A Neuron for Classification

Introduction to Neural Networks

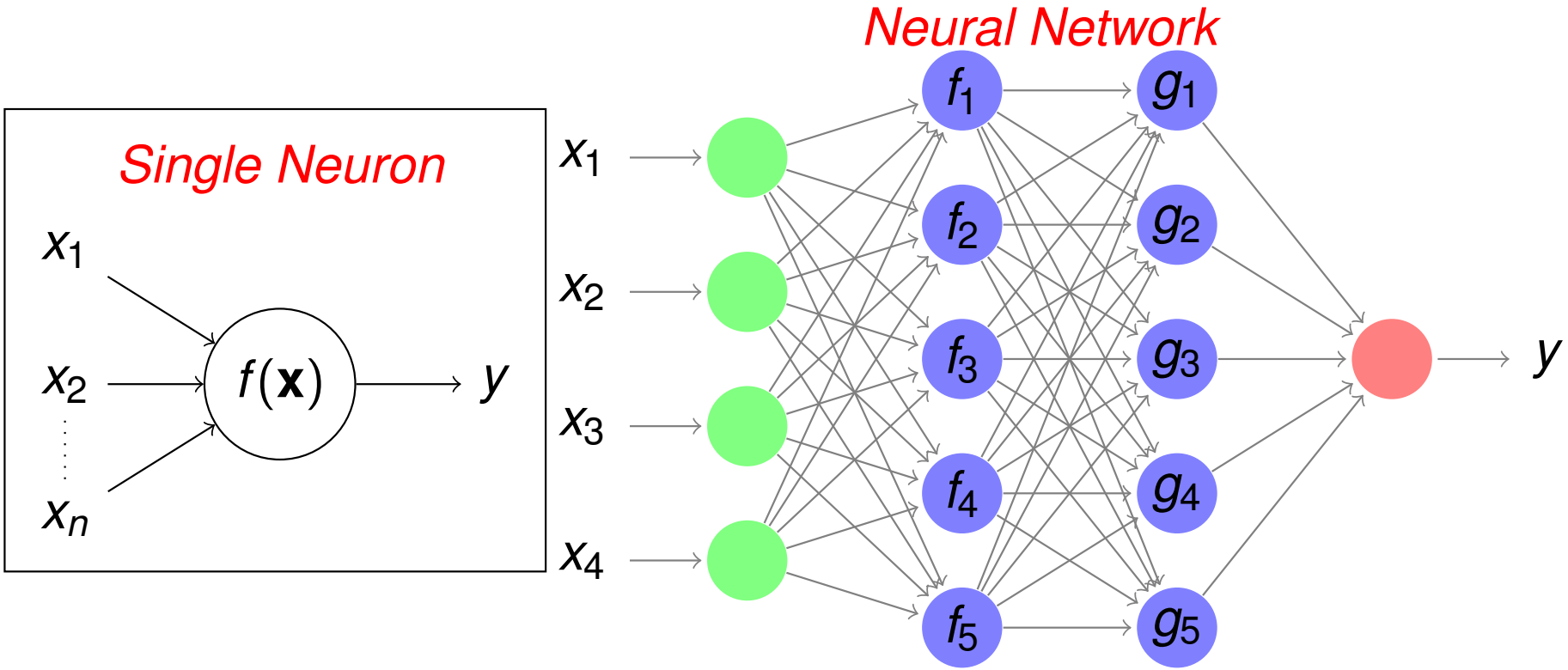
Prof Hans Georg Schaathun

Høgskolen i Ålesund

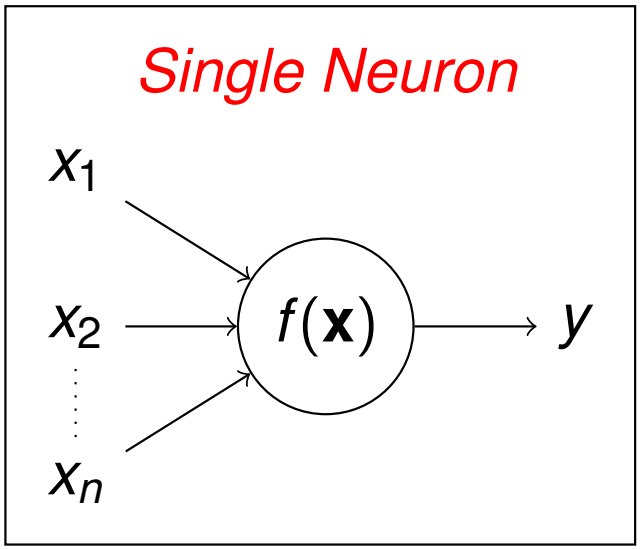
18th January 2016

Outline

Imitating a Human Brain



The Neuron as a Function



— Input observed features (data)

— Output decision

Many names: Classification problems, Pattern recognition, Machine learning, Artificial Intelligence

Data sets

`http://archive.ics.uci.edu/ml/`

Element	Purpose	Type
Feature vector	Input for the classifier (e.g. neural network)	Floating point vector (List of Double)
Class label	Correct output from the classifier	Discrete (String, Integer, Character)
Other data	Record ID, etc. (should be ignored)	Anything

