Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology

Chapter 4
Data Classification

Assoc. Prof. TRAN MINH QUANG
quangtran@hcmut.edu.vn
http://researchmap.jp/quang

2020/4/30

CONTENT

1. Overview
2. Logistic regression
3. Decision tree
4. Bayesian method
5. Artificial neural network (ANN)
6. Other classification methods
7. Evaluation and selection of models
8. Summary

REFERENCES

[1] Jiawei Han, Micheline Kamber, and Jian Pei, "Data Mining: Concepts and Techniques", 3rd Edition, Morgan Kaufmann Publishers, 2012.

[2] David Hand, Heikki Mannila, Padhraic Smyth, "Principles of Data Mining", MIT Press, 2001.

[3] David L. Olson, Dursun Delen, "Advanced Data Mining Techniques", Springer-Verlag, 2008.

[4] Graham J. Williams, Simeon J. Simoff, "Data Mining: Theory, Methodology, Techniques, and Applications", Springer-Verlag, 2006.

[5] ZhaoHui Tang, Jamie MacLennan, "Data Mining with SQL Server 2005", Wiley Publishing, 2005.

[6] Oracle, "Data Mining Concepts", B28129-01, 2008.

[7] Oracle, "Data Mining Application Developer's Guide", B28131-01, 2008.

[8] Ian H.Witten, Eibe Frank, "Data mining : practical machine learning tools and techniques", 2nd Edition, Elsevier Inc, 2005.

[9] Florent Messeglia, Pascal Poncelet & Maguelonne Teisseire, "Successes and new directions in data mining", IGI Global, 2008.

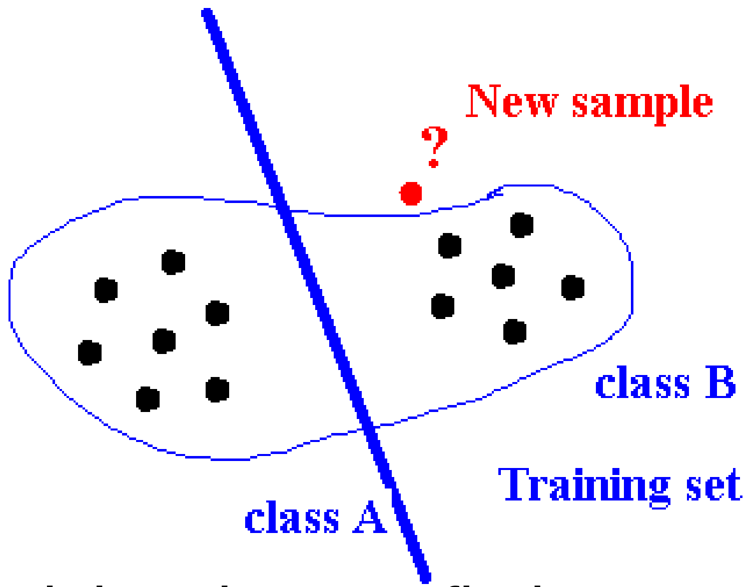
[10] Oded Maimon, Lior Rokach, "Data Mining and Knowledge Discovery Handbook", 2nd Edition, Springer Science + Business Media, LLC 2005, 2010.

1. OVERVIEW

- Situations
  - Email: *"spam"* or *"normal"*
  - Online transactions (e-commerce): *"fraud"* or *"normal"*
  - Healthcare: *"sick"* or *"not sick"*; tumor *"benign"* or *"malignant"*,…
  - Y $\in$ {0, 1}: 0: *"negative"* or 1: *"positive"* classes
  - Y $\in$ {0, 1, 2, 3}: multiple classes
  - Each class is labeled: Ex., *"spam"* or *"not spam"*

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

1. OVERVIEW



• Given a training dataset, find out models that describe class A and B

• Given a new pattern/object, identify the class it belongs to?

• Evaluate whether the selected class is really appropriate with the given pattern/object?

1. OVERVIEW

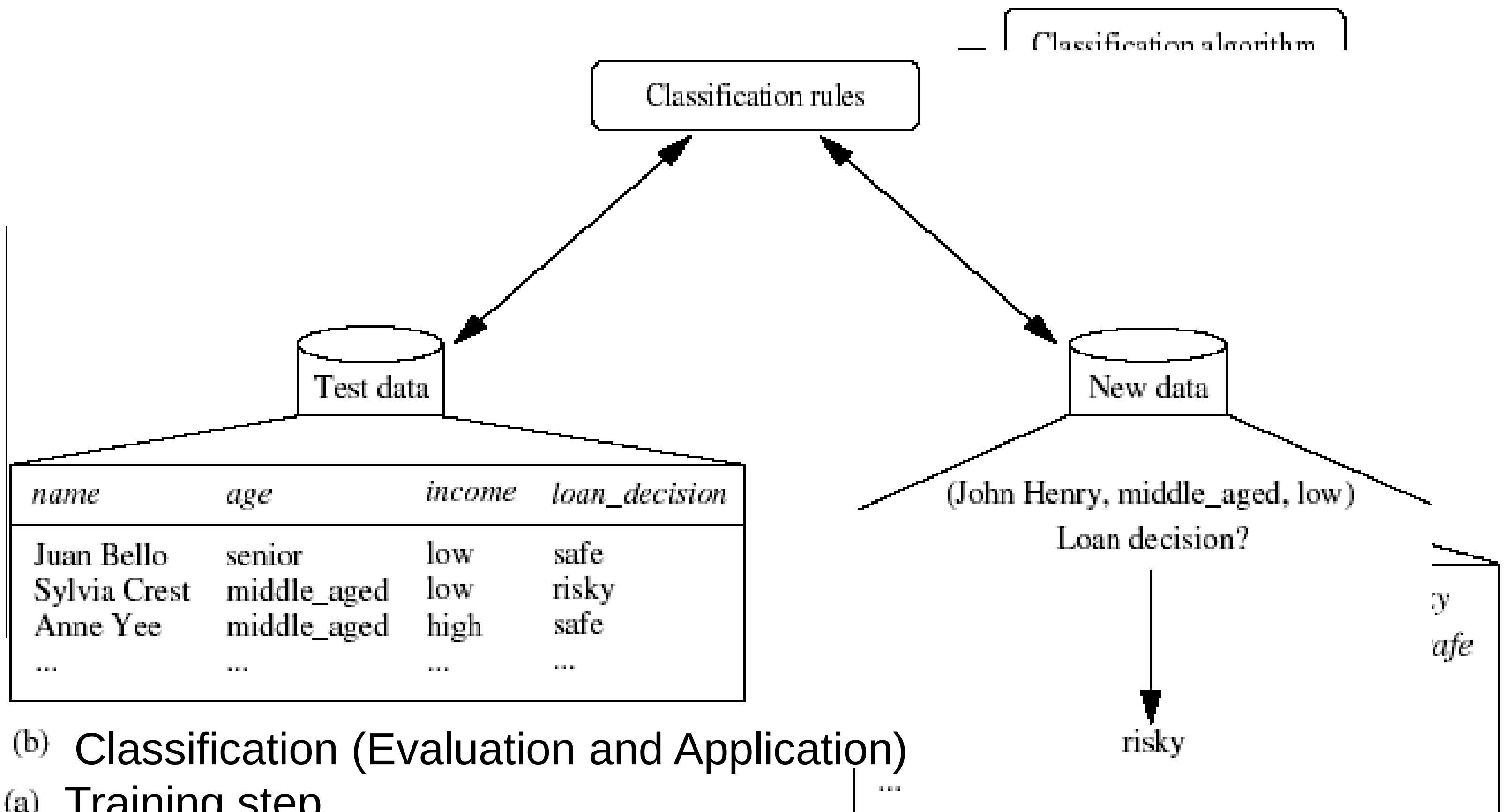
- Classification
  - Is a data analysis method that extract models that describe data classes or the trends of data
  - Is a two steps process:
    - **Training**: to build the classifier by analyzing (learning) the training dataset
    - **Classification**: classify new patterns/object, if the accuracy of the built classifier is acceptable

y = f (X): y is label (the description) of a class and X is the data/object

- **Training**: tuple <X,y> is given in the training dataset → identify f

- **Classification**: given a set of <X', y'>, where X' <> X (testing dataset) **->** evaluate f. If the accuracy of f is acceptable then use f to identify y'' for any new given X''

