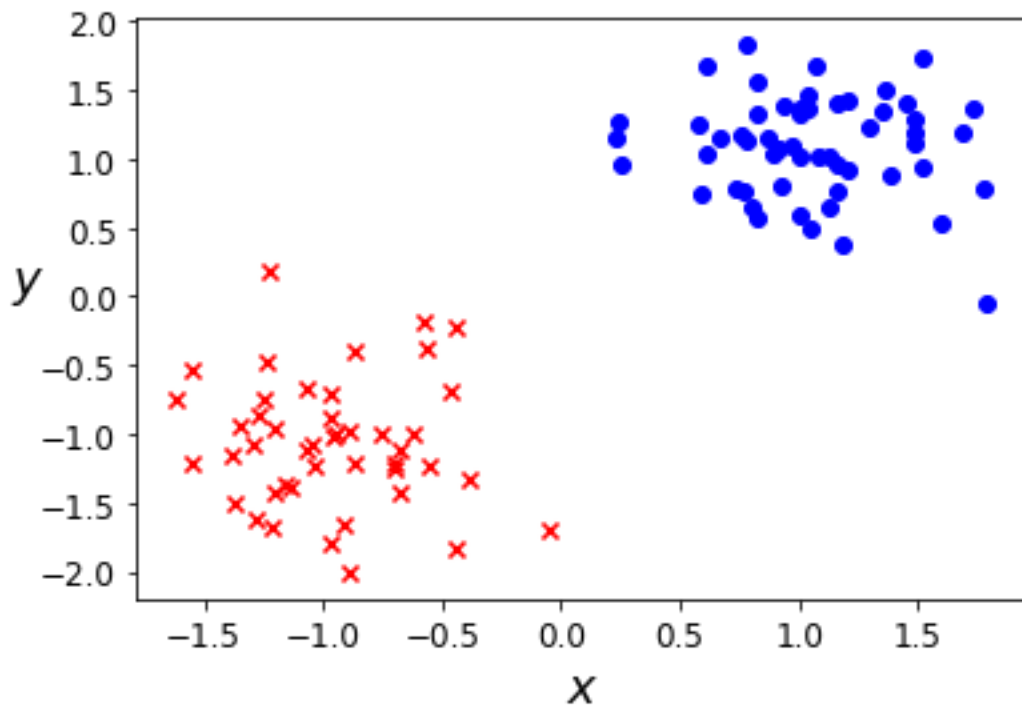
ج) فتمی کنیم که با ترکیب پیم ، دومی X_1 در همان لحظه در X_2 در لحظه بعدی فعال نگردد
 به کمک حس های شعری ، در ادامه توسط میم ② ، این نتیجه ذخیره شده در X_2 بعدی
 ابتدا AND می گردد ، سیر I ، در میب دم را ذخیره کرده در نهایت سیم ③
 سیمای میب دم را با همای سیمای رسیده AND کرده و فتمی می گیریم .

سوال ۲ – Perceptron

الف) بعد از جداسازی داده های تست و ترین نتیج زیر بدست آمد.

