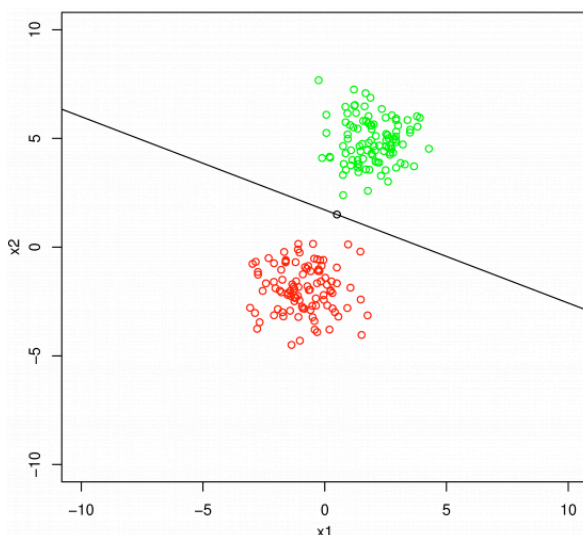
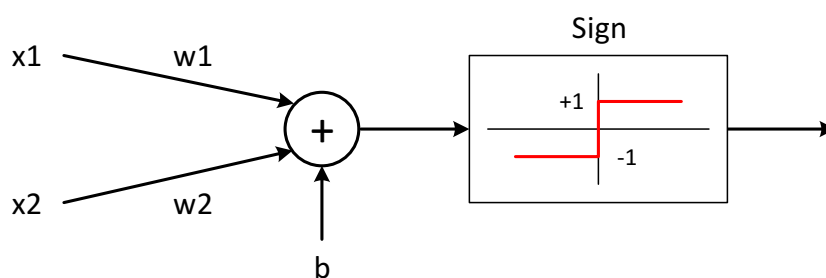




یکی از کاربردهای شبکه‌عصبی رده‌بندی (Classification) داده‌ها است. در رده‌بندی داده‌ها تلاش می‌کنیم با رسم یک خط در فضای دو بعدی، داده‌های موجود در صفحه را به دو رده تقسیم کنیم. مثلاً در شکل زیر یک نورون با رسم یک خط داده‌ها را به دو رده‌ی سبز و قرمز تقسیم کرده است.



المان پایه‌ی یک شبکه‌ی عصبی نورون (Neuron) است. شکل زیر یک نورون را نشان می‌دهد که دارای دو ورودی x_1 و x_2 ، یک ورودی بایاس (Bias) و دو مقدار وزنی w_1 و w_2 متناظر با دو ورودی x_1 و x_2 است (وزن ورودی بایاس همیشه برابر مقدار ۱ است). عملکرد نورون به این صورت است که ورودی‌های x_1 و x_2 را به ترتیب در دو وزن w_1 و w_2 ضرب کرده و با ورودی بایاس جمع می‌کند و روی مقدار به دست آمده یک تابع فعال‌ساز (Activation Function) اعمال می‌شود تا خروجی نورون به دست بیاید. برای یادگیری شبکه‌ی عصبی می‌توان از یادگیری با نظارت استفاده کرد. به این صورت که با اعمال یک ورودی (با خروجی مشخص) به نورون بر اساس میزان انحراف خروجی نورون از خروجی مطلوب ضرایب نورون (w_1 و w_2) به هنگام می‌شوند تا میزان انحراف از خروجی به حداقل برسد.