1. OVERVIEW

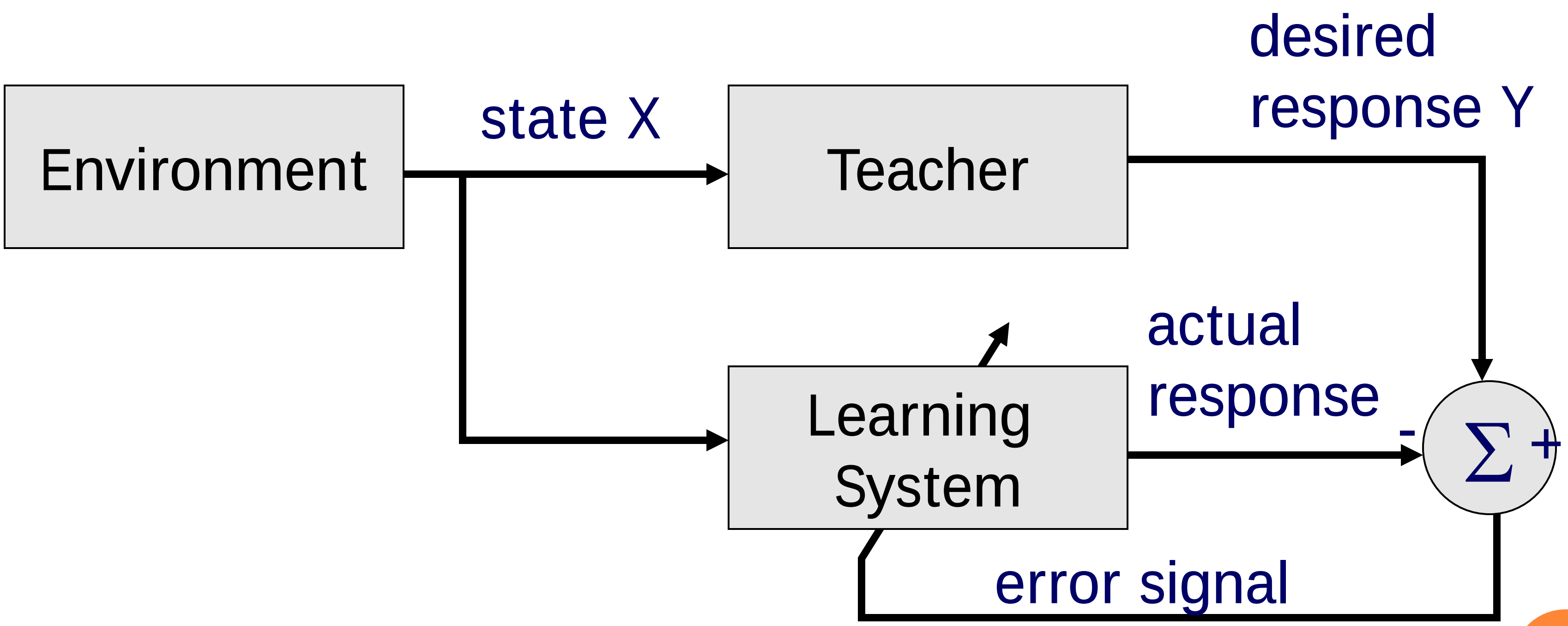


(b) Classification (Evaluation and Application)

(a) Training step

1. OVERVIEW

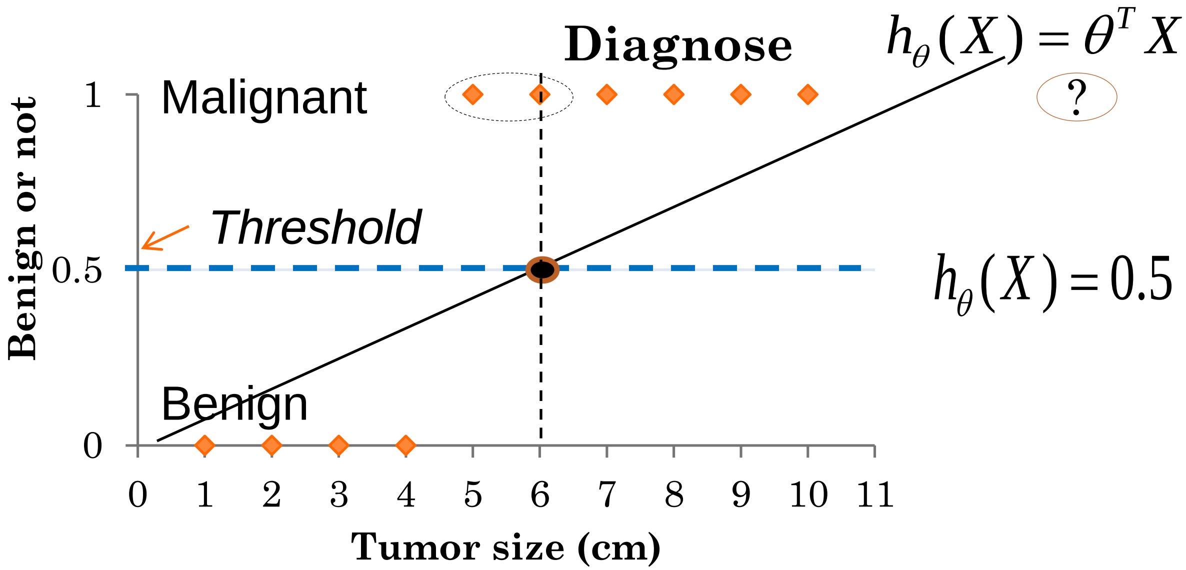
- Classification: is a supervised learning method



Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

1. OVERVIEW

- Estimate the tumor is "benign" or "malignant"based on its size



- If $h_\theta(X) \geq 0.5$ then: "Y=1" and vice versa "Y=0"
- In fact, $h_\theta(X) > 1$ or $h_\theta(X) < 0$
- *Logistic regression:* $0 \leq h_\theta(X) \leq 1$ => *Classification*

1. OVERVIEW

- Common classification algorithms
  - Logistic regression
  - Decision tree
  - Bayesian method
  - Artificial neural network (ANN)
  - K-nearest neighbor
  - Case-based reasoning
  - Genetic algorithms
  - Rough sets analysis
  - Fuzzy sets analysis …

2. Logistic regression

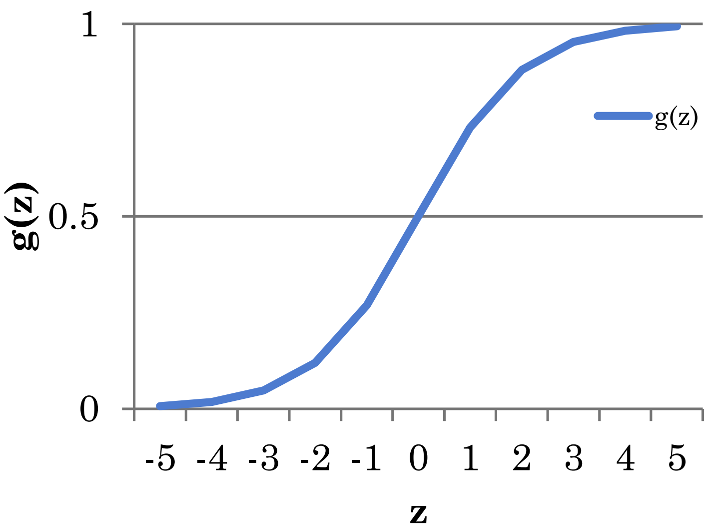
- $h_\theta(X)=\theta^T X$ *(may be >1 or <0)*
- We need $h_\theta(X)$ that is $0 \le h_\theta(X) \le 1$
- Re-modeling: $h_\theta(X)=g(\theta^T X)$

$$g(z)=\frac{1}{1+e^{-z}}$$

where, $$h_\theta(X)=\frac{1}{1+e^{-\theta^T X}}$$

- Sigmoid function

  or Logistic function



Related to coefficients θ

2. Logistic regression

- Explain the value of $$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}}$$
  - is the probability to predict "y=1" with input is x
  - Ex.,