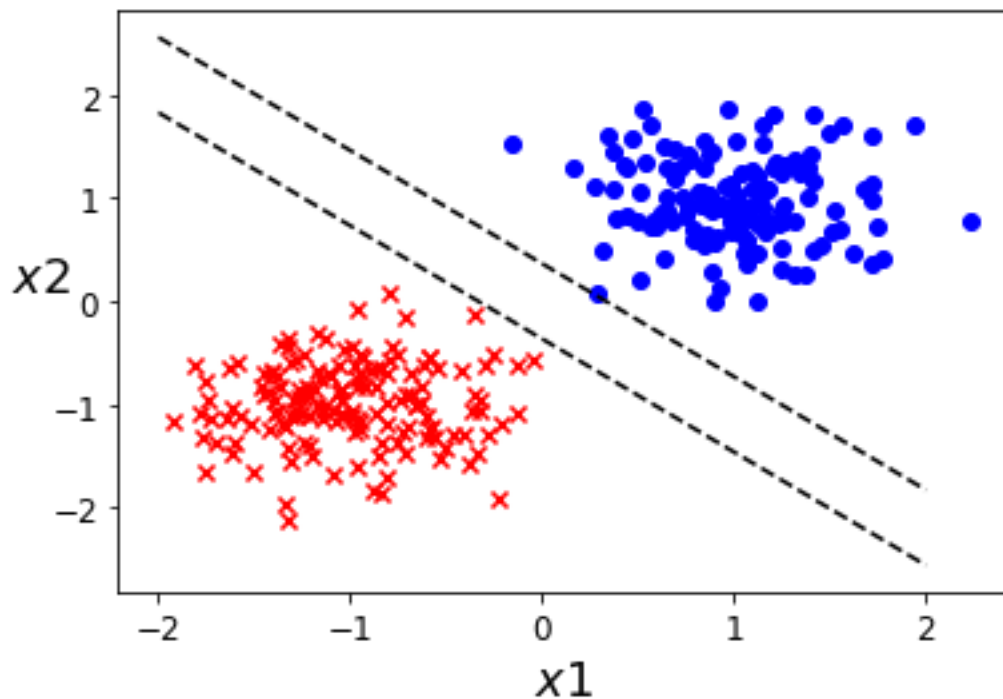


شکل ۱) داده های ترین در دو کلاس قرمز و آبی



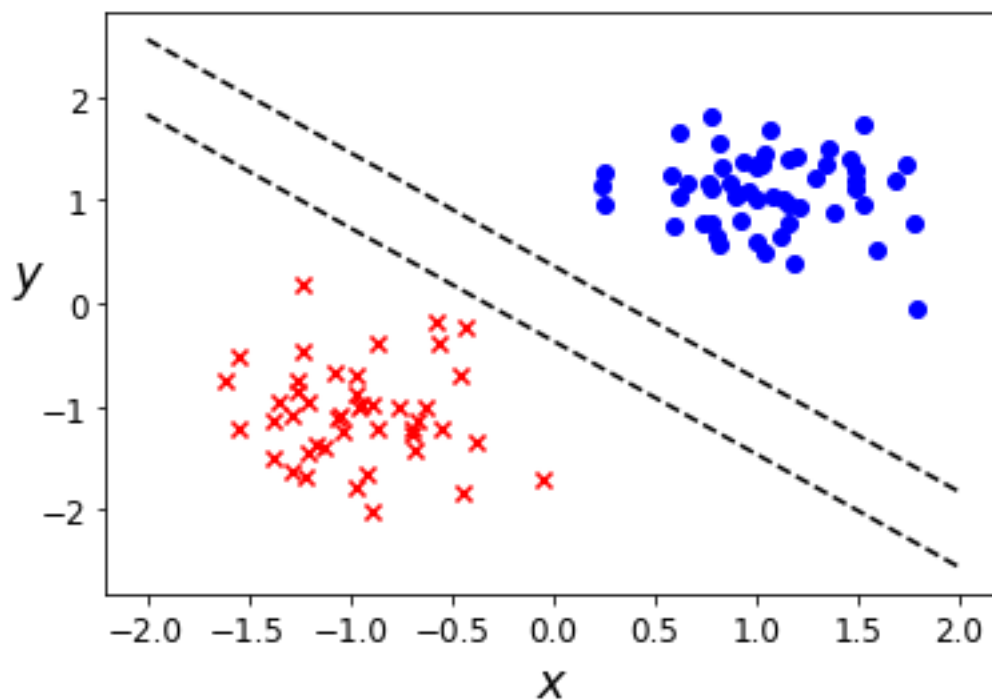
شکل ۲) داده های تست در دو کلاس قرمز و آبی

ب) داده های ترین را توسط یک نورون با قاعده پرسپترون با دو خط موازی جدا سازی کردیم برای این منظور $\text{threshold}=0.2$ و لرنینگ ریت $\text{Alpha}=0.1$ در نظر گرفته شد، عملیات یادگیری در ۷ اپیاک صورت می گیرد. شکل نهایی در شکل ۳ آورده شده است:



شکل ۳) جداسازی داده های ترین به وسیله ی نورون با قاعده پرسپترون

ج) با رسم داده های تست و خط جداساز داریم :