Iris Data Set

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa (...)
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor (...)
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica
6.3,2.9,5.6,1.8,Iris-virginica
6.5,3.0,5.8,2.2,Iris-virginica
7.6,3.0,6.6,2.1,Iris-virginica
4.9,2.5,4.5,1.7,Iris-virginica (...)
```

Objective

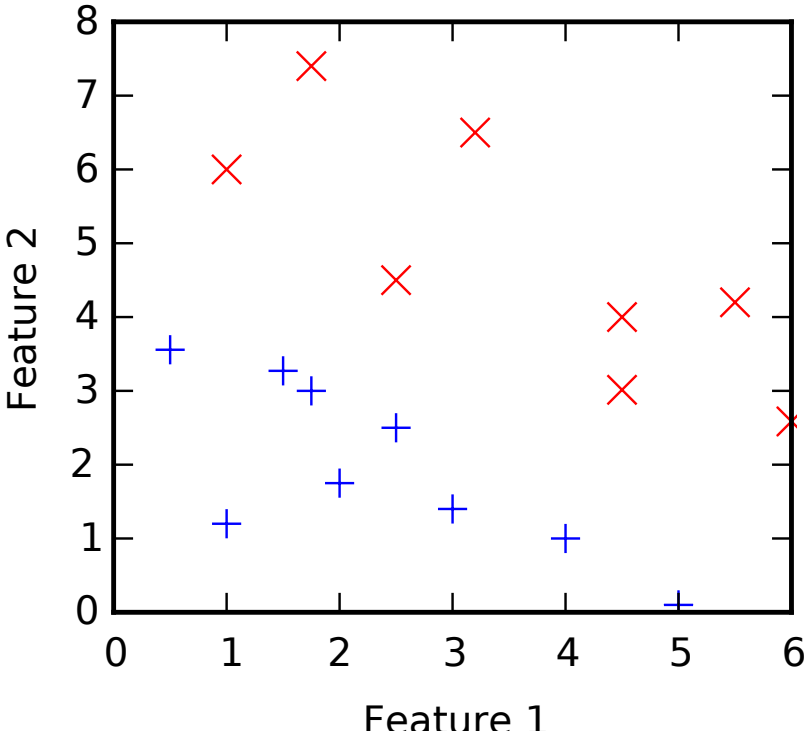
1. Input: feature vector $\mathbb{R}^4$
   - in Haskell: `[Double]` of length four
2. Output: species
   - Either Iris-setosa, Iris-versicolor, or Iris-virginica

Objective