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ kt.khoi\_u \end{bmatrix}$$

$h_\theta(x)=0.7$

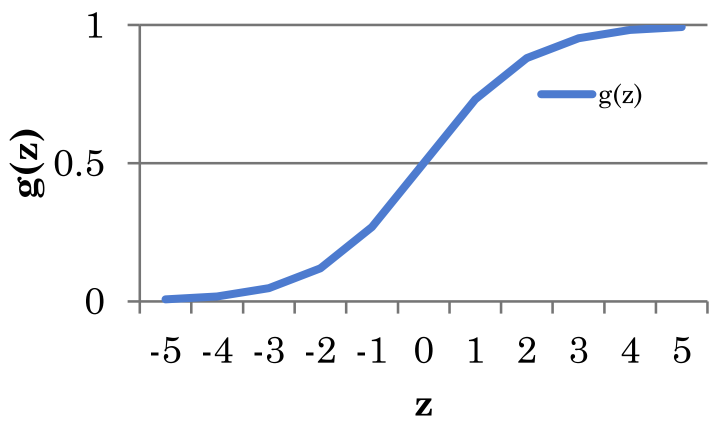
$\Rightarrow$ 70% tumors with given size could be "malignant"

$\Rightarrow$ *$h_\theta(x)=P(y=1|x,\theta)$ (probability for y=1, with a given x and parameterred by θ)*

2. LOGISTIC REGRESSION

- Note $h_\theta(X)=g(\theta^T X)$ where $g(z)=\dfrac{1}{1+e^{-z}}$ or

$$h_\theta(X)=\frac{1}{1+e^{-\theta^T X}}$$



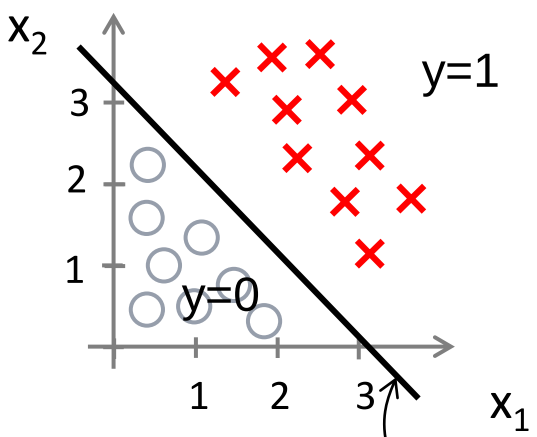
- $g(z) \geq 0.5$, when $z \geq 0$
- $g(z) < 0.5$, when $z < 0$

- Predict y=1 when $h_\theta(X) \geq 0.5$ or $\theta^T X \geq 0$
- Predict y=0 when $h_\theta(X) < 0.5$ or $\theta^T X < 0$

2. Logistic regression

- Decision boundary
  - $h_\theta(X) = g(\theta^T X) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
  - *Select*

$$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$



Decision boundary

  - Predict "y=1" if $\theta^T X \geq 0$

or $-3 + x_1 + x_2 \geq 0$

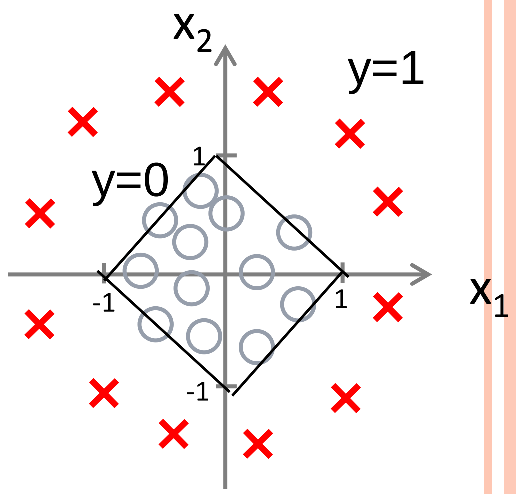
$\Rightarrow x_1 + x_2 \geq 3$

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. LOGISTIC REGRESSION

- Decision boundary
  - $h_\theta(X)=g(\theta^T X)=g(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x^2{}_1+\theta_4 x^2{}_2)$
  - Predict "y=1" if $\theta^T X \geq 0$

  or $-1+x^2{}_1+x^2{}_2 \geq 0$



2. Logistic regression

- Cost function of the **logistic regression** function
  - Training set: $\{(x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}),\ldots, (x^{(N)},y^{(N)})\}$
  - N examples

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}; x_0 = 1; y \in \{0,1\}$$

$$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}}$$

  - How to identify the set of coefficients θ?

2. Logistic regression

- Refer to the linear regression: $$J(\theta)=\frac{1}{2N}\sum_{i=1}^{N}(h_\theta(x^{(i)})-y^{(i)})^2$$

- In non-linear regression

  $J(\theta)=cost(h_\theta(x),y)$

  $cost(h_\theta(x^{(i)}),y^{(i)}) = ½(h_\theta(x^{(i)})-y^{(i)})^2$

To simplify we can write:

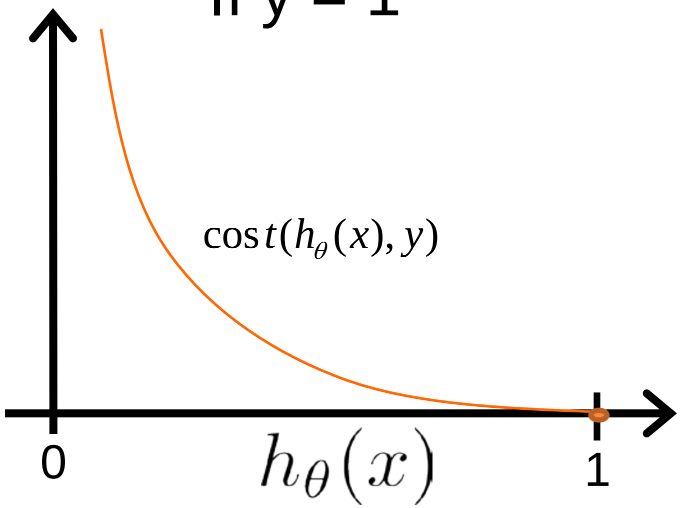
$$cost(h_\theta(x),y) = ½(h_\theta(x)-y)^2$$

2. LOGISTIC REGRESSION

- Cost function of the logistic regression function

$$\cos t(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) : y = 1 \\ -\log(1 - h_\theta(x)) : y = 0 \end{cases}$$

If y = 1



- Cost = 0 if y=1, $h_\theta(x)$=1
- When $h_\theta(x)$->0 then cost -> ∞

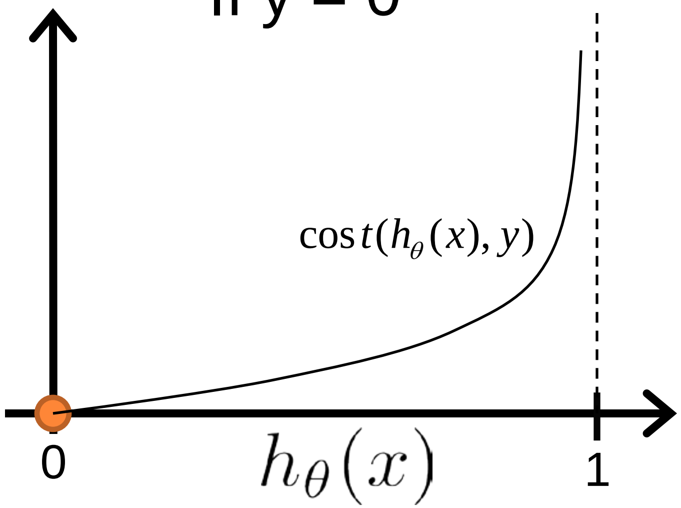
$\Rightarrow$ when $h_\theta(x)$=0 i.e., we predict: P(y=1|x, θ)=0, meanwhile y=1, hence the cost of the algorithm in this case must be large

2. Logistic regression

- Cost function of the logistic regression function

$$\mathrm{cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) : y = 1 \\ -\log(1 - h_\theta(x)) : y = 0 \end{cases}$$

If y = 0



- Cost = 0 if y=0, $h_\theta(x)$=0
- When $h_\theta(x)$->1 then cost -> ∞

⇒ When $h_\theta(x)$=1, i.e., we predict : P(y=1|x, θ)=1, meanwhile y=0, hence the cost of the algorithm in this case is large

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. Logistic regression

- Simplify the cost function and gradient descent algorithm

$$\cos t(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) : y = 1 \\ -\log(1 - h_\theta(x)) : y = 0 \end{cases}$$

