Week 4
Classification (Part III)

Seokho Chi
Professor | Ph.D.
SNU Construction Innovation Lab

Source: *Tan, Kumar, Steinback (2006)*

서울대학교 건설혁신연구실
Construction Innovation Lab SNU

Neural Networks

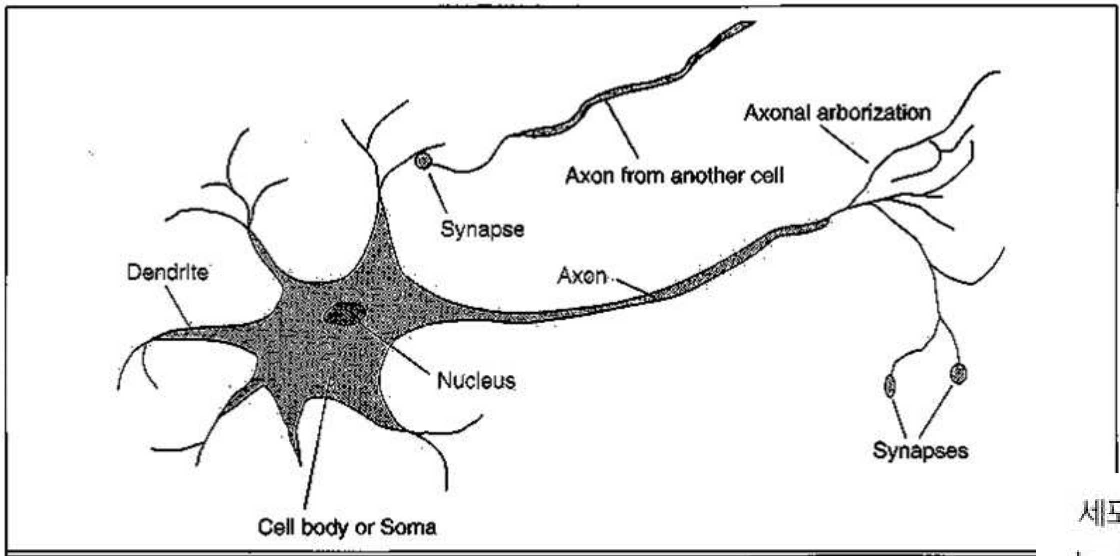
- Machine learning algorithm
  - Computer learning mechanism based on experiences, examples, and predictions
  - Upgrade performance as learning time goes by

- Analogy to Biological Systems “Human Brain”
- Massive Parallelism allowing for computational efficiency

Neuron

- The cell that perform information processing in the brain
- Fundamental functional unit of all nervous system tissue
- Each neuron consists of :

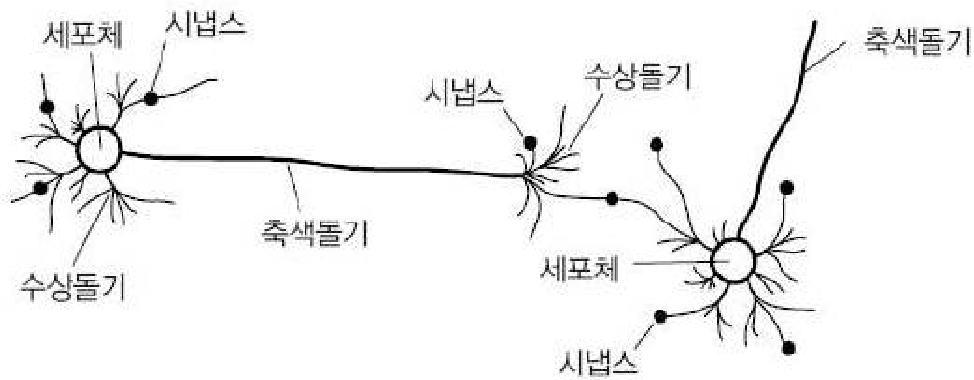
  CELL BODY, DENDRITES, AXON, and SYNAPSE.



Neuron:
Basic unit for information processing

Brain: complex, non-linear, and parallel connections of neurons

Adaptation (weighting):
Weaken "wrong" connections &
strengthen "correct" connections

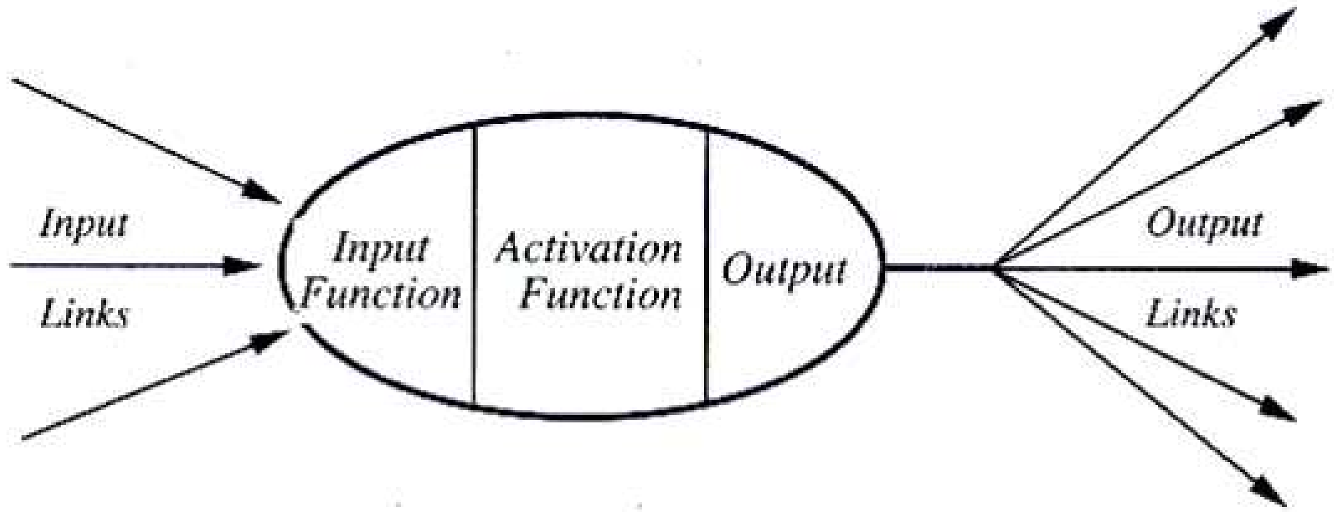


Definition of Neural Network

- A Neural Network is a system composed of many simple processing elements operating in parallel which can acquire, store, and utilize experiential knowledge

- Components
  - Each element is a node called unit
  - Units are connected by links
  - Each link has a numeric weight
    - Weight explains the importance of links
    - Machine learning controlling and training weights repeatedly

