



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
شبکه های عصبی و یادگیری عمیق

تمرین سری اول

نام و نام خانوادگی	میلاد محمدی
شماره دانشجویی	810100462
تاریخ ارسال گزارش	1400/12/21

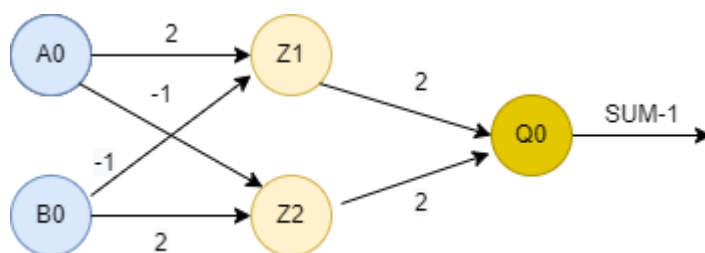
فهرست گزارش سوالات

3- 6.....	Mcculloch Pitts – 1 سوال
7-10	Adaline – ۲ سوال
11-12	Madaline – 3 سوال
13-14	Perceptron – 4 سوال

سوال 1 – Mcculloch Pitts

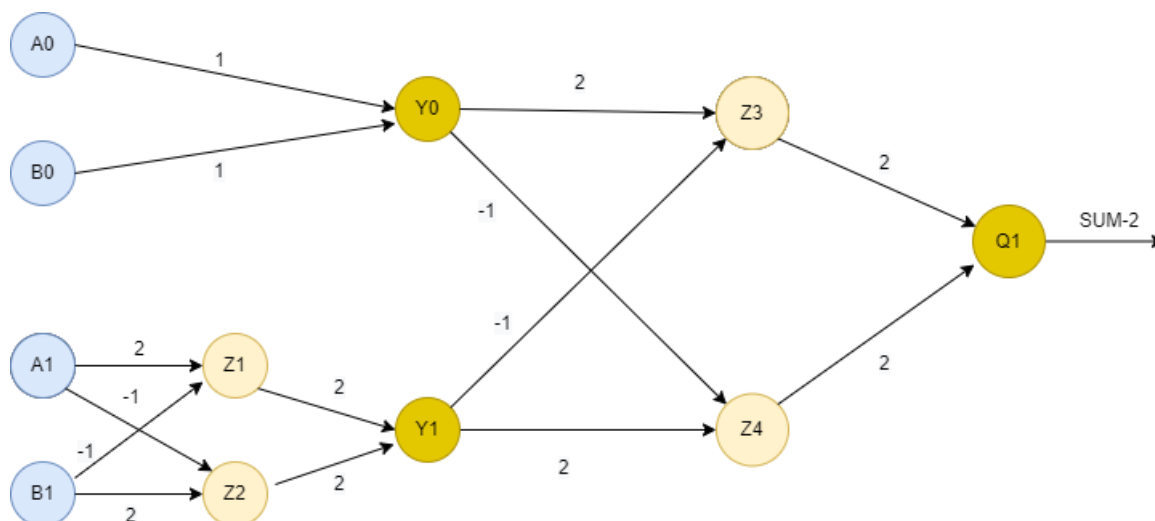
با فرض اینکه دو عدد ورودی ما به شکل $A = A_0A_1$ و $B = B_0B_1$ و خروجی به شکل CQ_1Q_0 است که در آن Q_1Q_0 حاصل جمع و C رقم نقلی است.

شبکه برای محاسبه‌ی رقم یکان حاصل جمع که برابر XOR ارقام یکان ورودی‌ها می‌باشد :



