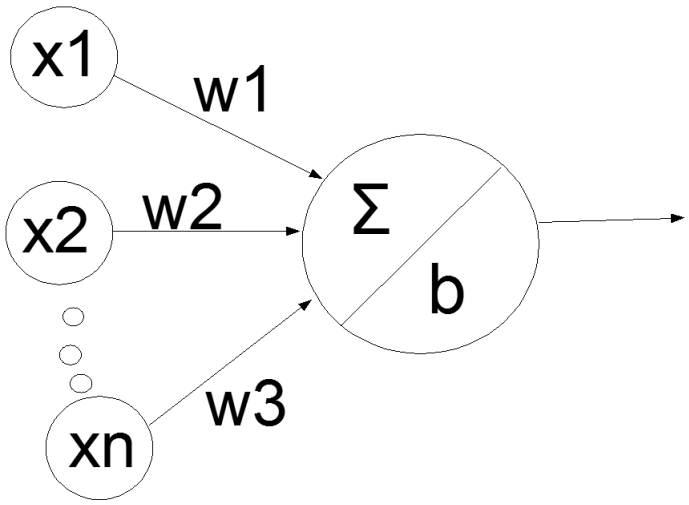
Learning

We would like to be able to create a something that can classify each object in a set of objects as, for example, a car (+1) or not a car (−1).

We can create a learning device that trains on a sample of the entire set of objects, with each object in the sample being labeled with the correct classification.

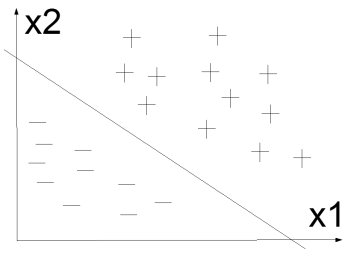
Let's represent an object as a vector of values: $a = (x_1, x_2, ..., x_n)$. With this, we can try to learn a linear classifier:



Σ is the sum $x_1w_1 + x_2w_2 + ... + x_nw_n$, and b is the threshold value of the classifier. If $\Sigma > b$, the classifier outputs (+1), if $\Sigma < b$, it outputs (−1).

Training the classifier:

- We would like to be able to use the classifier to actually classify objects
- So we'll train the classifer on a set of m examples: $(a_1, a_2, ...a_m)$
- Each a_y has an associated value l_y, that is either (+1) or (−1).
- We want to find weights $w = (w_1, w_2, ..., w_n)$ such that $w \cdot a_y < b$ if $l_y = -1$, and $w \cdot a_y > b$ if $l_y = +1$
- This is possible if the set is linearly separable, like the set of examples below:



Learning continued on next page...

- This is not always possible though:

- However we may be able to take the data into higher dimensions where it is linearly separable (eg. use x_1x_2) using kernels, but this is for another time.

To phrase our goal in another way, let $\hat{a_i} =< a_i, -1 >$, and let $\hat{w} =< w, b >$. (We scale $\hat{a_i}$ so that $|\hat{a_i}| \leq 1$). With this, we want $(\hat{w} \cdot \hat{a_i})l_i > 0$ for all i.

How can we find these weights $\hat{w}$? We could use linear programming, but there is a faster way:
- The idea is to start looking at the patterns that are the samples and their classifications (a_il_i).
- If the weights cause a mismatch for the pattern, add the pattern to the weights
- Now the weights are closer to matching the given pattern (an error in a positive classification for some vector (a_y will increase the weights in w for the variables that are more significant in a_y)

Now we can write out the algorithm:
- Set b to 0, and scale all a_i so that $|a_i| \leq 1$
- Set $w = a_1l_1$ (Must be correct for a_i because $a_1l_1 \cdot a_1l_1 > 0$).
- While $w \cdot a_il_i \not> 0$ for all i, iterate through i
  - If $w \cdot a_il_i < 0$, add a_il_i to w

We can show that the above algorithm will find a solution if the data is linearly separable. First let's define the margin of a linear classifier, δ, as the distance of the closest sample point to the line. So, over all i

$$\delta = \frac{min(wa_il_i)}{|w|}$$

Note that we divide by $|w|$ in order to prevent the scaling of w from affecting the margin.

Theorem

Suppose there exists some w^* with margin $\delta > 0$. Then the algorithm finds some solution w within $\frac{1}{\delta^2} - 1$ updates of w.

Proof

- Assume that $|w^*| = 1$ (Scaling does not make a difference)
- Examine the cosine of the angle between w as found by the algorithm, and w^*:

$$cos = \frac{w \cdot w^*}{|w|}$$

- We can show that the two values converge

Learning continued on next page...

- First of all, *cos* never increases beyond 1.
- We can show that $cos \rightarrow 1$:
- How much does the numerator grow with each update?
  - At each update, the new value for the numerator is, for some i

$$(w + a_i l_i)(w^*) = ww^* + w^* a_i l_i$$

  - Since we assumed that w^* classified correctly with a margin of δ, $w^* a_i l_i \geq \delta$
  - So the numerator increases by this amount on each update, which is at least δ
- How much does the denominator grow with each update?
  - New magnitude of w:

$$|w + a_i l_i|^2 = |w|^2 + 2wa_i l_i + (a_i l_i)^2$$

  - Since a_i was misclassified, $2wa_i l_i$ must be less than 0.
  - Since a_i was normalized, $(a_i l_i)^2 \leq 1$
  - At most:

$$|w + a_i l_i|^2 = |w|^2 + 1$$

  - So the most the denominator can increase on any update is 1
- After t updates:
  - $|w \cdot w^*| \geq (t+1)\delta$
  - $|w|^2 \leq (t+1) \rightarrow |w| \leq \sqrt{(t+1)}$
  - We then have:

$$cos \geq \frac{(t+1)\delta}{\sqrt{(t+1)}}$$

  - When is

$$\frac{(t+1)\delta}{\sqrt{(t+1)}} \leq 1$$

  - Solve this out

$$(t+1)^2\delta^2 \leq (t+1)$$

$$(t+1)\delta^2 \leq 1$$

$$t\delta^2 \leq 1 - \delta^2$$

$$t \leq \frac{1-\delta^2}{\delta^2} = \frac{1}{\delta^2} - 1$$