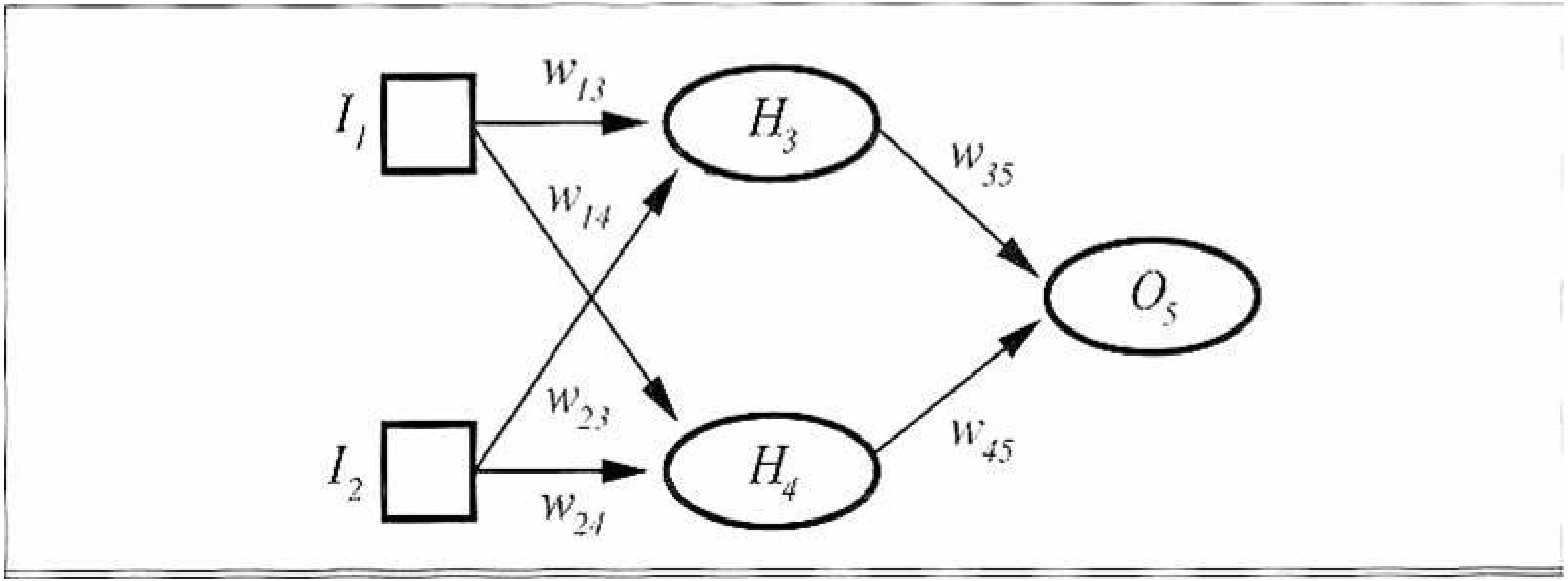
Computing Elements



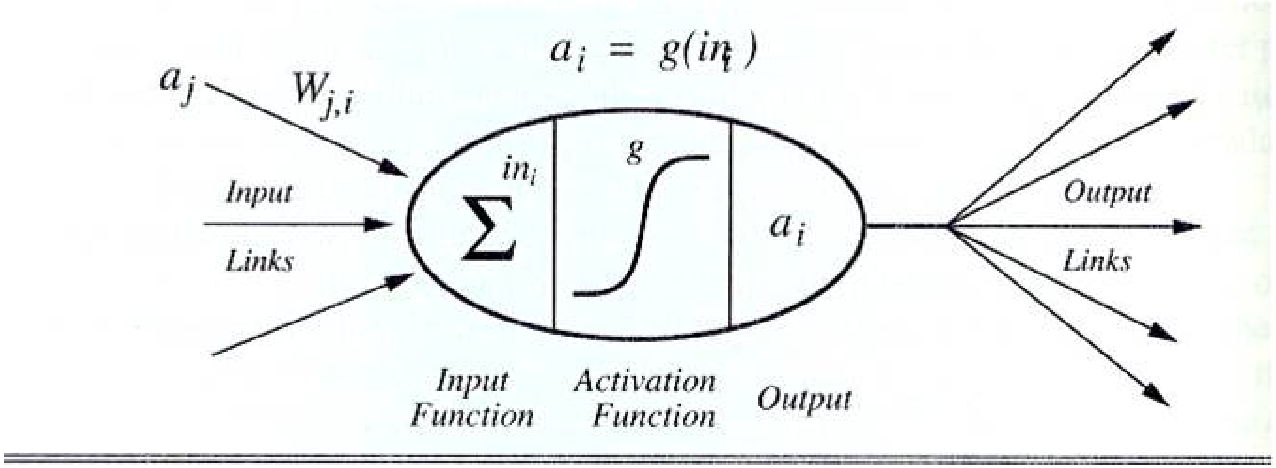
Require decisions on:

- The number of units to use
- The type of units required
- Connection between the units

Neural Network Example



NN Computing Unit



- 2 components: Linear and Non-linear
- Linear: Input function
    - calculate weighted sum of all inputs

$$in_i = \sum_j W_{j,i} a_j = \mathbf{W}_i \cdot \mathbf{a}_i$$

- Non-linear: Activation function
    - transform sum into activation level

$$a_i \leftarrow g(in_i) = g\left(\sum_j W_{j,i} a_j\right)$$

Activation Functions

- Use different functions to obtain different models
  1) Compare sum of inputs with the threshold value
  2) Output becomes "0", "-1", etc. if sum is smaller
  3) Output becomes "+1" if sum is same or larger
     → "activated"

- 3 most common choices
  1) Step function
  2) Sign function
  3) Sigmoid function

Activation Functions

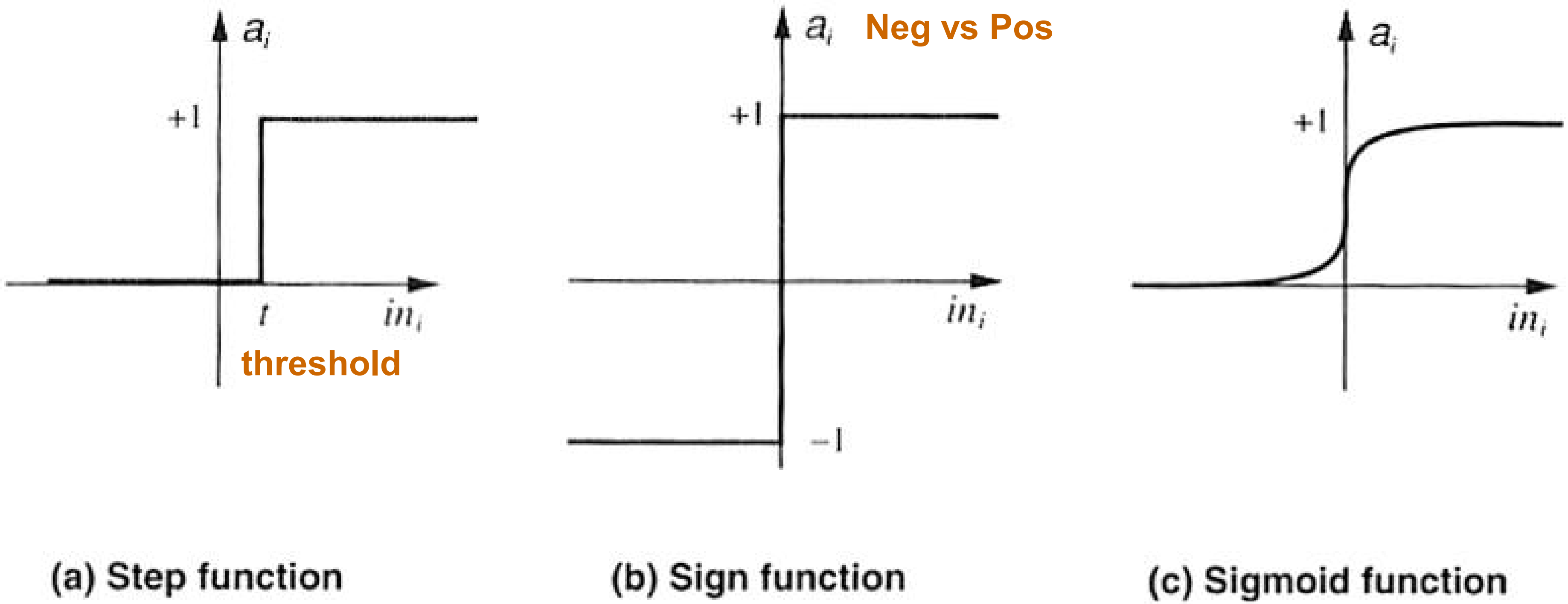


Figure 19.5 Three different activation functions for units.

$$\text{step}_t(x) = \begin{cases} 1, & \text{if } x \geq t \\ 0, & \text{if } x < t \end{cases} \quad \text{sign}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \quad \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Feed-forward Neural Networks (FF NN)

