# Inf2b Learning and Data

## Lecture 12: Single layer Neural Networks (1)

*Hiroshi Shimodaira*
*(Credit: Iain Murray and Steve Renals)*

Centre for Speech Technology Research (CSTR)
School of Informatics
University of Edinburgh

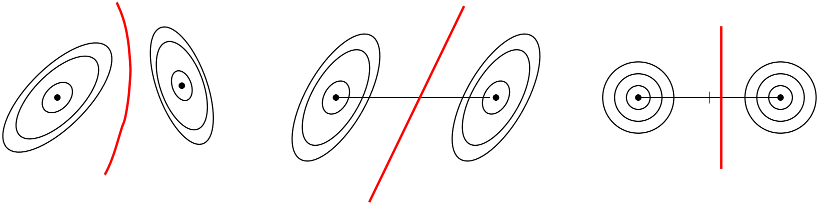
Jan-Mar 2014

# Today's Schedule

1. Discriminant functions (recap)
2. Decision boundary of linear discriminants
3. Discriminants for multiple classes
4. Training of linear discriminant functions
5. Perceptron

# Discriminant functions (recap)

$$
\begin{aligned}
y_c(\mathbf{x}) &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_c| + \ln P(c) \\
&= -\frac{1}{2}\mathbf{x}^T \boldsymbol{\Sigma}_c^{-1}\mathbf{x} + \boldsymbol{\mu}_c^T \boldsymbol{\Sigma}_c^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}_c^{-1}\boldsymbol{\mu}_c - \frac{1}{2}\ln|\boldsymbol{\Sigma}_c| + \ln P(c
\end{aligned}
$$



(a) (b) (c)

# Linear discriminants for a 2-class problem

$$y_1(\mathbf{x}) = \mathbf{w}_1^T \mathbf{x} + w_{10}$$
$$y_2(\mathbf{x}) = \mathbf{w}_2^T \mathbf{x} + w_{20}$$

Combined discriminant function:

$$\begin{aligned} y(\mathbf{x}) = y_1(\mathbf{x}) - y_2(\mathbf{x}) &= (\mathbf{w}_1 - \mathbf{w}_2)^T \mathbf{x} + (w_{10} - w_{20}) \\ &= \mathbf{w}^T \mathbf{x} + w_0 \end{aligned}$$

Decision:

$$C = \begin{cases} 1, & \text{if } y(\mathbf{x}) \geq 0, \\ 2, & \text{if } y(\mathbf{x}) < 0 \end{cases}$$

# Decision boundary of linear discriminants

- Decision boundary:

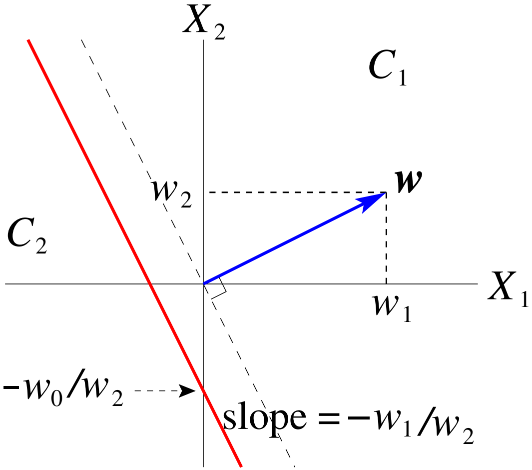
  $$\mathbf{w}^T\mathbf{x} + w_0 = 0$$

| Dimension | Decision boundary | |
|---|---|---|
| 2 : | line | $w_1x_1 + w_2x_2 + w_0 = 0$ |
| 3 : | plane | $w_1x_1 + w_2x_2 + w_3x_3 + w_0 = 0$ |
| d : | hyperplane | $\left(\sum_{i=1}^{d} w_ix_i\right) + w_0 = 0$ |