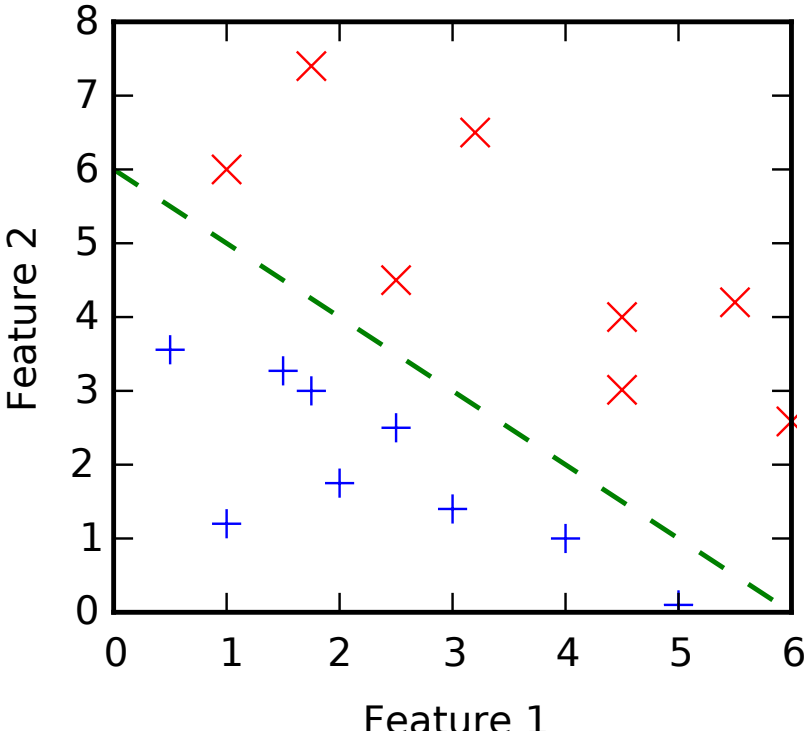
1. Input: feature vector $\mathbb{R}^4$
   - in Haskell: `[Double]` of length four
2. Output: species
   - Either Iris-setosa, Iris-versicolor, or Iris-virginica
3. String labels may be awkward

Iris-setosa	$+1$	$(1, 0, 0)$
Iris-versicolor	0	$(0, 1, 0)$
Iris-virginica	-1	$(0, 0, 1)$

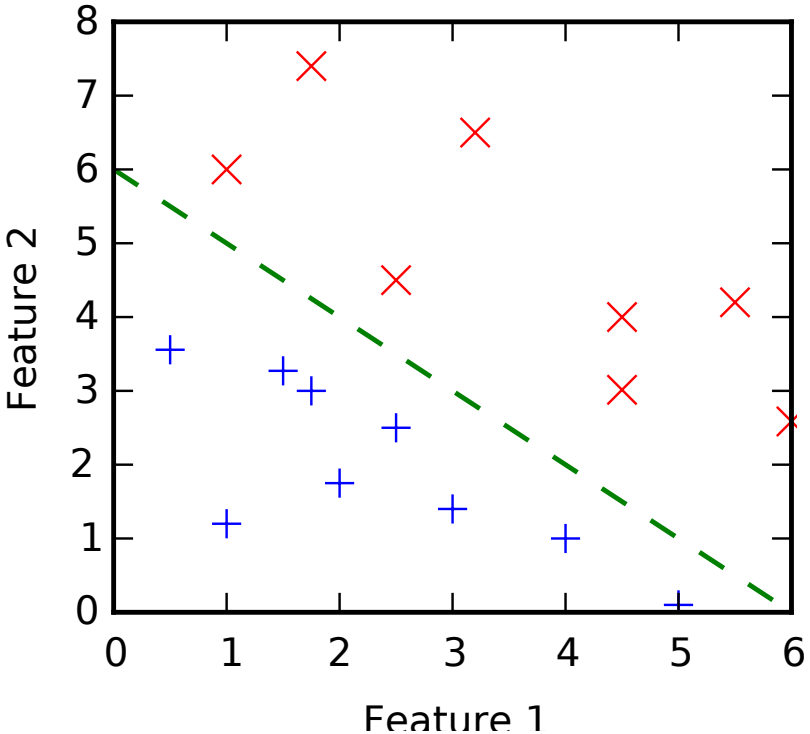
Classification in Graphics

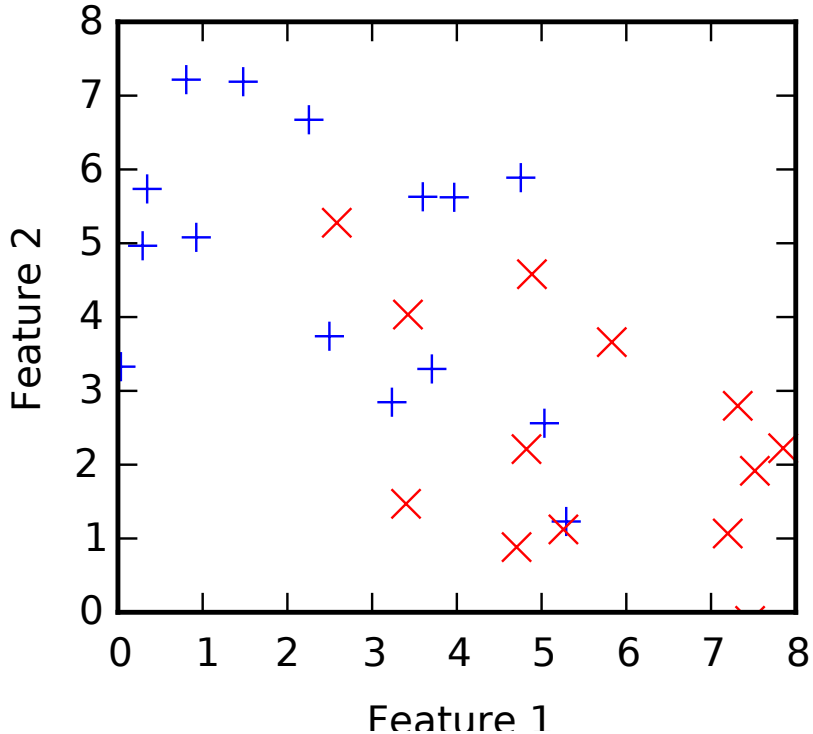


Classification in Graphics

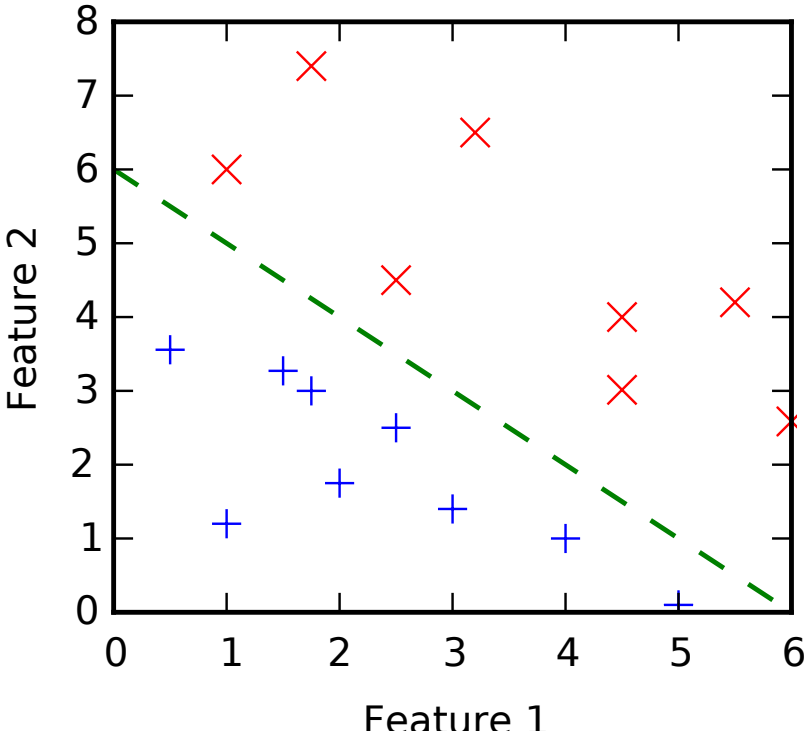


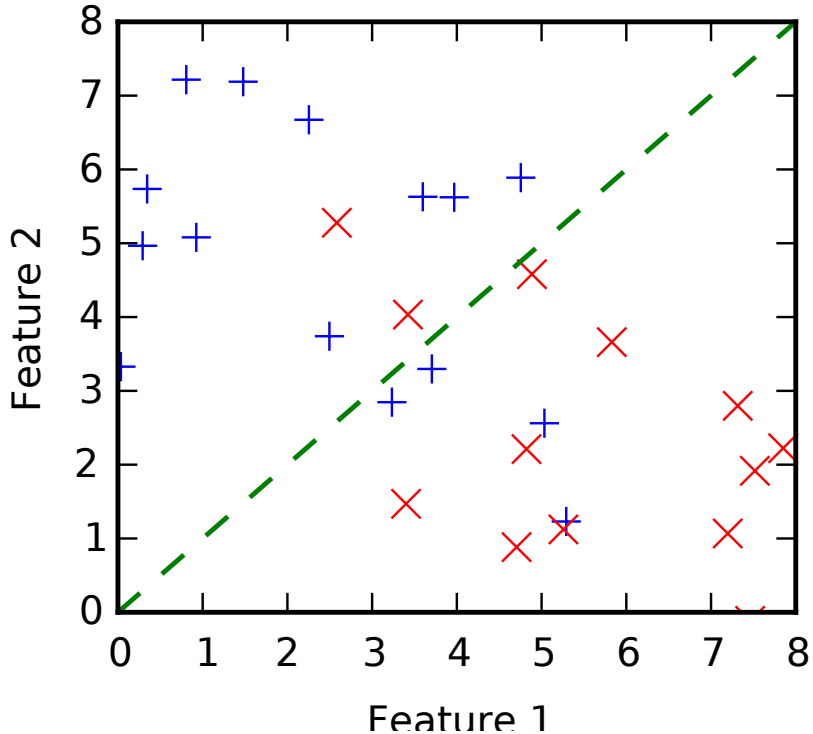
Classification in Graphics





Classification in Graphics

