Introduction

to

Machine Learning Algorithms

บุญเสริม กิจศิริกุล

ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

The Needs for Machine Learning

- Recently, full-genome sequencing has blossomed.
- Several other technologies, such as DNA microarrays and mass spectrometry have considerably progressed.
- The technologies rapidly produce terabytes of data.

The Problem of Diabetes

- To predict whether the patient shows signs of diabetes
- Data contains 2 classes, i.e.

  + : tested positive for diabetes

  – : tested negative for diabetes

The Problem of Diabetes (cont.)

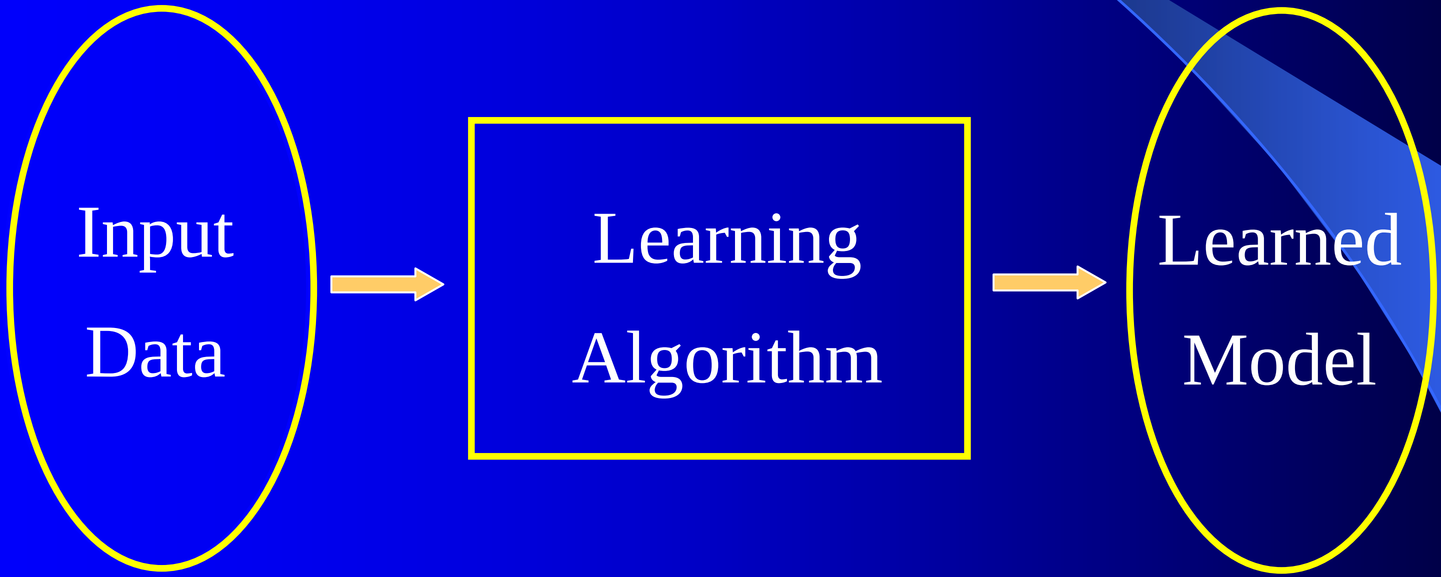
Data contains 8 attributes

1. Number of times pregnant
2. Plasma glucose concentration after?? 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)

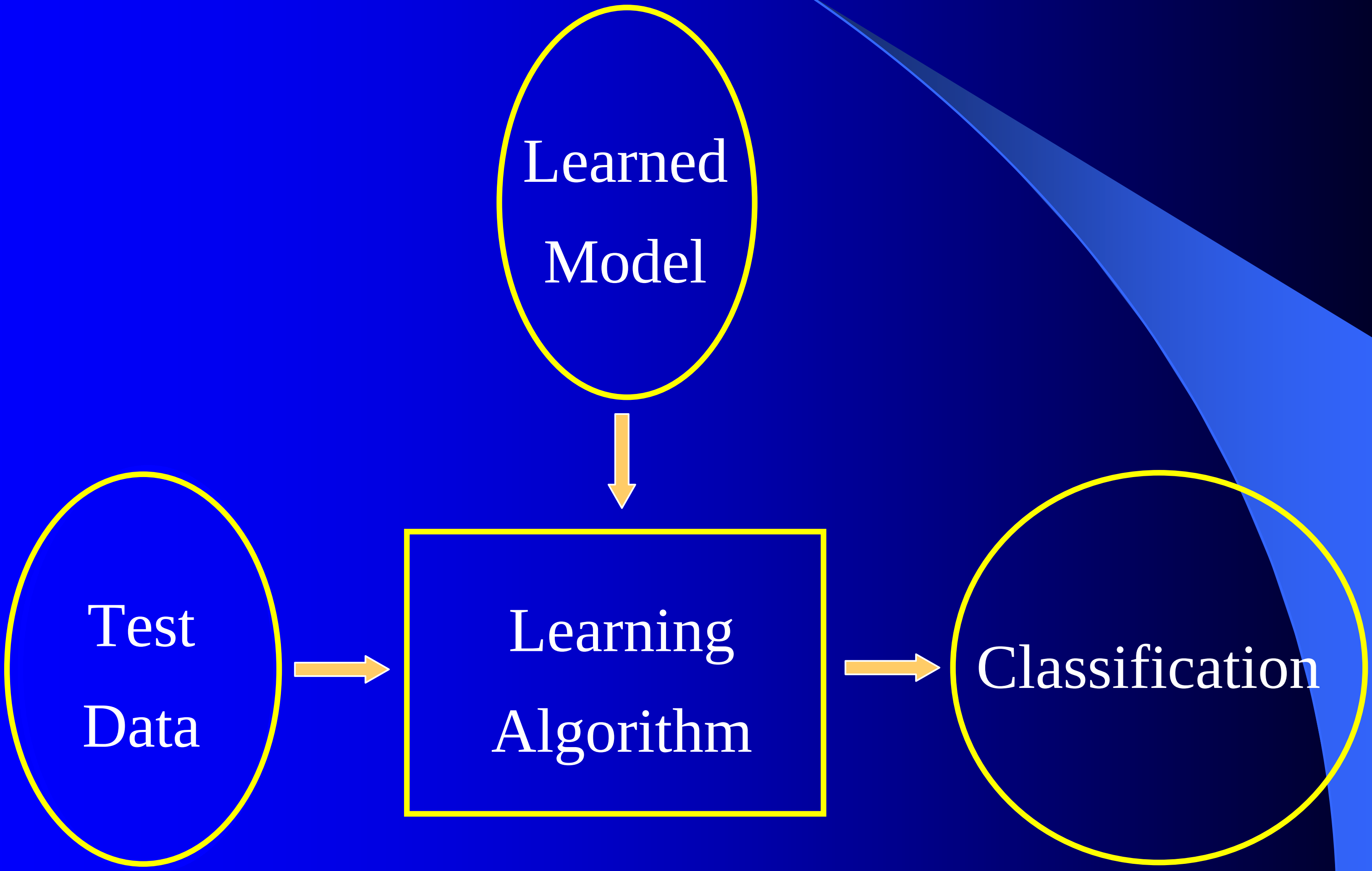
The Data of Diabetes (Table 1)

No. Time Pregnant	Plasma Glucose	Blood Pressure	Fold Thickness	Serum Insulin	BMI	Pedi-gree	Age	Class
1	189	60	23	846	30.1	0.398	59	+
5	166	72	19	175	25.8	0.587	51	+
1	103	30	38	83	43.3	0.183	33	−
1	89	66	23	94	28.1	0.167	21	−
0	137	40	35	168	43.1	2.288	33	+
3	126	88	41	235	39.3	0.704	27	−
13	145	82	19	110	22.2	0.245	57	−
1	97	66	15	140	23.2	0.487	22	−
2	197	70	45	543	30.5	0.158	53	+

Inductive Learning form Data (Classification Tasks)



Inductive Learning form Data – Cont. (Classification Tasks)



Machine Learning Algorithms

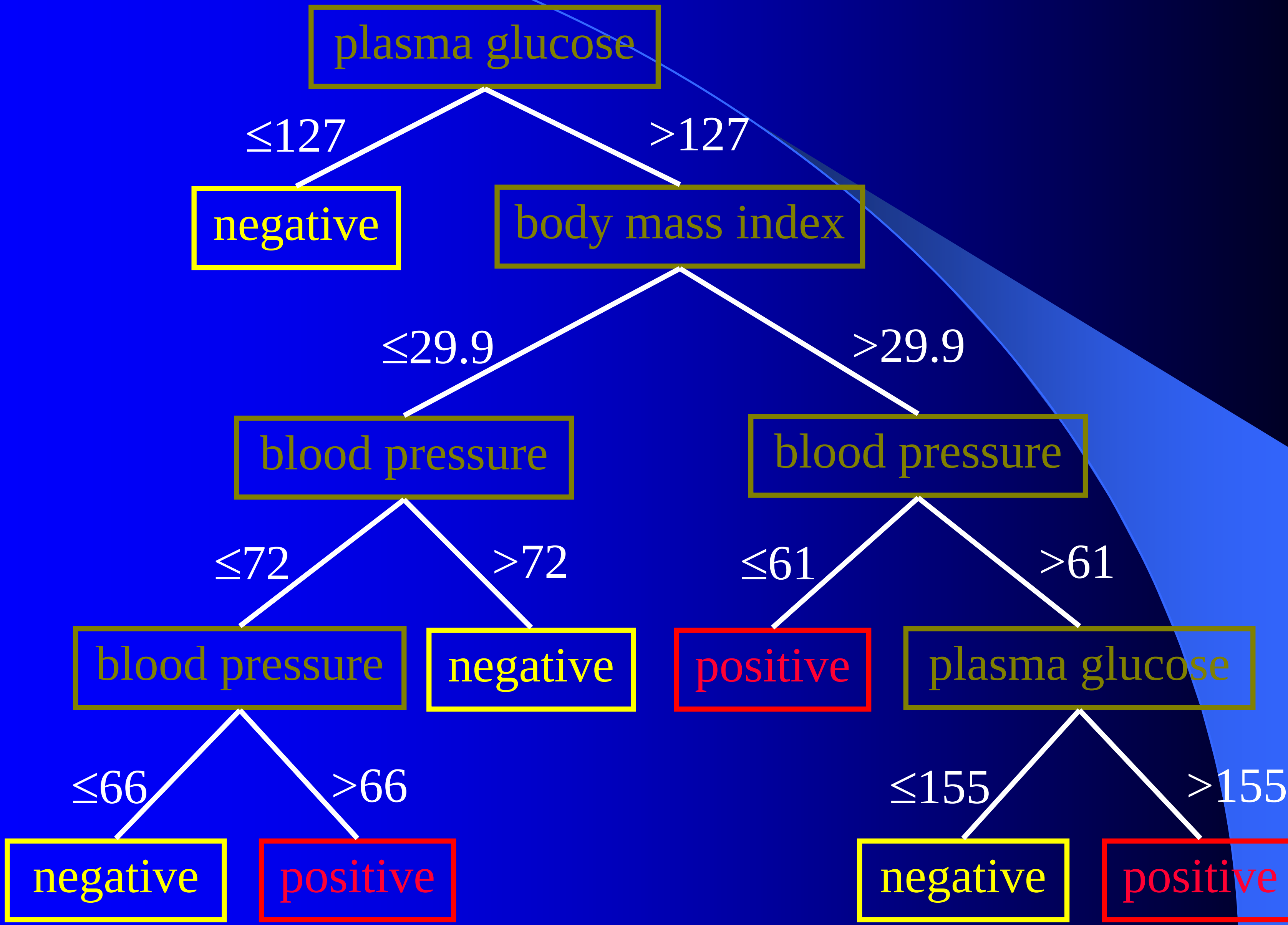
- Decision Tree Learning
- Neural Networks
- Support Vector Machines
- Etc.