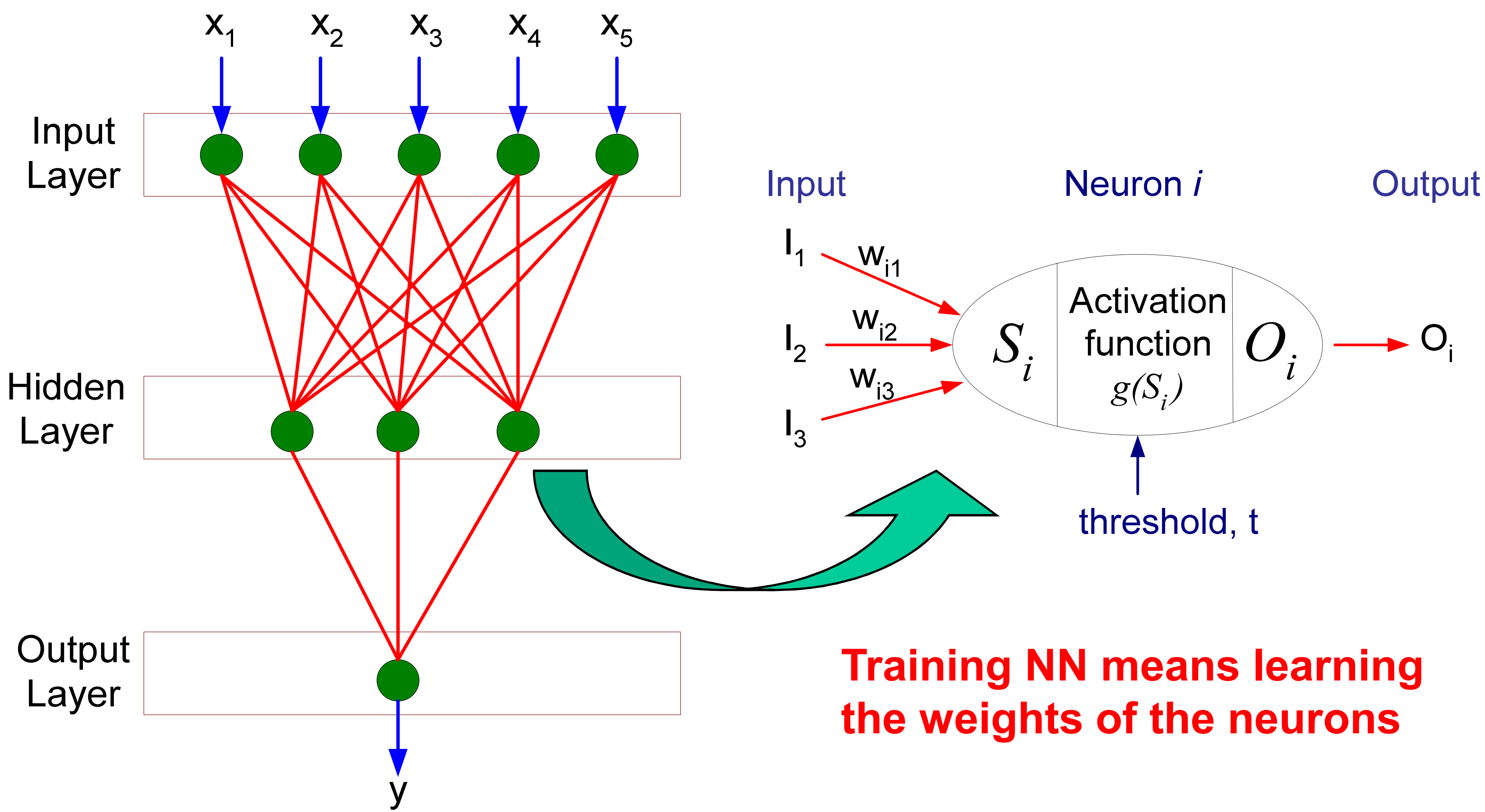
- Arranged in layers

- Each unit is linked only in the unit in next layer

- No units are linked between the same layer, back to the previous layer or skipping a layer (only go in one direction)

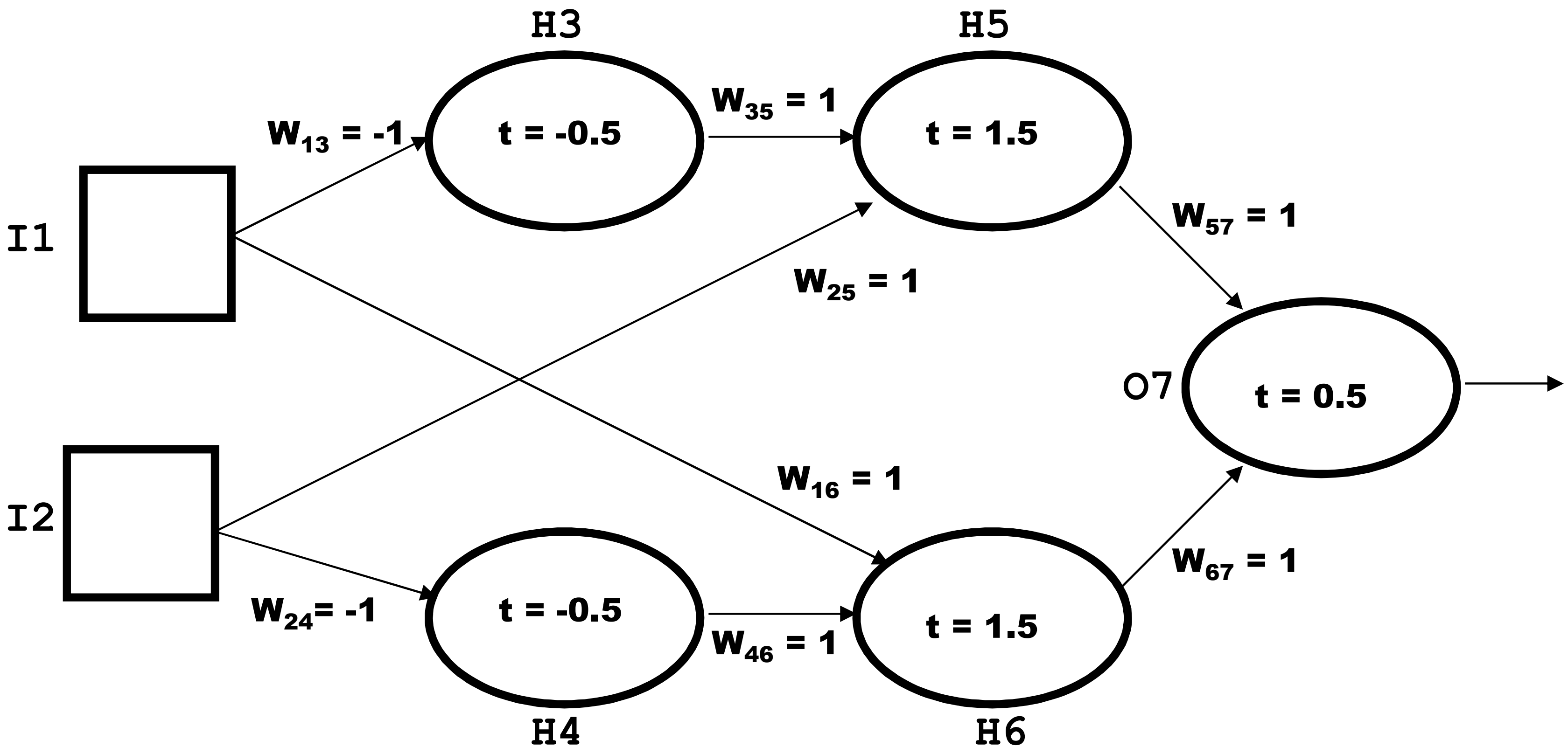
- Computations can proceed uniformly from input to output units

General Structure of FF NN



Training NN means learning the weights of the neurons

Feed-Forward Example



Network Layers

- Networks without hidden layer are called a "perceptron"
- Multi-layer Networks: Have one or more layers of hidden units
- With two possibly very large hidden layers, it is possible to implement any function

NN Learning

- Initial network has a randomly assigned weights

- Weight adjustments are made to reduce the difference between the observed and predicted values

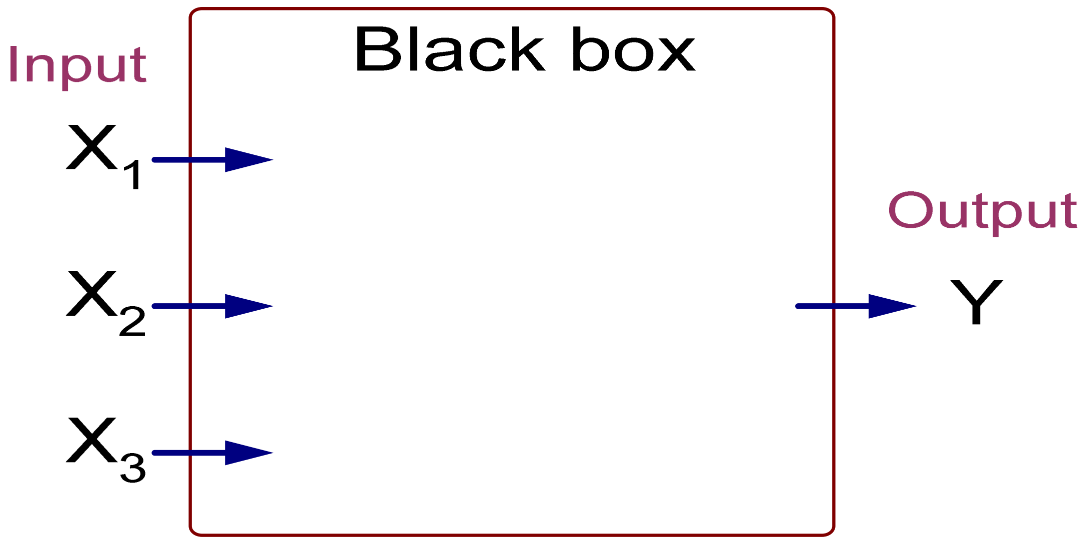
- Need to repeat the update phase several times in order to achieve convergence.

- Updating process is divided into epochs, each epoch updates all the weights of the process.