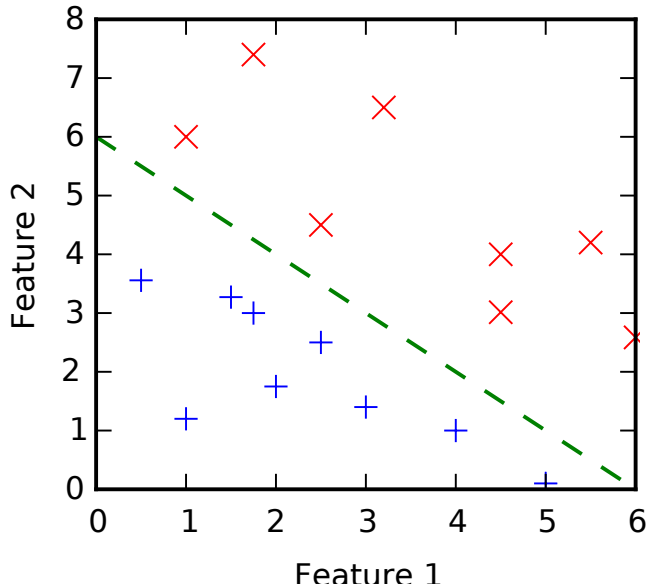




Outline

Approaches to classification

1. Analytic solutions
2. Statistics
3. Machine learning



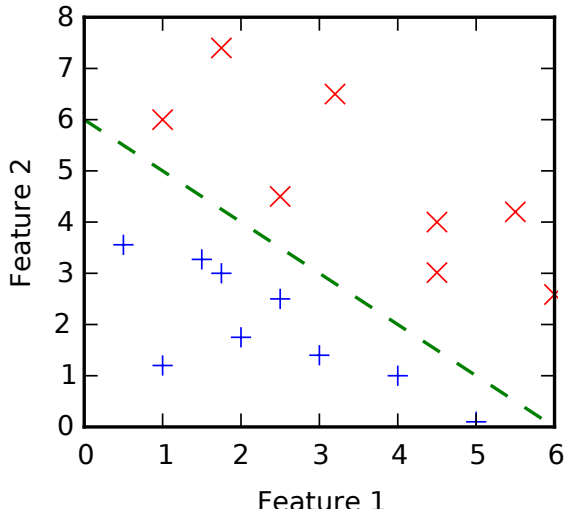
Machine learning

Phase 1 Training

1. data set with known class labels.
2. the algorithm learns the patterns.
3. output a model

Phase 2 Recall

1. data with unknown class label.
2. the algorithm predicts the class label.
3. uses the model from training

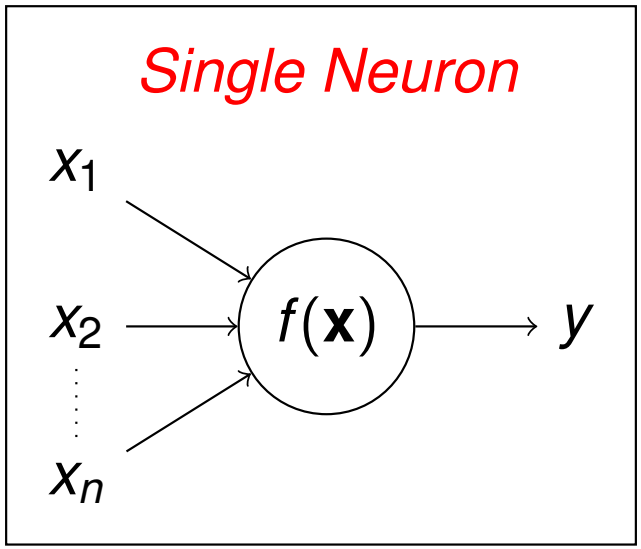


Testing

1. data set with known class labels.
2. do recall (ignoring class labels).