روش یادگیری

روش یادگیری یک نورون را در زیر می‌بینیم:

- **مرحله ۰:** مقداردهی اولیه وزن‌ها و بایاس که صفر است
تنظیم نرخ یادگیری که برای ساده‌سازی آلفا را ۰/۷۵ در نظر می‌گیریم
- **مرحله ۱:** تا زمانی که شرط توقف نرسیده است مراحل ۲ تا ۶ را انجام بده (شرط توقف تغییر نکردن همه وزن‌ها در یک مرحله است)
- **مرحله ۲:** برای هر جفت داده آموزش $(s:t)$ (s ورودی و t خروجی مطلوب است) مراحل ۳ تا ۵ را انجام بده

- **مرحله ۳:** فعال کردن واحدهای ورودی

$$x_i = S_i$$

- **مرحله ۴:** محاسبه خروجی نورون

$$y_{in} = b + \sum x_i w_i$$

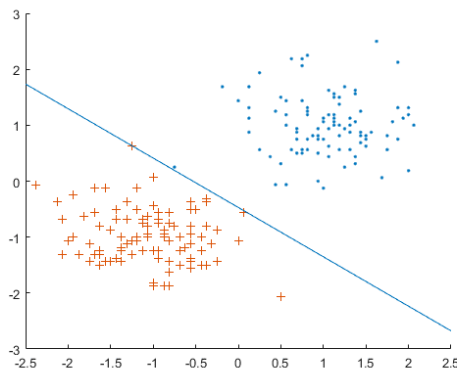
$$y = \text{sign}(y_{in})$$

- **مرحله ۵:** وزن‌ها و بایاس را در صورتی که خروجی شبکه برای این نمونه خطا داشت ($y \neq t$) به‌روز رسانی کن

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

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

- **مرحله ۶:** ارزیابی شرایط توقف (اگر هیچ وزنی در یک مرحله برای هیچ جفت داده آموزش تغییر نکرد)



نتیجه مانند شکل مقابل خواهد بود. نورون شما باید قادر به جدا سازی این نقاط باشد. برای ساده کردن طراحی سخت افزاری نقاط داده شده Quantize شده اند که ۴ بیت کسری و ۳ بیت صحیح دارند. تمام اعداد به صورت علامت دار هستند.

با طراحی مسیر داده و واحد کنترل یک نورون برای جداسازی دو کلاس طراحی کنید و به روش گفته شده آن را آموزش دهید و آنها را با Verilog مدل‌سازی کنید و در انتها با داده‌های داده شده، عملکرد نورون را تست کنید. برای درک بهتر این مکانیزم می‌توانید به فایل Matlab ضمیمه شده مراجعه کنید.

نکات مهم:

- وزن‌ها و بایاس و ثبات y_{in} را ۱۴ بیت علامت‌دار فرض کنید که ۸ بیت کسری و ۶ بیت صحیح دارند.
- ورودی‌ها به صورت ۷ بیت علامت‌دار با ۴ بیت کسری فرض شوند.
- خروجی نورون همان گونه که گفته شد دوقطبی است و مقدار ۱ و -۱ را می‌گیرد.

```

۱۱۱۰۰۰۰ ۱۱۱۰۰۰۰ ۱۱
۱۱۱۰۰۰۰ ۰۰۱۰۰۰۰ ۰۱
۰۰۱۰۰۰۰ ۱۱۱۰۰۰۰ ۰۱
۰۰۱۰۰۰۰ ۰۰۱۰۰۰۰ ۰۱

```

اگر تست عملکرد نورون با این حجم داده، مشکل است می‌توانید در ابتدا با مثلاً ۴ ورودی تست کنید. نورون روبرو یک گیت OR را مدل می‌کند:

روش ارزیابی:

- پیاده سازی نورون و آموزش آن ۱۰۰ نمره دارد
 - ۲۵ نمره طراحی مسیر داده و واحد کنترل (در زمان تحویل حضوری به صورت کتبی تحویل داده شود)
 - ۲۵ نمره روش کدینگ (مسیر داده به صورت ساختاری و واحد کنترل به روش هافمن)
 - ۲۵ نمره صحت طراحی (به دست آوردن خط جداساز) با داده‌های ضمیمه شده
 - ۲۵ نمره صحت طراحی با داده‌های آزمون توسط دستیاران آموزشی