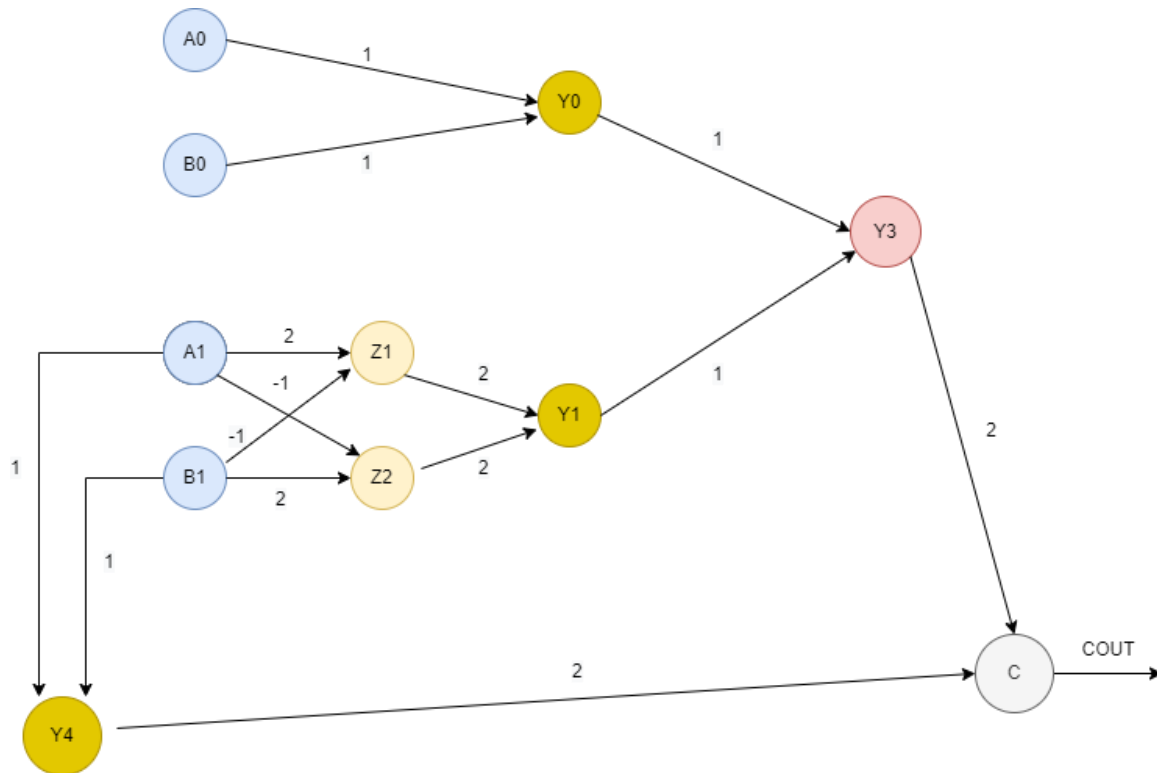
شکل 1. شبکه محاسبه رقم یکان

شبکه برای محاسبه‌ی رقم دهگان حاصل جمع که برابر $A_0B_0 \oplus (A_1 \oplus B_1)$ است :



شکل 2. شبکه محاسبه رقم دهگان

شبکه برای محاسبه‌ی رقم نقلی که برابر $A_0B_0 (A_1 \oplus B_1) + A_1B_1$ می‌باشد :