3. compare prediction to known label
4. Estimate the error probabilities (statistics)

Outline

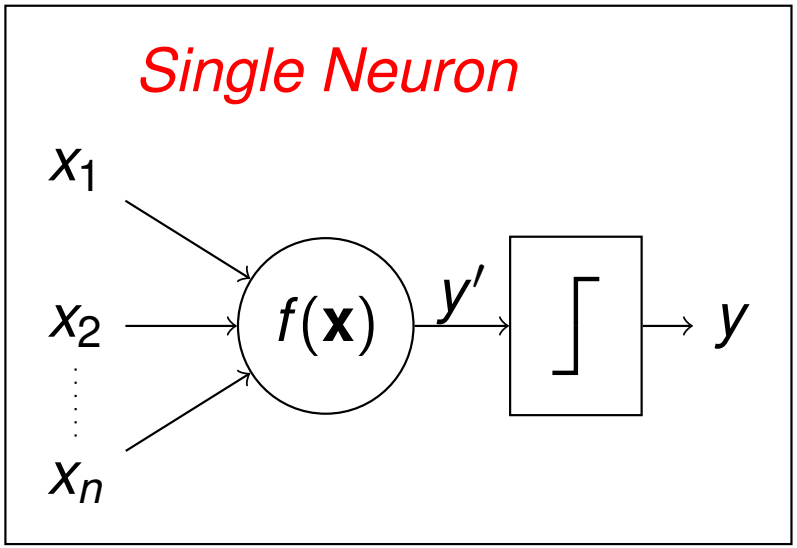
The neuron recall function

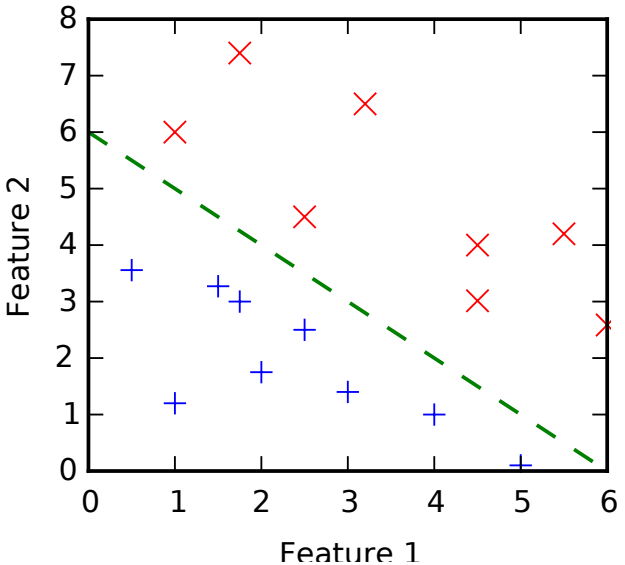


$$f(x) = \mathbf{w} \cdot \mathbf{x} = \sum w_i x_i$$

Neuron weights:
$\mathbf{w} = (w_1, \ldots, w_n)$

The Neuron as a Classifier

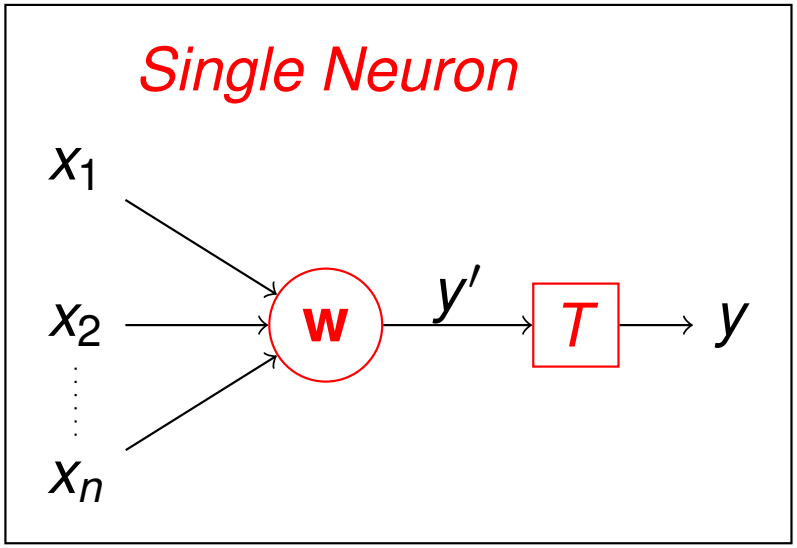




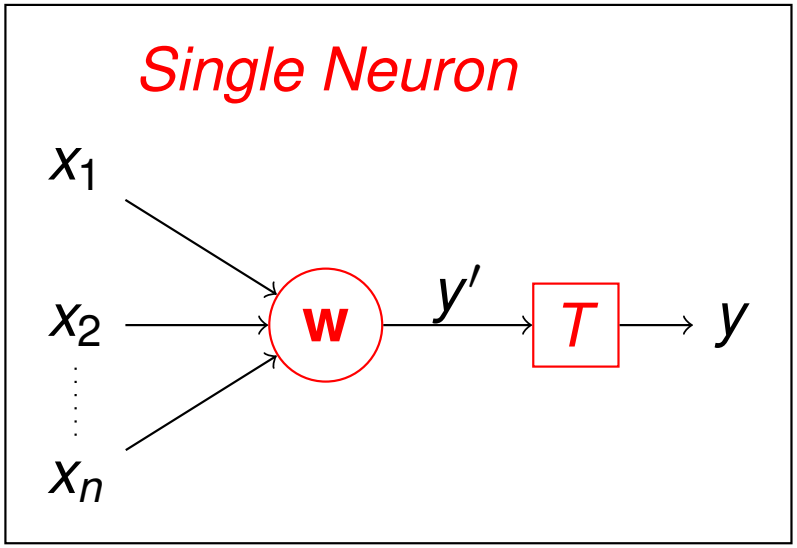
$$y' = \mathbf{w} \cdot \mathbf{x} \tag{1}$$

$$y = \begin{cases} +1, & \text{when } y' > T \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