Decision Tree Learning

Decision tree representation

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

An Example of Decision Trees



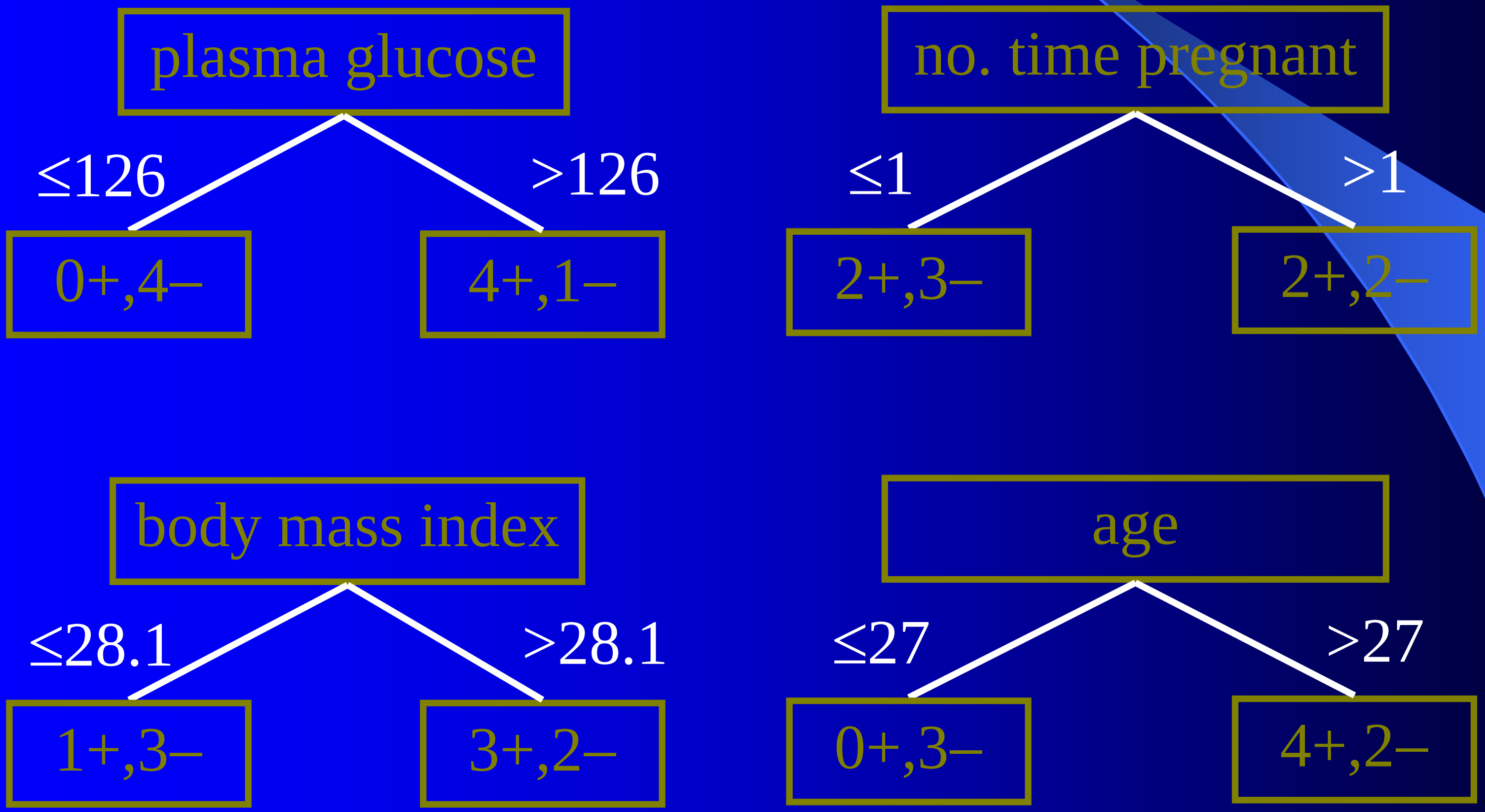
A Decision Tree Learning Algorithm

Main Loop:

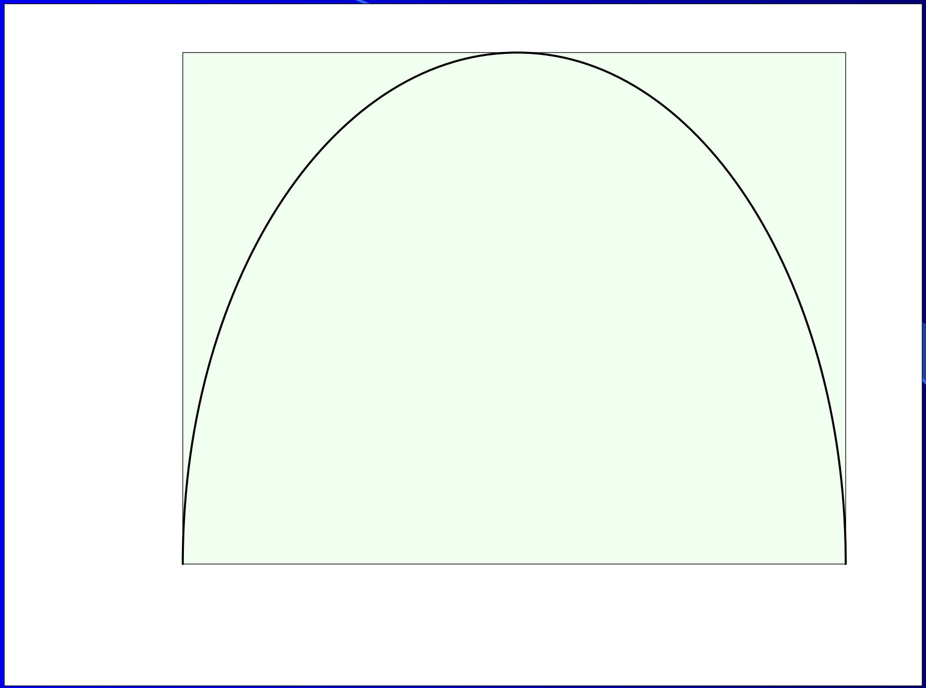
1. A $\leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples are perfectly classified, then STOP, else iterate over new leaf nodes

Selection of the "Best" Attribute

- Using Data in Table 1 (4+,5–), some candidate attributes are shown below.



Entropy



- S is a sample of training examples
- p^+ is the proportion of positive examples in S
- p^- is the proportion of negative examples in S
- Entropy measures the impurity of S

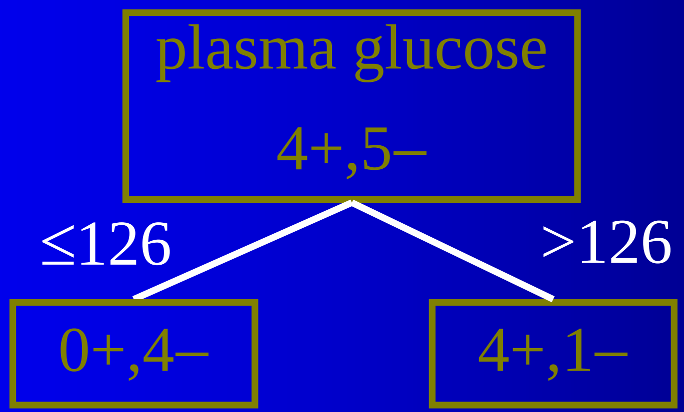
$$\text{Entropy}(S) = -\,p^+\log_2 p^+ - p^-\log_2 p^-$$