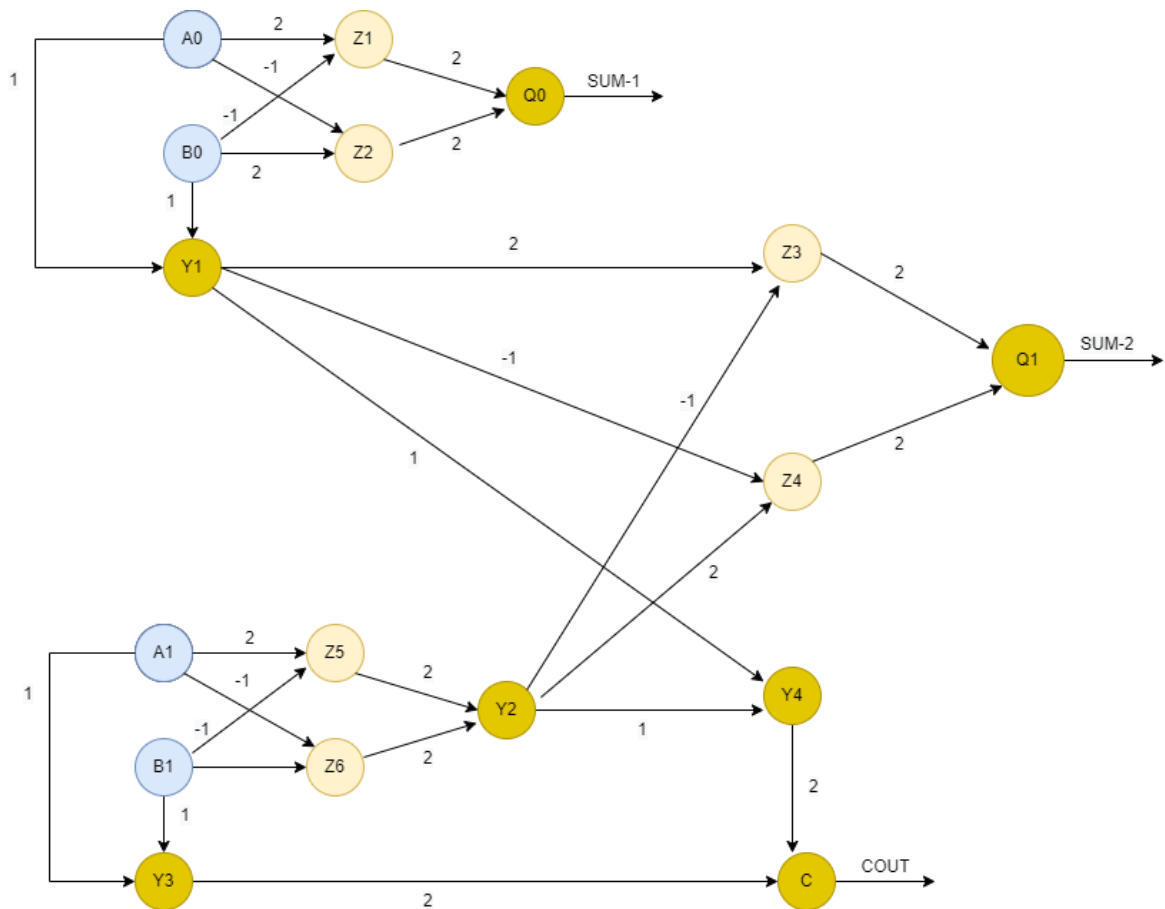


شکل 3. شبکه محاسبه رقم نقلی

شبکه کلی برای محاسبه حاصل جمع در صفحه بعد است. مقدار آستانه (threshold) برای تمام شبکه‌ها و گیت‌های and، or و xor برابر 2 می‌باشد.

تابع فعال‌ساز برای شبکه‌ها تابع زیر است :

$$f(y_{in}) = \begin{cases} 1 & y_{in} \geq 2 \\ -1 & y_{in} < 2 \end{cases}$$



شکل 4. شبکه کلی جمع کننده‌ی دو بیتی

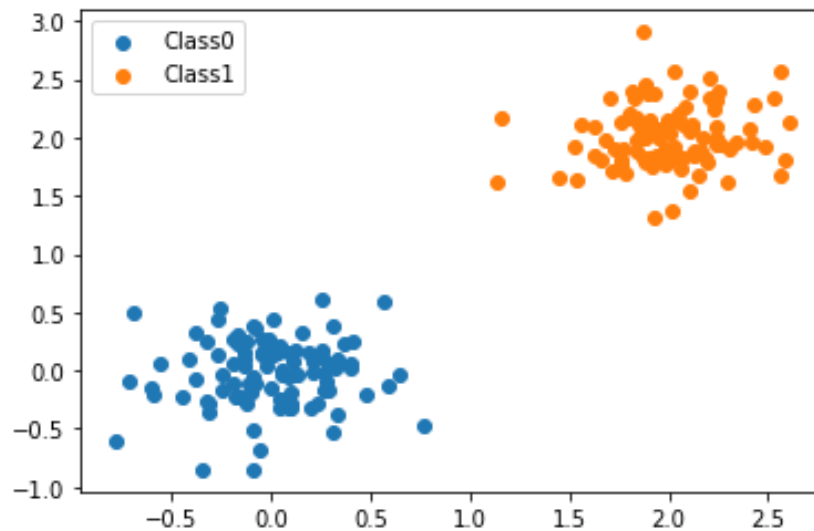
بخش پیاده‌سازی سوال در پوشه شماره یک قرار دارد و شبیه‌سازی شده است. خروجی کد به ازای تمام 16 حالت ممکن در صفحه بعدی است. چهار رقم ابتدایی به شکل $A_1 A_0 B_1 B_0$ است و در زیر حاصل جمع به صورت $(\text{Carry out}, \text{Sum}_2, \text{Sum}_1)$ نشان داده شده است.