- Since y=0|1, the cost function can be simplified as:

cost(h$_\theta$(x),y)=-ylog(h$_\theta$(x))-(1-y)log(1-h$_\theta$(x))

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \cos t(h_\theta(x^{(i)}) - y^{(i)}) = -\frac{1}{N}\sum_{i=1}^{N} y^{(i)} \log h_\theta(x^{(i)}) + (1-y)\log(1 - h_\theta(x^{(i)}))$$

- Finding $\min_\theta J(\theta)$, we can figure out θ (gradient descent)
- To predict y based on a new given x:

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

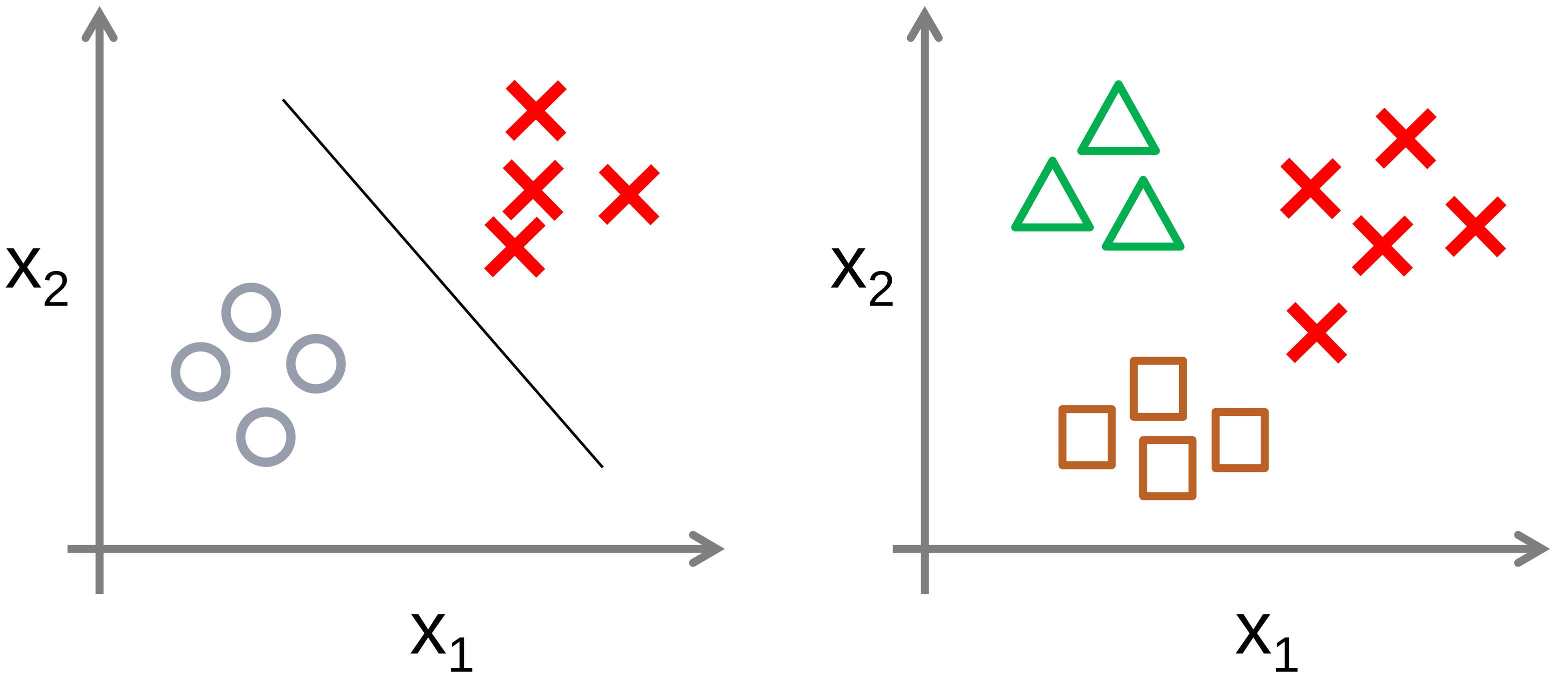
2. LOGISTIC REGRESSION

- Using logistic regression to classify multi-classes data set:
  - Email folder: *"business", "friend", "family", "hobby"* (y={1,2,3,4})
  - Diagnose: "flu", "fever due to virus", "rubella" (y={1,2,3})
  - Weather forecast: "sunny", "clouding", "rainy" (y={1,2,3})

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. Logistic regression

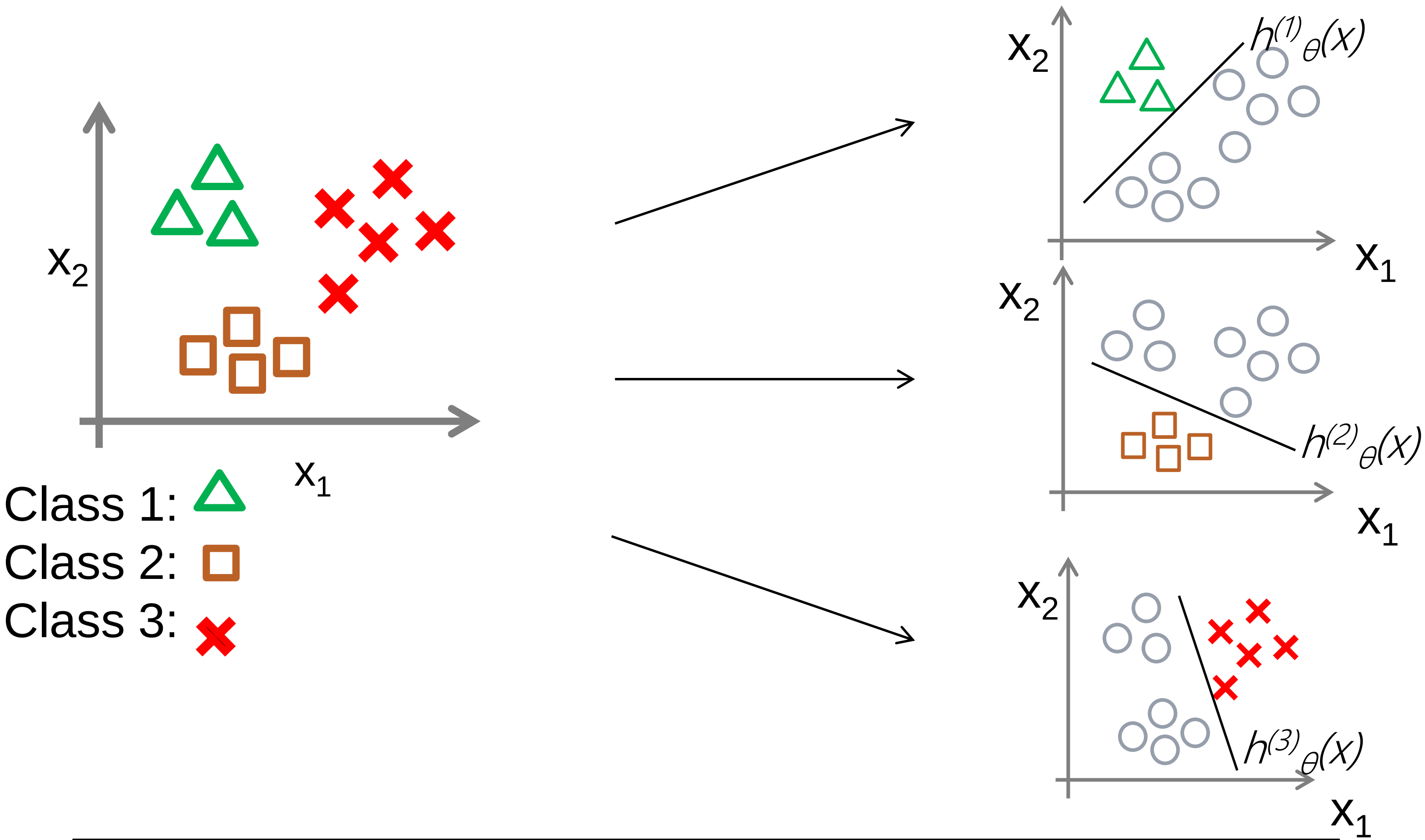
- multi-classes dataset



Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

2. Logistic regression

- multi-class dataset: One and the rest



$h^{(i)}{}_{\theta}(x)=P(y=1|x;\theta)$ *với i* $(i=1,2,..k)$, *k is the number of classes*