NB: $\mathbf{w}$ is a normal vector to the hyperplane

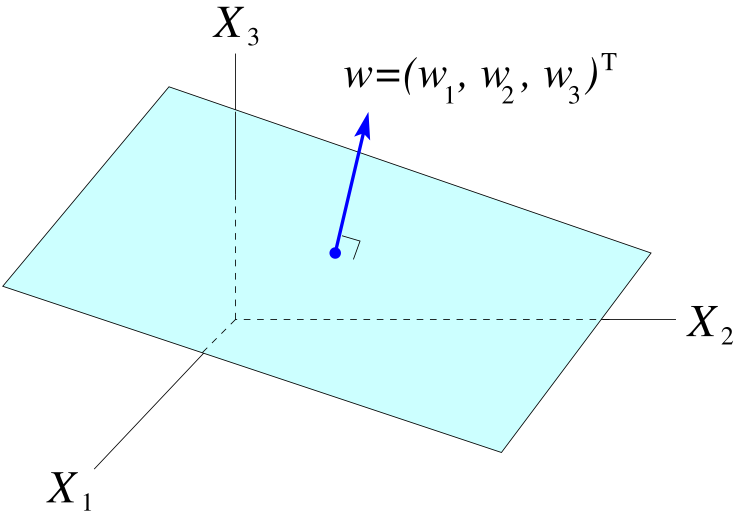
# Decision boundary of linear discriminant (2D)

$w_1 x_1 + w_2 x_2 + w_0 = 0$



# Decision boundary of linear discriminant (3D)

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0 = 0$$



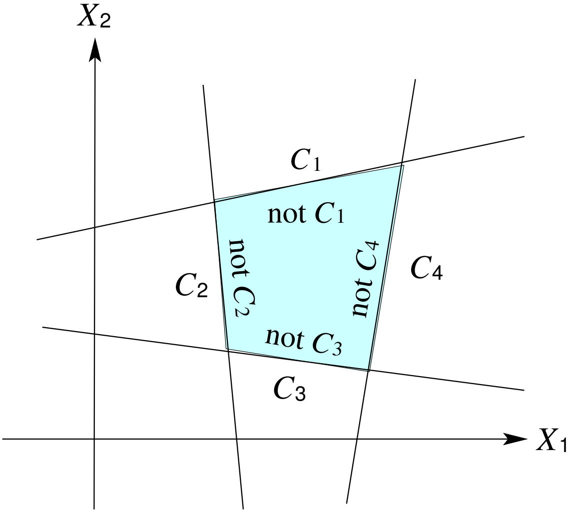
# Discriminants for multiple classes ($K > 2$)

- One-versus-the-rest classifiers $\cdots$ $K$ classifiers
  E.g. $K = 4$

$$
\begin{aligned}
y_1(\mathbf{x}) &= \mathbf{w}_1^T \mathbf{x} + w_{10} \\
y_2(\mathbf{x}) &= \mathbf{w}_2^T \mathbf{x} + w_{20} \\
y_3(\mathbf{x}) &= \mathbf{w}_3^T \mathbf{x} + w_{30} \\
y_4(\mathbf{x}) &= \mathbf{w}_4^T \mathbf{x} + w_{40}
\end{aligned}
$$

- What if $\forall i : \; y_i(\mathbf{x}) < 0$ ?

# Discriminants for multiple classes ($K > 2$)



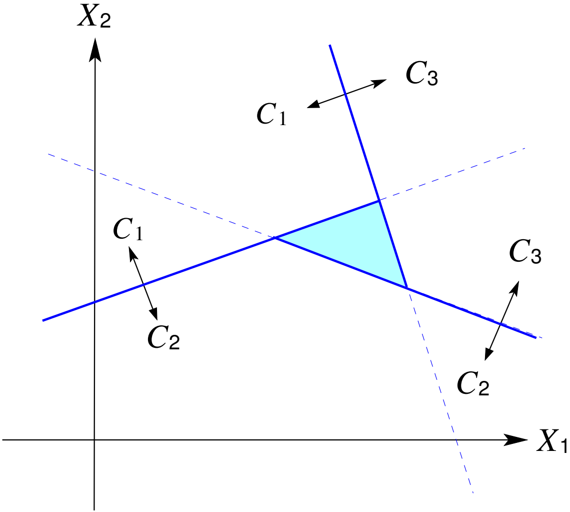
# Multi-class discriminants (one-vs-one)

- One-versus-one classifiers : $\cdots$ $K(K-1)/2$ classifiers
  E.g. $K = 3$

$$
\begin{aligned}
y_{12}(\mathbf{x}) &= \mathbf{w}_{12}^T\mathbf{x} + w_{12,0} \\
y_{23}(\mathbf{x}) &= \mathbf{w}_{23}^T\mathbf{x} + w_{23,0} \\
y_{31}(\mathbf{x}) &= \mathbf{w}_{31}^T\mathbf{x} + w_{31,0}
\end{aligned}
$$

- What if $y_{12}(\mathbf{x}) < 0,\ y_{23}(\mathbf{x}) < 0,\ y_{31}(\mathbf{x}) < 0$ ?