0000
 (0, 0, 0)
 0001
 (0, 0, 1)
 0010
 (0, 1, 0)
 0011
 (0, 1, 1)
 0100
 (0, 0, 1)
 0101
 (0, 1, 0)
 0110
 (0, 1, 1)
 0111
 (1, 0, 0)
 1000
 (0, 1, 0)
 1001
 (0, 1, 1)
 1010
 (1, 0, 0)
 1011
 (1, 0, 1)
 1100
 (0, 1, 1)
 1101
 (1, 0, 0)
 1110
 (1, 0, 1)
 1111
 (1, 1, 0)

شکل 5. خروجی کد

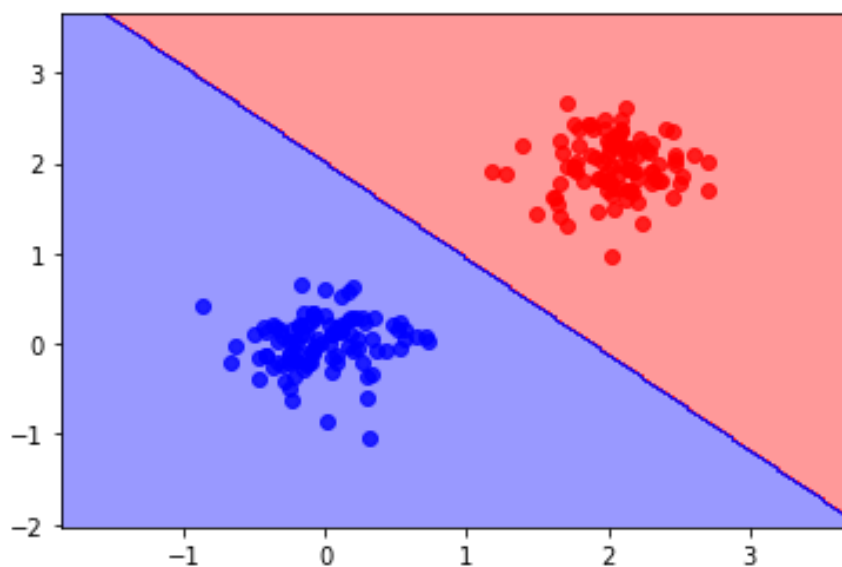
سوال ۲ – Adaline

قسمت الف) بخش پیاده‌سازی سوال در پوشه جداگانه قرار دارد. خروجی کد برای این قسمت :



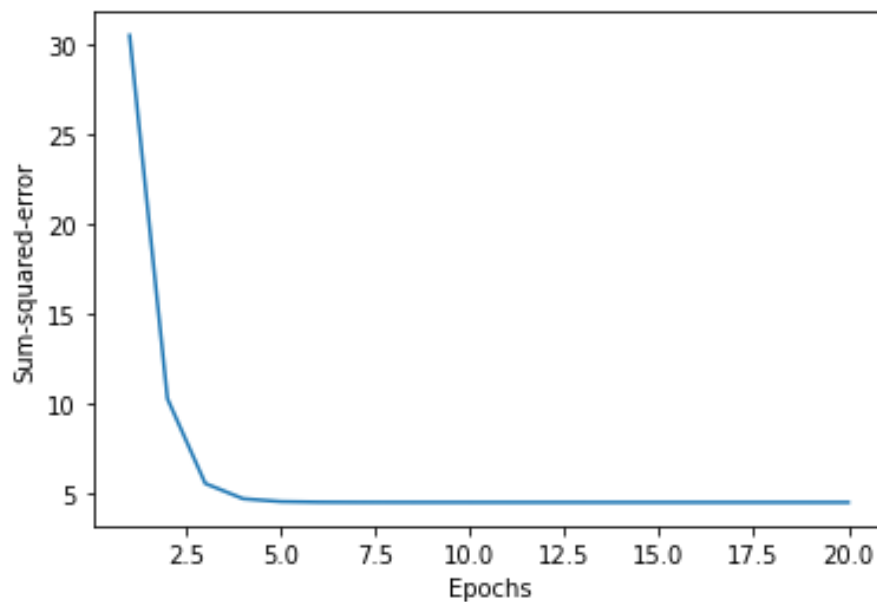
شکل 6. نمودار پراکندگی داده‌ها

قسمت ب) الگوریتم Adaline پیاده‌سازی شده و خروجی کد به صورت زیر است :