Information Gain

Gain(S,A) = expected reduction in entropy due to sorting on A

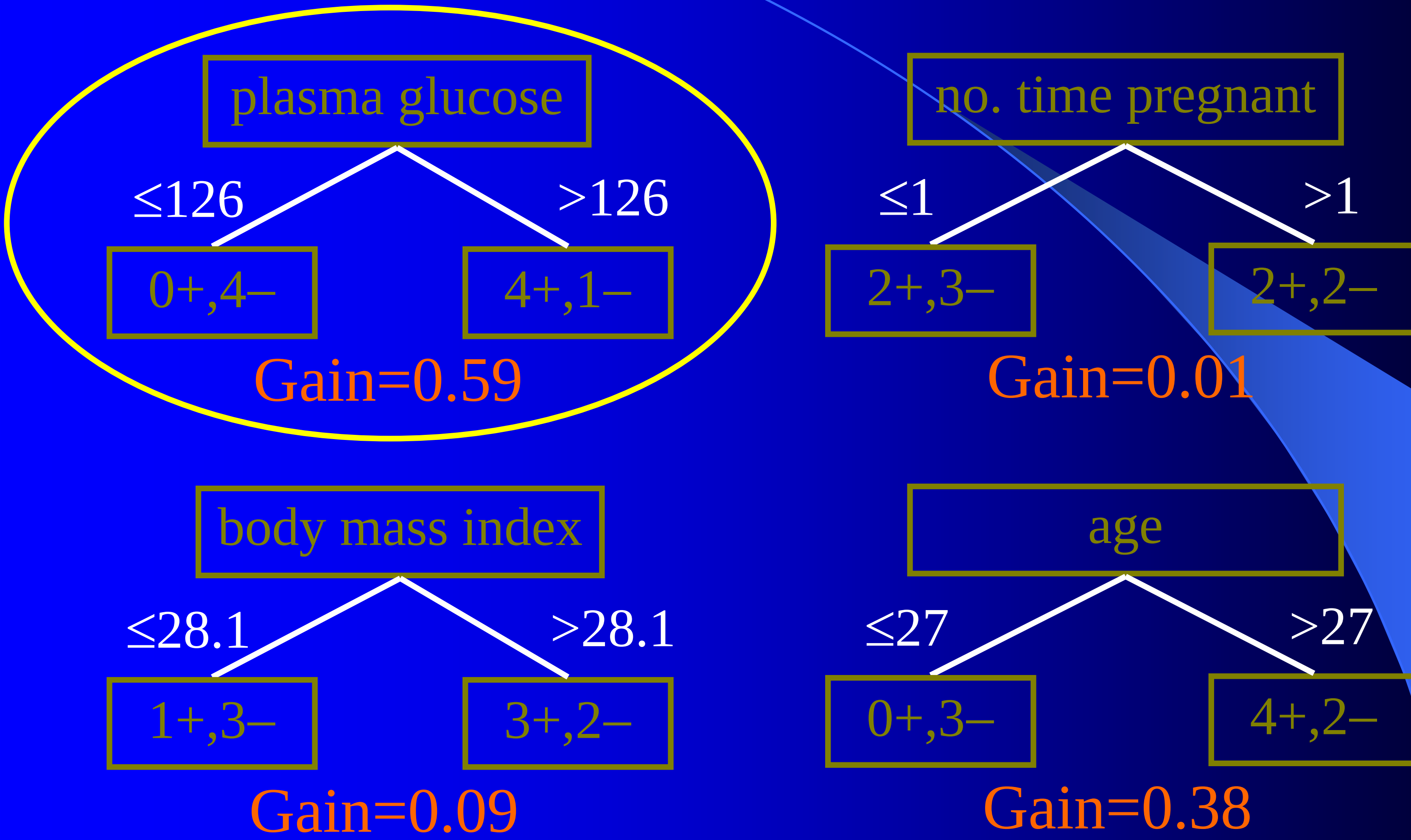
$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Example:



$$Gain(S, plasma\ glucose) \equiv \left(-\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9}\right) - \left[\begin{array}{l}\left(-\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4}\right) + \\ \left(-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right)\end{array}\right] = 0.59$$

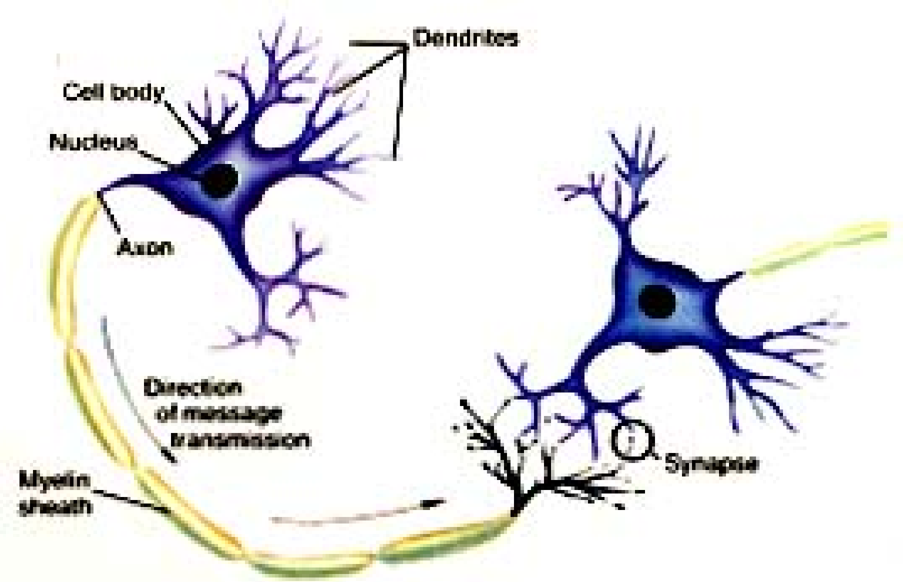
Selection of the “Best” Attribute by Gain



“Plasma glucose” is selected as the root node, and the process of adding nodes is repeated.

Artificial Neural Networks

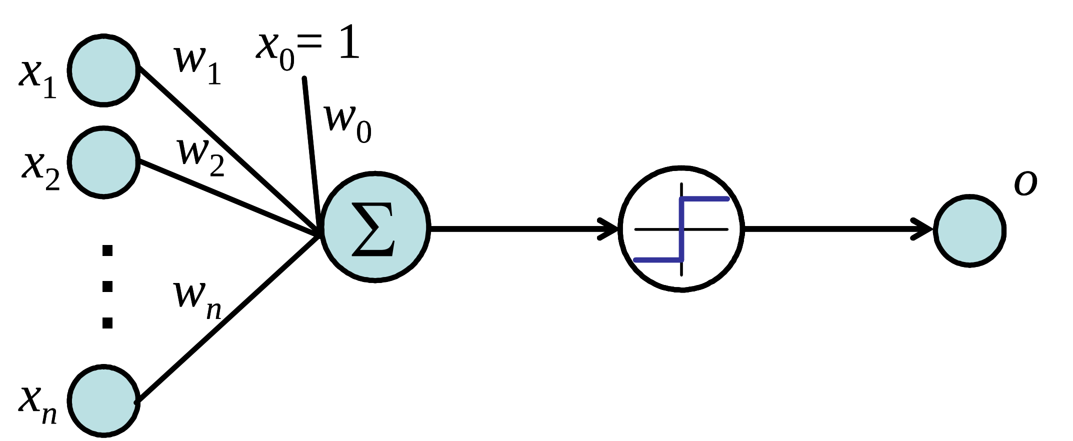
- Artificial neural networks (ANN) simulate units in human brain (neurons)
- Many weighted interconnections among units
- Learning by adaptation of the connection weights



Nerve cells, called neurons, transmit messages throughout the body. Most neurons have many branching dendrites, a large cell body, and a long, insulated axon.

Perceptron

- One of the earliest neural network models
- Input is a real valued vector ($x_1,\ldots,x_n$)
- Use an activation function to compute output value (o)
- Output is linear combination of the input with weights ($w_1,\ldots,w_n$)



$$o(x_1, x_2, \ldots, x_n) = \begin{cases} 1 & if \;\; w_1x_1 + w_2x_2 + \cdots + w_nx_n > \theta \\ -1 & if \;\; w_1x_1 + w_2x_2 + \cdots + w_nx_n < \theta \end{cases}$$

$$o(x_1, x_2, \ldots, x_n) = \begin{cases} 1 & if \;\; w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n > 0 \\ -1 & if \;\; w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n < 0 \end{cases}$$

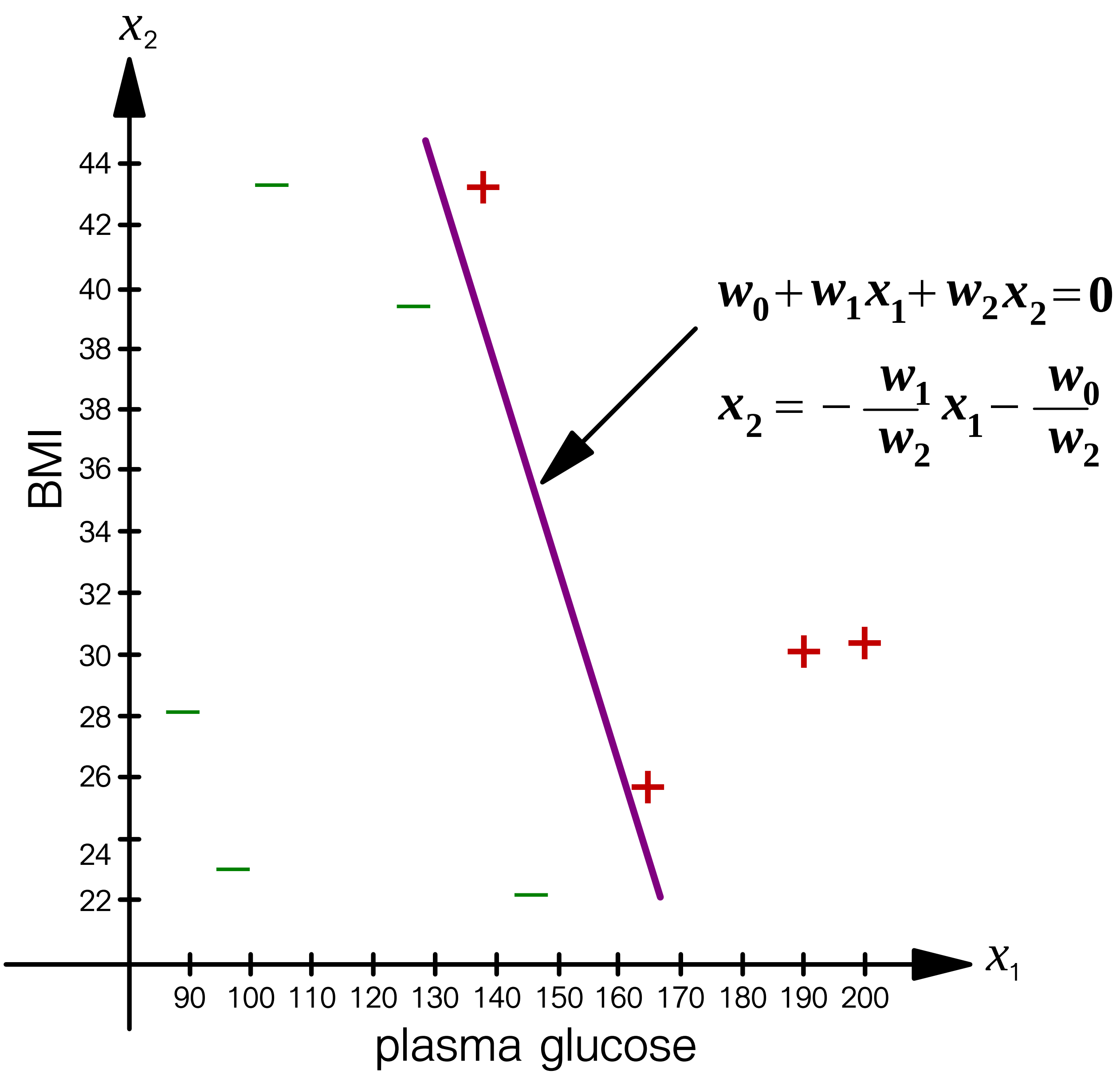
The Data of Diabetes (Table 1)