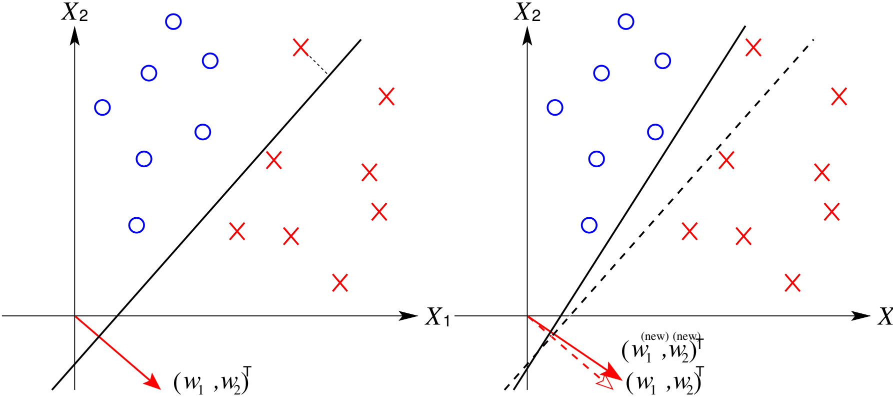
# Multi-class discriminants (one-vs-one)



# Training of linear discriminant functions

- A discriminant for a two-class problem:

$$
\begin{aligned}
y(\mathbf{x}) = y_1(\mathbf{x}) - y_2(\mathbf{x}) &= (\mathbf{w}_1 - \mathbf{w}_2)^T \mathbf{x} + (w_{10} - w_{20}) \\
&= \mathbf{w}^T \mathbf{x} + w_0
\end{aligned}
$$



# Error correction algorithm

$$a(\dot{\mathbf{x}}) = \mathbf{w}^T\mathbf{x} + w_0 \;=\; \dot{\mathbf{w}}^T\dot{\mathbf{x}}$$

where $\dot{\mathbf{w}} = (w_0, \mathbf{w}^T)^T,\; \dot{\mathbf{x}} = (1, \mathbf{x}^T)^T$

*Let's just use* $\mathbf{w}$ *and* $\mathbf{x}$ *to denote* $\dot{\mathbf{w}}$ *and* $\dot{\mathbf{x}}$ *from now on!*

$$y(\mathbf{x}) = g(\,a(\mathbf{x})\,) \;=\; g(\,\mathbf{w}^T\mathbf{x}\,)$$

$$\text{where } g(a) = \begin{cases} +1, & \text{if } a \geq 0, \\ -1, & \text{if } a < 0 \end{cases}$$

- Training set : $D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(N)}, t^{(N)})\}$
  where $t^{(i)} \in \{-1, +1\}$
- If $y(\mathbf{x}^{(i)}) = -1$ for $t^{(i)} = +1$,
  $$\mathbf{w}^{(\mathrm{new})} \leftarrow \mathbf{w} + \eta\,\mathbf{x}^{(i)} \qquad (\eta > 0)$$
- If $y(\mathbf{x}^{(i)}) = +1$ for $t^{(i)} = -1$,
  $$\mathbf{w}^{(\mathrm{new})} \leftarrow \mathbf{w} - \eta\,\mathbf{x}^{(i)} \qquad (\eta > 0)$$

# The Perceptron criterion

- The number of misclassification:

$$E = \sum_{i=1}^{N} |\, t^{(i)} - y(\mathbf{x}^{(i)}) \,| \,/\, 2$$

- Instead of calculating the number of misclassification, use the following error measure — Perceptron criterion:

$$E_p(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^T \mathbf{x}^{(n)}\, t^{(n)}$$

where $\mathcal{M}$ : a set of all misclassified samples.

| | |
|---|---|
| No misclassification | $E_p(\mathbf{w}) = 0$ |
| Misclassification | $E_p(\mathbf{w}) > 0$ |

$$\nabla E_p(\mathbf{w}) = (\partial E_p / \partial w_i) = -\nabla \sum_{n \in \mathcal{M}} \mathbf{w}^T \mathbf{x}^{(n)}\, t^{(n)}$$

$$= -\sum_{n \in \mathcal{M}} \mathbf{x}^{(n)}\, t^{(n)}$$

# The Perceptron learning algorithm

Batch Perceptron algorithm:

$$
\begin{aligned}
&\mathcal{M} \leftarrow \phi \\
&\text{for } i = 1, \ldots, N \\
&\quad \text{if } \; g(\mathbf{w}^T \mathbf{x}^{(i)}) \neq t^{(i)} \\
&\quad\quad \mathcal{M} \leftarrow \{\mathcal{M}, i\} \\
&\mathbf{w}^{(\mathrm{new})} \leftarrow \mathbf{w} + \eta \sum_{n \in \mathcal{M}} \mathbf{x}^{(n)} \, t^{(n)}
\end{aligned}
$$