A Parameter too Many

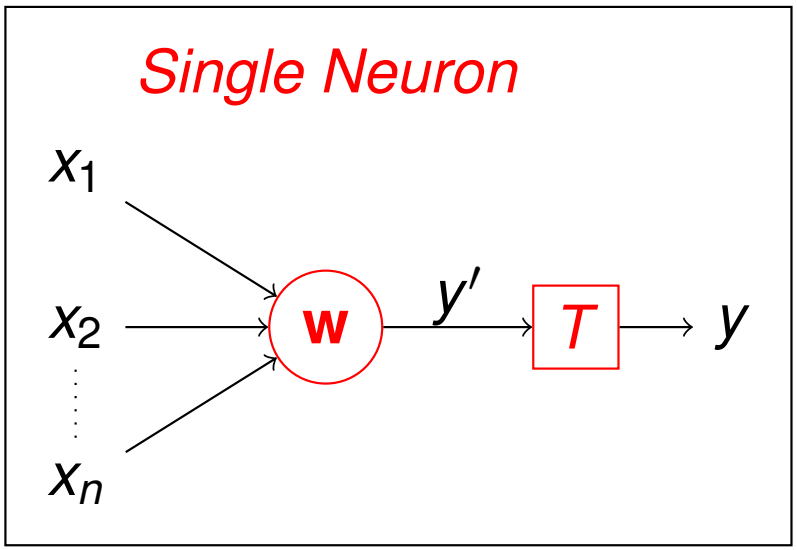


A Parameter too Many



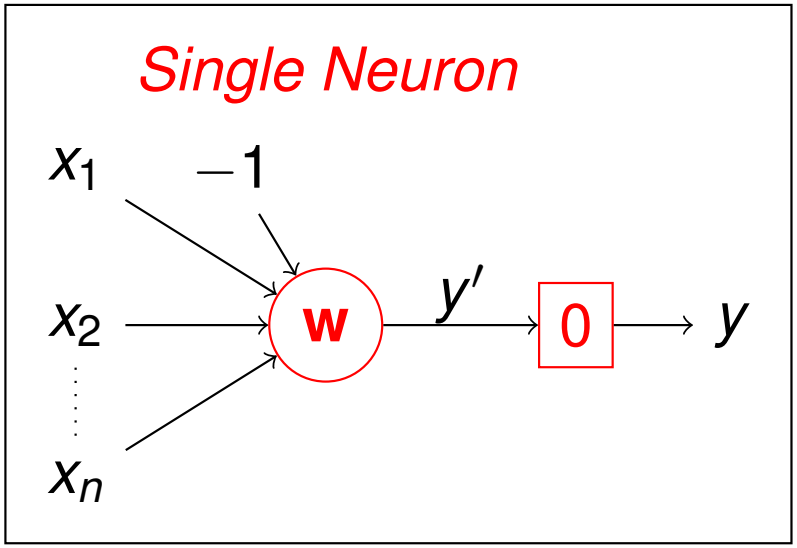
1. $T = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i$
2. $0 = -T + \sum_{i=1}^{n} w_i x_i$

A Parameter too Many



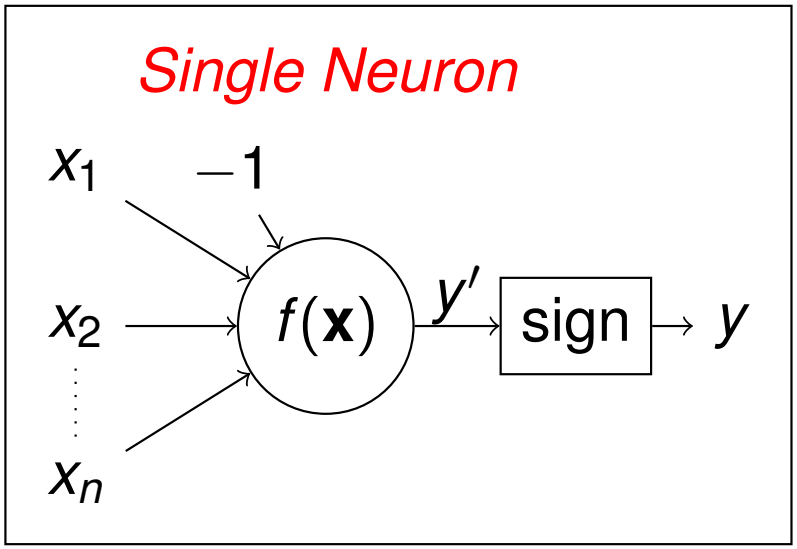
1. $T = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i$
2. $0 = -T + \sum_{i=1}^{n} w_i x_i$
3. $0 = \sum_{i=0}^{n} w_i x_i$
   - where $x_0 = -1$ and $w_0 = T$

A Parameter too Many



1. $T = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i$
2. $0 = -T + \sum_{i=1}^{n} w_i x_i$