No. Time Pregnant	Plasma Glucose	Blood Pressure	Fold Thickness	Serum Insulin	BMI	Pedi-gree	Age	Class
1	189	60	23	846	30.1	0.398	59	+
5	166	72	19	175	25.8	0.587	51	+
1	103	30	38	83	43.3	0.183	33	−
1	89	66	23	94	28.1	0.167	21	−
0	137	40	35	168	43.1	2.288	33	+
3	126	88	41	235	39.3	0.704	27	−
13	145	82	19	110	22.2	0.245	57	−
1	97	66	15	140	23.2	0.487	22	−
2	197	70	45	543	30.5	0.158	53	+

The Data of Diabetes (Table 1)

No. Time Pregnant	Plasma Glucose	Blood Pressure	Fold Thickness	Serum Insulin	BMI	Pedi-gree	Age	Class
1	189	60	23	846	30.1	0.398	59	+
5	166	72	19	175	25.8	0.587	51	+
1	103	30	38	83	43.3	0.183	33	−
1	89	66	23	94	28.1	0.167	21	−
0	137	40	35	168	43.1	2.288	33	+
3	126	88	41	235	39.3	0.704	27	−
13	145	82	19	110	22.2	0.245	57	−
1	97	66	15	140	23.2	0.487	22	−
2	197	70	45	543	30.5	0.158	53	+

