Introduction to Machine Learning

Perceptrons

Mingchen Gao

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
mgao8@buffalo.edu
Slides Adapted from Varun Chandola

Outline

Artificial Neurons

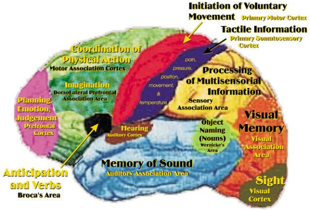


Figure: *Src: http://brainjackimage.blogspot.com/*

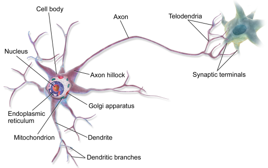
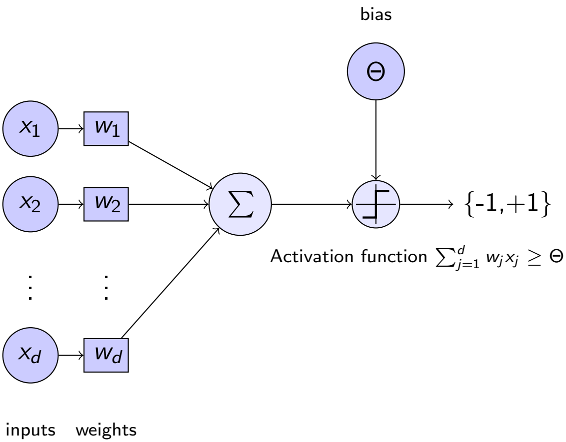


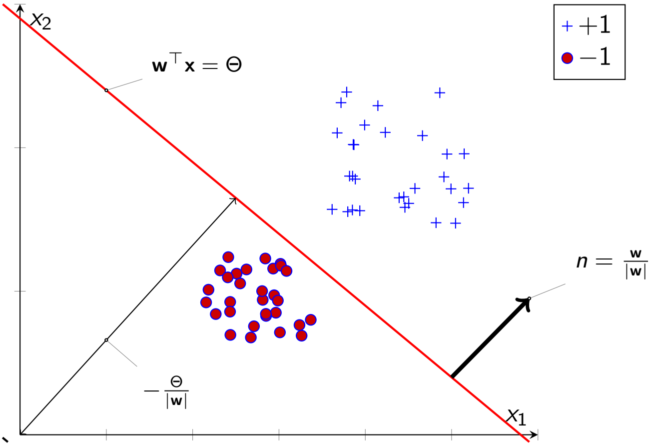
Figure: *Src: Wikipedia*

- Number of neurons 10^{10-11}
- Connections per neuron 10^{4-5}
- Switching time 0.001 seconds
- Scene recognition time 0.1 seconds
- Number of cycles per scene recognition 100

Perceptron [3, 1]



Geometric Interpretation



Eliminating Bias

- Add another attribute $x_{d+1} = 1$.
- w_{d+1} is $-\Theta$
- Desired hyperplane goes through origin in $(d+1)$ space

Hypothesis Space

- **Assumption**: $\exists \mathbf{w} \in \mathbb{R}^{d+1}$ such that **w** can *strictly* classify all examples correctly.
- *Hypothesis space*: Set of all hyperplanes defined in the $(d+1)$-dimensional space passing through origin
  - The target hypothesis is also called **decision surface** or **decision boundary**.

Perceptron Training - Perceptron Learning Rule

$\mathbf{w} \leftarrow (0, 0, \ldots, 0)_{d+1}$
for i=$1, 2, \ldots$ **do**
  if $\mathbf{w}^\top \mathbf{x}^{(i)} > 0$ **then**
    $c(\mathbf{x}^{(i)}) = +1$
  else
    $c(\mathbf{x}^{(i)}) = -1$
  end if
  if $c(\mathbf{x}^{(i)}) \neq c_*(\mathbf{x}^{(i)})$ **then**
    $\mathbf{w} \leftarrow \mathbf{w} + c_*(\mathbf{x}^{(i)})\mathbf{x}^{(i)}$
  end if
end for

- Every mistake *tweaks* the hyperplane
  - Rotation in $(d+1)$ space
  - Accomodate the offending point
- Stopping Criterion:
  - Exhaust all training examples, or
  - No further updates

Convergence Assumptions

1. Linearly separable examples
2. No errors
3. $|\mathbf{x}| = 1$
4. A positive δ gap exists that "contains" the target concept (hyperplane)
   - $(\exists\delta)(\exists\mathbf{v})$ such that $(\forall\mathbf{x})\mathbf{v}^\top\mathbf{x} > c_*(\mathbf{x})\delta$.

Perceptron Convergence Theorem

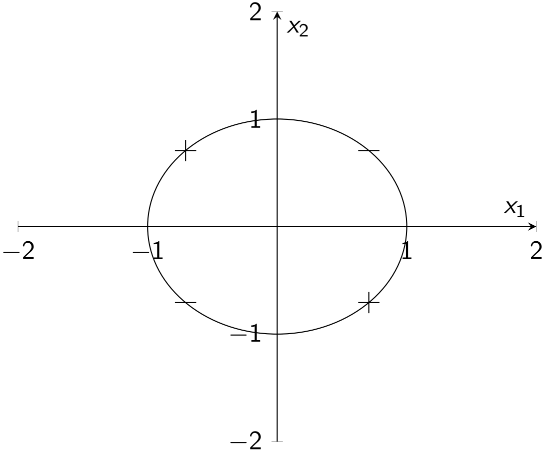
Theorem

For a set of unit length and linearly separable examples, the perceptron learning algorithm will converge after a finite number of mistakes (at most $\frac{1}{\delta^2}$).

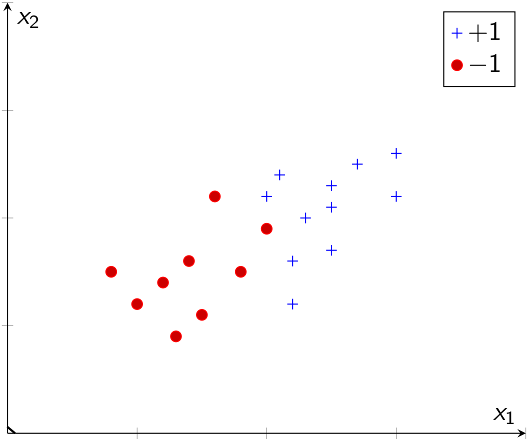
Proof discussed in Minsky's book [2].

Target concept $c_* \notin \mathcal{H}$

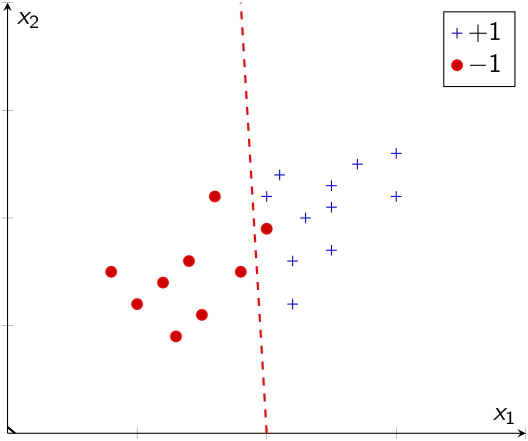
- Expand $\mathcal{H}$?
- Lower expectations
  - *Principle of good enough*



Perceptron Learning in Non-separable Case



Perceptron Learning in Non-separable Case

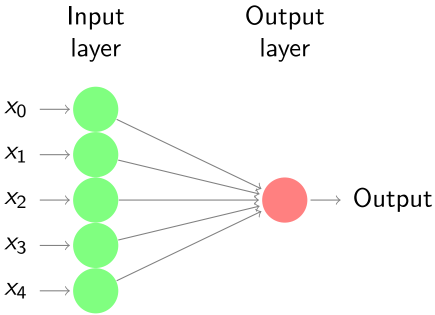


Gradient Descent and Delta Rule

- Which hyperplane to choose?
- Gives **best performance** on training data
  - Pose as an optimization problem
  - Objective function?
  - Optimization procedure?

Objective Function for Perceptron Learning

- An unthresholded perceptron (a linear unit)



- Training Examples: $\langle \mathbf{x}_i, y_i \rangle$
- Weight: $\mathbf{w}$

$$E(\mathbf{w}) = \frac{1}{2} \sum_i (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

References

W. Mcculloch and W. Pitts.
A logical calculus of ideas immanent in nervous activity.
Bulletin of Mathematical Biophysics, 5:127–147, 1943.

M. L. Minsky and S. Papert.
Perceptrons: An Introduction to Computational Geometry.
MIT Press, 1969.

F. Rosenblatt.
The perceptron: A probabilistic model for information storage and organization in the brain.
Psychological Review, 65(6):386–408, 1958.