3. $0 = \sum_{i=0}^{n} w_i x_i$
   - where $x_0 = -1$ and $w_0 = T$

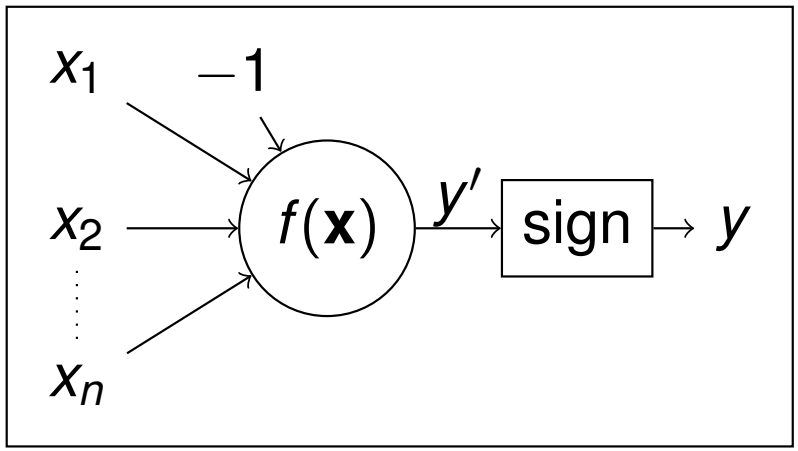
The neuron recall function



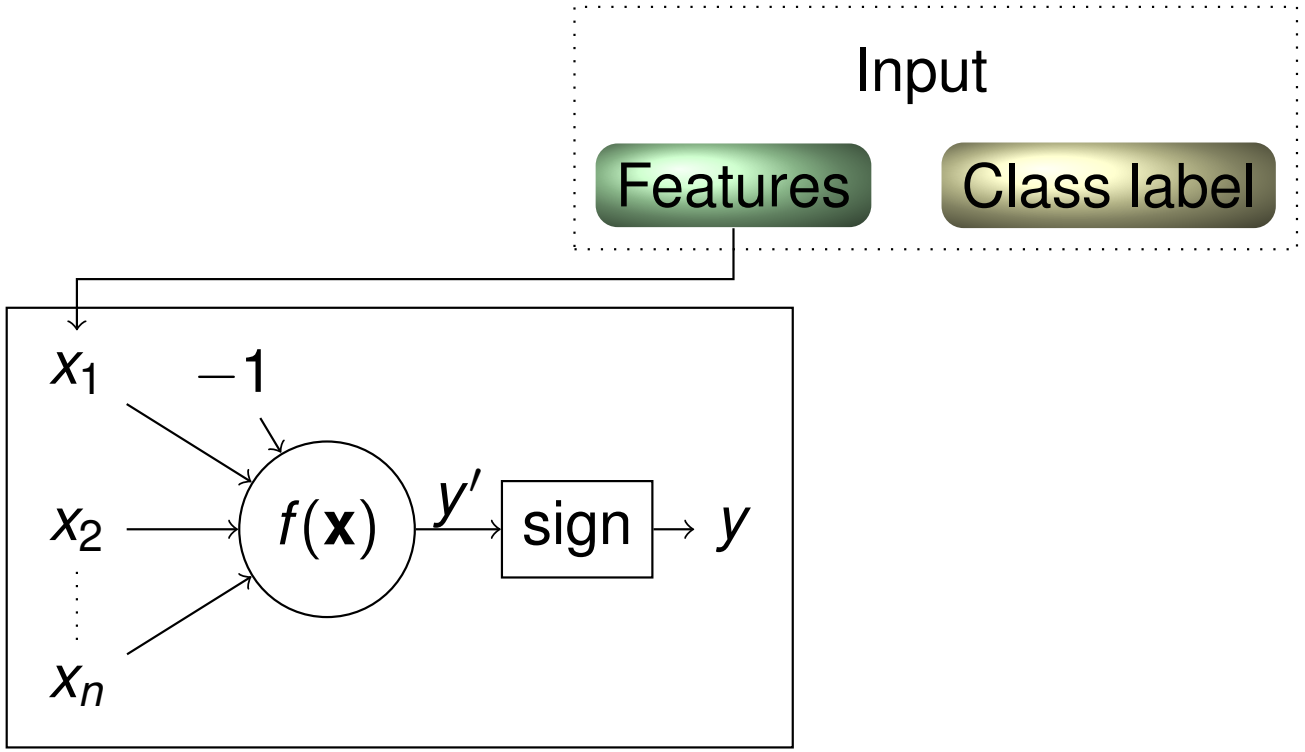
- Weights
  $\mathbf{w} = (w_0, w_1, w_2, \ldots, w_n)$
- Input $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_n)$
  1. $x_0 = -1$
  2. $x_1, x_2, \ldots$ are feature values
- $y = \operatorname{sign} \mathbf{w} \cdot \mathbf{x}$
- Floating point in
- Binary value out

Outline

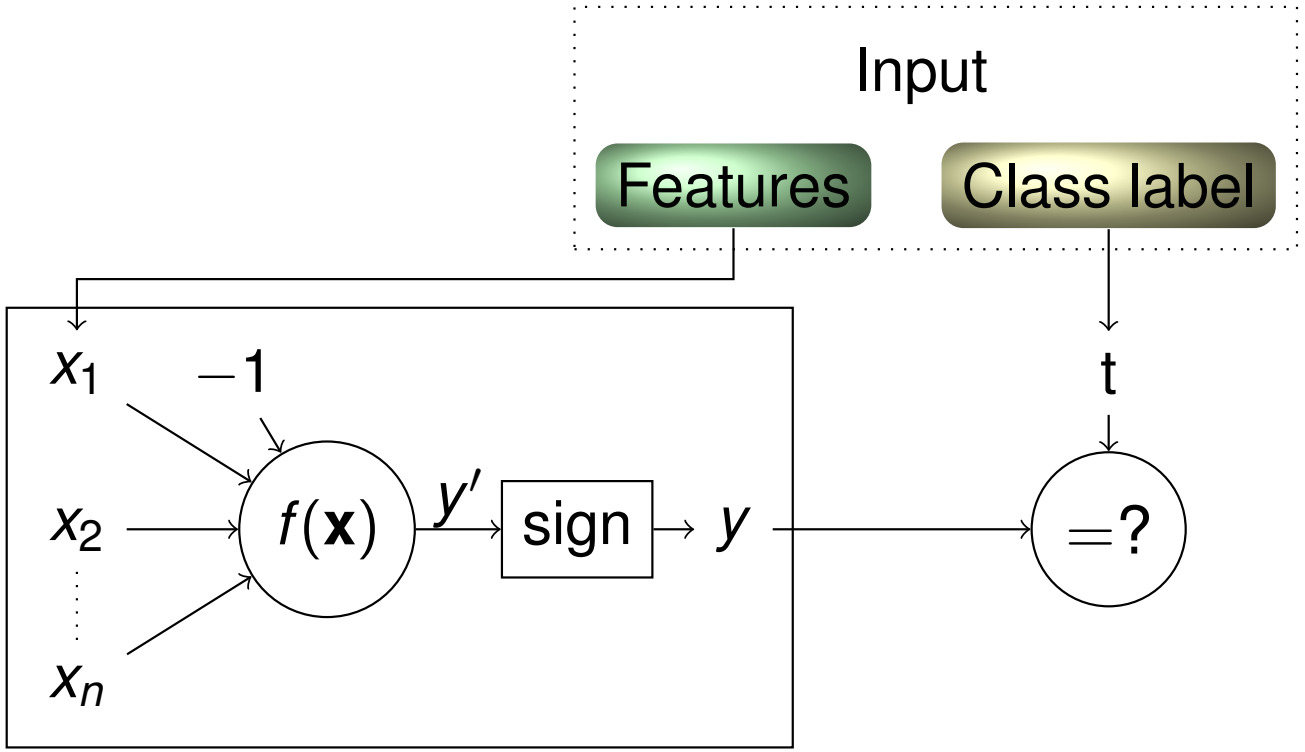
Training the Neuron



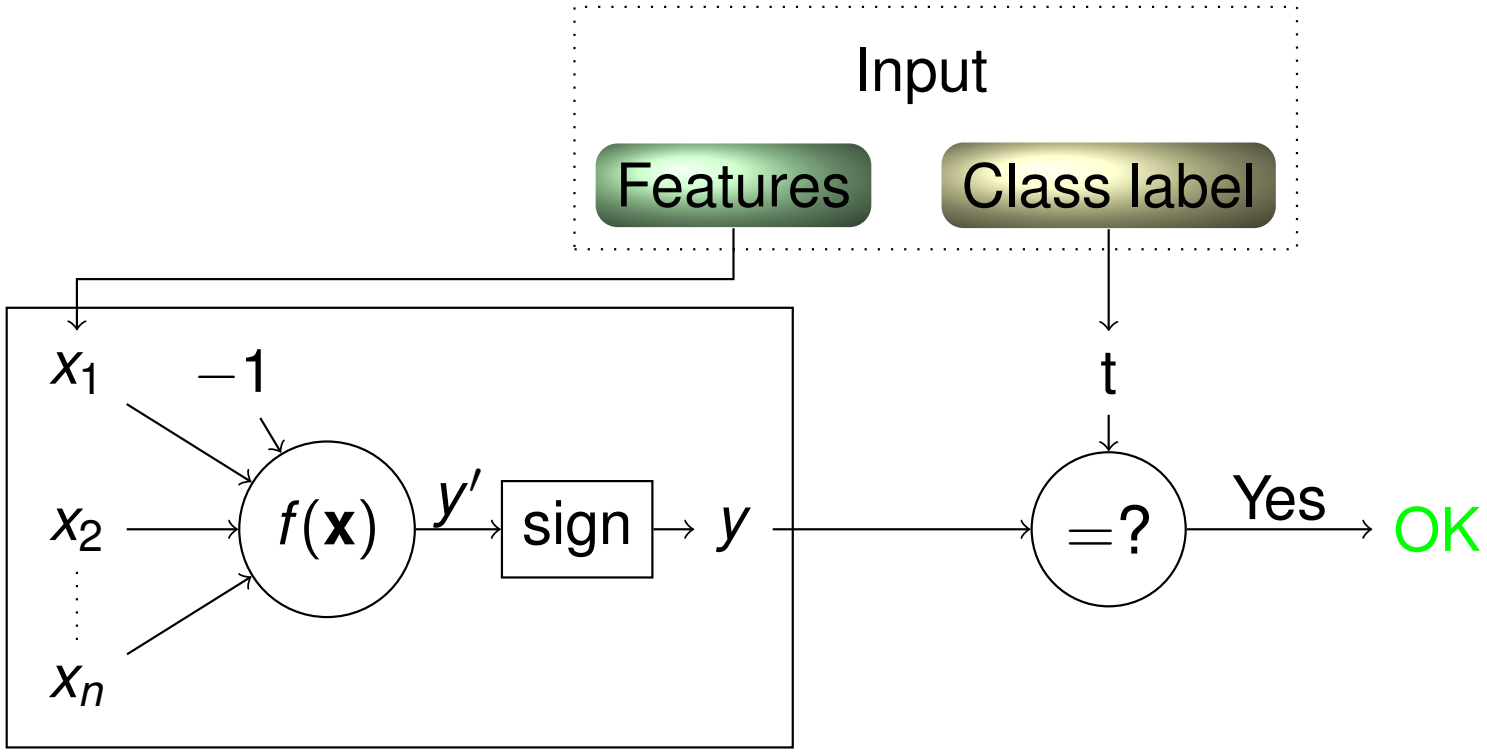