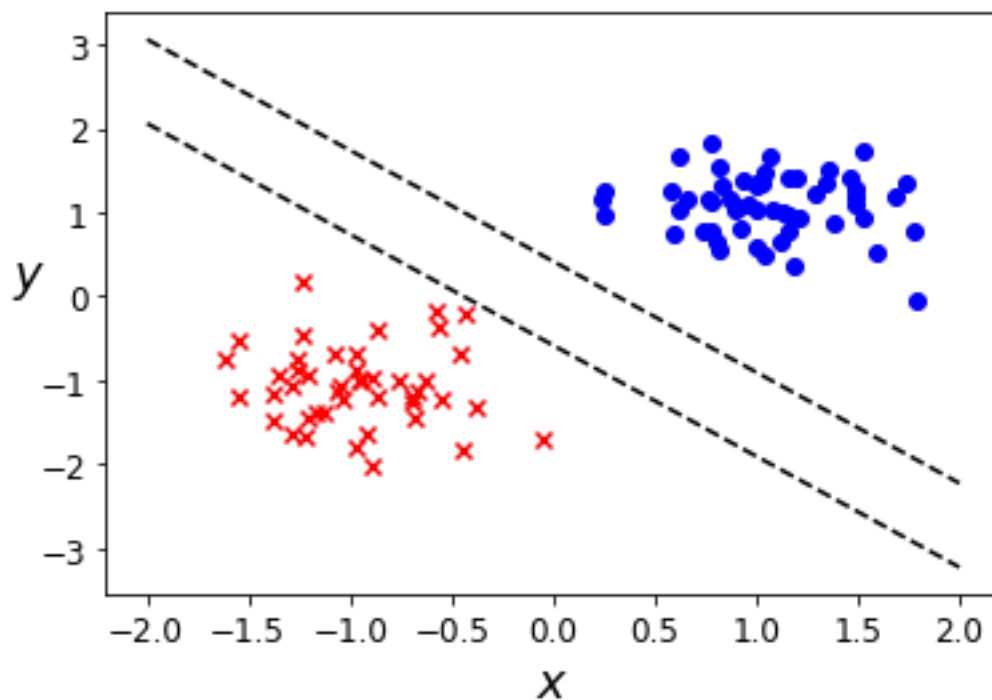


شکل ۴) جداسازی داده های تست توسط نورون با قاعده پرسپترون

دقت این جداساز برای داده ی تست ما ۱۰۰ درصد می باشد و خطایی ندارد

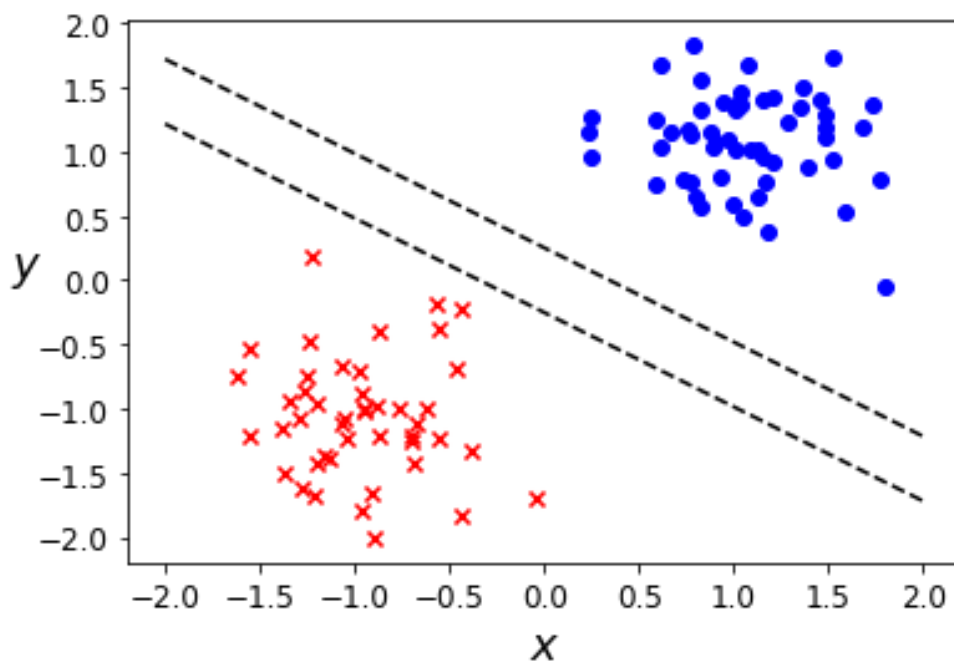
د) این بار یک بار با $\text{threshold}=0.6$ ، و بار دیگر با $\text{threshold}=0.05$ نتایج را برای داده ی تست مقایسه می کنیم.