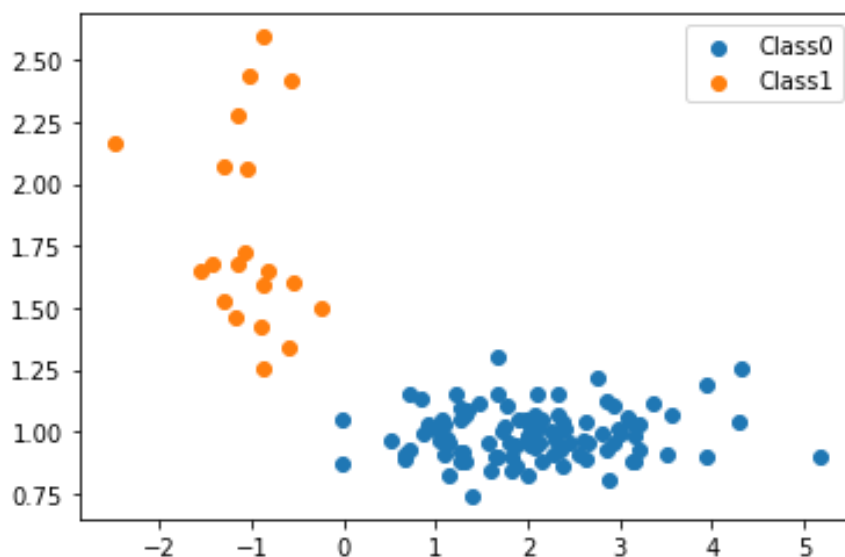
شکل 7. جدا کردن دسته داده‌ها

با توجه به شکل داده‌ها به خوبی از هم جدا شده‌اند. دلیل جدا شدن خوب داده‌ها این است که چون نقاط کاملاً از هم جدا هستند و پیچیدگی خاصی ندارند و داده‌ها دارای نویز نیز نمی‌باشند با یک مدل خطی ساده می‌توان آنها را از هم جدا کرد. در زیر نمودار تغییر خطا را می‌بینیم:

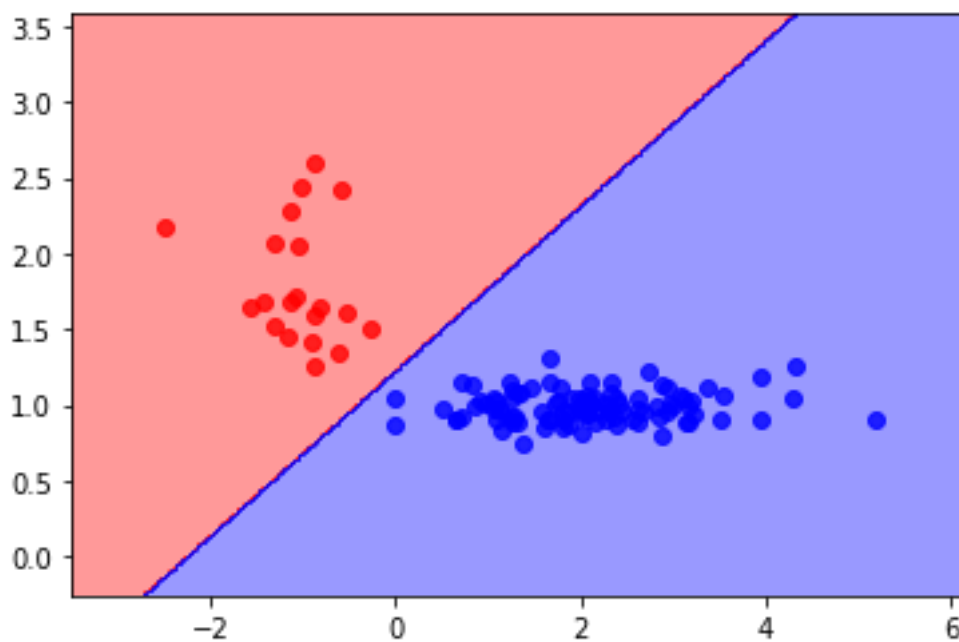


شکل 8. نمودار تغییرات خطا

قسمت ج :