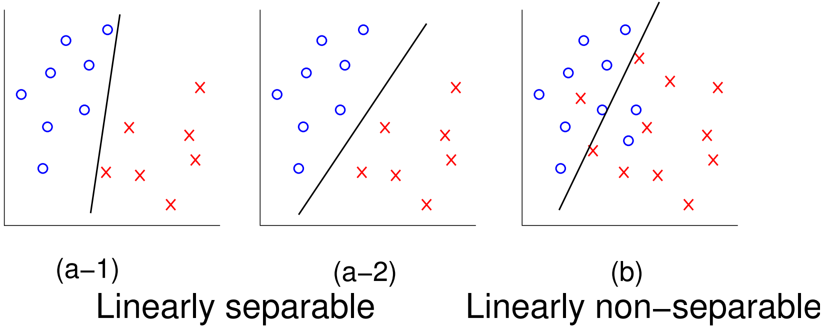
Incremental (online) Perceptron algorithm:

$$
\begin{aligned}
&\text{for } i = 1, \ldots, N \\
&\quad \text{if } \; g(\mathbf{w}^T \mathbf{x}^{(i)}) \neq t^{(i)} \\
&\quad\quad \mathbf{w}^{(\mathrm{new})} \leftarrow \mathbf{w} + \mathbf{x}^{(i)} \, t^{(i)}
\end{aligned}
$$

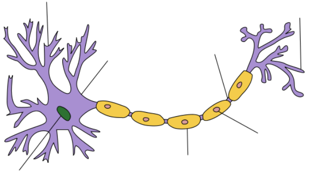
What about convergence?

The Perceptron learning algorithm terminate if training samples are linearly separable.

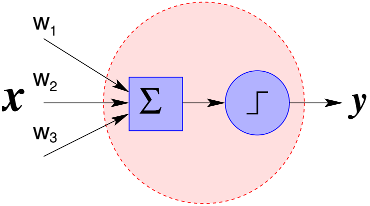
# Linearly separable vs linearly non-separable



# Background of Perceptron



(http://en.wikipedia.org/wiki/File:Neuron_Hand-tuned.svg)

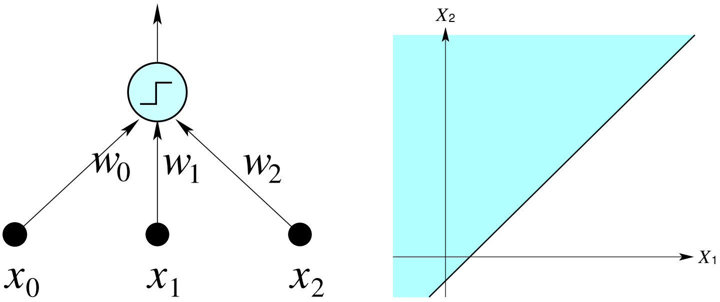



(a) function unit

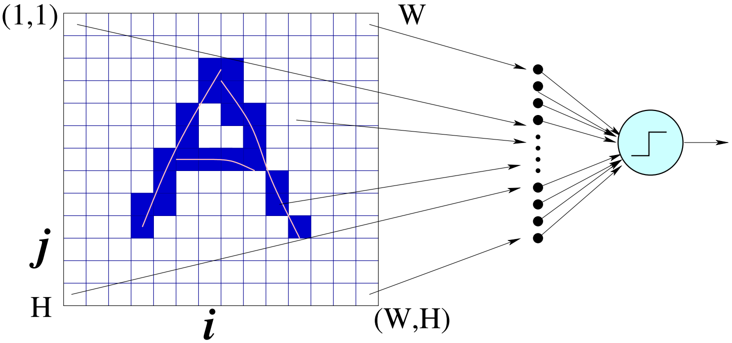
| | |
|---|---|
| 1940s | Warren McCulloch and Walter Pitts : 'threshold logic' |
| | Donald Hebb : 'Hebbian learning' |
| 1957 | Frank Rosenblatt : 'Perceptron' |