Perceptron’s Decision Surface



Perceptron Training

$w=[0.25\ –0.1\ 0.5]$

$x_2 = 0.2\ x_1 – 0.5$

$(x,t)=([-1,-1],1)$

$o=sgn(0.25+0.1-0.5)$

$=-1$

$\Delta w=[0.2\ –0.2\ –0.2]$

$(x,t)=([2,1],-1)$

$o=sgn(0.45-0.6+0.3)$

$=1$

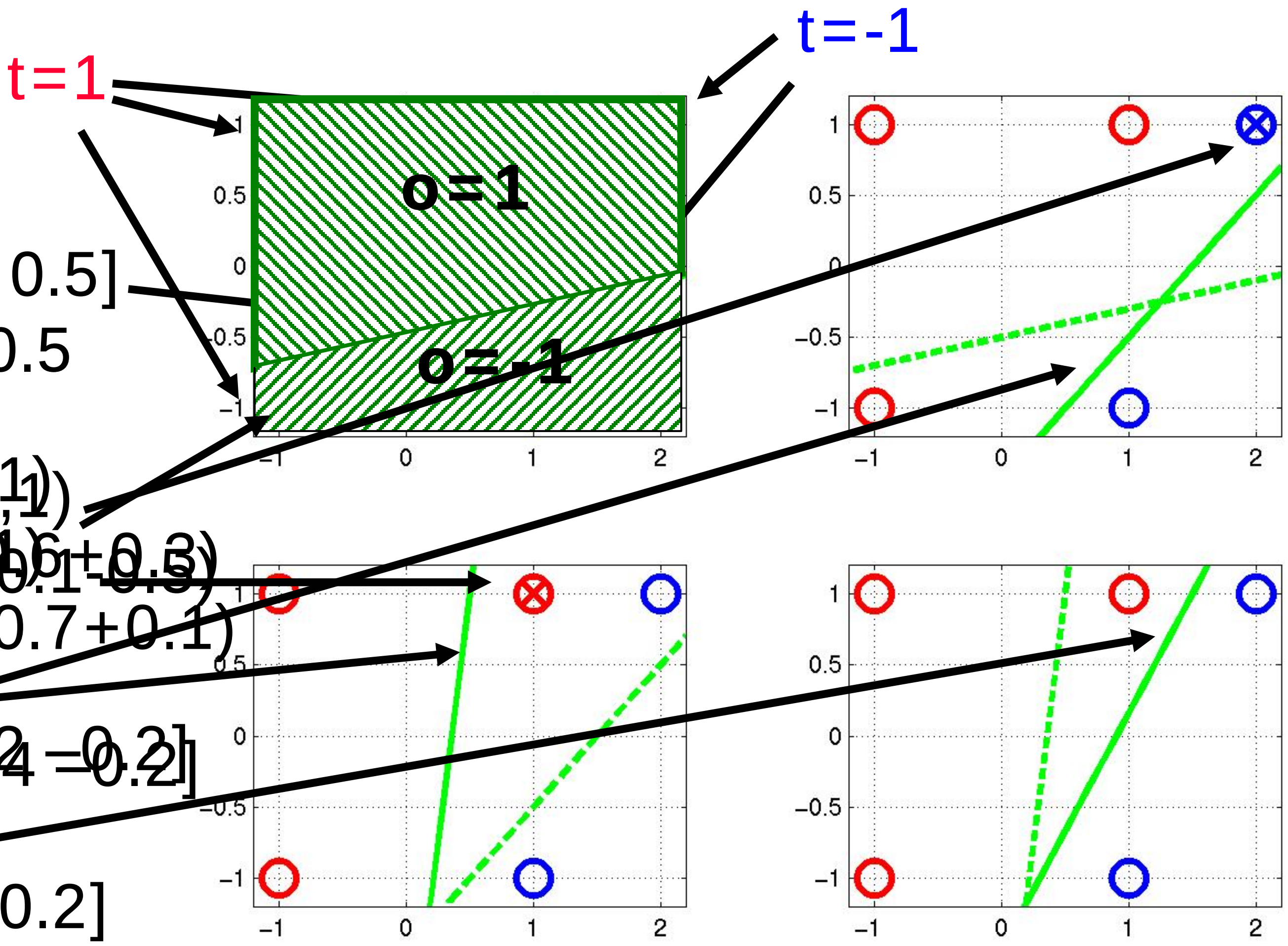
$\Delta w=[-0.2\ –0.4\ –0.2]$

$(x,t)=([1,1],1)$

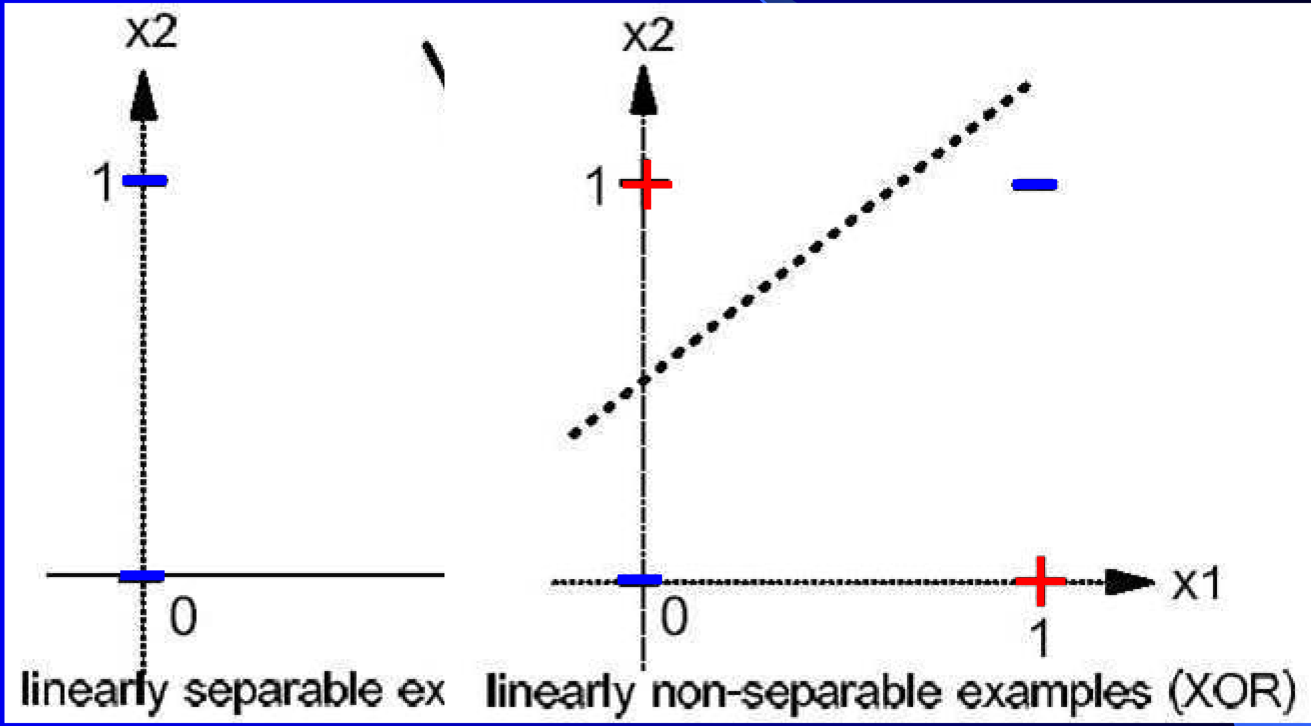
$o=sgn(0.25-0.7+0.1)$

$=-1$

$\Delta w=[0.2\ 0.2\ 0.2]$

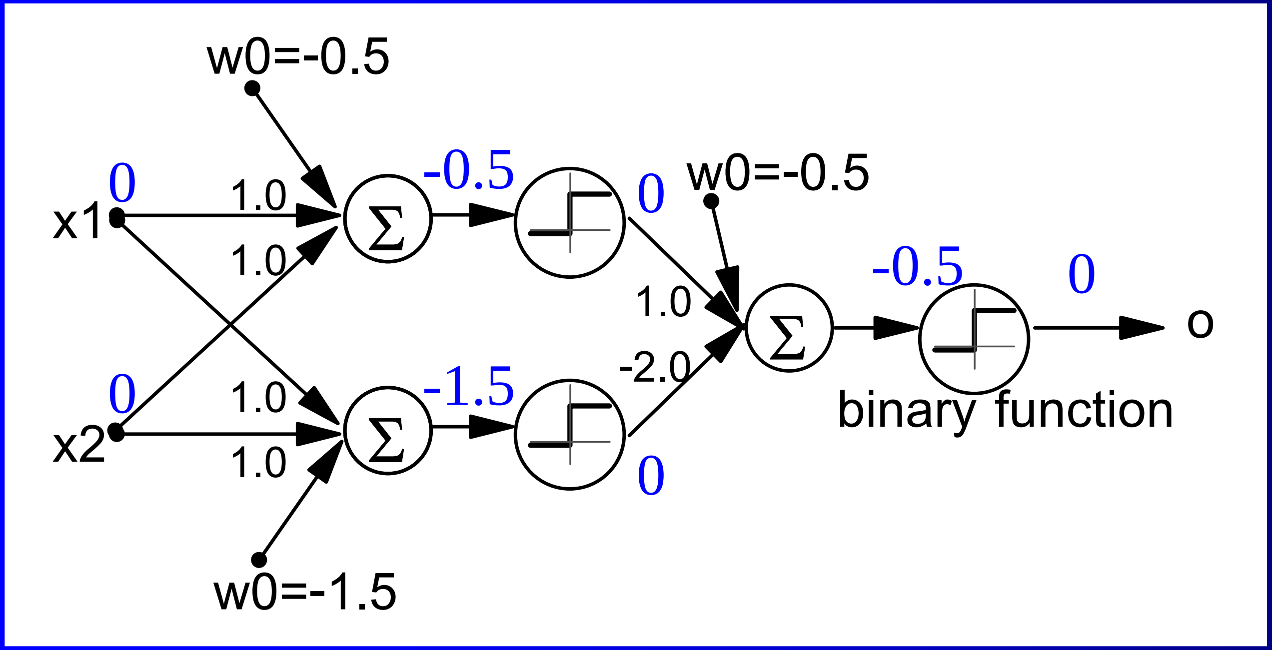


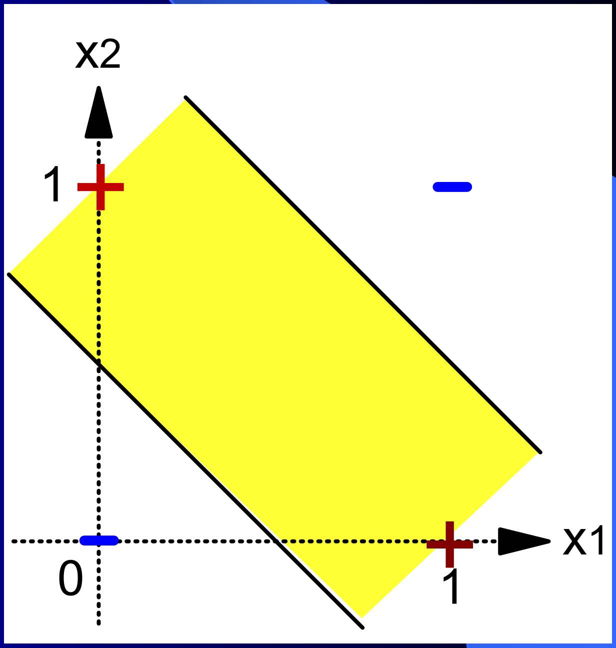
Linearly Non-Separable Examples



Multilayer Networks

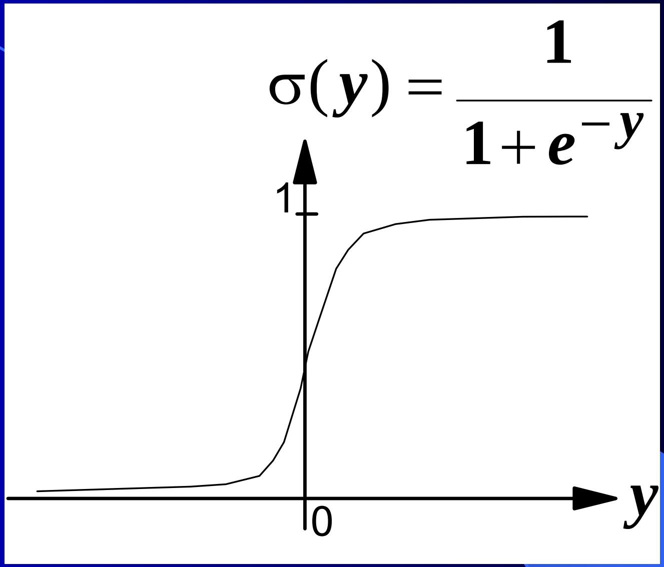
- Multilayer Feedforward Neural Networks can represent non-linear decision surface
- An Example of multilayer networks