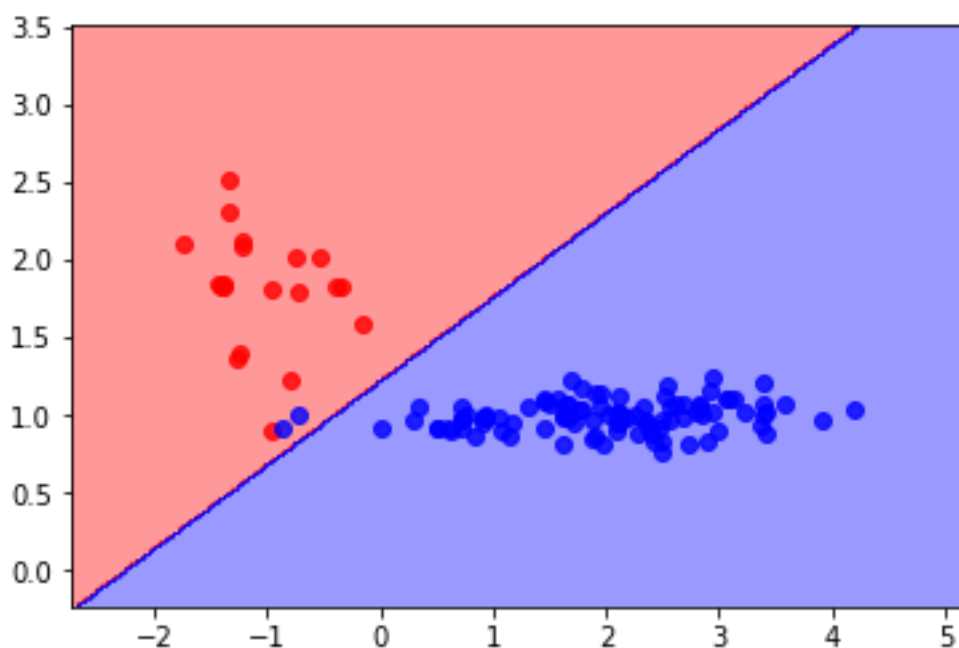


شکل 9. نمودار پراکندگی داده‌ها



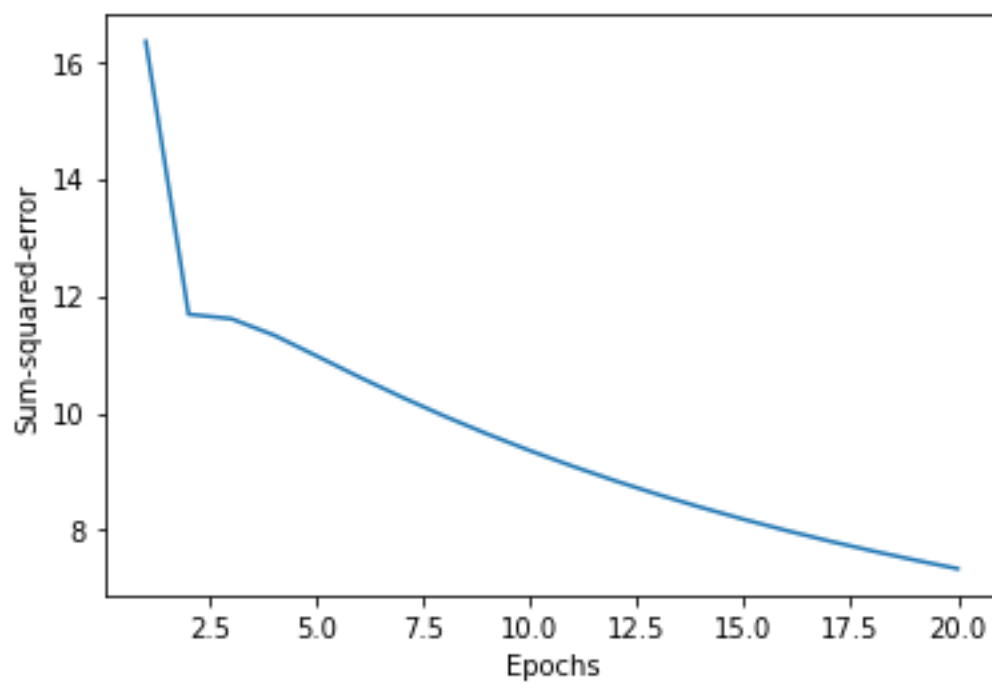
شکل 10. جدا کردن دسته داده‌ها

در این شکل با توجه به اینکه نقاط از هم فاصله دارند الگوریتم به خوبی آنها را جدا کرده است ولی در شکل زیر که نقاط به هم نزدیک‌تر هستند و در واقع کمی پیچیدگی زیاد شده است الگوریتم نتوانسته به خوبی آنها را جدا کند و باید از مدل دیگری استفاده کرد چون مدل خطی ادلاین خروجی مورد نظر را در صورت وجود نویز یا پیچیدگی تولید نخواهد کرد.