Epoch: 1 training

Artificial Neural Networks (ANN)

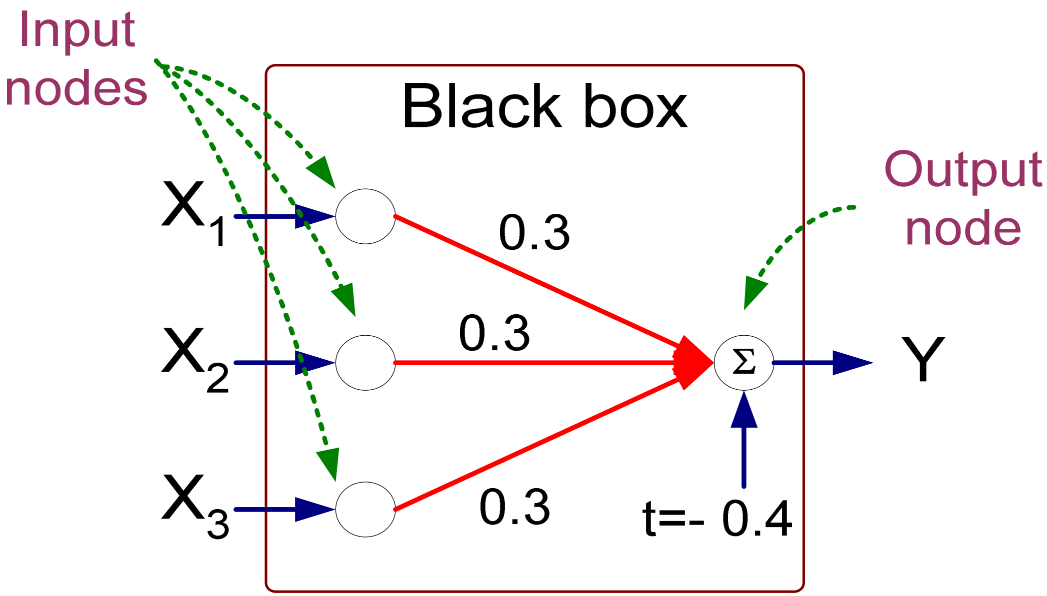
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



Output Y is 1 if at least two of the three inputs are equal to 1.

Artificial Neural Networks (ANN)

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

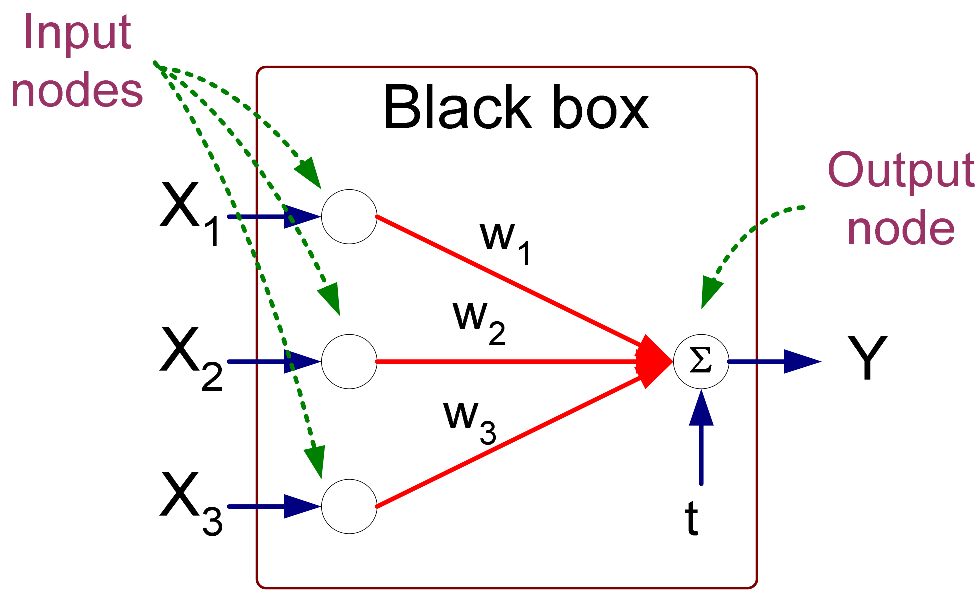


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{where } I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links
- Output node sums up each of its input value according to the weights of its links
- Compare output node against some threshold t



Perceptron Model

$$Y = sign(\sum_i w_i X_i - t)$$

Introduction to Backpropagation

- In 1969 a method for learning in multi-layer network, Backpropagation, was invented by Bryson and Ho.
- The Backpropagation algorithm is a sensible approach for dividing the contribution of each weight.
- Can model complex functions

Backpropagation Algorithm

The ideas of the algorithm can be summarized as follows:

- Computes the error term for the output units using the observed error

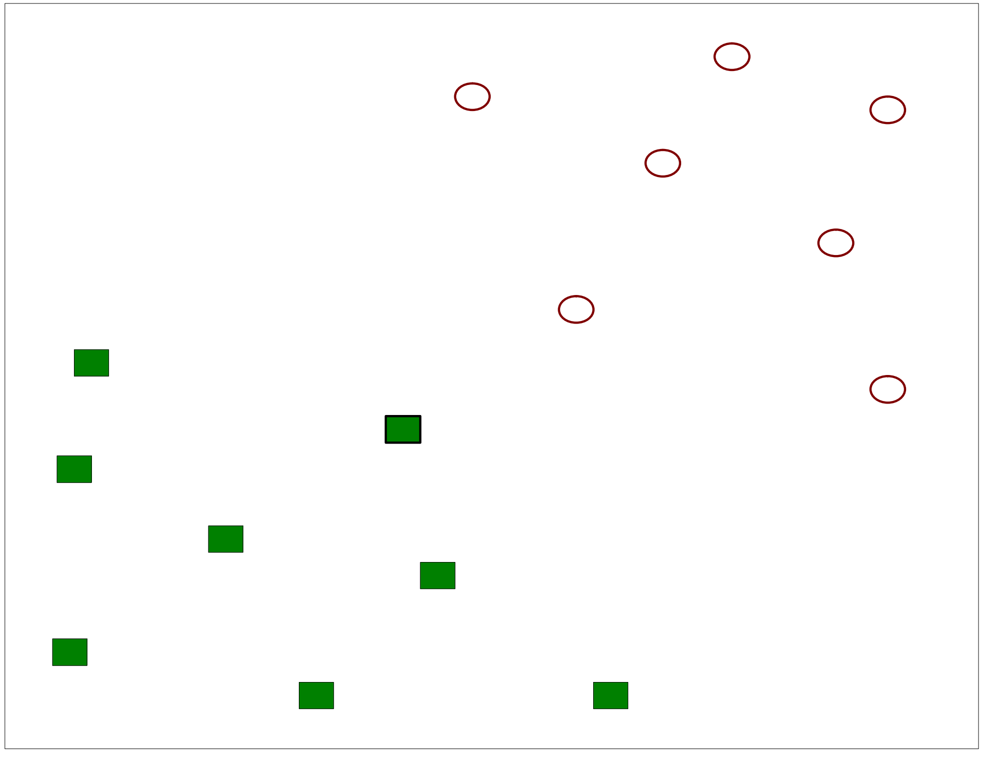
- From output layer, repeat propagating the error term back to the previous layer and updating the weights between the two layers until the earliest hidden layer is reached

References

- Han, J. and Kamber, M. (2001) Data Mining: Concepts and Techniques, 1st edition, Morgan Kaufmann.
- Russel, S. and P. Norvig (1995). Artificial Intelligence: A Modern Approach. Prentice Hall.
- Tan, P., Steinbach, M., and Kumar, V. (2005) Introduction to Data Mining, 1st edition, Addison-Wesley.

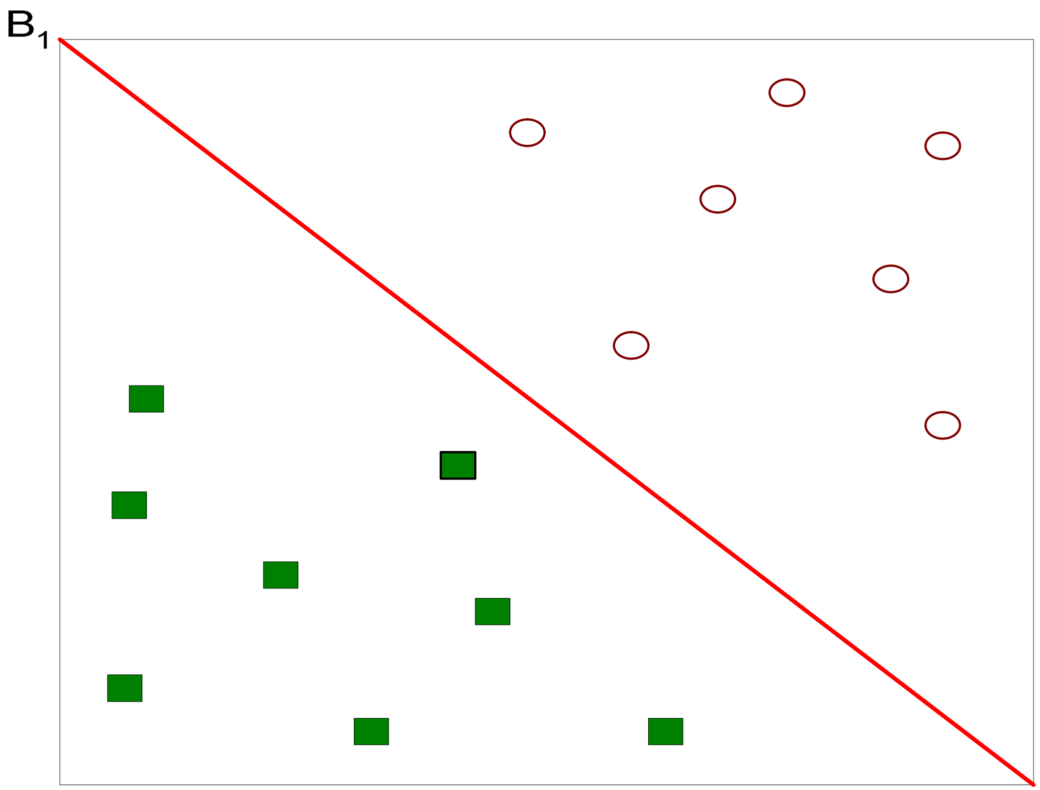
Support Vector Machines (SVM)

Non-probabilistic binary linear classifier



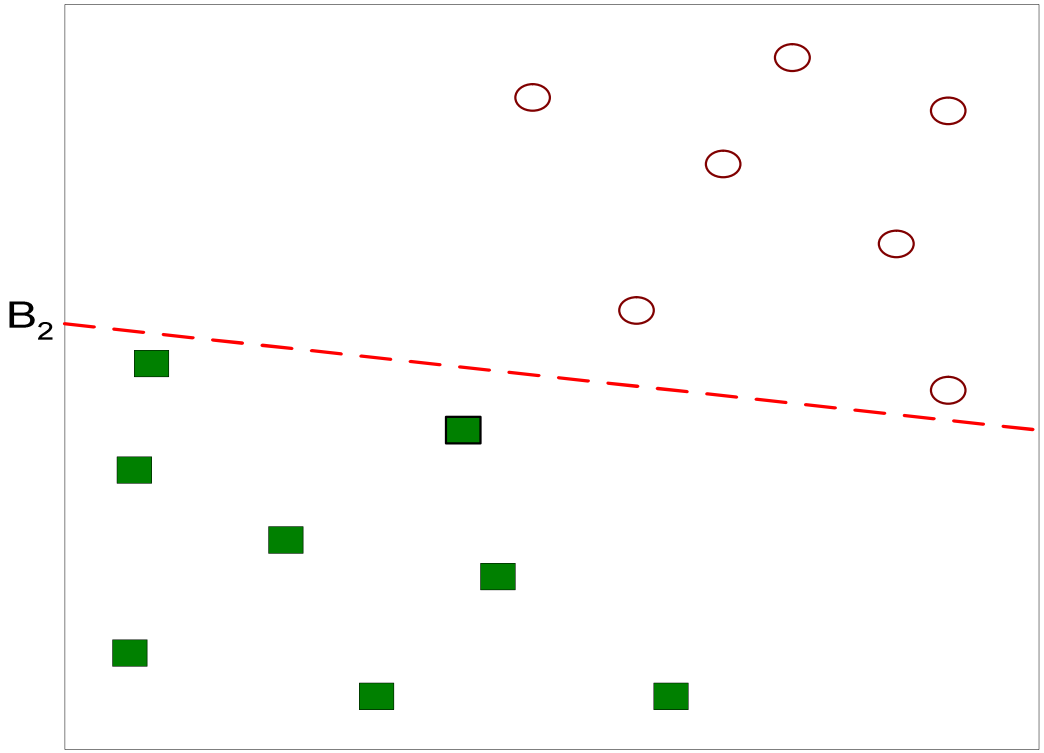
- Find a linear hyperplane (decision boundary) that will separate the data

SVM



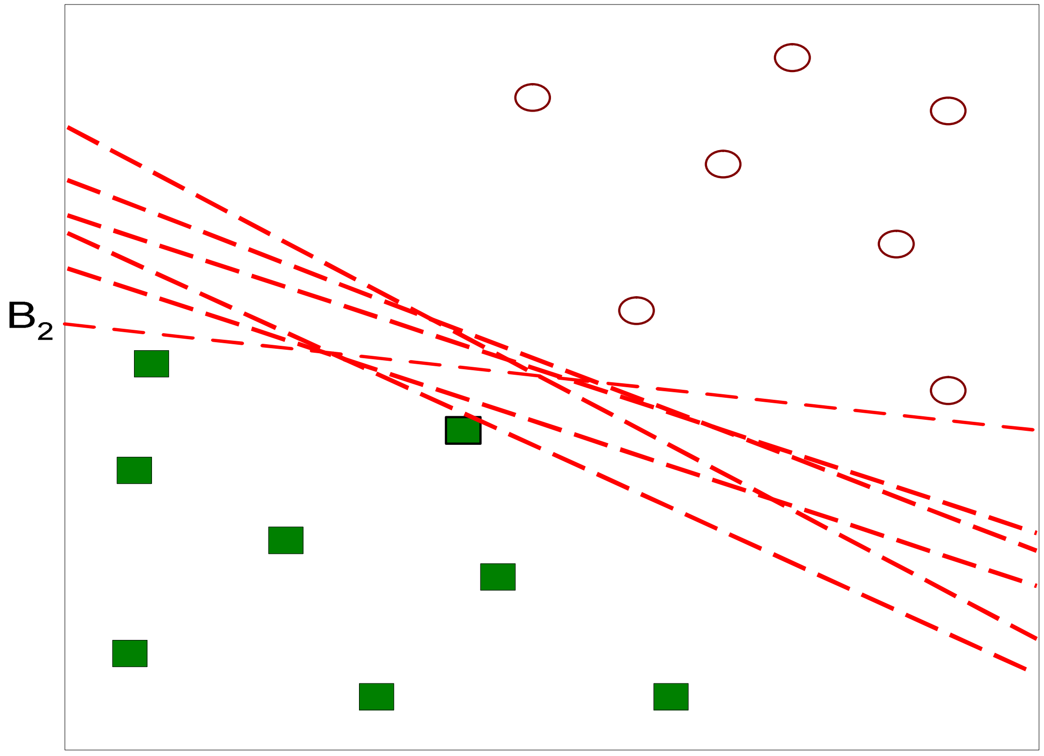
- One Possible Solution

SVM



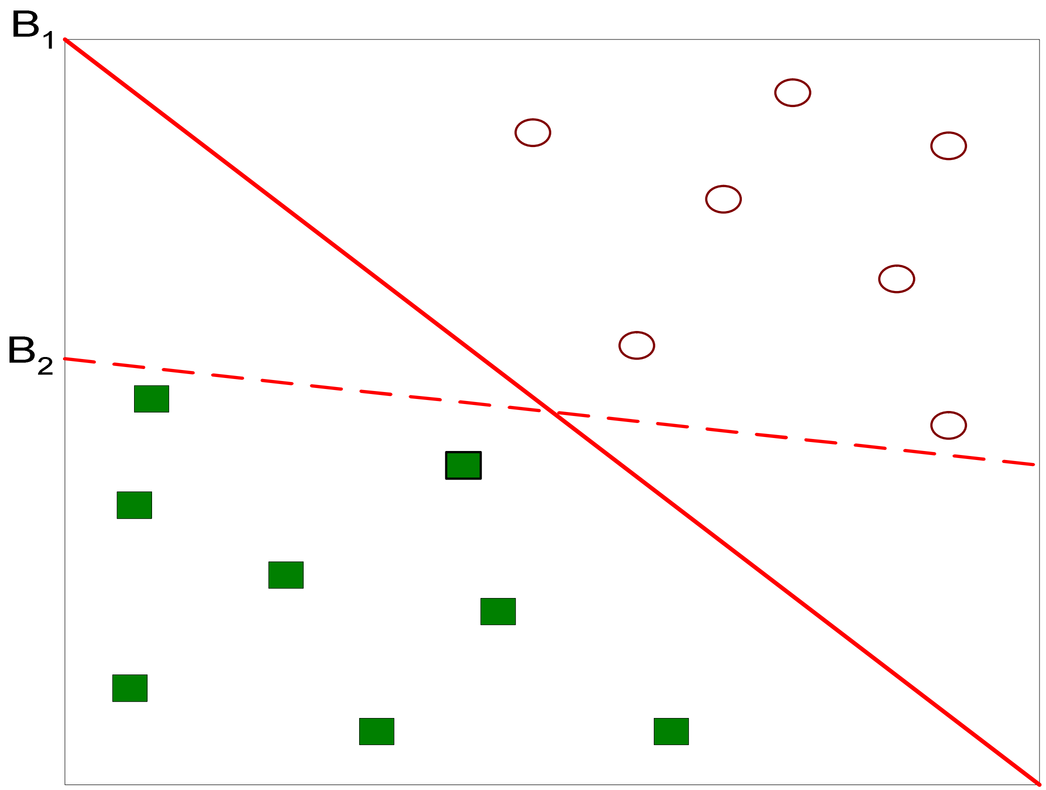
- Another possible solution

SVM



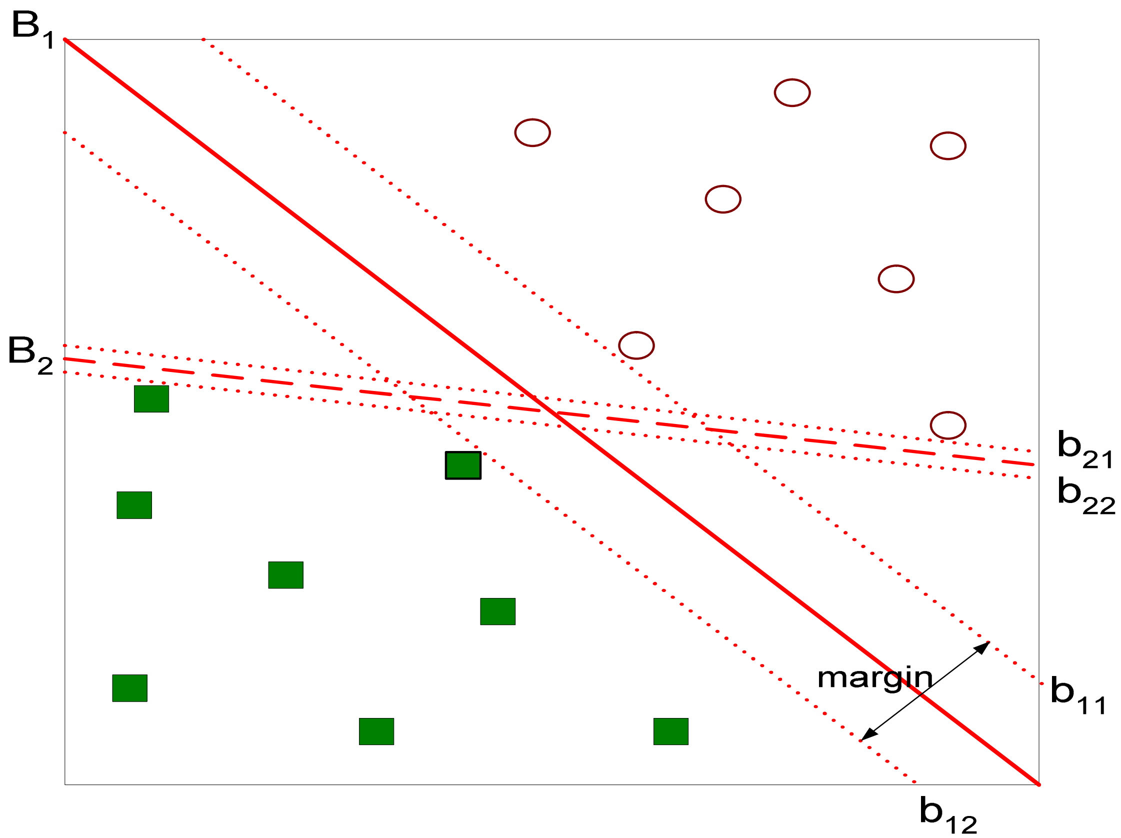
- Other several possible solutions

SVM



- Which one is better? B1 or B2?
- How do you define better?

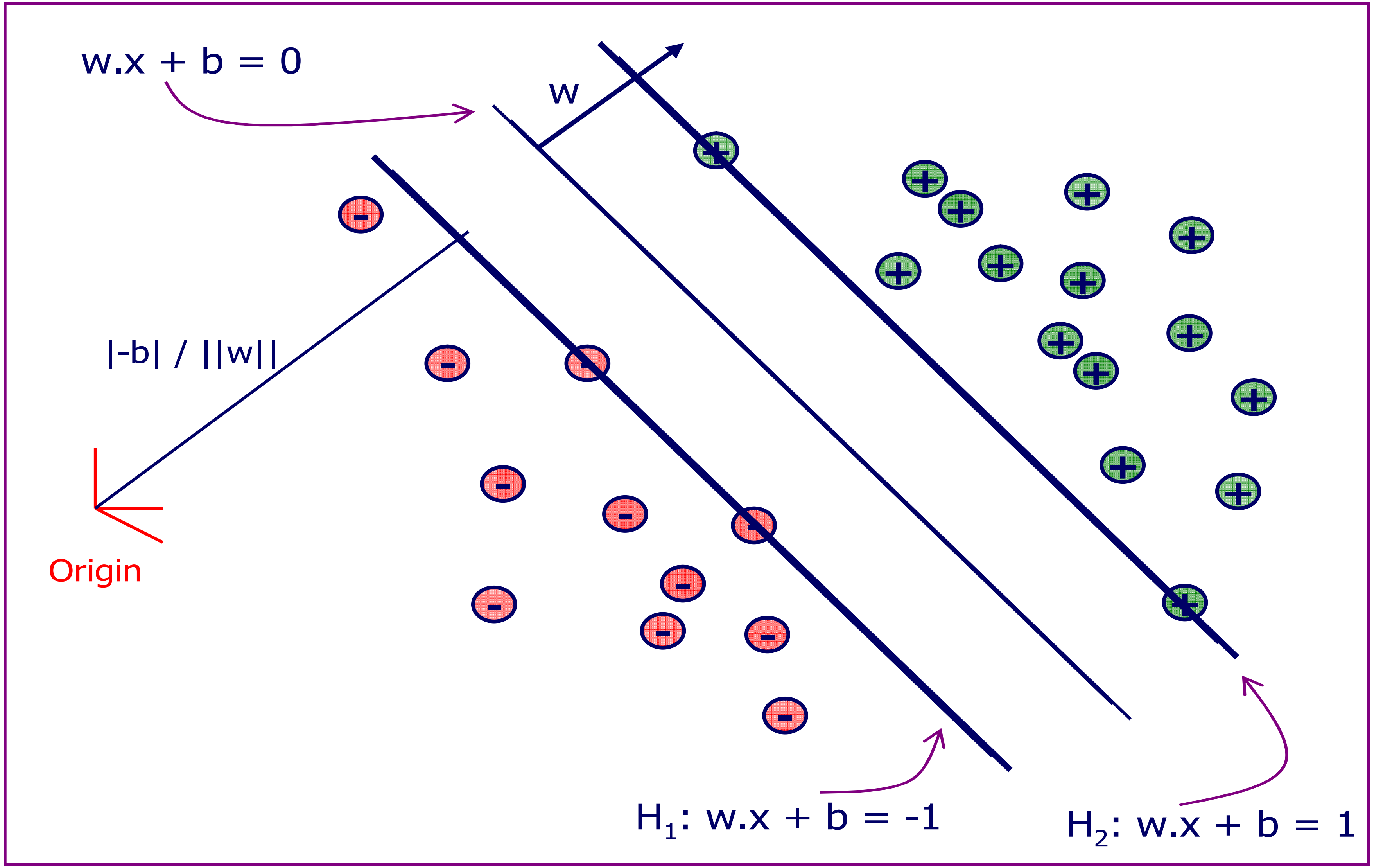
SVM



- Find hyperplane maximizes the margin
  → Better generalization errors → B1 is better than B2

(smaller margin → slight perturbations to the decision boundary can have a significant result)

SVM – Model Building



SVM – Model Building

- Formulation:

Suppose that all training data satisfy the following constraints

$$x_i \cdot w + b \geq +1 \quad for\ y_i = +1$$

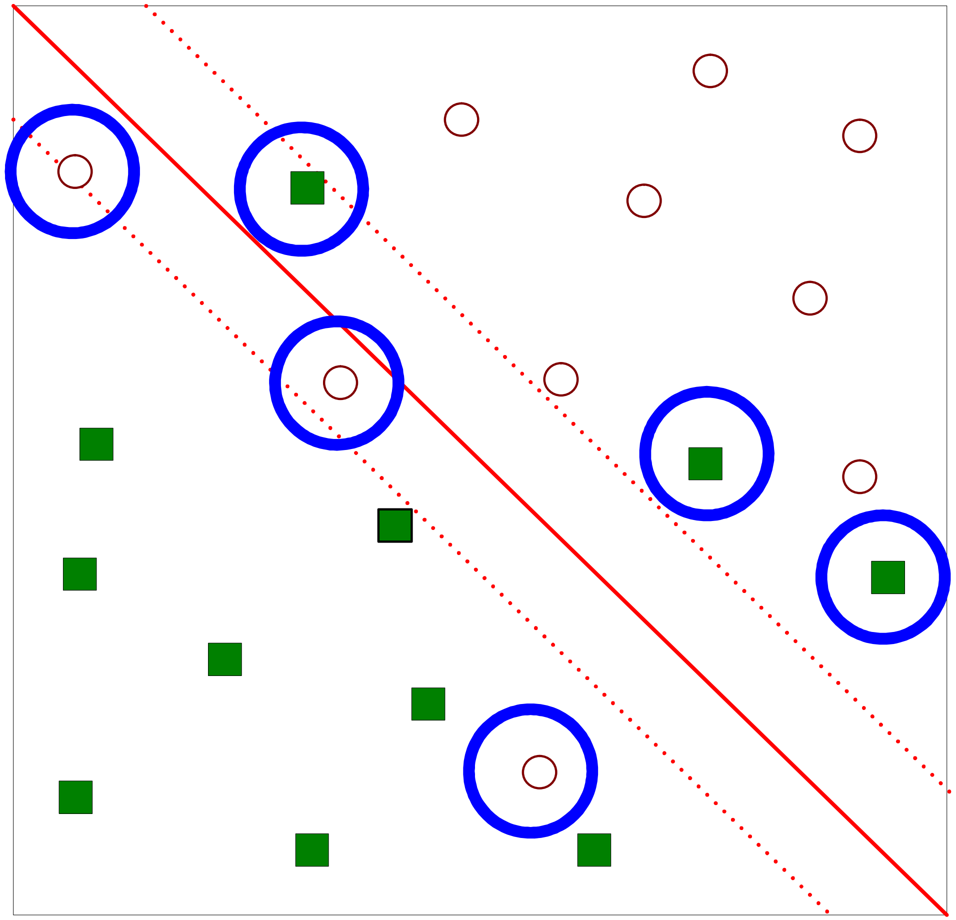
$$x_i \cdot w + b \leq -1 \quad for\ y_i = -1$$

These can be combined into one set of inequalities

$$y_i (x_i \cdot w + b) - 1 \geq 0 \quad \forall i$$

Support Vector Machines

- What if the problem is not linearly separable?



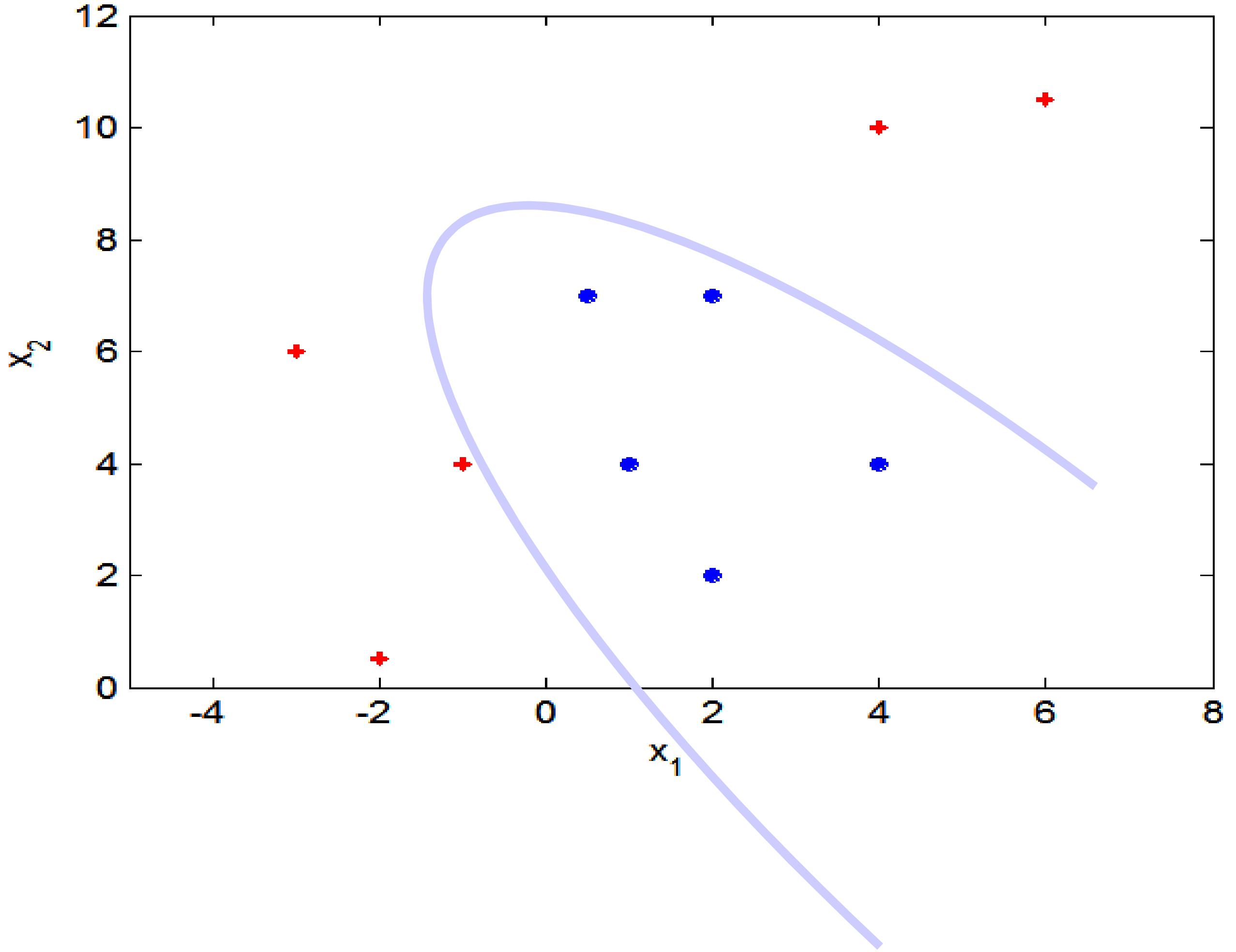
Support Vector Machines

- What if the problem is not linearly separable?
  - Introduce slack variables ξ_i (크사이)
    - Soft Margin method
    - Which measures the degree of misclassification of the data
    - Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

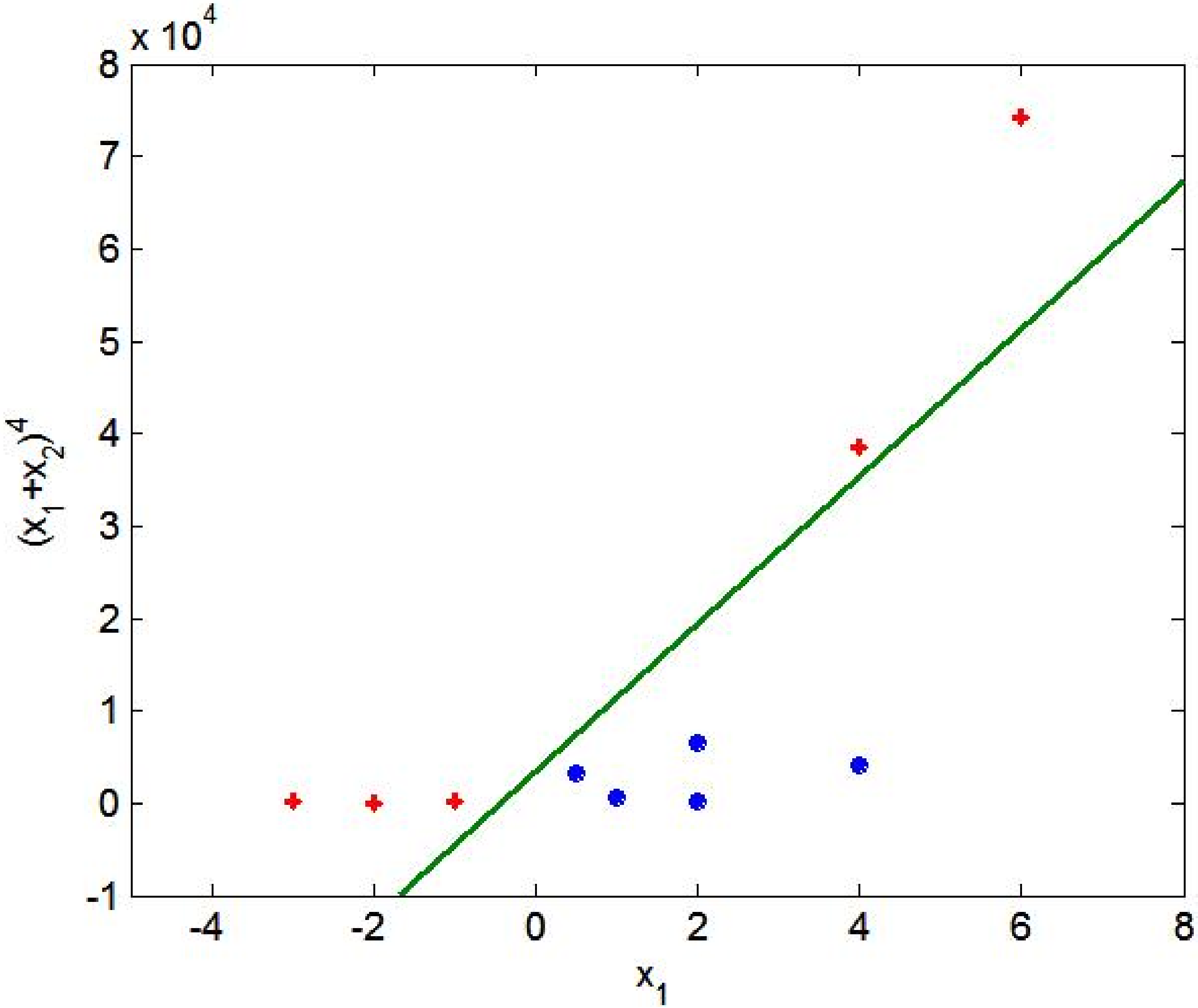
Nonlinear Support Vector Machines

- What if decision boundary is not linear?



Nonlinear Support Vector Machines

- Transform data into higher dimensional space



SVM - Advantages

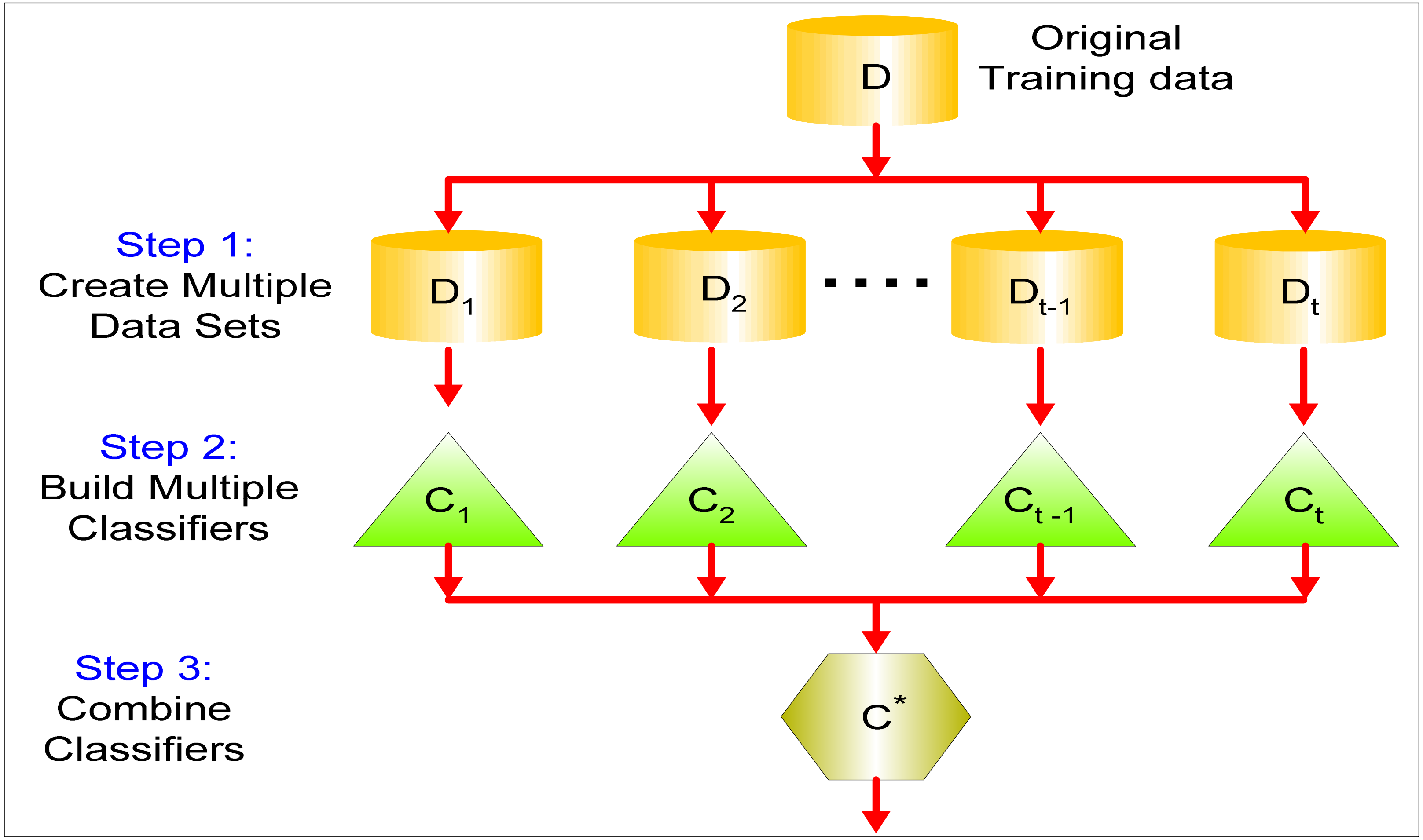
- Mathematical formulation → Objective quantification
- High dimensional feature spaces → Advantages for multi-dimensional analysis
- Easier generalization

Ensemble Methods

- Construct a set of classifiers from the training data (also known as classifier combination)

- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

General Idea



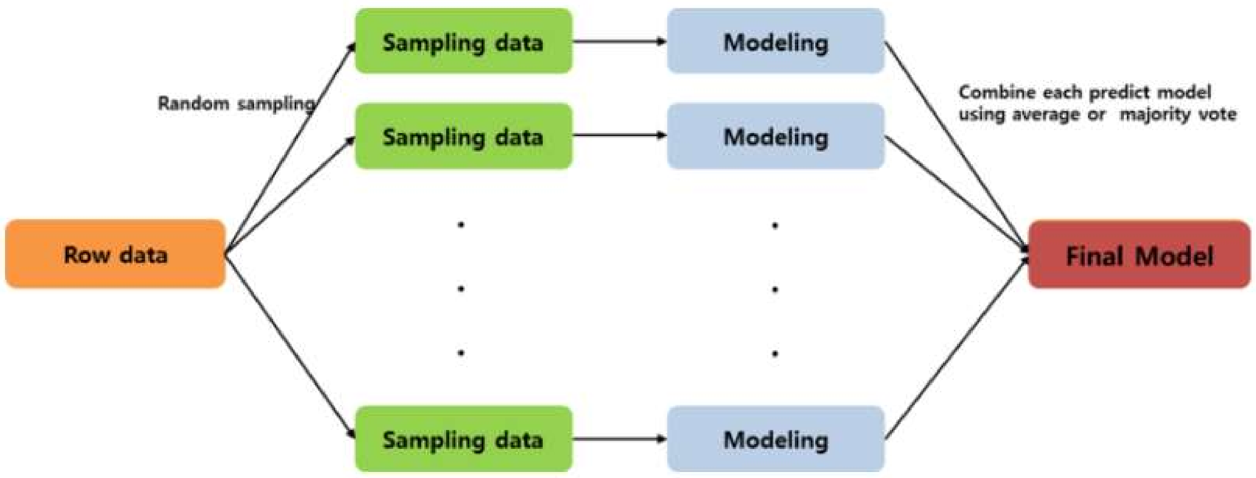
Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - The ensemble classifier predicts the class label of a test example by taking a majority vote on the predictions
  - Assume classifiers are independent (thus their errors are uncorrelated)
  - Probability that the ensemble classifier makes a wrong prediction: more than half of the classifiers (13 and more) predict wrong

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

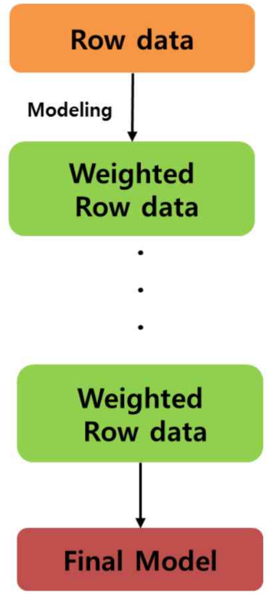
Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
  - Bagging (Bootstrap Aggregating): repeatedly samples from a data set with replacement according to a uniform probability distribution (모집단의 성질에 대해 표본을 통해 추정할 수 있는 것처럼 표본의 성질에 대해서도 재표본을 통해 추정할 수 있다는 개념: 무작위 추출 → 표본통계 → 이를 반복)



Examples of Ensemble Methods

– Boosting: an iterative procedure used to adaptively change the distribution of training examples so that the base classifiers will focus on examples that are hard to classify → add erroneous samples into the training data for model building



XGBoost (eXtreme Gradient Boosting)

"Generate a tree → Weight more to misclassified samples to be selected for next → Repeat generating a tree"