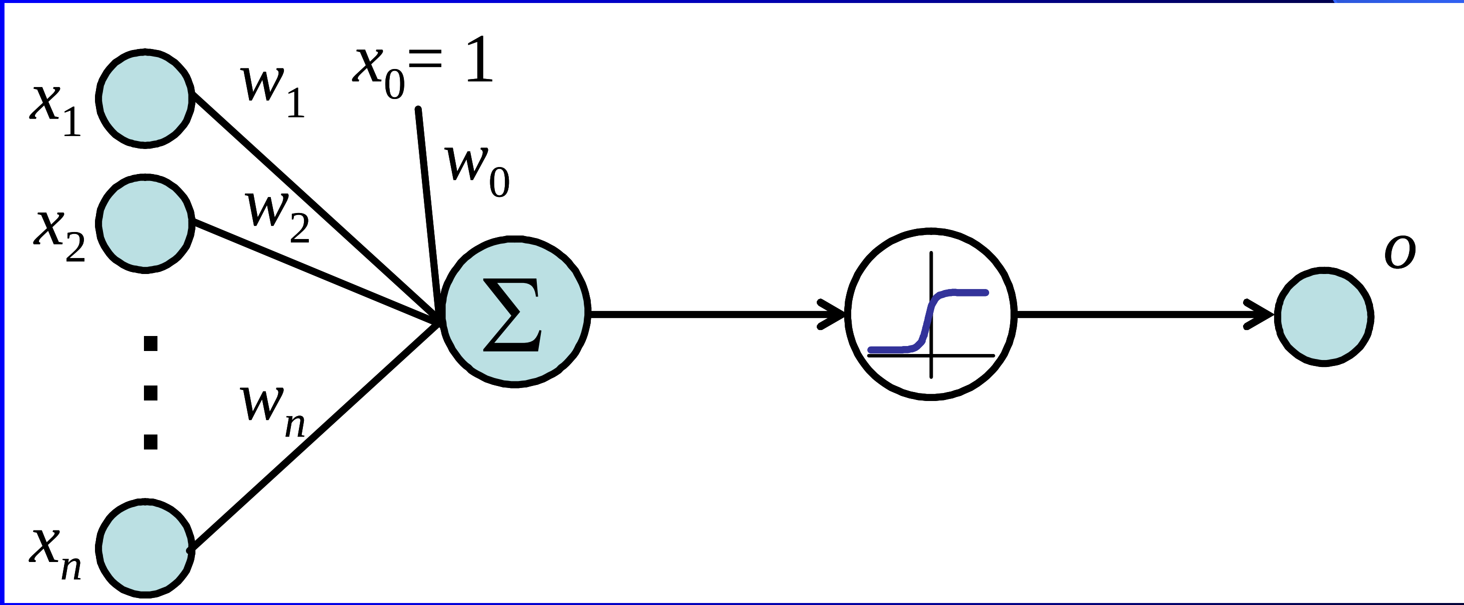




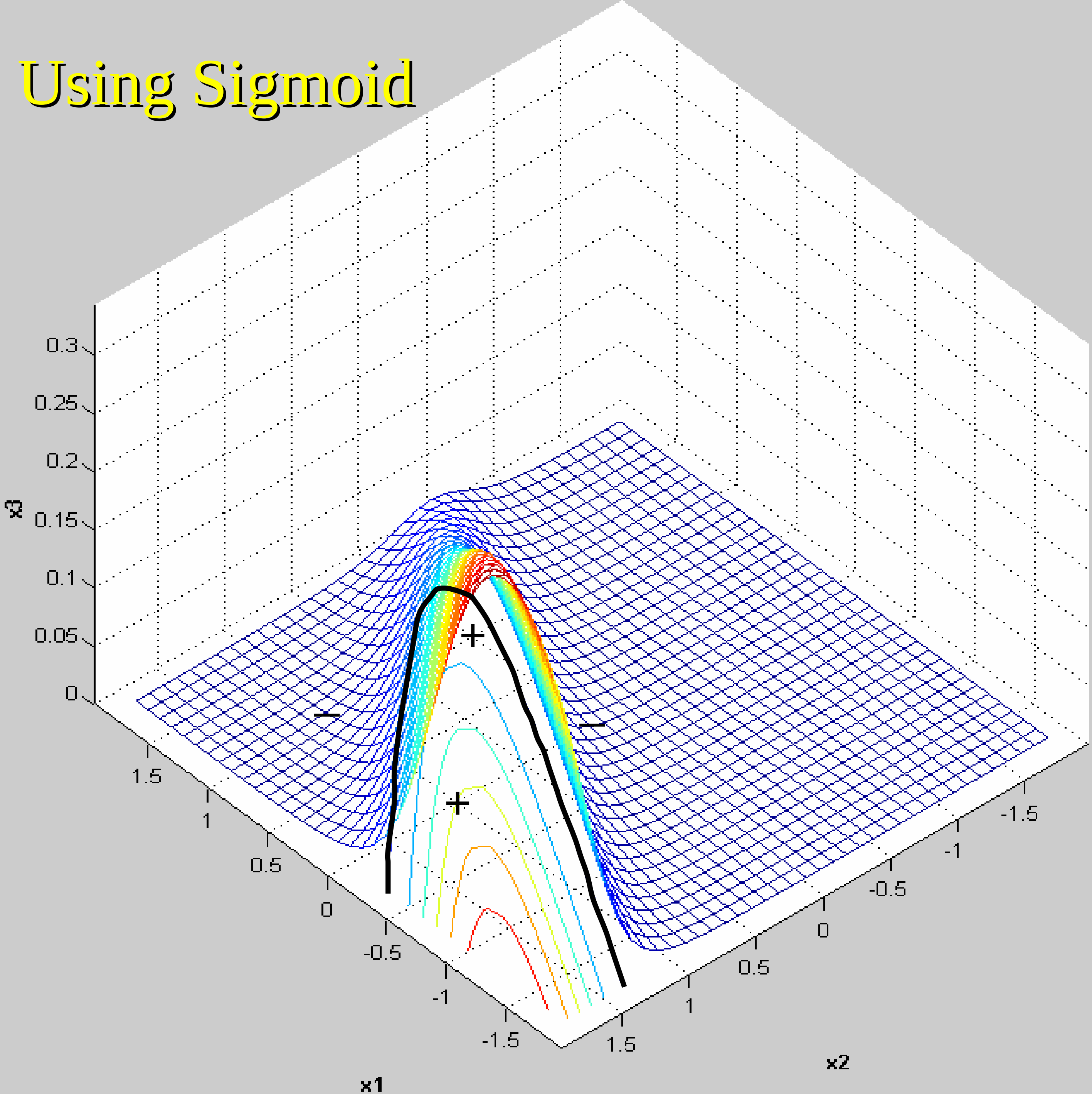
Sigmoid Function

- Sigmoid function is used as activation function in multilayer networks

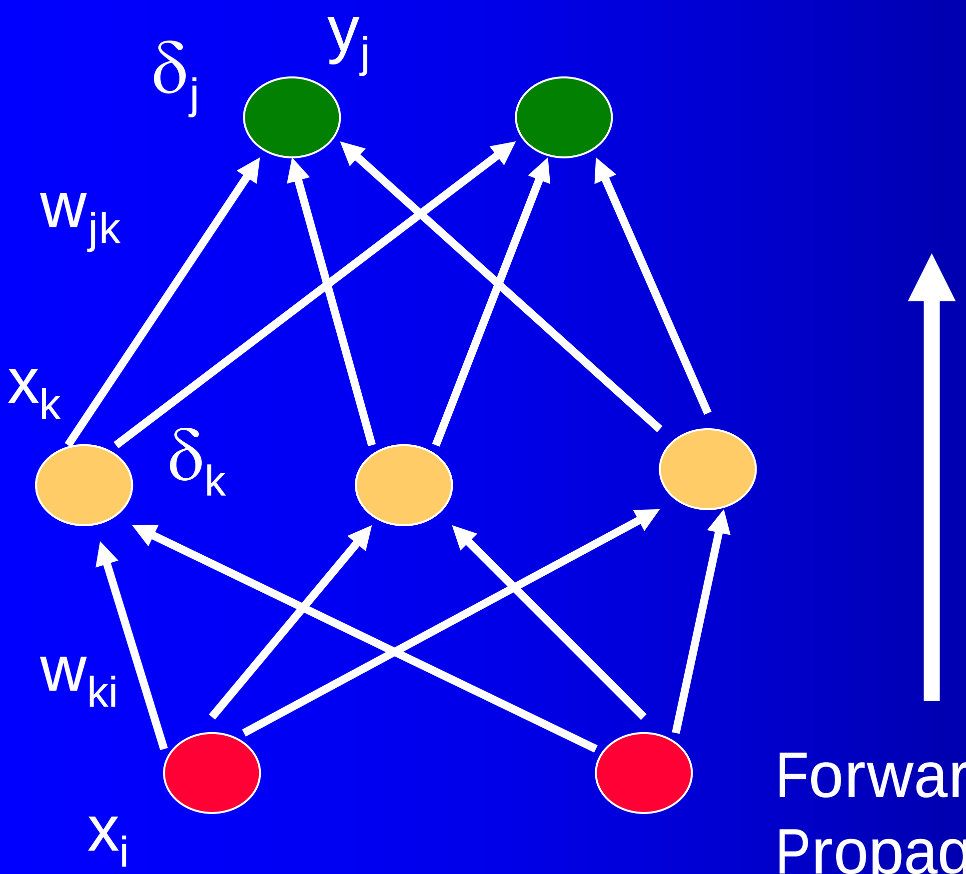




Using Sigmoid



Backpropagation Algorithm



Backward step:
propagate errors from output to hidden layer

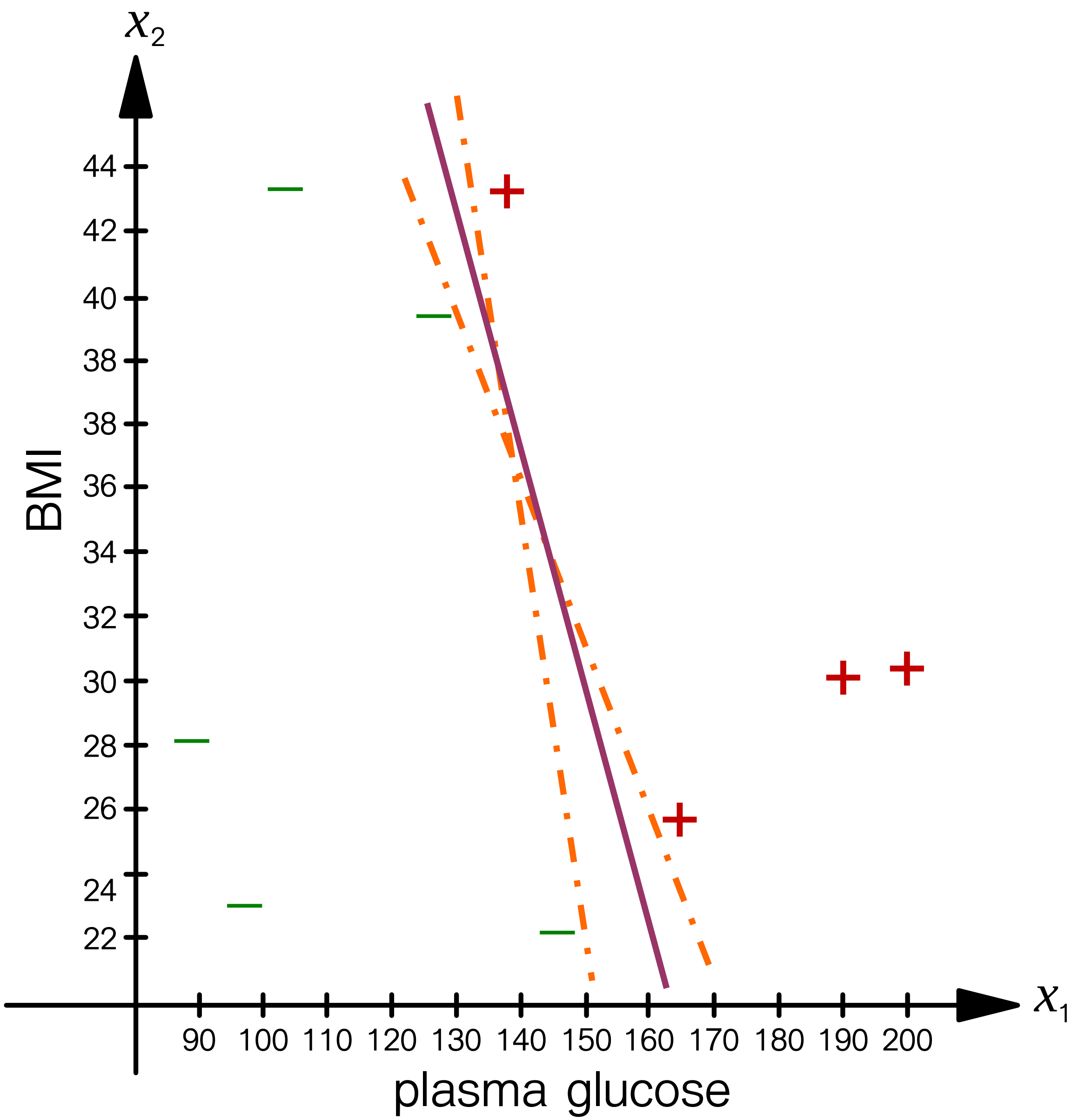
Forward step:
Propagate activation from input to output layer