شکل 11. جدا کردن دسته داده‌ها

نمودار تغییرات خطا را در این قسمت می بینیم :



شکل 12. نمودار تغییرات خطا

قسمت الف) الگوریتم MRI : در این الگوریتم که فرم اصلی MADALINE می باشد، فقط وزن های لایه ی پنهان تنظیم می شوند و وزن های واحد خروجی ثابت هستند. واحد خروجی مانند or منطقی عمل می کند بدین ترتیب که اگر یکی (یا بیشتر) از سیگنال های ورودی یک باشد، خروجی یک خواهد بود و در صورتی که همه آنها 1- باشند خروجی 1- خواهد بود. تابع فعال ساز این الگوریتم به صورت زیر است :

$$F(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

این الگوریتم 8 مرحله دارد که شامل تنظیم وزن ها و بایاس ها می باشد و پس از آن برای هر کدام از نودهای لایه های پنهان مقدار net محاسبه می شود و به تابع فعال ساز داده می شود. سپس مقدار net برای خروجی محاسبه می شود. اگر $t = 1$ باشد و به هدف نرسیم وزن ها و بایاس به صورت زیر آپدیت می شوند :

$$W(\text{new}) = W(\text{old}) + \alpha \times (1 - Z_{in}) X_i$$

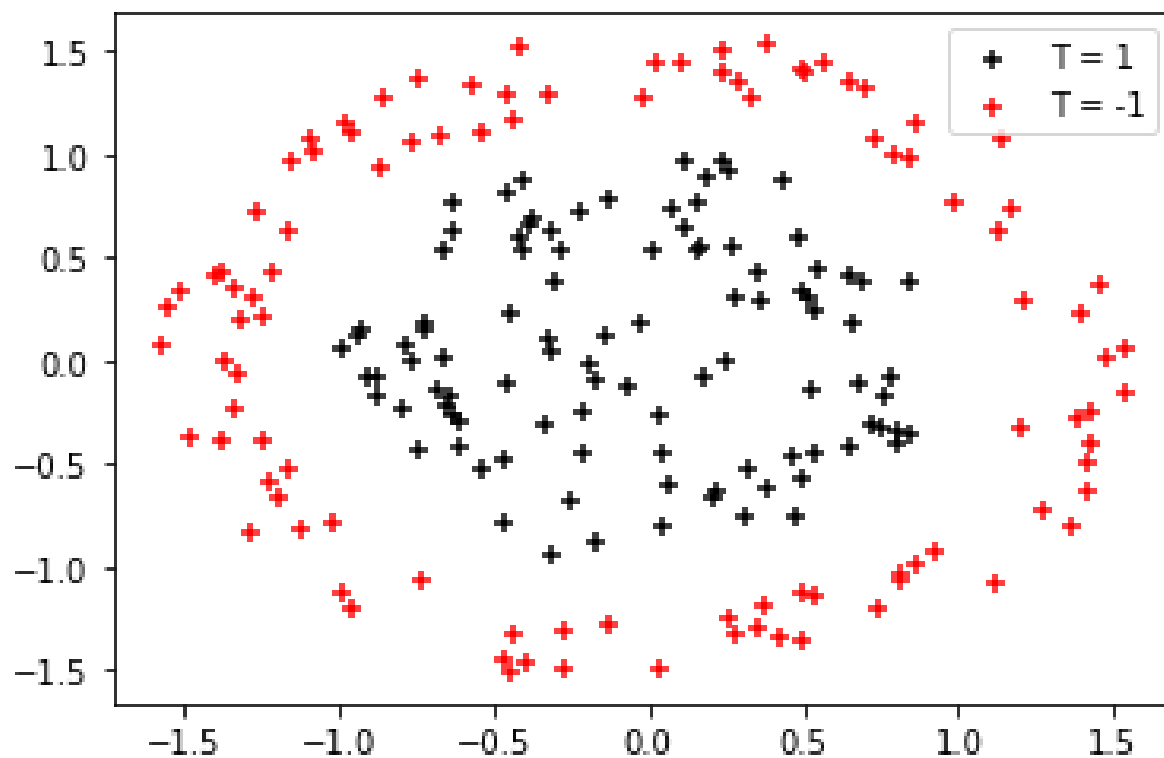
$$b(\text{new}) = b(\text{old}) + \alpha \times (1 - Z_{in})$$