2. Logistic regression

- Train the classifier using logistic regression $h^{(i)}_{\theta}(x)$ for each class i
- Given a new object x, we predict y by selecting class i with the highest $h^{(i)}_{\theta}(x)$:

$$\boxed{h^{(i)}_{\theta}(x) = P(y=1|x;\ \theta) \ \ \text{với } i\ (i=1,2,..k),\ k \text{ is the number of classes}}$$

3. DECISION TREE

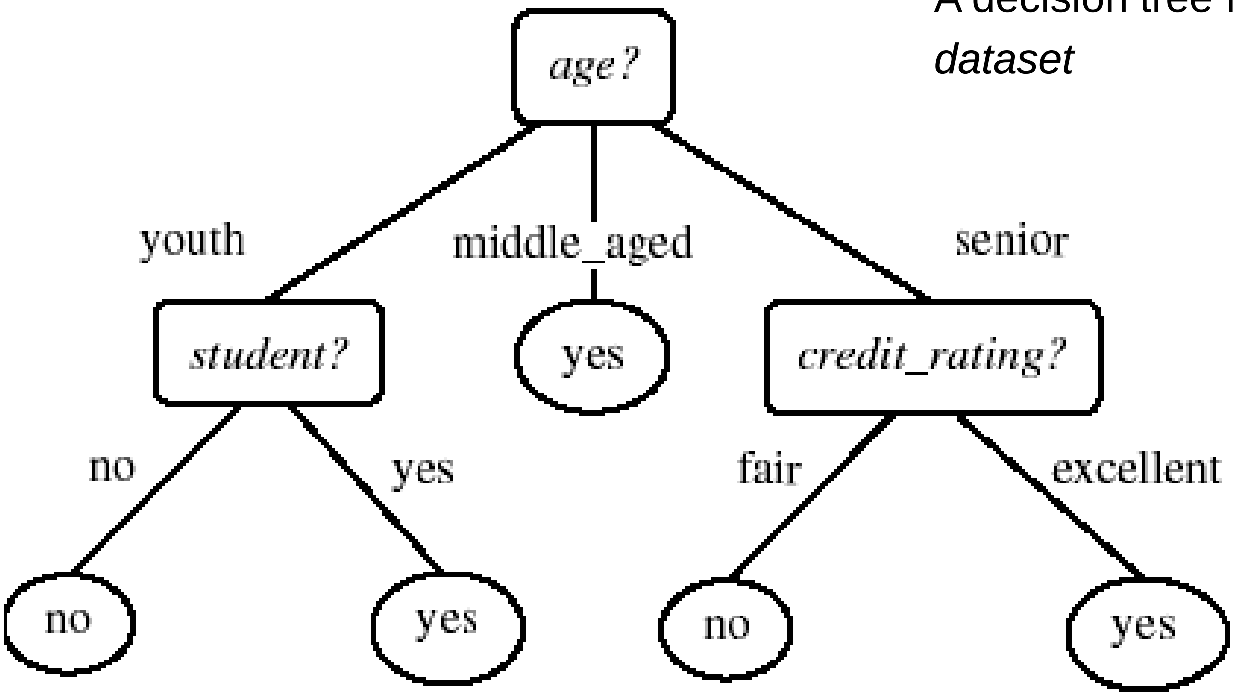
RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

AllElectronics database used for training

3. DECISION TREE

- Internal node: is a test on a specific feature
- Leaf node: class label
- Path from an internal node: the result of a test on the corresponding feature

A decision tree from *AllElectronics dataset*



3. DECISION TREE

- Algorithm for building decision tree
  - ID3, C4.5, CART (Classification and Regression Trees – binary decision trees)

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D.

Input:

- Data partition, D, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

3. DECISION TREE

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C then
(3)      return N as a leaf node labeled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let D_j be the set of data tuples in D satisfying outcome j; // a partition
(12)     if D_j is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(D_j, attribute_list) to node N;
     endfor
(15) return N;
```

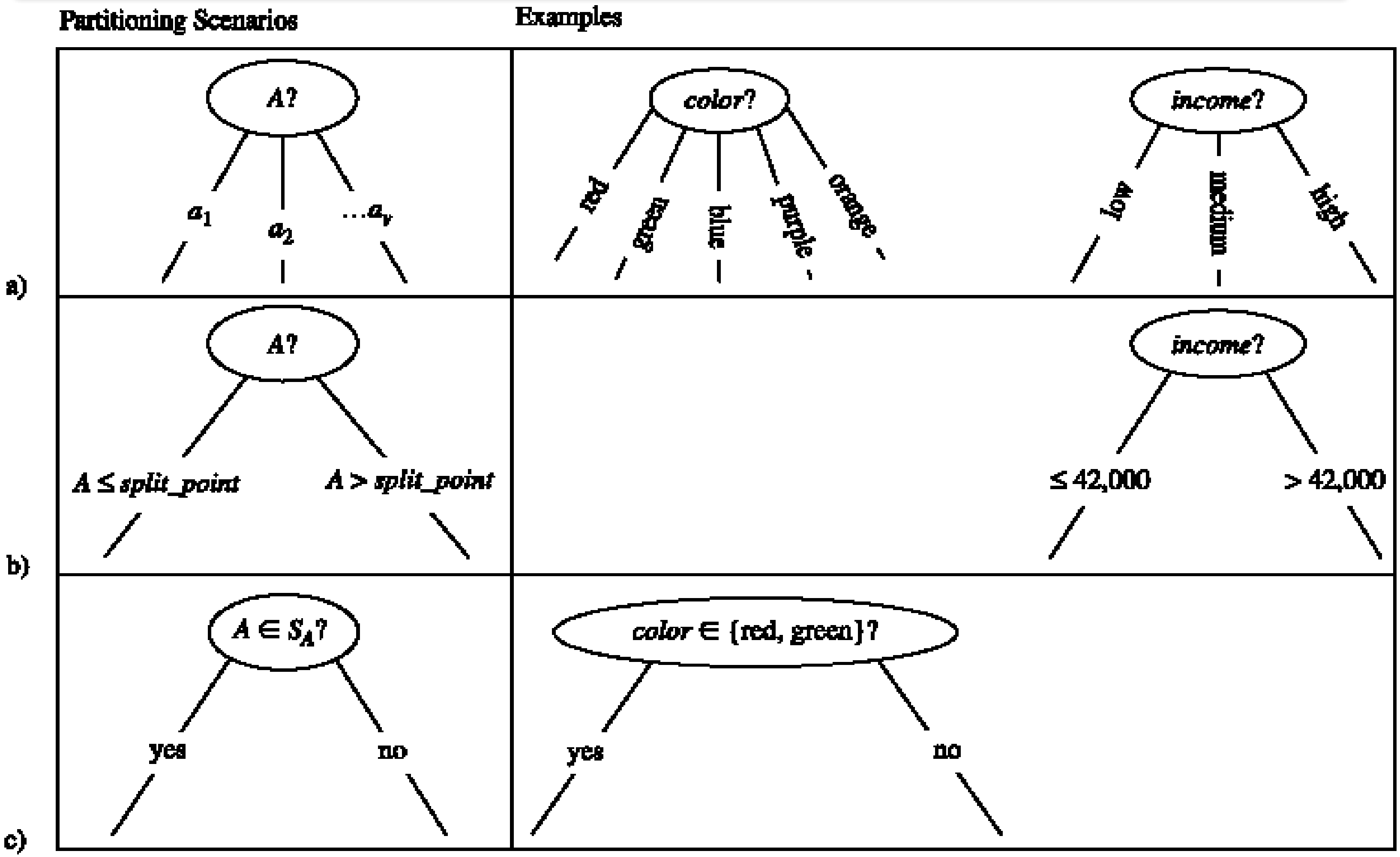
3. DECISION TREE

- Characteristics of the algorithm
  - A greedy algorithm (without backward), divide and conquer, recursive, top-down analysis
  - Complexity: $O(n^*|D|^*log|D|)$
    - Each feature corresponds to a level of the tree
    - At each level, |D| objects/patterns in the training data are examined
    - In-memory → ???

3. DECISION TREE

- **Attribute_selection_method**
  - Heuristic: to chose the partition criteria at a node, i.e. to divide *D* into smaller partitions with appropriate classes
    - Rank each attribute
    - The selected attribute is the one whose score is the highest
    - Measure for attribute splitting : **information gain**, **gain ratio**, **gini index**

3. DECISION TREE



A là thuộc tính phân tách (splitting attribute).

3. DECISION TREE

- **Information Gain**
  - Based on information theory introduced by Claude Shannon about the value/content of information
  - The attribute whose information gain is the highest is selected as splitting attribute for the current node N
    - N: current node where D is partitioned
    - Splitting attribute: assure that the impurity/randomness is minimized in the resulted partitions
    - This approach helps minimizing the number of tests in order to classify a given object

3. Decision tree

- **Information Gain**
  - **Info(D):** The necessary information used to classify an object in D (= Entropy(D))
    - p_i: the probability for an object in D that belongs to a specific class C_i (where i = 1..m)
    - $C_{i,D}$: a set of objects belong to C_i in D

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$p_i = |C_{i,D}| / |D|$$

3. DECISION TREE

- **Information Gain**
  - **$Info_A(D)$:** The necessary information used to classify an object in D based on attribute A
    - Attribute A is used to divide D into v partitions

      $\{D_1, D_2, \ldots, D_j, \ldots, D_v\}$
    - Each D_j has $|D_j|$ object in D
    - This information describes the level of chaos (impurity) in partitions
    - **It is better to have small $Info_A(D)$**

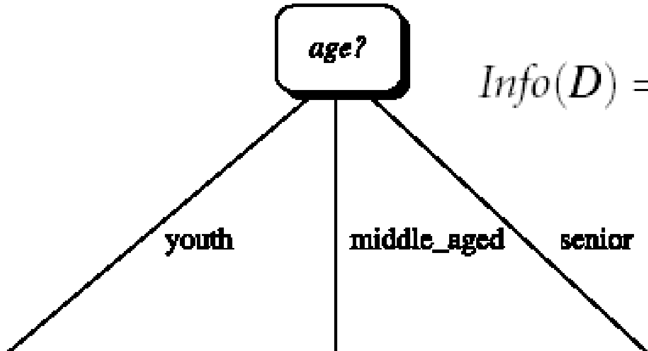
$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j)$$

3. Decision tree

- **Information Gain**
  - Information gain is the difference between **Info(D)** (before partitioning) and **$\mathbf{Info_A(D)}$** (after partitioning by using attribute A)

$$Gain(A) = Info(D) - Info_A(D)$$

3. DECISION TREE



youth:

income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

middle_aged:

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

senior:

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$$

Gain(age)=0.246 bits

Gain(income)?

Gain(student)?

Gain(credit_rating)?

→ Splitting attribute?

$$\begin{aligned} Info_{age}(D) = & \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\ & + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) \\ & + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\ = & 0.694 \text{ bits.} \end{aligned}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

3. Decision tree

- **GainRatio(A)**
  - Used in C4.5 algorithm
  - Problem in Information Gain: It may create many small partitions (even with only 1 object)

  => Normalize Information with split information: **SplitInfo$_A$(D)**
  - Splitting attribute A is the one whose **GainRatio(A)** is the maximum

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

3. DECISION TREE

$$\text{SplitInfo}_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right)$$

$$= 0.926.$$

Gain(income) = 0.029

GainRatio(income) = 0.029/0.926 = 0.031

GainRatio(age)?

GainRatio(student)?

GainRatio(credit_rating)?

→ Splitting attribute?

3. Decision tree

- **Gini Index**
  - Used with CART
  - Is a binary split for each attribute A
    - $A \in S_A$?
    - S_A is a subset of 1 or v - 1 values of attribute A
  - Gini index of an attribute is the minimum value in accordance with a subset S_A from $2^v - 2$ subsets
  - Splitting attribute is the one whose <u>gini index is minimum</u> (to maximize the reduction in duplication between partitions)

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

3. Decision tree

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

$$\begin{aligned} Gini_{income \in \{low, medium\}}(D) &= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) \\ &= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right) \\ &= 0.450 \\ &= Gini_{income \in \{high\}}(D). \end{aligned}$$

$Gini_{income \in \{low,high\}} = Gini_{income \in \{medium\}} = 0.315$

$Gini_{income \in \{medium,high\}} = Gini_{income \in \{low\}} = 0.300$

→ $Gini_{income \in \{medium,high\}/\{low\}} = 0.300$

$Gini_{age \in \{youth,senior\}/\{middle\_aged\}} = 0.375$

$Gini_{student} = 0.367$

$Gini_{credit\_rating} = 0.429$

