Perceptrons

Barna Saha

The Machine Learning Model

- Training set: A training set consists of a set of pairs (x,y), called training examples, where
  - x is a vector of values, often called a feature vector
    - Can be categorical or numerical
  - y is the label, the classification value for x.
- The objective of the ML process is to discover the function y=f(x) that best predicts the value of y associated with each vector x
  - Example:
    - y is a real number: regression
    - y is a boolean value: binary classification
    - y is a member of some finite set: multiclass classification

Example

- Training set ([1], 2), ([2],1), ([3],4), ([4],3)
- Learn a linear function f(x)=ax+b that best represents the points of the training set.
  - Minimize with respect to a and b

$$\sum_{x=1}^{4}(ax+b-y_x)^2$$

  - a=3/5 and b=1

Perceptrons

- Perceptrons are threshold functions applied to the components of the vector $\boldsymbol{x}=(x_1, x_2, \ldots\ldots, x_d)$. A weight w_i is associated with the i-th component for each $i=1,2,\ldots,d$ and there is a threshold θ. The output is +1 if

$$\sum_{i=1}^{d} w_i x_i > \theta$$

and -1 otherwise

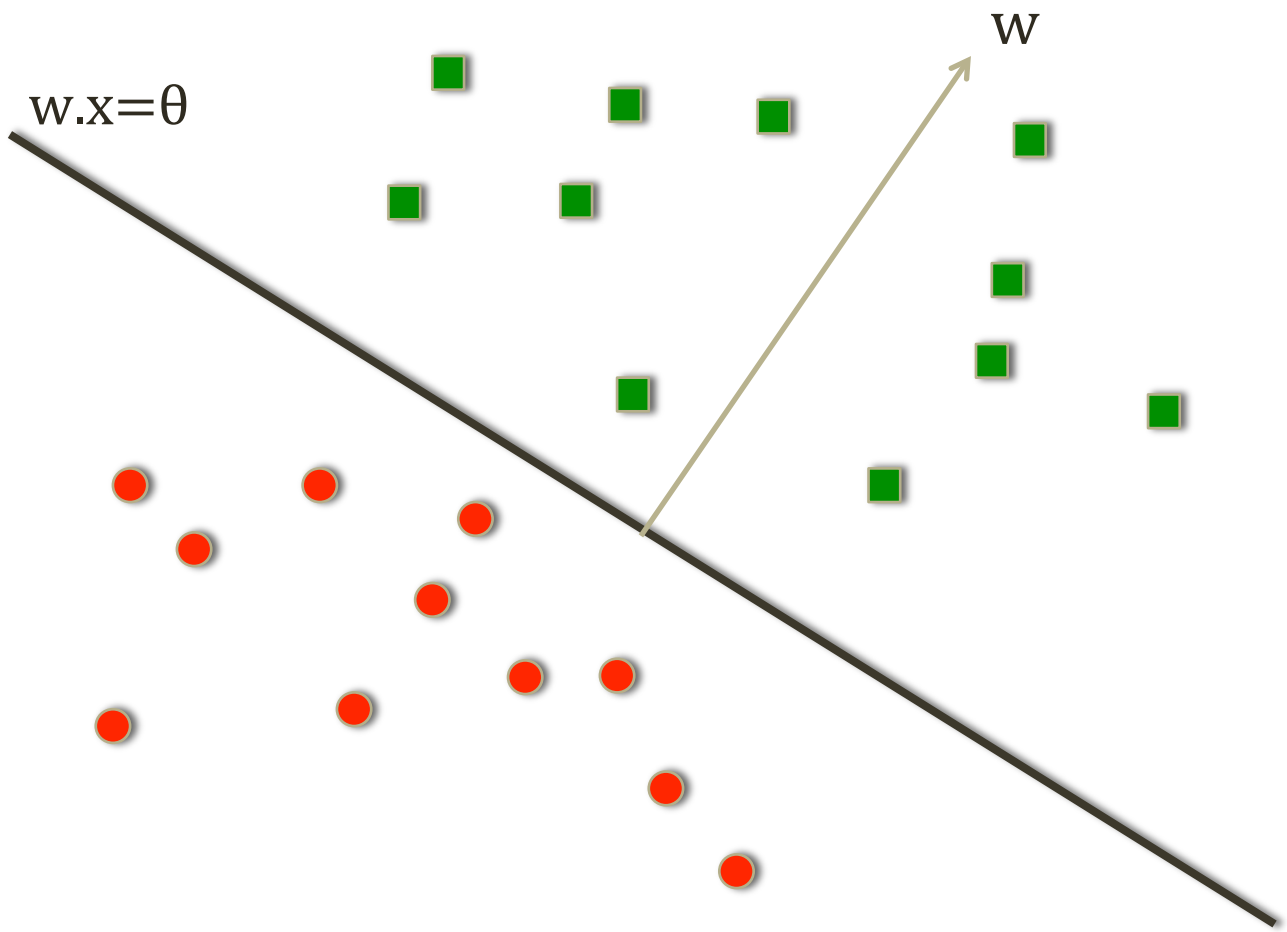
- Suitable for binary classification even when the number of features is very large.
- Neural nets are acyclic networks of perceptrons, with the outputs of some perceptrons used as inputs to others.

Exercise

- Exercise 12.1.1 of Leskovec et al.'s book
  - Requires f(x) to be a straight line passing through the origin
  - Requires f(x) to be quadratic

Perceptrons

- $w.x=\theta$



Perceptrons

- A perceptron classifier works only for data that is linearly separable, in the sense that there is some hyperplane that separates all the positive points from all the negative points.

- If there are many such hyperplanes, the perceptron will converge to one of them, and will thus correctly classify all the training data.

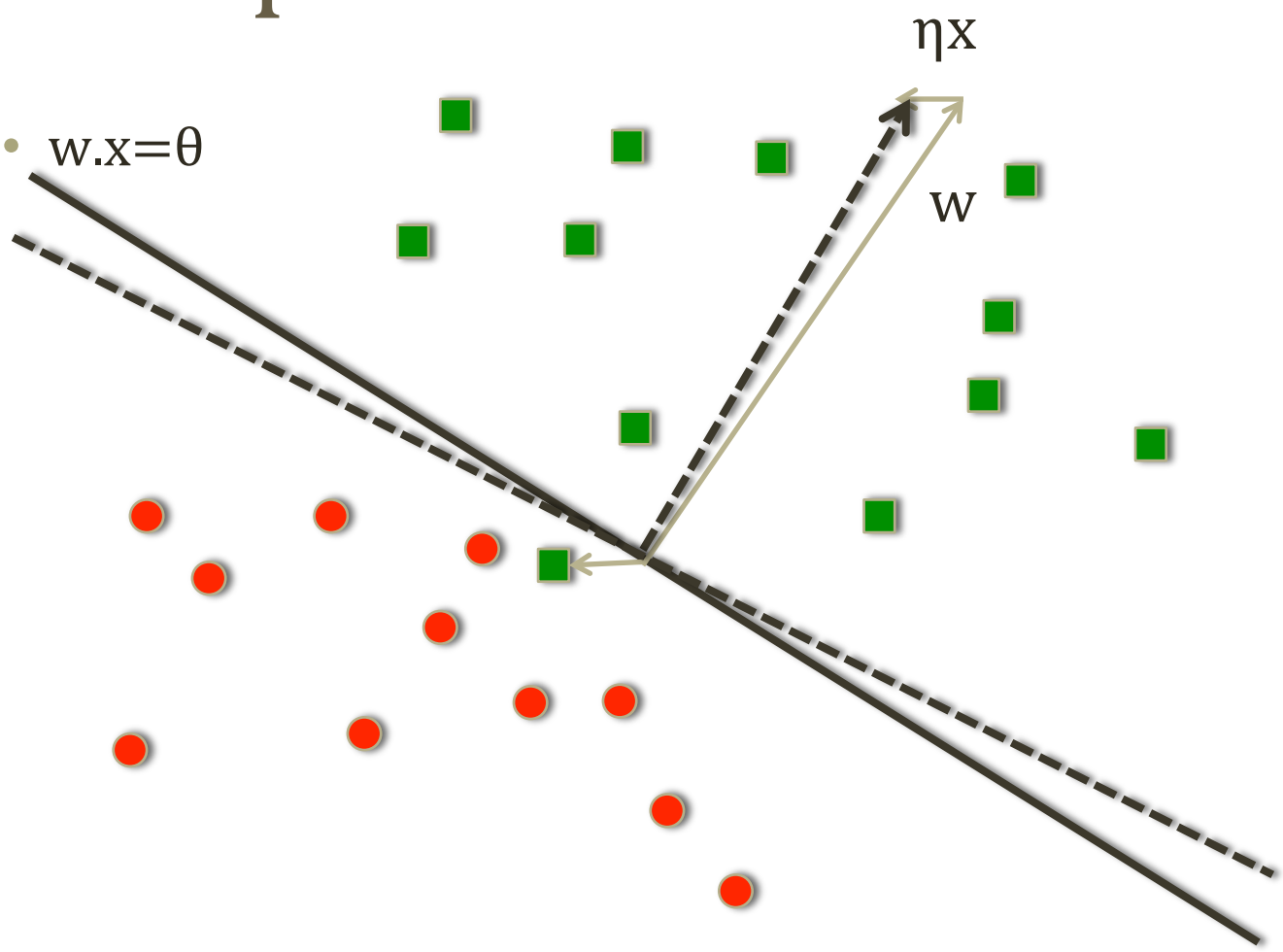
- If no such hyperplane exists, then the perceptron cannot converge to any particular one.

Training a Perceptron with Zero Threshold

- Initialize the weight vector w to all 0's.
- Pick a learning-rate parameter η, which is a small, positive real number.
- Consider each training example $t=(x,y)$ in turn
  - (a) Let $y'=w.x$
  - (b) If y' and y have the same sign, then do nothing; t is properly classified.
  - (c) However, if y' and y have different signs or $y'=0$, replace, w by $w=w+\eta yx$

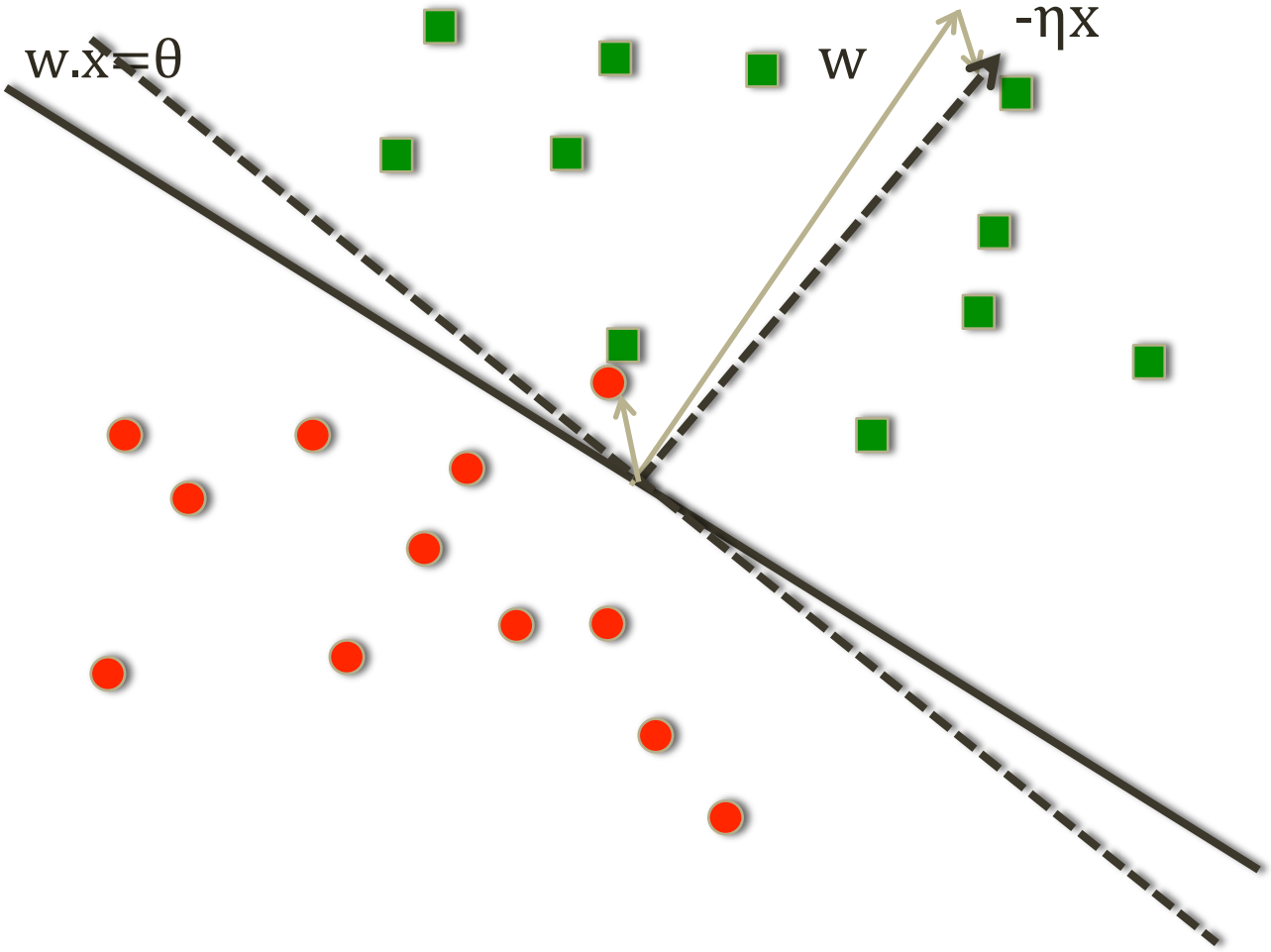
Perceptrons

- $w.x=\theta$



Perceptrons

- $w.x=\theta$



Example

- Training data:
- [1,1,0,1,1]→+1
- [0,0,1,1,0]→-1
- [0,1,1,0,0]→+1
- [1,0,0,1,0]→-1
- [1,0,1,0,1]→+1
- [1,0,1,1,0]→-1

Take $\eta=1/2$

Solution: w=[0,1,0,-1/2,1/2]

Convergence of Perceptrons

- Hard to tell if the data is linearly separable
  - Stop after a fixed number of iterations
  - Terminate when the number of misclassified points stop changing
  - Withhold a test set from the training data, and after each round, run the perceptron on the test data. Terminate the algorithm when the number of errors on the test set stops changing.
  - Lower the training rate with the number of iterations

Allowing the Threshold to Vary

- Replace the vector $\boldsymbol{w}=(w_1, w_2, \ldots\ldots, w_d)$ by

$$\boldsymbol{w}'=(w_1, w_2, \ldots\ldots, w_d, \theta)$$

- Replace every feature vector $\boldsymbol{x}=(x_1, x_2, \ldots\ldots, x_d)$ by

$$\boldsymbol{x}'=(x_1, x_2, \ldots\ldots, x_d, -1)$$

$w'.x' > 0$ is equivalent to $w.x-\theta > 0$

Why does Perceptron converge?

- Theorem: On any sequence of examples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t$, if there exists a vector w^* such that $x_t.w^* \geq 1$ for the positive examples and $x_t.w^* \leq -1$ for the negative examples, then the Perceptron algorithm makes at most $R^2|w^*|^2$ mistakes, where $R = \max_t |x_t|$

- Proof in board (pg 143-147 of Foundations of Data Science book by Blum et al.)

Why does Perceptron converge?

- Define "hinge-loss" of w* on a positive example x_t as $\max(0,1-x_t.w^*)$ and on a negative example x_t as $\max(0,1+x_t.w^*)$
- Define $L_{hinge}(w^*, S)$ as the sum of hinge-losses of w* on all examples in S.

- Theorem: On any sequence of examples $S=x_1,x_2,\ldots$, the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*|^2+2L_{hinge}(w^*,S)) \text{ mistakes,}$$

where $R=\max_t|x_t|$.