1) $\text{threshold}=0.6$: در ۱۴ ایپاک به جواب بهینه خود می رسیم ، در این حالت برخی داده ها کمی به مرز تصمیم گیری نزدیک شده اما هم چنان دقت ۱۰۰ درصد داریم.



شکل 5) جداسازی داده های تست توسط نورون با قاعده پرسپترون با $\text{threshold}=0.6$

2) $\text{threshold}=0.05$: در تنها ۲ ایپاک به جواب بهینه خود می رسیم ، در این حالت بخش عدم تصمیم گیری کوچک تر شده ، اما هم چنان دقت ۱۰۰ درصد داریم.



شکل 6) جداسازی داده های تست توسط نورون با قاعده پرسپترون با $\text{threshold}=0.05$

با توجه به نتایج بدست آمده در این داده ها آستانه تاثیری بر روی دقت نداشت (به علت اینکه توسط w_2 در نهایت به نوعی scale شده است.) اما ایپاک لازم برای انجام یادگیری با زیاد شدن آن افزایش پیدا کرد.

سوال 3 – Adaline

الف) به عنوان شباهت ساختار شبکه هر دو مدل یکسان می باشد،

$$\text{net} = \sum_{i=1}^n w_i x_i + b$$

$$h = f(\text{net}) = \begin{cases} +1 & \text{net} \geq 0 \\ -1 & \text{net} < 0 \end{cases}$$

تفاوت : قوانین آپدیت سازی وزن ها متفاوت بوده و مطابق رابطه فوق آستانه برای شبکه Adaline صفر می باشد(بر خلاف perceptron که دارای Threshold بود).

ب) در این قسمت پیاده سازی شبکه و بدست آوردن وزن ها یک بار برای شکل ۱ و یکبار برای شکل ۲ صورت گرفته است.

الگوریتم استفاده شده به شرح زیر می باشد:

(0) وزن ها و بایاس $= 0.1$ و لرنینگ ریت $= 0.1$ مقدار دهی اولیه شده است.

(1) در هر مرحله برای هر (s, t) مراحل بعدی را انجام می دهیم.

(2) ورودی های z را اعمال می کنیم

(3) خروجی h شبکه را پیدا کرده

(4) وزن ها را آپدیت می کنیم:

$$w_i^+ = w_i^- + \alpha(t - net)x_i$$

$\alpha > 0$: α small enough learning rate

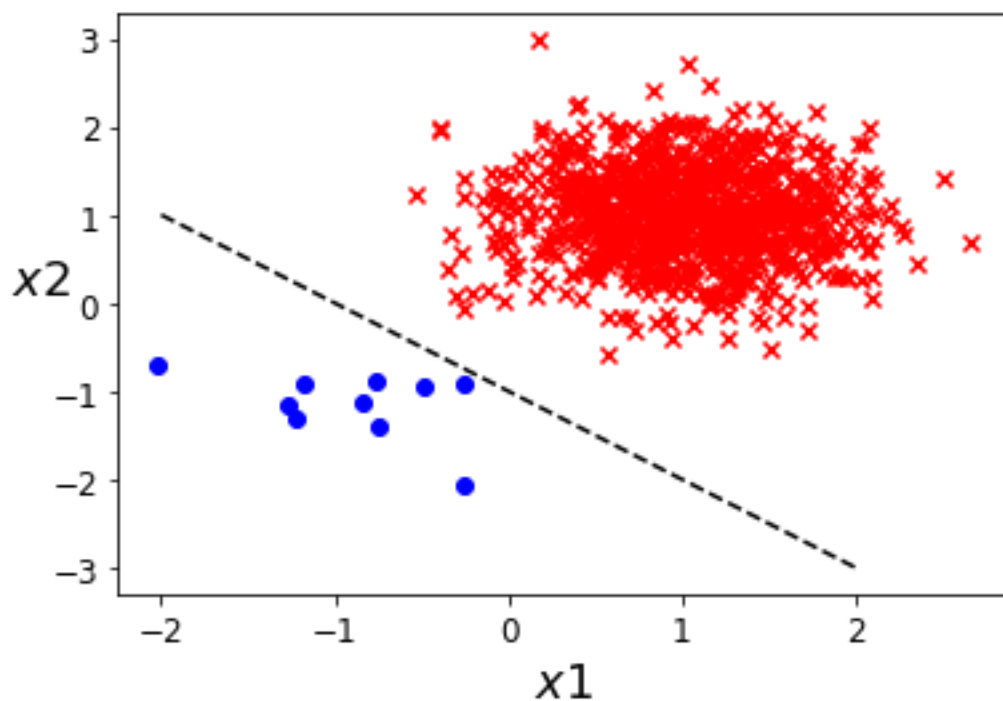
$$b^+ = b^- + \alpha(t - net)$$

5) در نهایت شرط استاپ را یا بر حسب تعداد ایپاک مشخص یا بر حسب اینکه تغییرات وزن ها در دو استپ متوالی از حد مشخص کمتر باشد قرار می دهیم (که من از OR کردن دو شرط فوق استفاده نمودم).

6) پایان

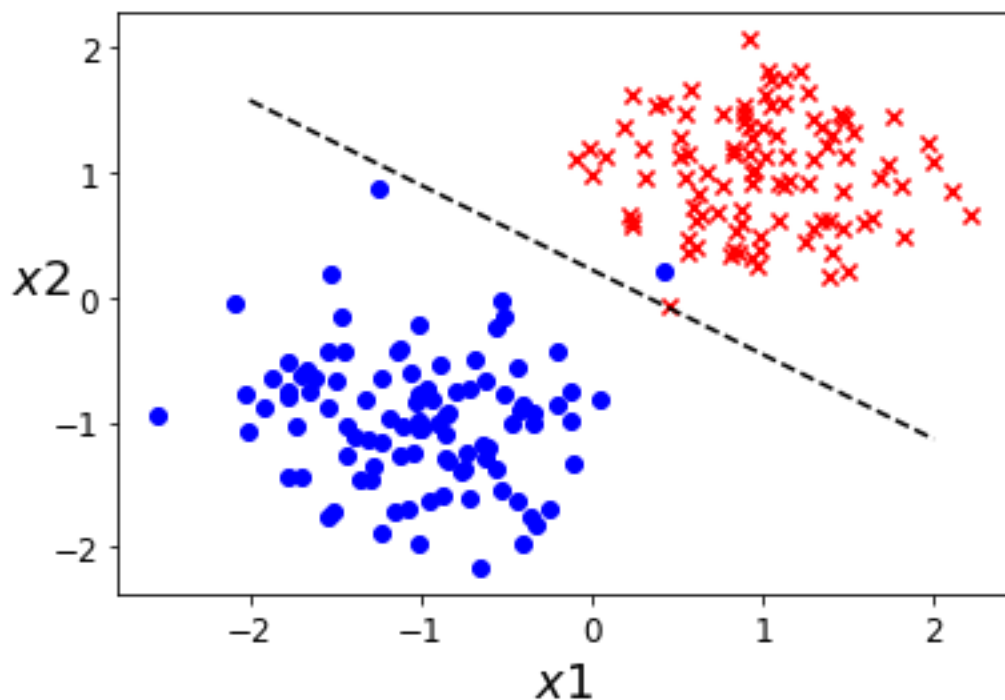
حال نتایج دو داده را بررسی می نماییم.

برای داده اول داریم :



شکل 7) خط جداساز برای داده اول

برای داده دوم :



شکل 8) خط جداساز برای داده دوم

ج) جواب این سوال بستگی به نوع داده دارد. در داده اول که تعداد جمعیت دو دسته برابر نمی باشد تضمین همگرایی برای این روش وجود ندارد اما برای داده دوم که جمعیت دو دسته برابر است Adaline عملکرد مناسب تری دارد.

در شکل ۱ برای robust بودن این الگوریتم می توان از $f = \tanh(\text{net})$ استفاده نمود که smooth بوده و در مشتق گیری مربوط به قاعده گرادیان عملکرد بهتری دارد، هم چنین می توان دسته قرمز را به نوعی کاهش جمعیت داد که ویژگی های آماری دسته حفظ شود اما عملکرد الگوریتم را خراب نکند.

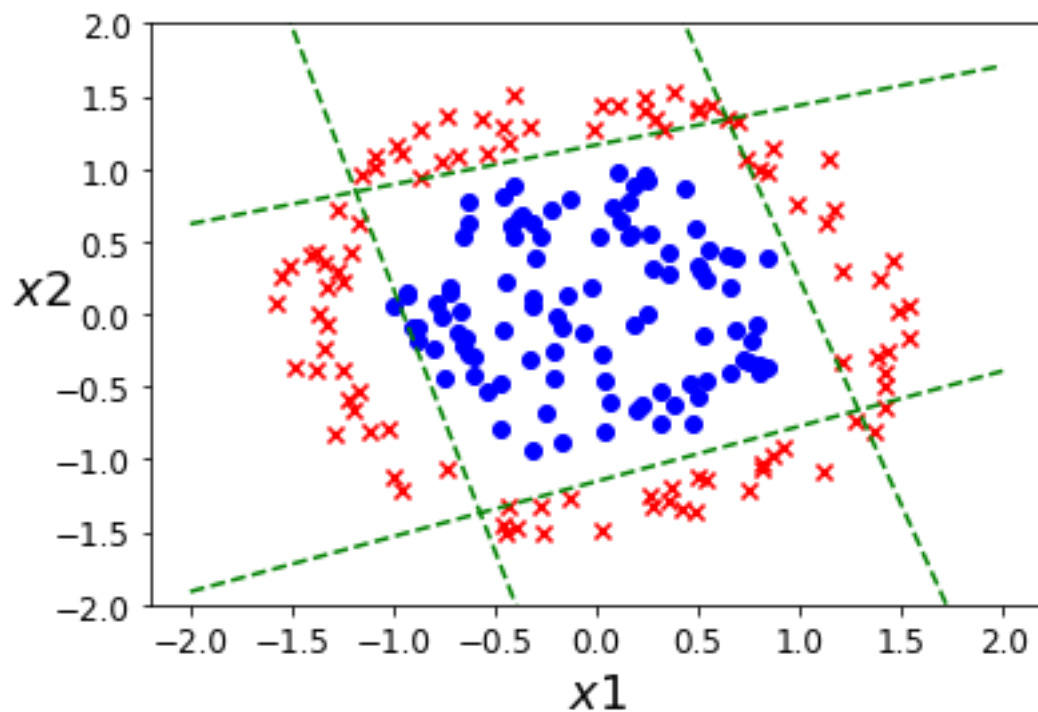
د) با توجه به گرادیان بیس بودن این الگوریتم ها تاثیر لرنینگ ریت مشابه تاثیر در روش های گرادیان می باشد بدین نحو که با بالا بردن این ضریب ممکن است پرش های زیادی رخ دهد و هیچ گاه به نقطه بهینه دست پیدا نکنیم و با زیاد از حد کوچک نمودن آن نیز ممکن است در یک مینیمم لوکال گیر نموده و الگوریتم در تعداد ایپاک بالا به مقدار بهینه خود برسد یا اصلا نرسد ، لذا برای عملکرد مناسب به نوعی trade off بین این دو موضوع داریم.

سوال 4 – Madaline

نکته : برای ران گرفتن ازین بخش نیاز دارید که برخی تغییرات را برای هر حالت اعمال کنید که در ادامه آورده شده است.