→ Splitting attribute?

3. DECISION TREE

- Home work: Build a decision tree from AllElectronics dataset using:
  - Information Gain
  - Gain Ratio
  - Gini Index

  → Are they similar ?

  → Practice the classification with the resulted Decision tree and discuss about their effectiveness

4. CLASSIFICATION WITH BAYESIAN

- Based Bayes's theorem
  - Assumption: class conditional independence
  - Is a classification based on probability

Reverend Thomas Bayes
(1702-1761)

4. Classification with Bayesian

- Bayes's theorem
  - X: a tuple/object (evidence)
  - H: hypothesis
    - X belongs to class C.

Given an RID, is it belongs to class "yes" (buys_computer = yes)

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

X

X is identified by values of its attributes

4. Classification with Bayesian

- Bayes's theorem
  - P(H | X): posterior probability
    - Ex: P(buys_computer=yes | age=young, income=high): **probability** of buying computer from a customer whose age is "young" and income is "high"
  - P(X | H): posterior probability, the conditional probability of X based on H (likelihood)
    - Ex: P(age=young,income=high | buys_computer=yes): probability of a customer who bough computer has age = "young" and income = "high"
      - P(age=young, income=high | buys_computer=yes) = 0
      - P(age=young, income=high | buys_computer=no) = 2/5 = 0.4

4. Classification with Bayesian

- Bayes's theorem
  - P(H): class prior probability
    - Ex: P(buys_computer=yes): **probability** of customer who buys computer in general
    - P(buys_computer=yes) = 9/14 = 0.643
    - P(buys_computer=no) = 5/14 = 0.357
  - P(X): predictor prior probability
    - Ex: P(age=young, income=high): **probability** of customer whose age = "young" and income = "high"
    - P(age=young, income=high) = 2/14 = 0.143

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

4. Classification with Bayesian

- Bayes's theorem
  - P(H), P(X|H), P(X): Calculated from given dataset
  - P(H|X): Inferred from Bayes's theorem

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

P(buys_computer=yes|age=young, income=high) = P(age=young, income=high|buys_computer=yes)P(buys_computer=yes)/P(age=young, income=high) = 0

P(buys_computer=no|age=young, income=high) = P(age=young, income=high|buys_computer=no)P(buys_computer=no)/P(age=young, income=high) = 0.4*0.357/0.143 = 0.9986

4. Classification with Bayesian

- Given a training dataset D with class labels for C_i, i=1..m, the classification process of an object/tuple $X = (x_1, x_2, \ldots, x_n)$ with Bayesian method:

  X is classified into C_i iff

  $$P(C_i|X) > P(C_j|X), \text{ where } j=1..m,\ j \neq i$$

  $$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

  → Maximize $P(C_i|X)$ (i.e. select C_i if $P(C_i|X)$ is the maximum value)

  → Maximize $P(X|C_i)P(C_i)$, since $P(X)$ is a similar and, we have $P(C_i) = |C_{i,D}|/|D|$ …

4. Classification with Bayesian

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) * P(x_2 \mid C_i) * .. * P(x_n \mid C_i)$$

- $P(X|C_i)$ is calculated with *class conditional independence* assumption
- x_k, k = 1..n: value of attribute A_k in object X
- $P(x_k|C_i)$ is calculated as follows:

4. Classification with Bayesian

- A_k is a categorical attribute
  - $P(x_k|C_i) = |\{X'|x'_k = x_k \wedge X' \in C_i\}|/|C_{i,D}|$

- A_k is a continuous attributes
  - We assume $P(x_k|C_i)$ follows a particular distribution (Ex: Gauss distribution with μ and σ)

$$g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \Rightarrow P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

4. Classification with Bayesian

- Problem: if $P(x_k | C_i) = 0$ then $P(X | C_i) = 0$!!!
  - Original approach
    - $P(x_k | C_i) = |\{X' | x'_k = x_k \wedge X' \in C_i\}| / |C_{i,D}|$
  - Laplace (Pierre Laplace, 1749-1827)
    - $P(x_k | C_i) = (|\{X' | x'_k = x_k \wedge X' \in C_i\}| \mathbf{+1})/(|C_{i,D}| \mathbf{+ m})$

      where, m is the number of different values in the domain of attribute A_k
  - z-estimate
    - $P(x_k | C_i) = (|\{X' | x'_k = x_k \wedge X' \in C_i\}| \mathbf{+ z*P(x_k)})/(|C_{i,D}| \mathbf{+ z})$

4. Classification with Bayesian

$$X = (age = youth, income = medium, student = yes, credit\_rating = fair)$$

C_1 = {X'|X'.buys_computer = yes}

C_2 = {X''|X''.buys_computer = no}

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

$$P(buys\_computer = yes) = 9/14 = 0.643$$
$$P(buys\_computer = no) = 5/14 = 0.357$$

$$\begin{aligned} P(\boldsymbol{X} \mid buys\_computer = yes) = {} & P(age = youth \mid buys\_computer = yes) \times \\ & P(income = medium \mid buys\_computer = yes) \times \\ & P(student = yes \mid buys\_computer = yes) \times \\ & P(credit\_rating = fair \mid buys\_computer = yes) \\ = {} & 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044. \end{aligned}$$

$$P(\boldsymbol{X} \mid buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$

$$P(\boldsymbol{X} \mid buys\_computer = yes)P(buys\_computer = yes) = 0.044 \times 0.643 = 0.028$$
$$P(\boldsymbol{X} \mid buys\_computer = no)P(buys\_computer = no) = 0.019 \times 0.357 = 0.007$$

→ $X \in C_1$

4. Classification with Bayesian – Categorical Data

- Weather dataset:
  (Outlook, Temp, Humidity, Windy) => Play (Yes/No)