و اگر $t = -1$ باشد وزن ها و بایاس ها به صورت آپدیت می شوند :

$$W(\text{new}) = W(\text{old}) + \alpha \times (-1 - Z_{in}) X_i$$

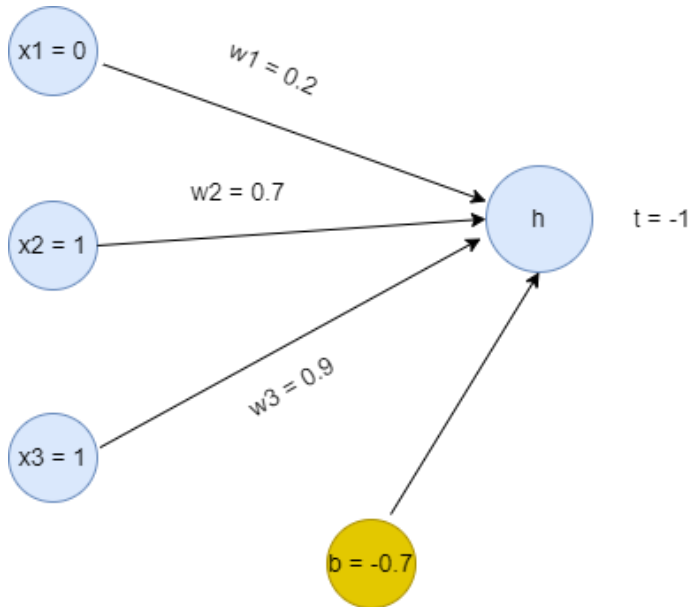
$$b(\text{new}) = b(\text{old}) + \alpha \times (-1 - Z_{in})$$

قسمت ب) در صفحه بعد نمودار پراکندگی داده ها داده نشان داده شده است که پیاده سازی آن در پوشه شماره 3 قرار دارد.



شکل 13. نمودار پراکندگی داده‌ها

سوال 4 – Perceptron



شکل 14. شبکه پرسپترون

Learning Rate = 0.3 , Activation Function : if $x \geq 0$: 1 else -1

مرحله 1 :

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 + b$$

$$\text{net} = (0 * 0.2) + (1 * 0.7) + (1 * 0.9) - 0.7 = 0.9$$

$$\text{sign}(0.9) = 1$$

چون به target نرسیدیم باید وزن ها و بایاس را آپدیت کنیم :

$$w_i(\text{new}) = w_i(\text{old}) + \alpha * x_i * t$$

$$b(\text{new}) = b(\text{old}) + \alpha * t$$

$$w_3 = 0.9 + 0.3 * 1 * (-1) = 0.6$$

$$w_2 = 0.7 + 0.3 * 1 * (-1) = 0.4$$

$$w_1 = 0.2 + 0.3 * 0 * (-1) = 0.2$$

$$b = -0.7 + 0.3 * (-1) = -1$$

مرحله 2 :

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 + b$$

$$\text{net} = (0 * 0.2) + (1 * 0.4) + (1 * 0.6) - 1 = 0$$

$$\text{sign}(0) = 1$$

چون به target نرسیدیم باید وزن‌ها و بایاس را آپدیت کنیم :

$$w_3 = 0.6 + 0.3 * 1 * (-1) = 0.3$$

$$w_2 = 0.4 + 0.3 * 1 * (-1) = 0.1$$

$$w_1 = 0.2 + 0.3 * 0 * (-1) = 0.2$$

$$b = -1 + 0.3 * (-1) = -1.3$$

مرحله 3 :

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 + b$$

$$\text{net} = (0 * 0.2) + (1 * 0.1) + (1 * 0.3) - 1.3 = -0.9$$

$$\text{sign}(-0.9) = -1 = t$$

$$w_3 = 0.3 + 0.3 * 1 * (-1) = 0$$

$$w_2 = 0.1 + 0.3 * 1 * (-1) = -0.2$$

$$w_1 = 0.2 + 0.3 * 0 * (-1) = 0.2$$

$$b = -1.3 + 0.3 * (-1) = -1.6$$

$$\text{net} = (0 * 0.2) + (1 * -0.2) + (1 * 0) - 1.6 = -1.8$$

$$\text{sign}(-1.8) = -1 = t$$