Support Vector Machines

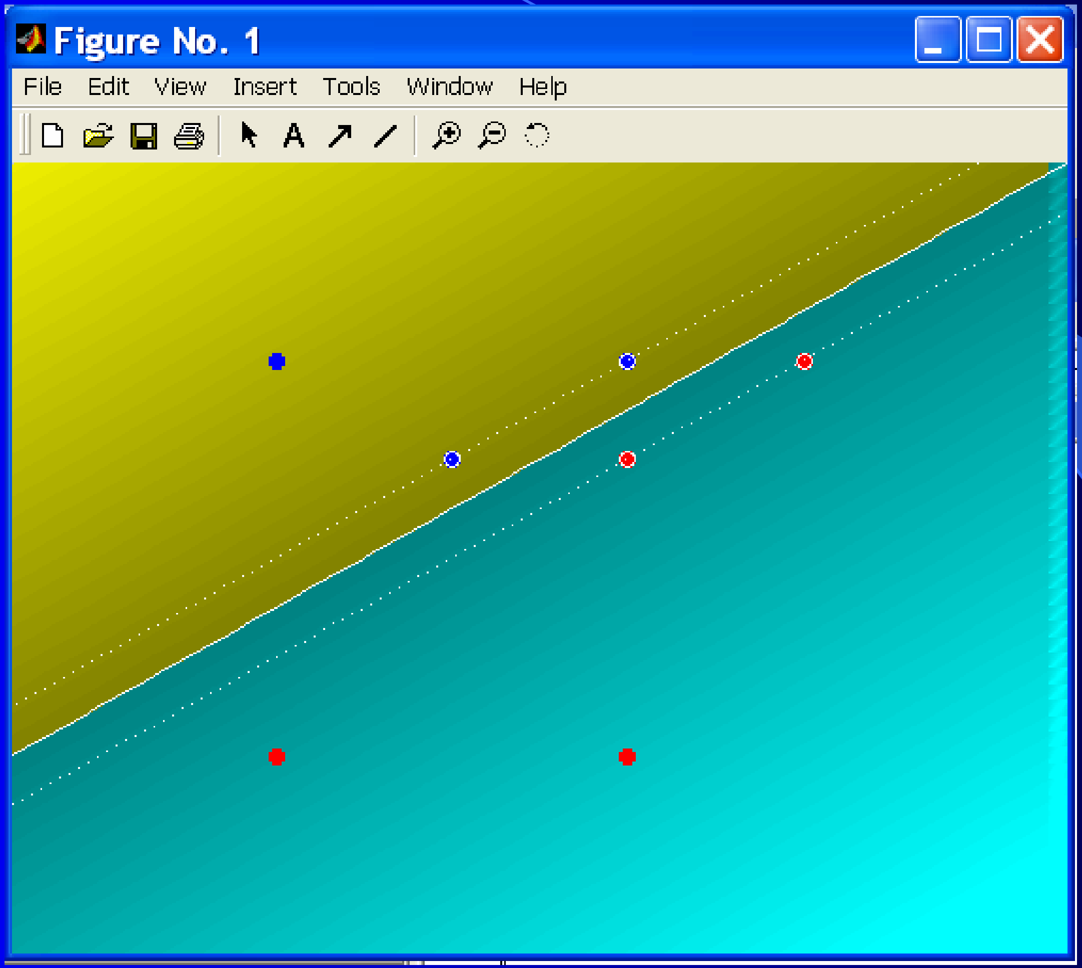
- An SVM constructs an optimal hyperplane that separates the data points of two classes as far as possible

No. Time Pregnant	Plasma Glucose	Blood Pressure	Fold Thickness	Serum Insulin	BMI	Pedi-gree	Age	Class
1	189	60	23	846	30.1	0.398	59	+
5	166	72	19	175	25.8	0.587	51	+
1	103	30	38	83	43.3	0.183	33	–
1	89	66	23	94	28.1	0.167	21	–
0	137	40	35	168	43.1	2.288	33	+
3	126	88	41	235	39.3	0.704	27	–
13	145	82	19	110	22.2	0.245	57	–
1	97	66	15	140	23.2	0.487	22	–
2	197	70	45	543	30.5	0.158	53	+

Optimal Separating Hyperplane

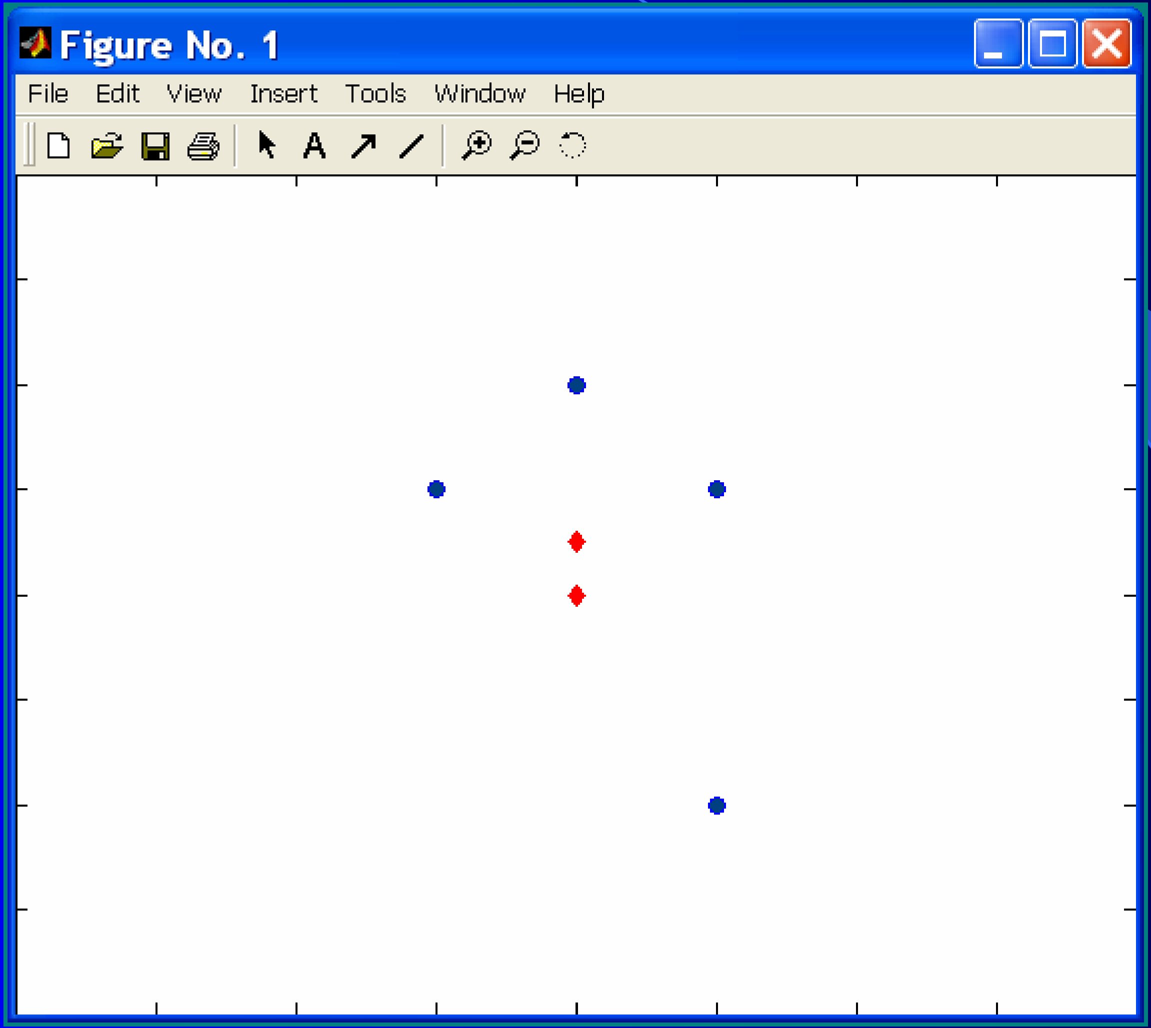


An Example of Linearly Separable Functions



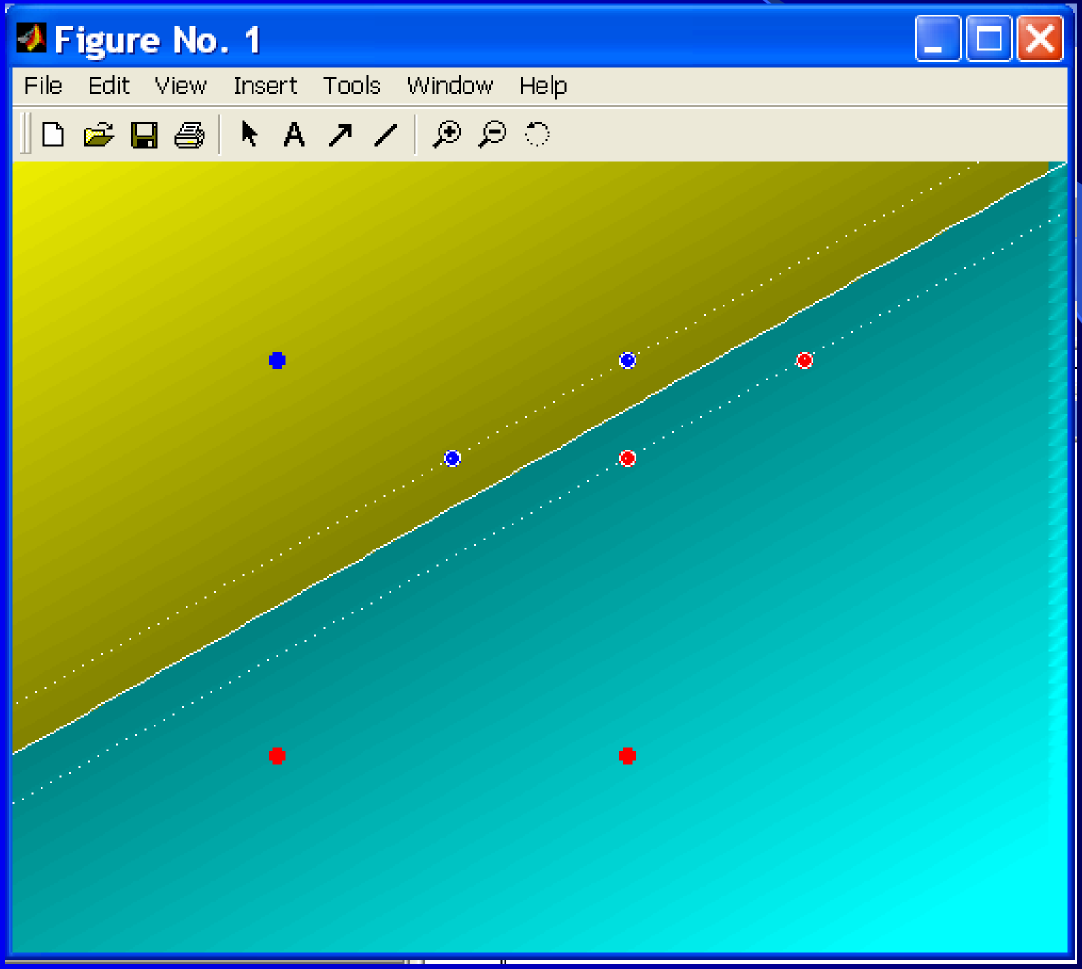
- No. of support vectors = 4

An Example of Linearly Non-Separable Functions



An Example of Linearly Separable Functions

- In case of using the input space



Feature Spaces

- For linearly non-separable function, it is very likely that a linear separator (hyperplane) can be constructed in higher dimensional space.
- Suppose we map the data points in the input space $\mathbf{R}^n$ into some feature space of higher dimension, $\mathbf{R}^m$ using function F

