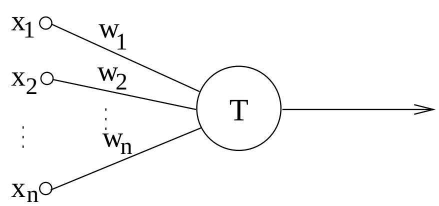
CS/ENGRI 172, Fall 2003: Computation, Information, and Intelligence
9/22/03: Perceptrons and Perceptron Learning

Perceptrons

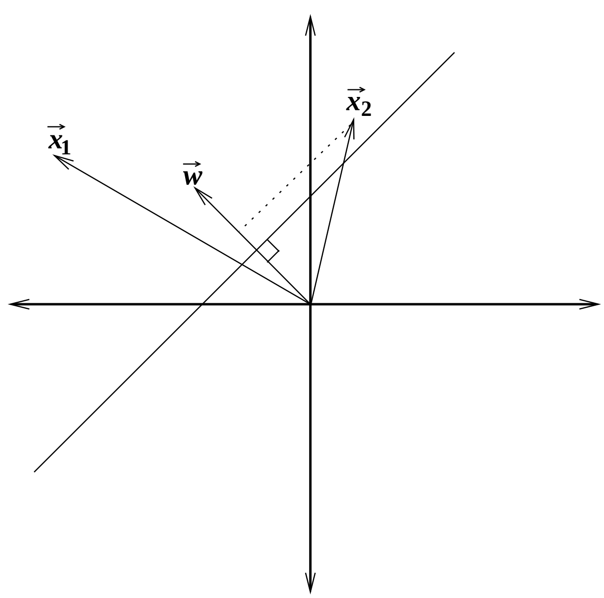
We will first restrict our attention to learning functions computable by *perceptrons*, which are idealizations of a single neuron.



Perceptrons are characterized by a weight vector $\vec{w}$ and a threshold T. Letting n be the dimensionality of $\vec{w}$, a perceptrons "fires" (outputs a one) on input $\vec{x} = (x_1, x_2, \ldots, x_n)$ if $w_1x_1 + w_2x_2 + \ldots + w_nx_n = \vec{w} \cdot \vec{x} \geq T$, and outputs 0 otherwise.

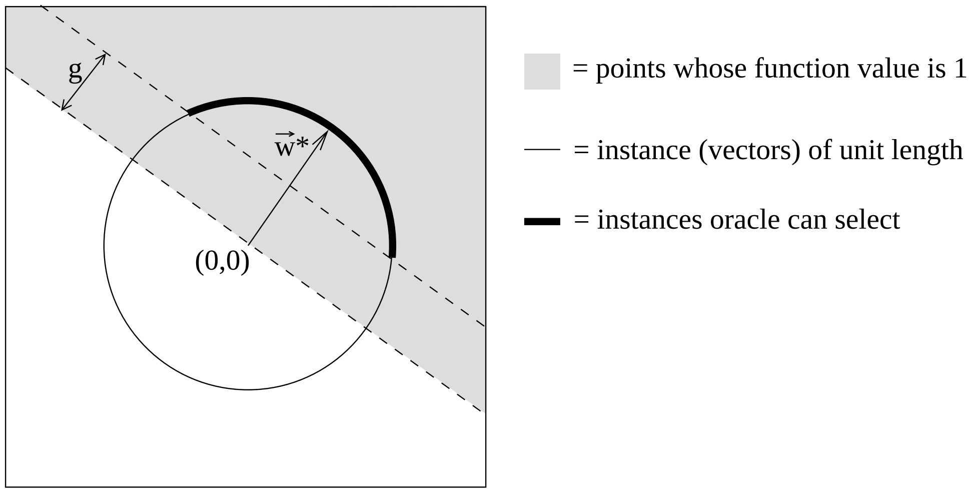
Perceptron Functions as Half-Plane Concepts

A perceptron outputs one when length($\vec{w}$)proj($\vec{x}$,$\vec{w}$) $\geq T$, or proj($\vec{x}$,$\vec{w}$) $\geq \frac{T}{\text{length}(\vec{w})}$ (if length($\vec{w}$) $\neq$ 0). That is, when the projection of $\vec{x}$ onto $\vec{w}$ is longer than the cutoff value of $\frac{T}{\text{length}(\vec{w})}$. This cutoff defines a line percendicular to $\vec{w}$, and thus a half-plane of points on which the perceptron outputs one.



Oracle Restrictions for Learning

Without some restrictions on our adversarial oracle, perceptron learning is impossible. We will set some restrictions on the examples the oracle is allowed to present, in order to permit learning. These restrictions can be illustrated through the following figure, where length($\vec{w}^*$) = 1:



Terminology Note: It is traditional to call an instance $\vec{x}$ for which label($\vec{x}$) = 1 a *positive instance*, and otherwise a *negative instance*. This terminology should not be confused with with positive and negative numbers, and we will try to refer to the label of an instance rather than this "positive/negative example" terminology as much as possible to avoid confusion.

Perceptron Learning Algorithm

An on-line algorithm for learning perceptron functions [Rosenblatt 1958]:

Set $\vec{w}$ to all zeros
For each example $\vec{x}^{(i)}$
    If $\vec{w} \cdot \vec{x}^{(i)} \leq 0$
        change $\vec{w}$ to $\vec{w} + \vec{x}^{(i)}$

The *Perceptron Learning Algorithm Convergence Theorem* states that after a finite number of examples, the PLA will eventually always label all future examples correctly. Specifically, under our oracle restrictions, the PLA makes at most $\frac{1}{g^2}$ mistakes.