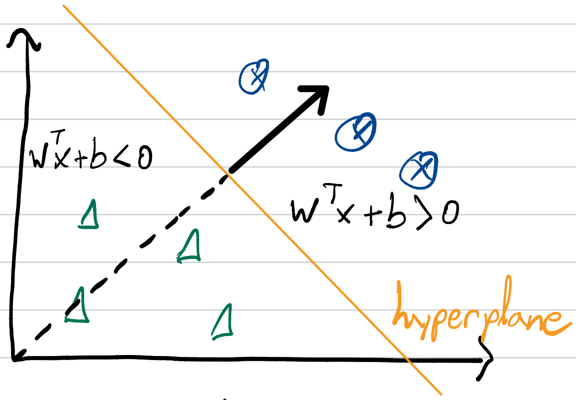
Perceptrons

Dot Products: $a^T b = \sum_i a_i b_i$

Linear Classifier Assumptions:

① Binary Classification (i.e. $y_i \in \{-1, +1\}$)
② Data is linearly seperable



Hyperplane: Is defined by weight vector w and bias b.
- w defines the direction of the hyperplane, passes through the origin and is perpendicular to the hyperplane.
- b determines how far from the origin the hyperplane lies.

Classification:

+ve sample if $w^Tx + b > 0$
-ve sample if $w^Tx - b < 0$

So, $h(x_i) = \text{sign}(w^T x_i + b)$

Absorbing the bias:

Would be nice if all you had to do was a dot product.

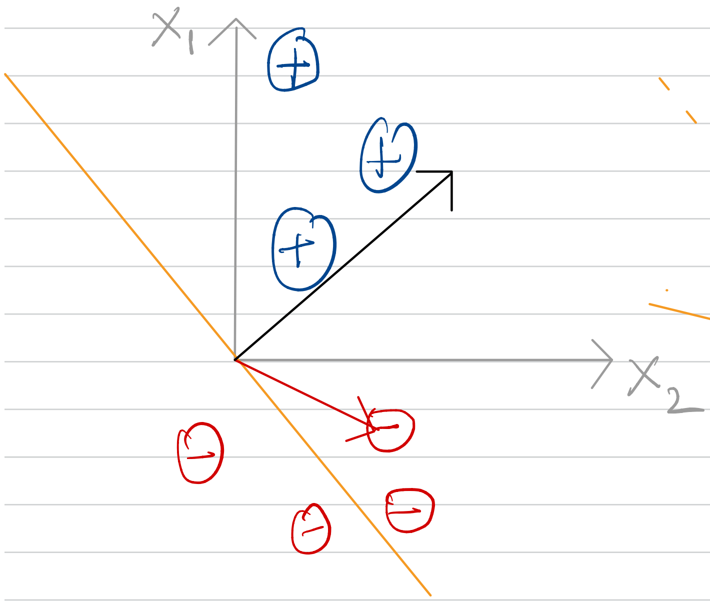
Quiz: By adding one or more dimensions, can you think of a way to compute w^Tx+b without having an explicit 'b' term?

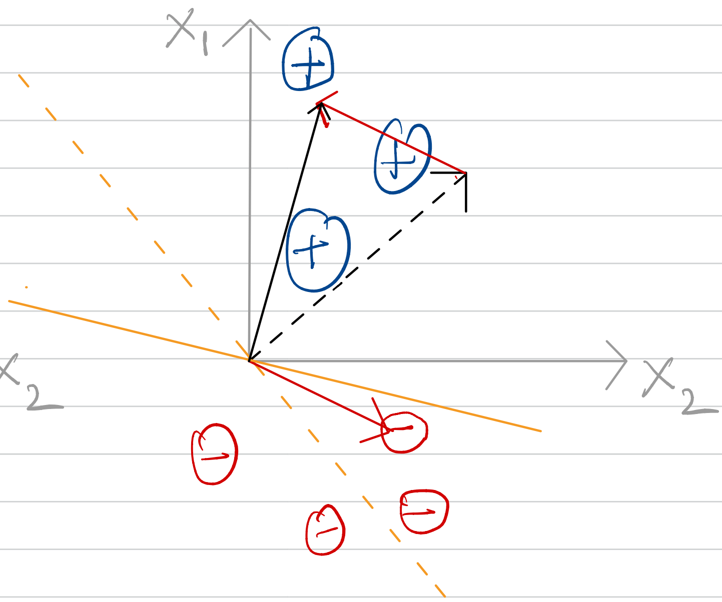
Geometric Intuition:

Quiz: How do you know a point x is misclassified?

Perceptron Algorithm:

```
Initialize w = 0⃗
while True:
    m = 0
    for (x_i, y_i) ∈ D:
        if y_i (w^T x_i) ≤ 0:
            w = w + yx
            m = m + 1
        end if
    end for
    if m == 0:
        break
    end if
end while
```





Notes:

* multiple update steps might be required
* more updates are required if the margin is small

Does the perceptron algorithm always converge?

Yes! If there is a separating hyperplane, in a finite number of steps, a valid hyperplane will be found.

Example of a function a perceptron cannot learn: