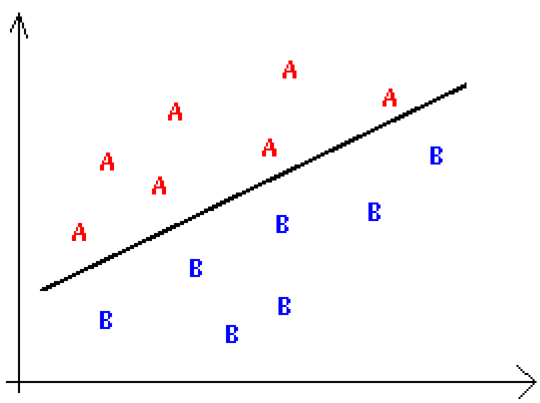
Transcription of Linear Classifier

1. Introduction

- Linear classifier use a decision hyperplane to perform classification.
- Linear classifier is simply and computational attractive.
- For getting a best classifier, we assume that all the giving vectors are linearly separable.
- For generalization, we will design an optimal linear classifier in which a linear classifier can not classify correctly all vectors by adopting an appropriate optimality criterion.
- An example of linear classifier for two class (class A, and class B):



2. Linear Discrimination Function and Decision Hyperplanes

- Consider a linear discriminant function for the two-class case. The respective decision hyperplane in the l-dimentional feature space is a hyperplane, that is

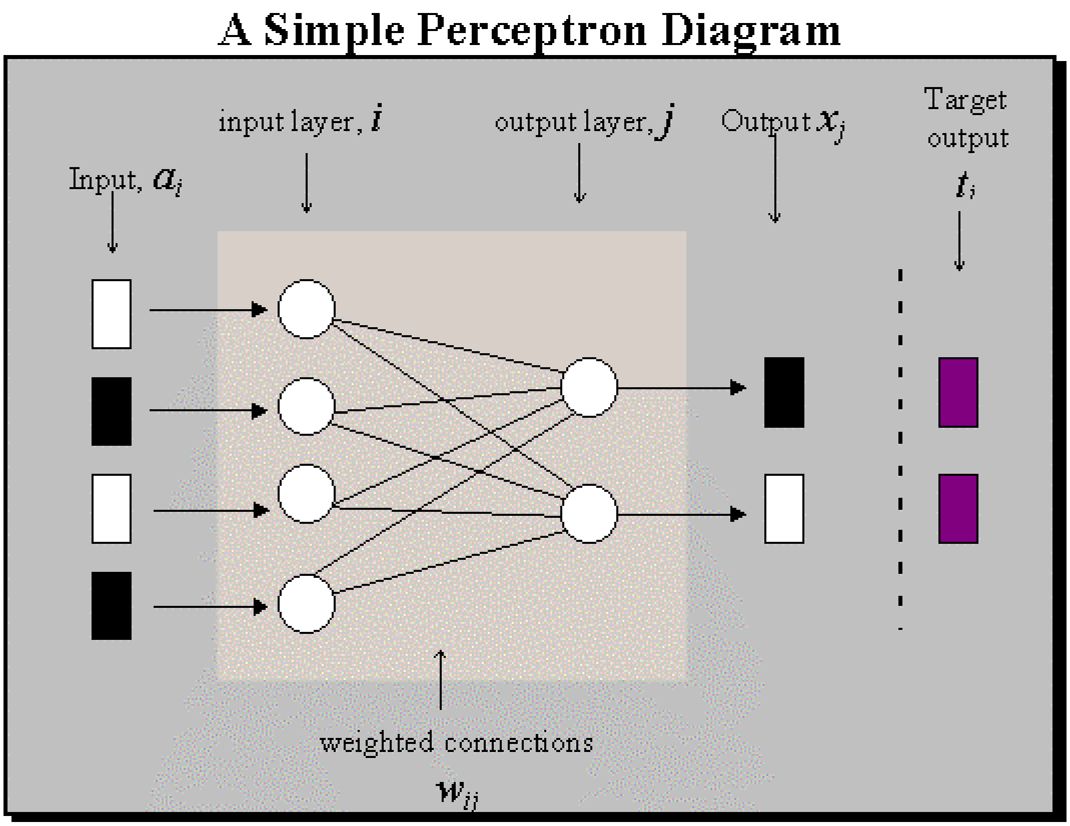
$$g(x) = w^T x + w_0 = 0$$

where $w = [w_1, w_2 ... w_l]^T$ is known as the weight vector and w_0 is the threshold. We predict w_2 if $\sum_i^l w_i x_i > w_0$, otherwise predict w_1 (assume we only have two classes).

. we use perceptron and winnow to find w_i's wieght.

2. The percetron algorithm

. Assume we have two classes and they are linear separatble, and assume we have a two layer perceptron classifier:



For a given w_0, we have error correction formula:

$$w(t+1) = w(t) + \rho(y - \hat{y})x(t)$$

where ρ is learning rate, $(y - \hat{y})$ moves the weights toward correct prediction for x.

. we have a cost function to measuer the correcteness and conservativeness of perceptron algorithm . Here the correctiveness is the tendency to improve on x(t), the conservativeness is the tendency to keep w(t+1) close to w(t).

$$U(w) = \| w(t+1) - w(t) \|_2^2 + \eta(y(t) - w(t+1)x(t))^2$$
$$= \sum_{i=1}^{l} (w_i(t+1) - w(t))^2 + \eta(y - \sum w_i(t+1)x(t))^2$$

take gradient w.r.t. $w(t+1)$ and set to 0. Approximate with

$0 = 2(w_i(t+1) - w_i(t)) - 2\eta(y - \sum_{i=1}^{l} w_i(t)x_i(t))x_i(t)$, which yield

$w_i(t+1) = w_i(t) + \eta(y - \sum_{i=1}^{l} w_i(t)x_i(t))x_i(t)$, applying threshold to summation yield

$$w(t+1) = w(t) + \rho(y - \hat{y})x(t)$$

. If classes linearly separable, then by cycling through vectors, we can guaranteed to converge in a finite numble of steps.

3. Winnow algorithm

. winnow is same with perceptron, but error correction function:

$$w_i(t+1) = w_i \exp(-2\eta(\hat{y}(t) - y(t))x_i(t)$$

. the cost function:

$$U(w) = \sum (w_i(t) - w_i(t+1) + w_i(t+1)\ln\frac{w_i(t+1)}{w_i(t)} + \eta(y - w(t+1)x(t))^2$$

take gradient w.r.t. w(t+1) and set to 0, approximate with

$0 = \ln\frac{w_i(t+1)}{w_i(t)} - 2\eta(y(t) - \sum_{i=1}^{l} w_i(t)x_i(t))x_i(t)$,

which yield:

$$w_i(t+1) = w_i \exp(-2\eta(\hat{y}(t) - y(t))x_i(t)$$

- winnow works well when most attributes are irrelevent.