Lecture 8.
Learning (V):
Perceptron Learning
OUTLINE

• Perceptron Model
• Perceptron learning algorithm
• Convergence of Perceptron learning algorithm
• Example
PERCEPTRON

• Consists of a single neuron with threshold activation, and binary output values.

• Net function $u(x) = w_0 + \sum_{i=1}^{M} w_i x_i = 0$ defines a hyper plane that partitions the feature space into two half spaces.

\[
y = \begin{cases} 
1 & \sum_{i=0}^{M} w_i x_i > 0; \\
0 & \text{Otherwise.}
\end{cases}
\]
Perceptron Learning Problem

• Problem Statement: Given training samples \( D = \{(x(k); t(k)); 1 \leq k \leq K\}, t(k)\in\{0, 1\} \) or \{−1, 1\}, find the weight vectors, \( W \) such that the number of outputs which match the target value, that is,

\[
\sum_{k=1}^{K} y(k) \oplus t(k)
\]

is maximized.

• Comment: This corresponds to solving \( K \) linear in-equality equations for \( M \) unknown variables – A linear programming problem.

Example. \( D = \{(1;1), (3;1), (-0.5;-1), (-2;-1)\}. 4 \) inequalities:

| \( (1,1) \) | \( w_1 \cdot 1 + w_0 > 0 \) | \( (3,1) \) | \( w_1 \cdot 3 + w_0 > 0 \) |
| \( (-0.5,-1) \) | \( w_1 \cdot (-0.5) + w_0 < 0 \) | \( (-2,-1) \) | \( w_1 \cdot (-2) + w_0 < 0 \) |
**Perceptron Example**

- **A linear-separable problem**: If patterns can be separated by a linear hyper-plane, than the solution space is a non-empty set.

<table>
<thead>
<tr>
<th>Data Space</th>
<th>Solution space</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1): ( w_1 \cdot 1 + w_0 &gt; 0 );</td>
<td>( w_1 \cdot 1 + w_0 &gt; 0 )</td>
</tr>
<tr>
<td>(-0.5,-1): ( w_1 \cdot (-0.5) + w_0 &lt; 0 );</td>
<td>( w_1 \cdot (-0.5) + w_0 &lt; 0 )</td>
</tr>
<tr>
<td>(3,1): ( w_1 \cdot 3 + w_0 &gt; 0 )</td>
<td>( w_1 \cdot (-2) + w_0 &lt; 0 )</td>
</tr>
</tbody>
</table>
Perceptron Learning Rules and Convergence Theorem

• Perceptron learning rule: (\( \eta > 0 \): Learning rate)

\[
W(k+1) = W(k) + \eta (t(k) - y(k)) x(k)
\]

**Convergence Theorem** – If \((x(k), t(k))\) is linearly separable, then \(W^*\) can be found in finite number of steps using the perceptron learning algorithm.

• Problems with Perceptron:
  – Can solve only linearly separable problems.
  – May need large number of steps to converge.
Proof of Perceptron Learning Theorem

• Assume $w^*$ is the optimal weights, then the reduction of errors in successive iterations are:
\[
|w(k+1) - w^*|^2 - |w(k) - w^*|^2 = A \eta^2 + 2 \eta B
\]
\[
= \eta^2 |x(k)|^2 [t(k) - y(k)]^2 + 2 \eta [w(k) - w^*][t(k) - y(k)]x(k)
\]  

• If $y(k) = t(k)$, RHS of (*) is 0. Hence only consider $y(k) \neq t(k)$. That is, only $t(k) - y(k) = \pm 1$ need to be considered. Thus, $A = |x(k)|^2 > 0$.

• Case I. $t(k) = 1$, $y(k) = 0$ \(\Rightarrow w^T(k)x(k) < 0\), and $(w^*)^T x(k) > 0$

• Case II. $t(k) = 0$, $y(k) = 1$ \(\Rightarrow w^T(k)x(k) > 0\), and $(w^*)^T x(k) < 0$.

In both cases, $B < 0$. Hence for $0 < \eta < -2B/A$, (*) < 0
Perceptron Learning Example

4 data points: \( \{x_1(i), \ x_2(i); \ t(i); \ i = 1,2,3,4\} = (-1,1;0), (-.8, 1; 1), (0.8, -1; 0), (1, -1; 1) \).

Initialize randomly, say, \( \mathbf{w}(1) = [0.3 \ -0.5 \ 0.5]^T \). \( \eta = 1 \)

\[
y(1) = \text{sgn}([1 -1 1] \cdot \mathbf{w}(1)) = \text{sgn}(1.3) = 1 \neq t(1) = 0
\]

\[
\mathbf{w}(2) = \mathbf{w}(1) + 1 \cdot (t(1)-y(1)) \mathbf{x}(1)
= [0.3 \ -0.5, \ 0.5]^T + 1 \cdot (-1) \cdot [1 \ -1 \ 1]^T = [-0.7 \ 0.5 \ -0.5]^T
\]

\[
y(2) = \text{sgn}([1 -0.8 \ 1] \cdot [-0.7 \ 0.5 \ -0.5]^T) = \text{sgn}[-1.6]=0
\]

\[
\mathbf{w}(3)=[-0.7 \ 0.5 \ -0.5]^T + 1 \cdot (1-0) \cdot [1 -0.8 \ 1]^T = [.3 \ -.3 \ .5]^T
\]

\[
y(3), \mathbf{w}(4), \ y(4), \cdots \text{ can be computed in the same manner.}
\]
Perceptron Learning Example

(a) Initial decision boundary
(b) Final decision boundary
Perceptron and Linear Classifier

• Perceptron can be used as a pattern classifier:

For example, sort eggs into medium, large, jumble. Features: weight, length, and diameter.

• A linear classifier forms a (loosely speaking) linear weighted function of feature vector $x$:

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

and then makes a decision based on if $g(x) \leq 0$. 
Linear Classifier Example

- Jumble Egg Classifier decision rule:
  If \( w_0 + w_1 \times \text{weight} + w_2 \times \text{length} > 0 \) then Jumble egg
  Let \( x_1: \text{weight}, x_2: \text{length} \), then
  \[ g(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2 = 0 \]
  is a hyperplane (straight line in 2D space).

\[
PQ \text{ (Distance)} = \frac{w_0}{\sqrt{w_1^2 + w_2^2}}
\]

\([w_1 \ w_2]\) is the normal vector perpendicular to the straight line \( g(x_1, x_2) = 0 \)
Limitations of Perceptron

- If the two classes of feature vectors are *linearly separable*, then a linear classifier, implemented by a perceptron, can be applied.
- **Question**: How about using perceptron to implement the Boolean XOR function that is linearly non-separable!

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>