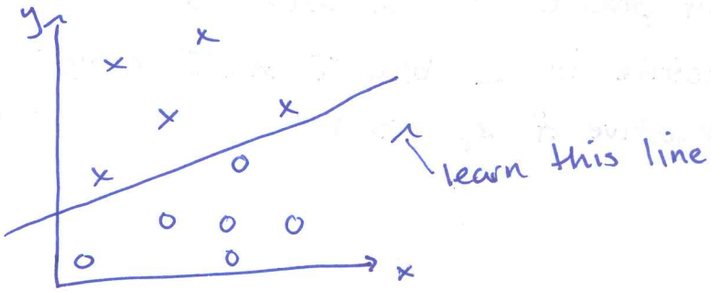
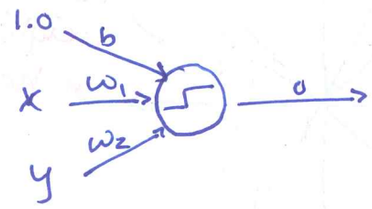
① Perceptron as classifier:





$$o = \begin{cases} 0 & \text{if } b + w_1 \cdot x + w_2 \cdot y \leq 0 \\ 1 & \text{if } b + w_1 \cdot x + w_2 \cdot y > 0 \end{cases}$$

$$p = \begin{bmatrix} 1.0 \\ x \\ y \end{bmatrix} \begin{bmatrix} b & w_1 & w_2 \end{bmatrix}$$

whoops — this does outer product. switch the two vectors

② Learning weights

input $i = (x_i, y_i)$ predicted ~~output~~ $p_i = \begin{bmatrix} 1.0 \\ x_i \\ y_i \end{bmatrix} \begin{bmatrix} b_t & w_{1t} & w_{2t} \end{bmatrix}$

real output $i = o_i$

update weights:

$$b_{t+1} = b_t + (o_i - p_i) \cdot (1.0)$$

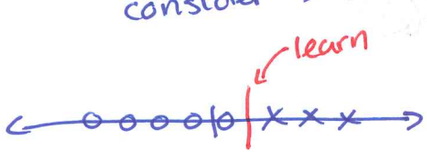
$$w_{1\,t+1} = w_{1t} + (o_i - p_i) \cdot x_i$$

$$w_{2\,t+1} = w_{2t} + (o_i - p_i) \cdot y_i$$

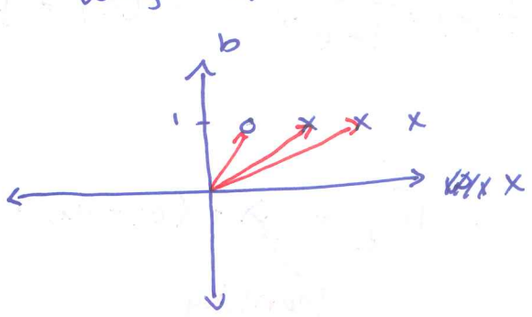
$$\overline{w_{t+1}} = \overline{w_t} + \text{error} \cdot \overline{i}$$ ← move weights in direction of input if input is ∉ "true", away if input is "false"

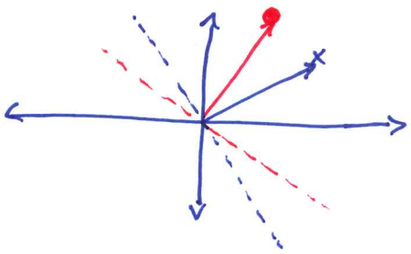
③ Intuition

consider 1-D classifier



weight space

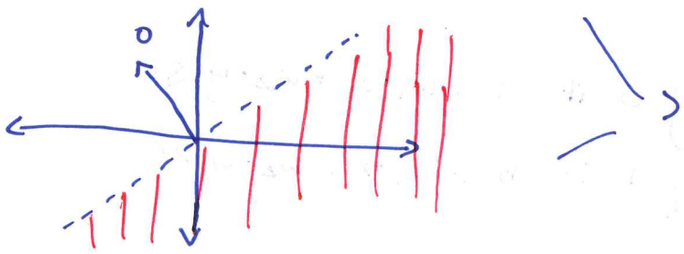


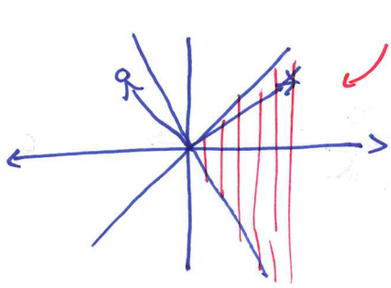


p = dot product of $\bar{x}$ and $\bar{w}$

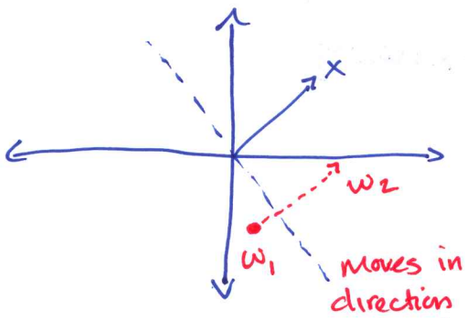
=> positive if $\angle$ b/w $\bar{x}$ and $\bar{w}$ $< 90°$

=> negative if $\angle$ $> 90°$

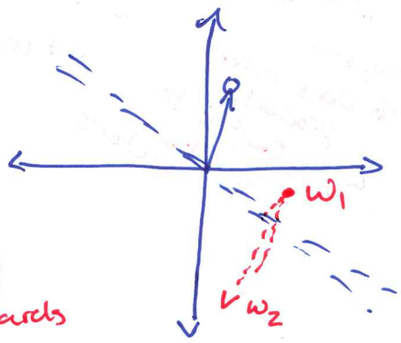




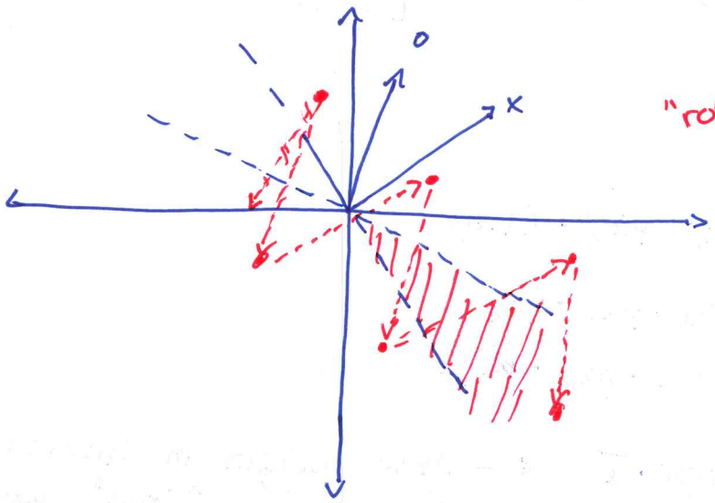
(4) intuition for training:



moves in direction of input — towards "right" side



moves away from input — towards "right" side



"rolls downhill toward shaded valley"

(5) why can't we learn xor? not linearly separable!

(6) more general activation f'n: sigmoid:

$$f(x) = \frac{1}{1+e^{-x}} \qquad f'(x) = f(x) \cdot (1 - f(x))$$

training:

$$\bar{w}_{t+1} = \bar{w}_t + \alpha \cdot (o_i - p_i) \cdot f'(\bar{w}_t \cdot \bar{x}) \cdot \bar{x}$$

α: learning rate

$f'(\bar{w}_t \cdot \bar{x}) \cdot \bar{x}$: gradient toward minimizing error

gradient descent