Training the Neuron



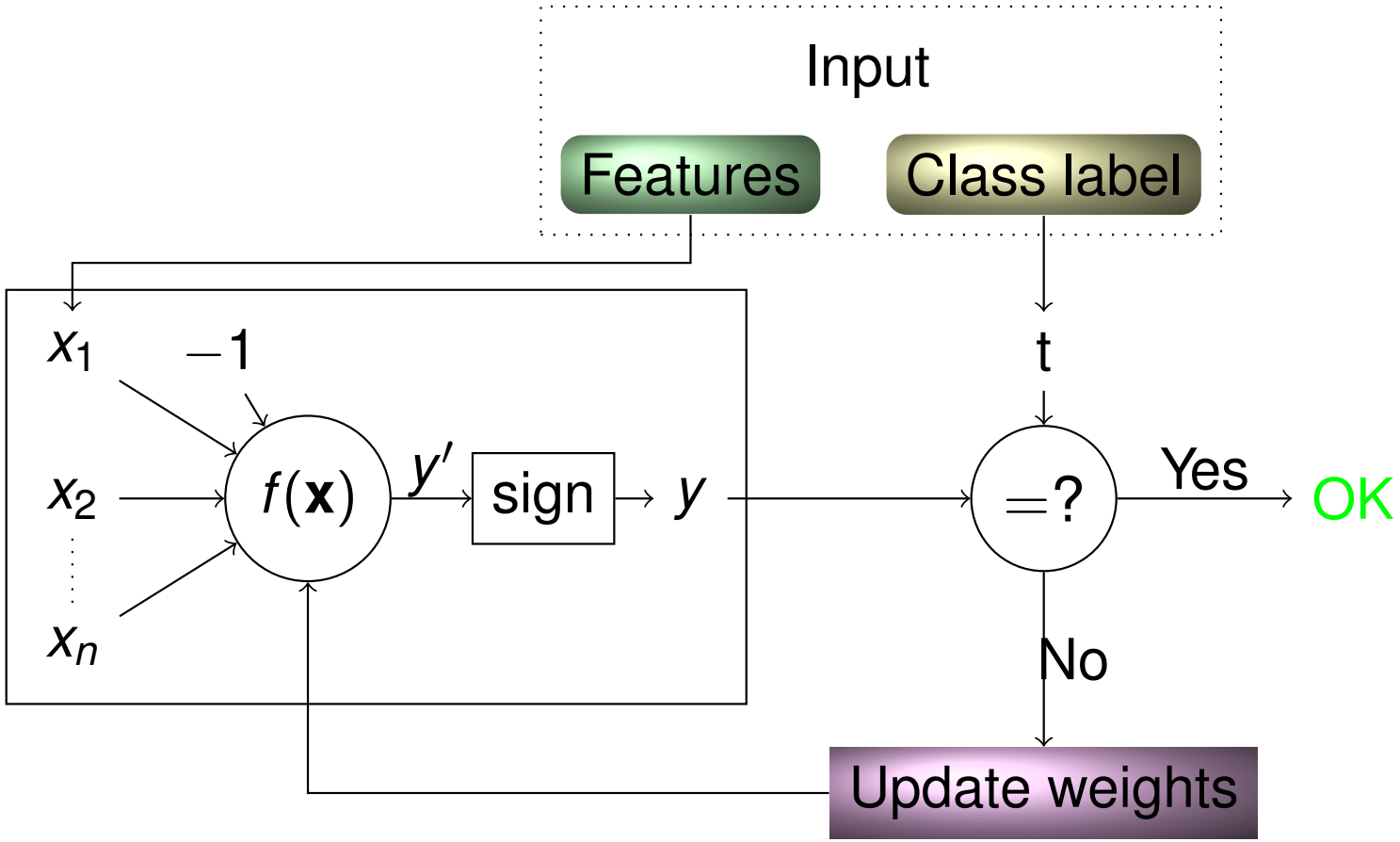
Training the Neuron



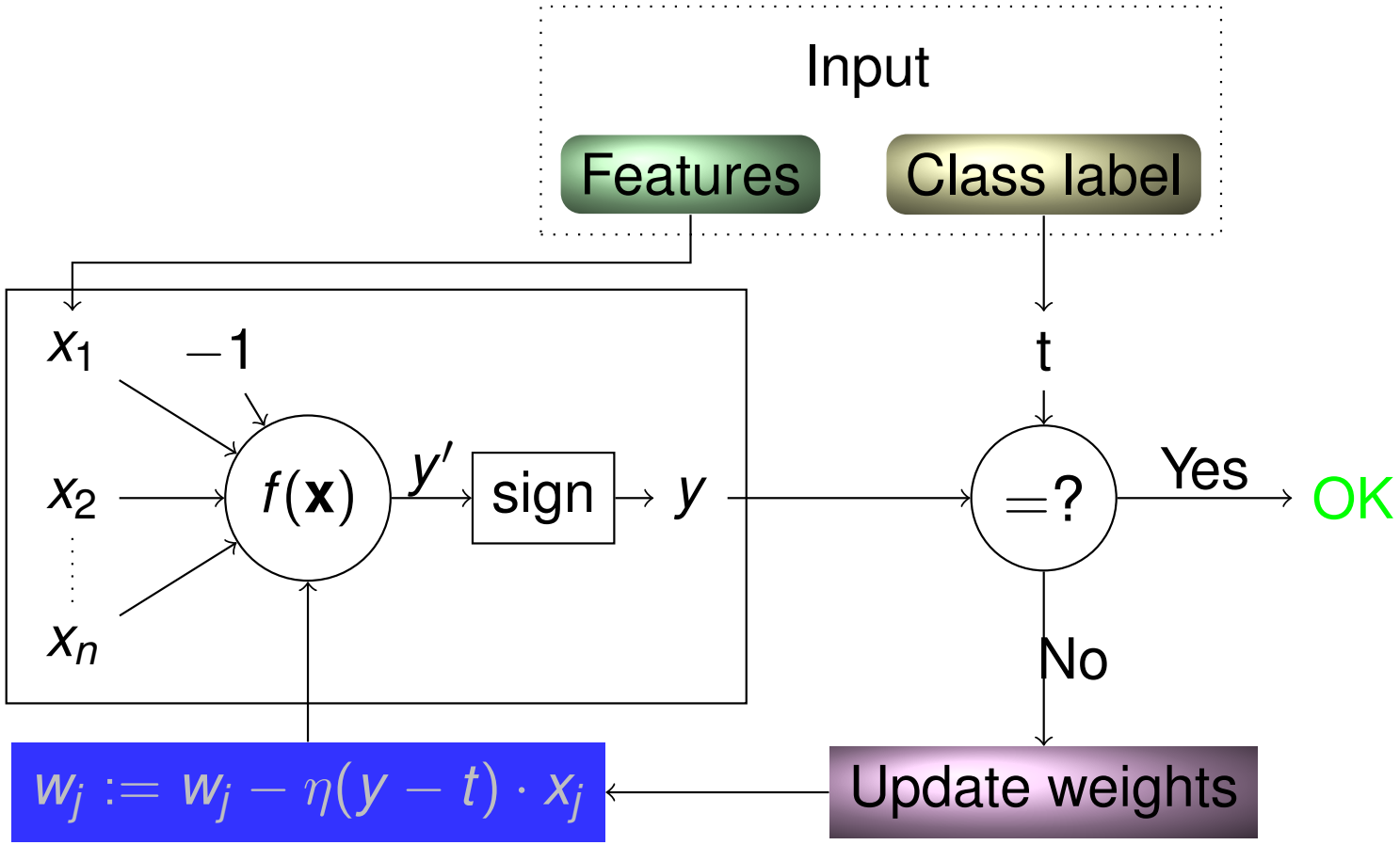
Training the Neuron



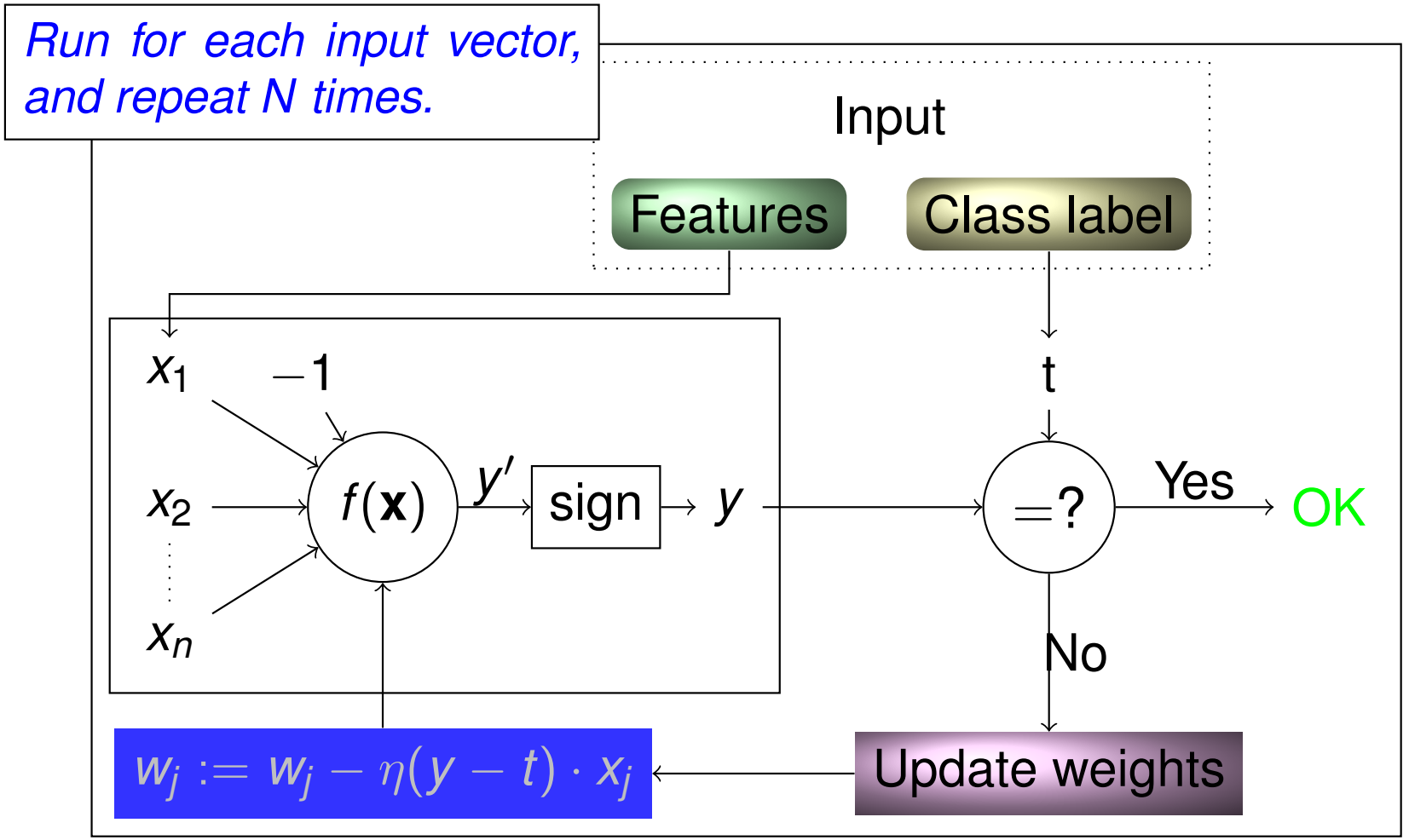
Training the Neuron



Training the Neuron



Training the Neuron



Outline

Summary

- Classification determines if X is fowl or fish
- Selected features are used
- Models define plausible feature values for fowl and for fish
- Machine learning generate models too complex for human comprehension
- The single neuron is a linear classifier
- Return to details next week