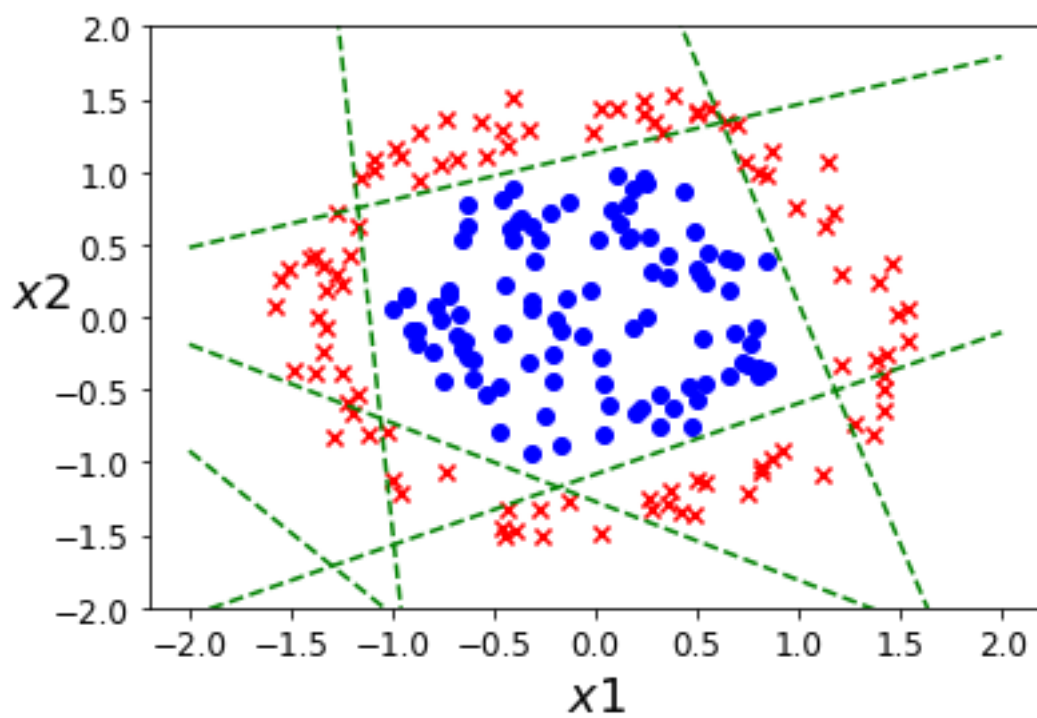
الف , ب) با پیاده سازی الگوریتم Madaline مطرح شده در کتاب (MR I) به نتایج زیر در شبیه سازی دست پیدا کردیم.

(1) $n=4$:



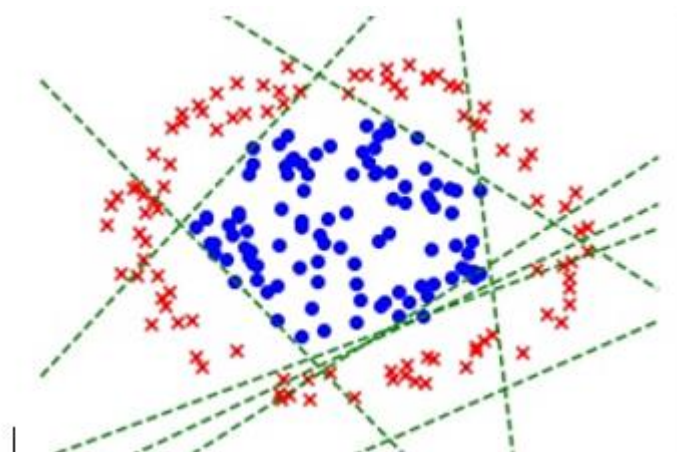
شکل 9) با 4 خط جداساز Madaline

2) برای $n=6$ نتایج زیر را داریم :



شکل 10) با 6 خط جداساز Madaline

3) برای $n=8$ داریم :



شکل 11) با 8 خط جداساز Madaline

ج) در این قسمت مشاهده می شود که با ۴ خط جداساز عملکرد بسیار خوب و بدون خطا در حدود ۵۰ ایپاک می باشد در حالیکه برای حالت ۶ نورون در حدود ۴۰۰ ایپاک و با کاهش لرنینگ ریت به جواب بدون خطا دست پیدا کردیم. در نهایت برای ۸ نورون و ۸ خط جداساز مشاهده می شود که با قرار دادن ۸۰۰ ایپاک ، بعضی از خطوط کمکی به جداسازی نکرده اند و به نوعی نورون ها ی اضافی عملکرد مشابهی با قبلی ها داشته و به نوعی افزونگی پیش آمده است.