# Perceptron architectures and decision boundaries

$$y(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$$



# Character recognition by Perceptron



# Perceptron as a logical function

**NOT**

| $x_1$ | $y$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

**OR**

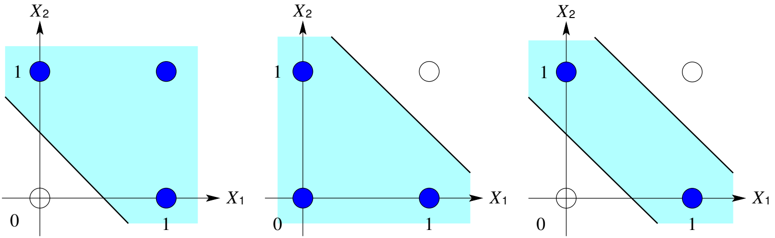
| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**NAND**

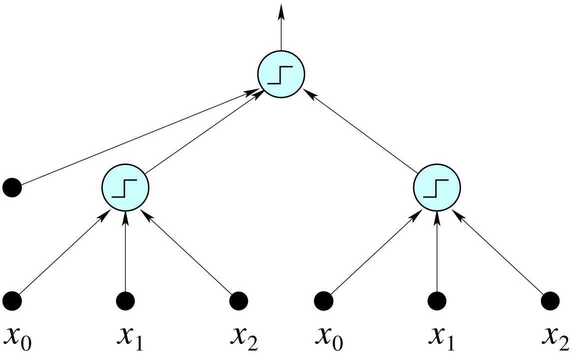
| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

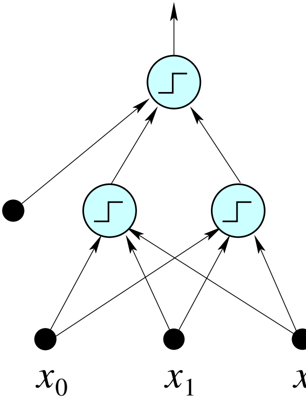
**EXOR**

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

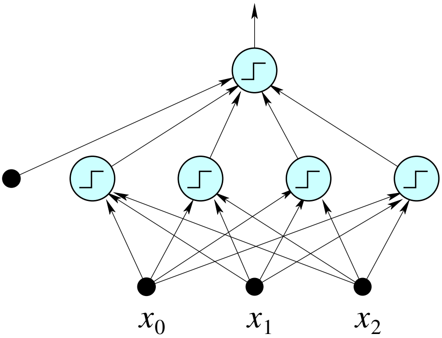


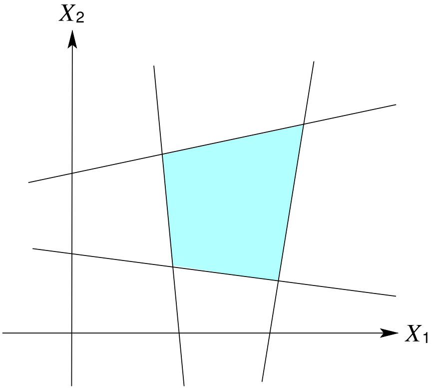
# Perceptron architectures and decision boundaries



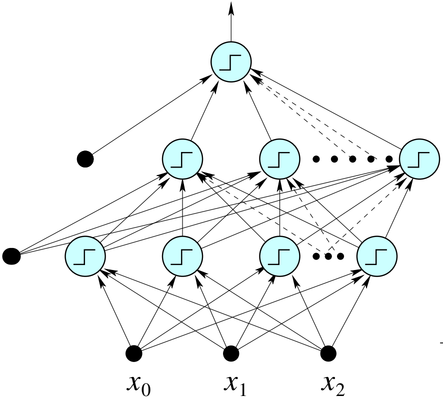



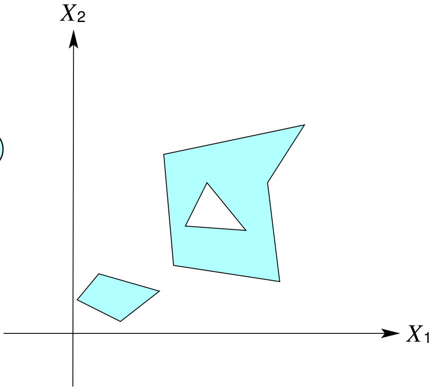
# Perceptron architectures and decision boundaries



# Perceptron architectures and decision boundaries



# Problems with Perceptron

- Non convergence if the training data are linearly non-separable
- Difficulty training multiple-layer Perceptrons
- Piecewise-linear decision boundaries

# Summary

- Decision boundaries of linear discriminant functions
- Discriminant functions for multiple classes
- Training discriminant functions directly
- Perceptrons