$$F : \mathbf{R}^n \rightarrow \mathbf{R}^m$$

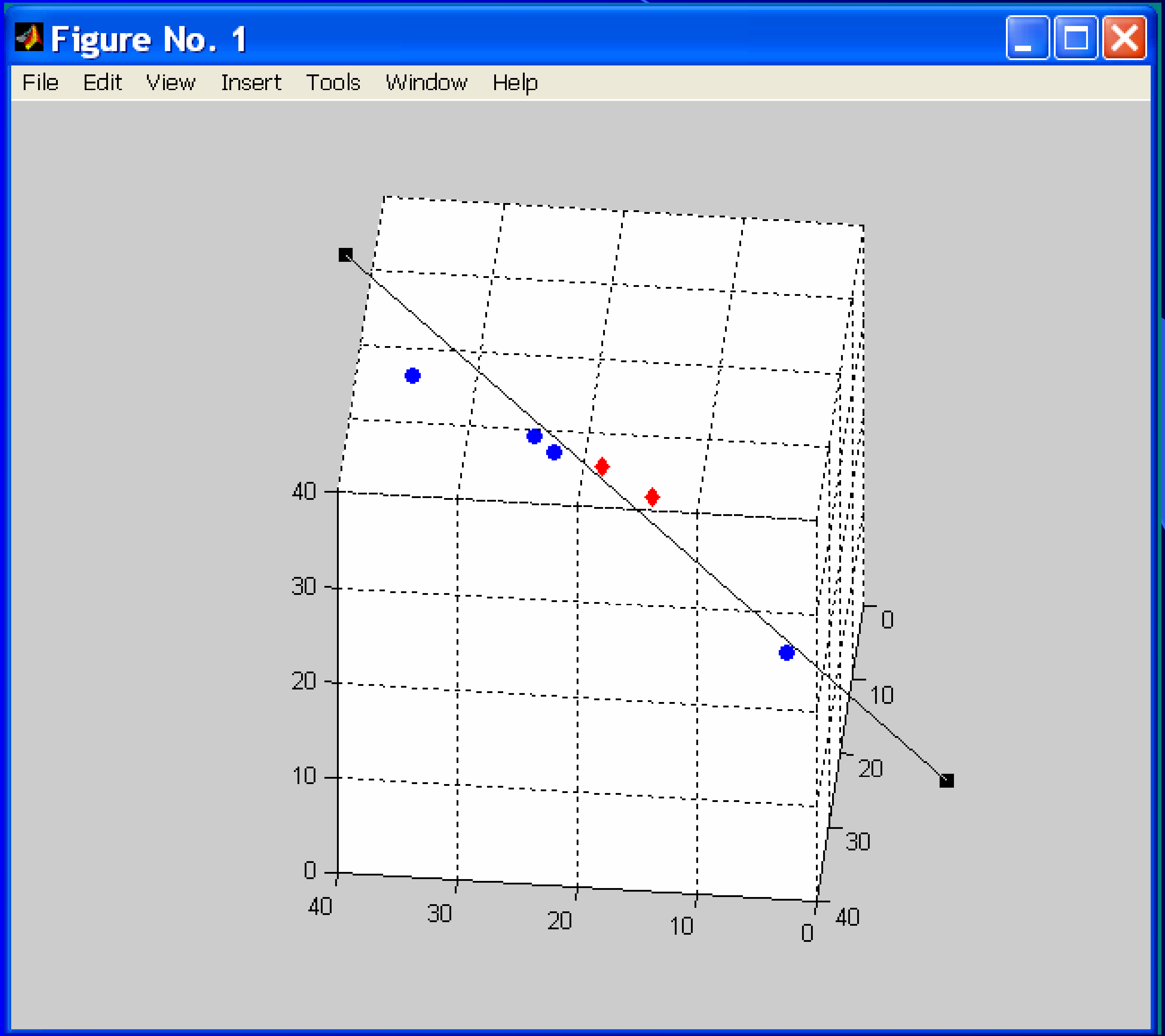
- Example:

$$F : \mathbf{R}^2 \rightarrow \mathbf{R}^3$$

$$\mathbf{x} = (x_1, x_2) \, , \quad F(\mathbf{x}) = (x_1, x_2, \sqrt{2}\, x_1 x_2)$$

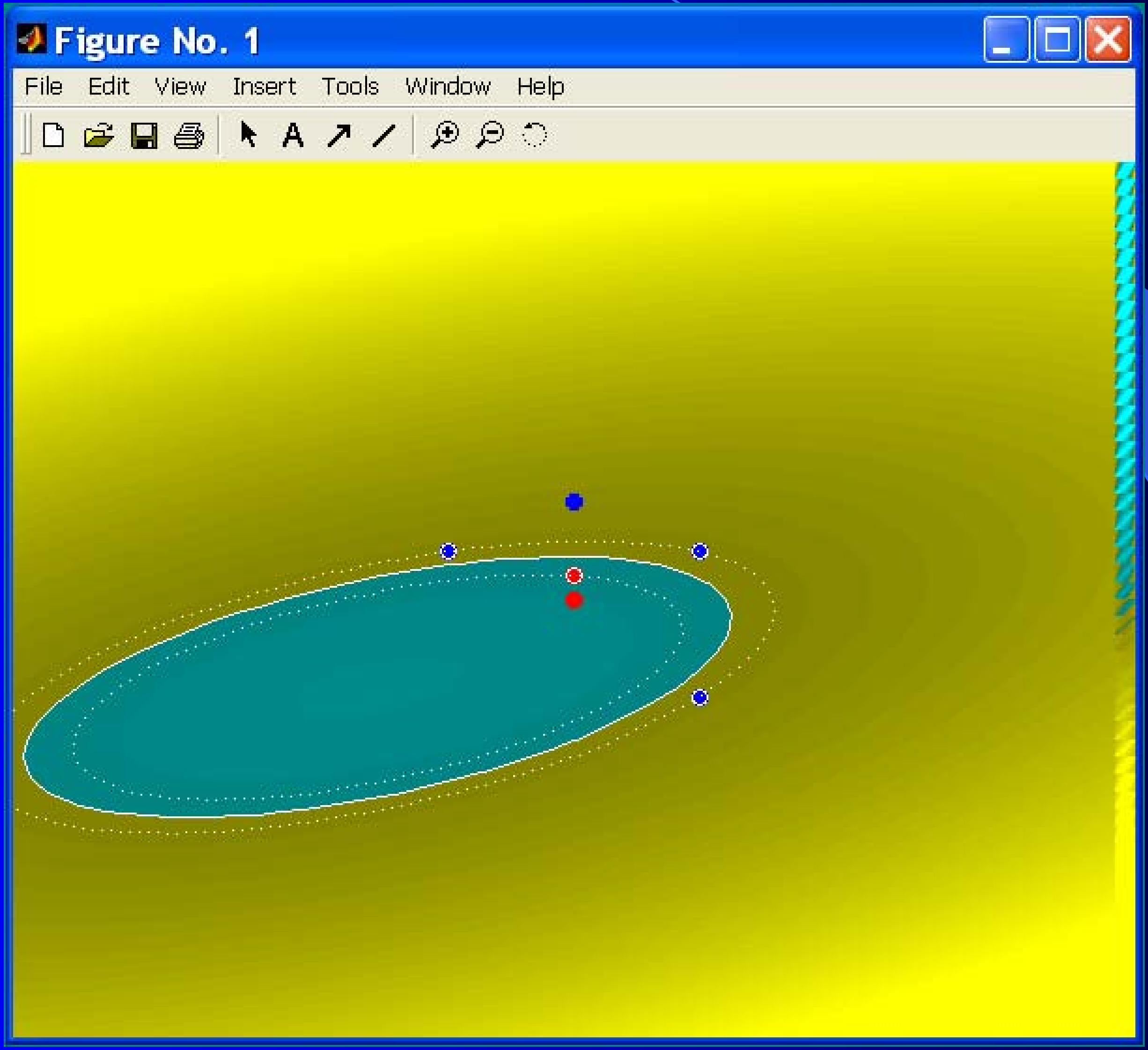
An Example of Linearly Non-Separable Functions

- In case of using the feature space: $F(\mathbf{x}) = (x_1, x_2, \sqrt{2}\, x_1 x_2)$



An Example of Linearly Non-Separable Functions

- The corresponding non-linear function in the input space



Multiclass SVMs

- The Max Win algorithm is an approach for constructing multiclass SVMs

  Training:
  - Construct all possible binary SVMs
  - For N classes, there will be $N(N\text{-}1)/2$ binary SVMs
  - Each classifier is trained on 2 out of N classes

  Testing:
  - A test example is classified by all classifiers
  - Each classifier provides one vote for its preferred class
  - The majority vote is the final output

Comparison of the Algorithms on Diabetes

Algorithm	Accuracy
Decision Tree	78.65%
Neural Network	79.17%
Support Vector Machine	78.65%