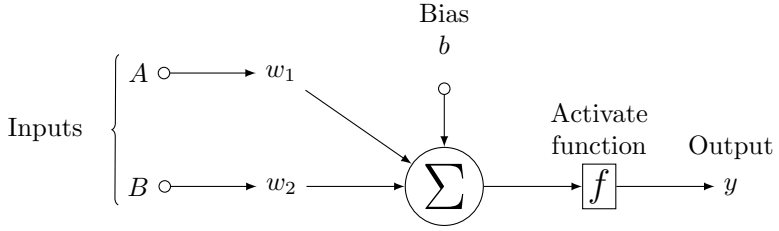
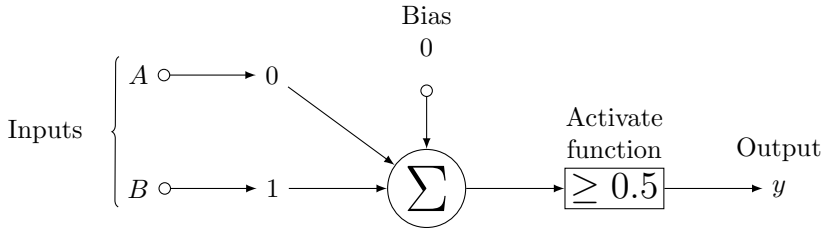
1 Exercise *(Perceptron (6p))*

1. Sketch a single neuron perceptron with weights and threshold θ for the following logical functions: Our network is set up like this:

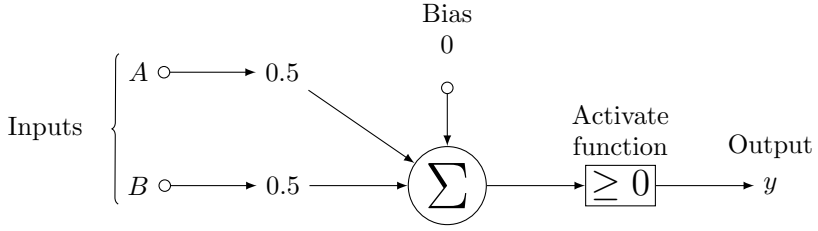


We assume $A, B \in \{-1, 1\}$

(a) $(A \wedge B) \vee (\neg A \wedge B)$
$\leftrightarrow^1$
B



(b) $(A \wedge B) \vee (\neg A \wedge B) \vee (A \wedge \neg B)$
$\leftrightarrow^2$
$A \vee B$

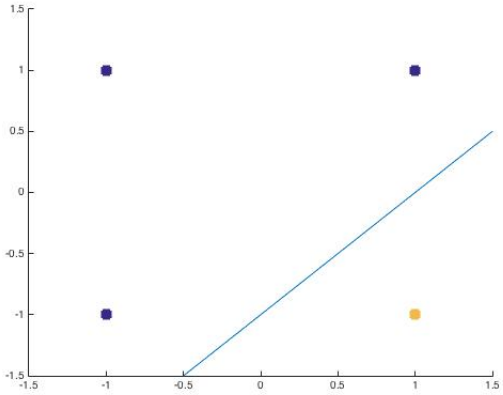


[1] since B is either 1 or 0
[2] again, bracketing out B and $\neg B$ and A and $\neg A$

2. Draw pictures similar to slide page 36 in chapter 7 for the boolean functions below, and give weights for the perceptron or argue if no such weight exist. The features x_1, x_2 and output y can only take values -1 or +1.

$$y = \begin{cases} +1 : x_1 = 1, x_2 = -1 \\ -1 : otherwise \end{cases}$$

$\rightarrow w_1 = 1, w_2 = -1, b = -1, f(x) = 1 \leftrightarrow x \geq 0.5$



2 Exercise *(Perceptron (6p))*

Simple classification problem for single neuron perceptron.

1. Draw a decision boundary in figure 2 (a) and (b) and find weights and bias for each network.

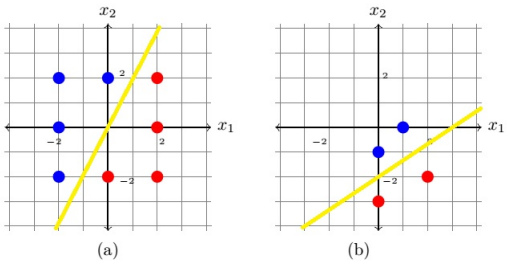


Figure 2: Simple classification problems. Blue dot represents positive and red dot for negative.

(a) $\vec{w} = (-1, \frac{1}{2}), b = 0$ (b) $\vec{w} = (-1, \frac{3}{2}), b = -2$

2. Determine the margin of the the separation line for each figure. (Hint: Compute the distance between each point in the figure to the selected decision boundary)

```
close all % plotting the data and calculating the point line distance by
hold on % projecting the points onto the unit normal vector
points = [-2,-2,-2,0,0,2,2,2 ; -2,0,2,-2,2,-2,0,2];
scatter(points(1,:),points(2,:),[],[0,0,0,10,0,10,10,10])
normal = [-1,0.5];
normal = normal * (1./sqrt(normal*normal.'));
quiver(0,0,normal(1),normal(2))
xlim([-2.5 2.5])
ylim([-2.5 2.5])

distances = abs(normal * points);
margin = min(distances)
---------------------------
margin =

    0.8944
---------------------------

close all;
hold on;
points2 = [0,0,1,2 ; -1,-3,0,-2];
scatter(points2(1,:),points2(2,:),[],[0,10,0,10])

xlim([-4 4])
ylim([-4 4])

normal2 = [-1,1.5];
normal2 = normal2 * (1./sqrt(normal2*normal2.'));

quiver(0,-2,normal2(1),normal2(2))
distances2 = abs(normal2 * [points2(1,:) ; points2(2,:) - 2]);

margin2 = min(distances2)

---------------------------

margin2 =

    2.2188
---------------------------
```