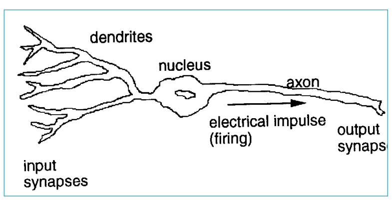
Machine Learning: Neural Networks

CSE 415: Introduction to Artificial Intelligence
University of Washington
Winter, 2018

Outline

- History of neural networks research
- The Perceptron
- Examples
- Training algorithm
- Fundamental training theorem

The Biological Neuron



The human brain contains approximately 10^{11} neurons.
Activation process:
Inputs are transmitted electrochemically across the input synapses
Input potentials are summed.
If the sum reaches a threshold, a pulse moves down the axon. (The neuron has "fired".)
The pulse is distributed at the axonal arborization to the input synapses of other neurons.
After firing, there is a refractory period of inactivity.

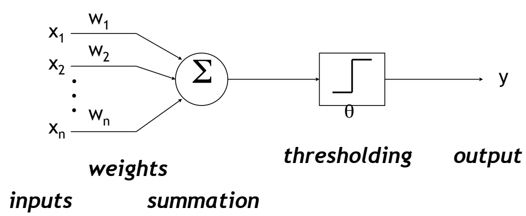
History of Neural Networks Research

1943 McCulloch & Pitts model of neuron.
$n_i(t+1) = \Theta(\Sigma_j w_{ij} n_j(t) - \mu_i)$, $\Theta(x) =$ 1 if $x \geq 0$;
0, otherwise.
1962 Frank Rosenblatt's book gives a training algorithm for finding the weights w_{ij} from examples.
Principles of neurodynamics: Perceptrons and the theory of brain mechanisms

1969 Marvin Minsky and Seymour Papert publish *Perceptrons*, and prove that 1-layer perceptrons are incapable of computing image connectedness.
1974-89, 1982: Associated content-addressable memory.

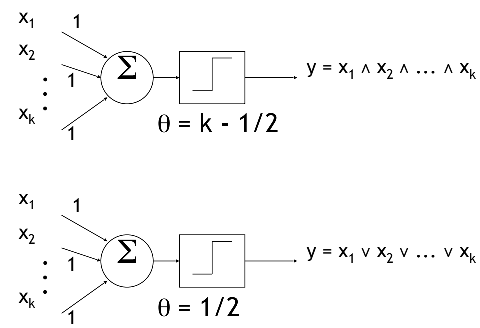
Backpropagation: Werbos 1974, Parker 1985, Rumelhart, Hinton, & Williams 1986.

The Perceptron



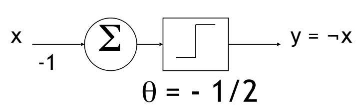
$$y = \begin{cases} 1 & \text{if } \Sigma\, w_i\, x_i \geq \theta; \\ 0, & \text{otherwise.} \end{cases}$$

Perceptron Examples: Boolean AND and OR.



$x_i \in \{0, 1\}$

Perceptron Examples: Boolean NOT



$x_i \in \{0, 1\}$

Perceptron Example: Template Matching

-1	-1	1	-1	-1
-1	1	-1	1	-1
1	1	1	1	1
1	-1	-1	-1	1
1	-1	-1	-1	1

weights w_1 through w_{25}

$x_i \in \{-1, 1\}$

$\theta = 25 - \varepsilon$

Recognizes the letter A provided the exact pattern is present.

Perceptron Training Sets

Let $X = X^+ \cup X^-$ be the set of training examples.

$S_X = \langle X_1, X_2, \ldots, X_k, \ldots \rangle$ is a *training sequence* on X, provided:

(1) Each X_k is a member of X, and

(2) Each element of X occurs infinitely often in S_X.

An element *e* occurs *infinitely often* in a sequence

$z = \langle z_1, z_2, \ldots \rangle$

provided that for any nonzero integer *i*, there exists a nonnegative integer *j* such that there is an occurrence of *e* in

$z_i, z_{i+1}, \ldots , z_j$.

Perceptron Training Algorithm

Let $X = X^+ \cup X^-$ be the set of training examples. and let $S_X = \langle X_1, X_2, \ldots, X_k, \ldots \rangle$ be a training sequence on X.

Let w_k be the weight vector at step *k*, and let θ_k be the threshold at step *k*.

Choose w_0 and θ_0 arbitrarily. For example. $w_0 = (0, 0, \ldots, 0)$, $\theta_0 = 1$.

Each each step *k*, *k* = 0, 1, 2, . . .

Classify X_k using w_k.

If X_k is correctly classified, take $w_{k+1} = w_k$.

If X_k is in X^- but misclassified, take $w_{k+1} = w_k - c_k X_k$ and $\theta_{k+1} = \theta_k + c_k$

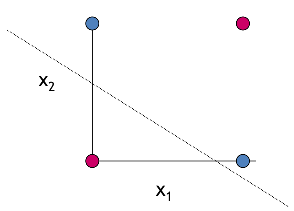
If X_k is in X^+ but misclassified, take $w_{k+1} = w_k + c_k X_k$ and $\theta_{k+1} = \theta_k - c_k$

The sequence c_k should be chosen according to the data. Overly large constant values can lead to oscillation during training. Values that are too small will increase training time. However, $c_k = c_0/k$ will work for any positive c_0.

Perceptron Limitations

Perceptron training always converges if the training data X^+ and X^- are linearly separable sets.

The boolean function XOR (exclusive or) is not linearly separable. (Its positive and negative instances cannot be separated by a line or hyperplane.) It cannot be computed by a single-layer perceptron. It cannot be learned by a single-layer perceptron.

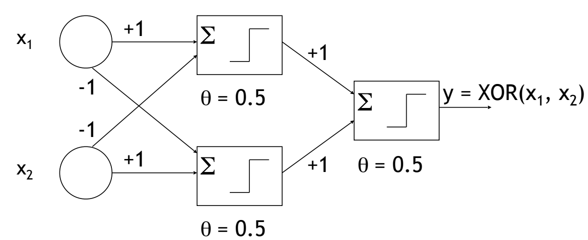


$X^+ = \{ (0, 1), (1, 0) \}$

$X^- = \{ (0, 0), (1, 1) \}$

$X = X^+ \cup X^-$

Two-Layer Perceptrons



Two-Layer Perceptrons (cont.)

Two-Layer perceptrons are computationally powerful.

However: they are not trainable with a method such as the perceptron training algorithm, because the threshold units in the middle level “block” updating information; there is no way to know what the correct updates to first-level weights should be.