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

4. Classification with Bayesian – Categorical Data

Outlook	Yes	No	Temperature	Yes	No	Humidity	Yes	No	Windy	Yes	No	Play Yes	Play No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

- Decision (play=yes/no)

  Calculate:

  P(Yes|E)

  P(No|E)

  where, E the input data (need to be classified)

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

4. CLASSIFICATION WITH BAYESIAN – CATEGORICAL DATA

- Quyết định (play=yes/no)

Outlook	Temp	Humidity	Windy	Play
Sunny	Cool	High	True	?

Evidence E

$$P(Yes \mid E) = P(Outlook = Sunny \mid Yes)$$
$$\times P(Temperature = Cool \mid Yes)$$
$$\times P(Humidity = High \mid Yes)$$
$$\times P(Windy = True \mid Yes)$$
$$\times \frac{P(Yes)}{P(E)}$$
$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{P(E)}$$

Probability of class Yes

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

4. Classification with Bayesian – Categorical data

- Quyết định (play=yes/no)

Outlook	Temp	Humidity	Windy	Play
Sunny	Cool	High	True	?

Evidence E

$$P(No \mid E) = P(Outlook = Sunny \mid No)$$
$$\text{x}\, P(Temperature = Cool \mid No)$$
$$\text{x}\, P(Humidity = High \mid No)$$
$$\text{x}\, P(Windy = True \mid No)$$
$$\text{x}\, \frac{P(No)}{P(E)}$$

Probability of class No

$$= \frac{\frac{3}{5} \text{x} \frac{1}{5} \text{x} \frac{4}{5} \text{x} \frac{3}{5} \text{x} \frac{5}{14}}{P(E)}$$

Outlook			Temperature			Humidity			Windy			Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

4. Classification with Bayesian – Categorical Data

- Decision (play=yes/no)

Outlook	Temp	Humidity	Windy	Play
Sunny	Cool	High	True	?

Likelihood "yes" = 2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.0053

Likelihood "no" = 3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.0206

Normalized:

P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

Since P("no") > P("yes") => Play ="No"

Outlook	Temp	Humidity	Windy	Play
Sunny	Cool	High	True	No

← **Result**

4. Classification with Bayesian – Continuous data

- Assumption: Attributes has Gauss distribution
- Probability distribution function is calculated as:
  - mean μ

$$\mu = \frac{1}{N}\sum_{j=1}^{N} x_j$$

  - Standard deviation σ

$$\sigma^2 = \frac{1}{N-1}\sum_{j=1}^{N}\left(x_j - \mu\right)^2$$

  - Distribution function f(x)

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

4. Classification with Bayesian – Continuous data

Outlook			Temperature			Humidity			Windy			Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3		83	85		86	85	False	6	2	9	5
Overcast	4	0		70	80		96	90	True	3	3		
Rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
Sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	std. dev.	6.2	7.9	std. dev.	10.2	9.7	True	3/9	3/5		
Rainy	3/9	2/5											

- *Ex:*

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi}6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

4. Classification with Bayesian – Continuous data

- Classification:

Outlook	Temp	Humidity	Windy	Play
Sunny	66	90	True	?

Likelihood ”yes” = 2/9 x 0.0340 x 0.0221 x 3/9 x 9/14 = 0.000036

Likelihood ”no” = 3/5 x 0.0291 x 0.0380 x 3/5 x 5/14 = 0.000136

P(“yes”) = 0.000036 / (0.000036 + 0. 000136) = 20.9

P(“no”) = 0.000136 / (0.000036 + 0. 000136) = 79.1

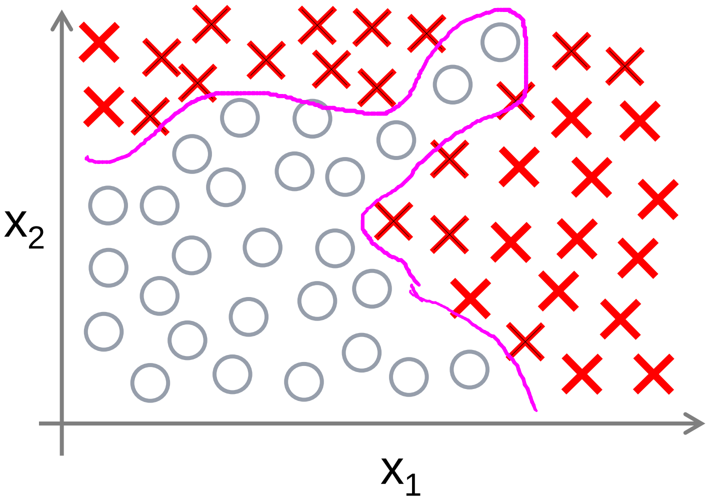
4. CLASSIFICATION WITH BAYESIAN

- **Advantage:**
  - Easy to implement, fast learning, easy to understand the results
  - Effective in many cases

- **Disadvantage:**
  - Assumption *class conditional independence* may not be satisfied -> carefully check this characteristic

5. CLASSIFICATION WITH NEURAL NETWORK

- Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

x1: Size
x2: No. of rooms
x3: floors
x4: age
….
X100:….

n=100

$x^2_1, x_1x_2, x_1x_3, \ldots x_1x_{100},$
$x^2_2, x_2x_3 \ldots$

$\Rightarrow$**5000 features (~O(n^2) parameters)**

$x^2_1, x^2_2, x^2_3, \ldots x^2_{10},$
$x_1x_2x_3, x^2_1x_2, \ldots$

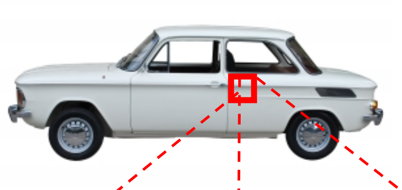
=>O(n^3) parameters

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

5. Classification with Neural network

What is this?

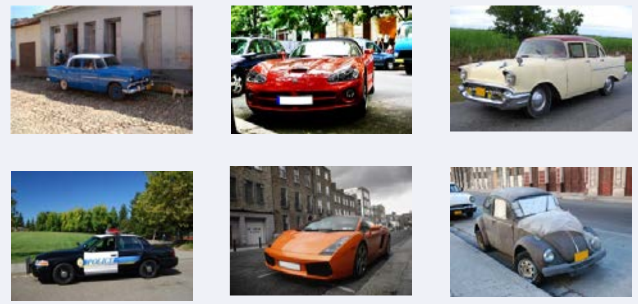
Human: a car



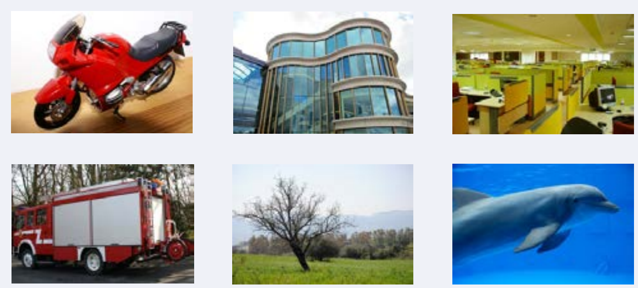
Camera can read pixels:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Training:



Cars



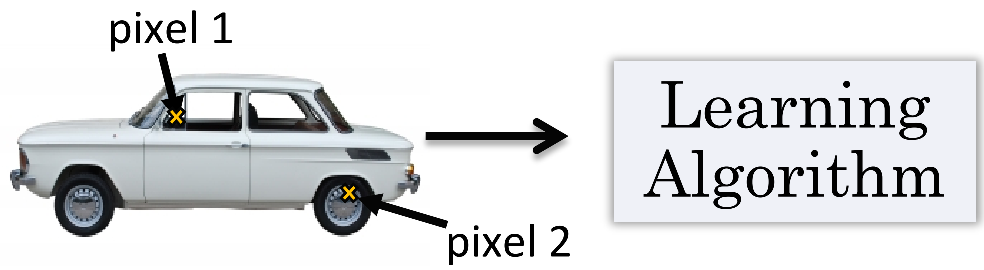
Not a car

Testing: What is this?

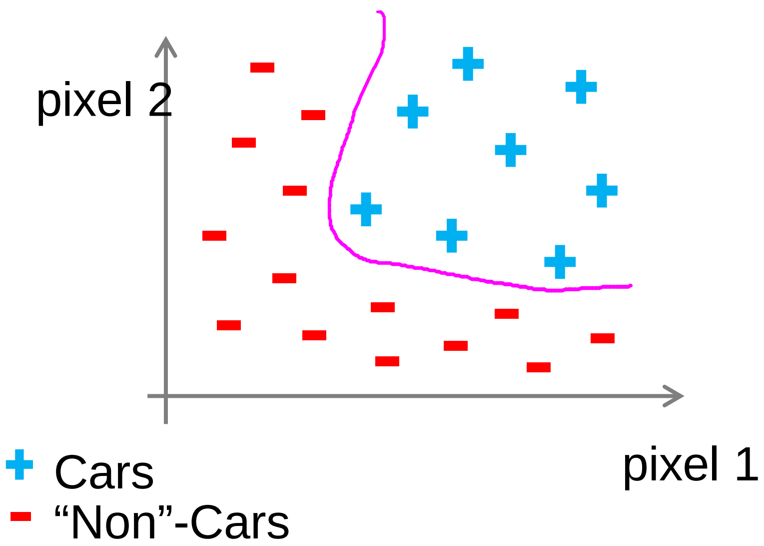
Source: Andrew Ng

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

5. Classification with Neural network



Hình 50 x 50 pixels → 2500 pixels (n=2500) (7500 if RGB)



0-255

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

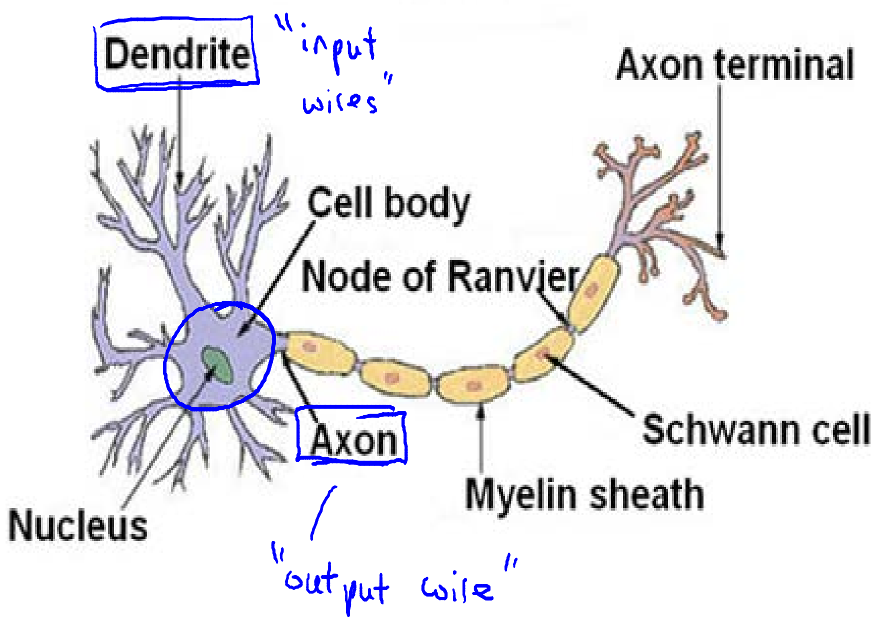
Kết hợp hàm bậc 2 (x_i*x_j): ≈3M features

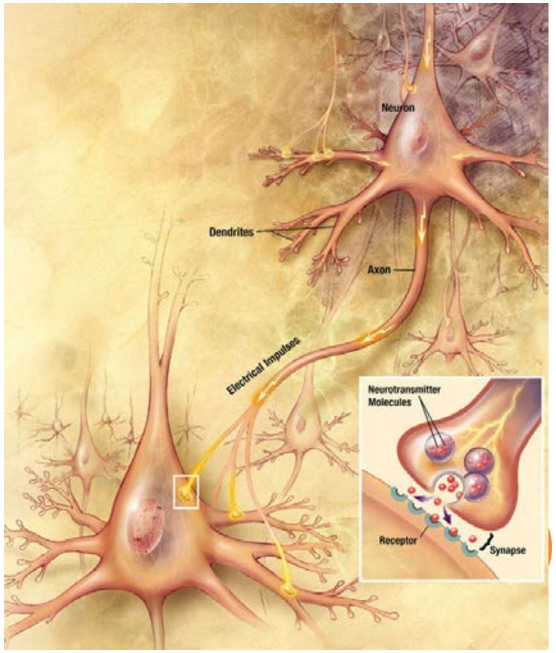
Source: Andrew Ng

5. Classification with Neural network

- Simulate the work of human brain
- Popular since 80s - 90s
- Now, it is applied in various applications

Neuron in the brain

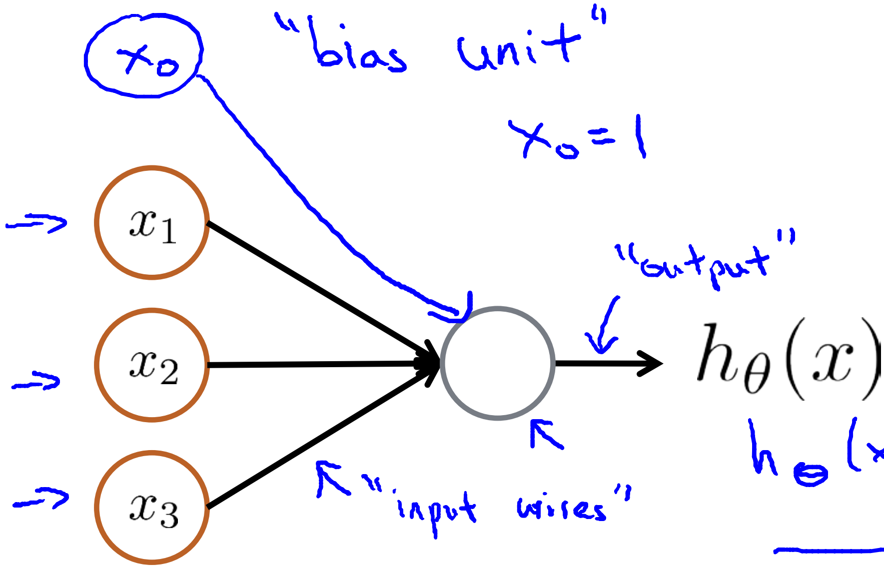




Source: Andrew Ng

5. Classification with Neural network

Modeling neuron: Logistic unit



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

"weights" (parameters)

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

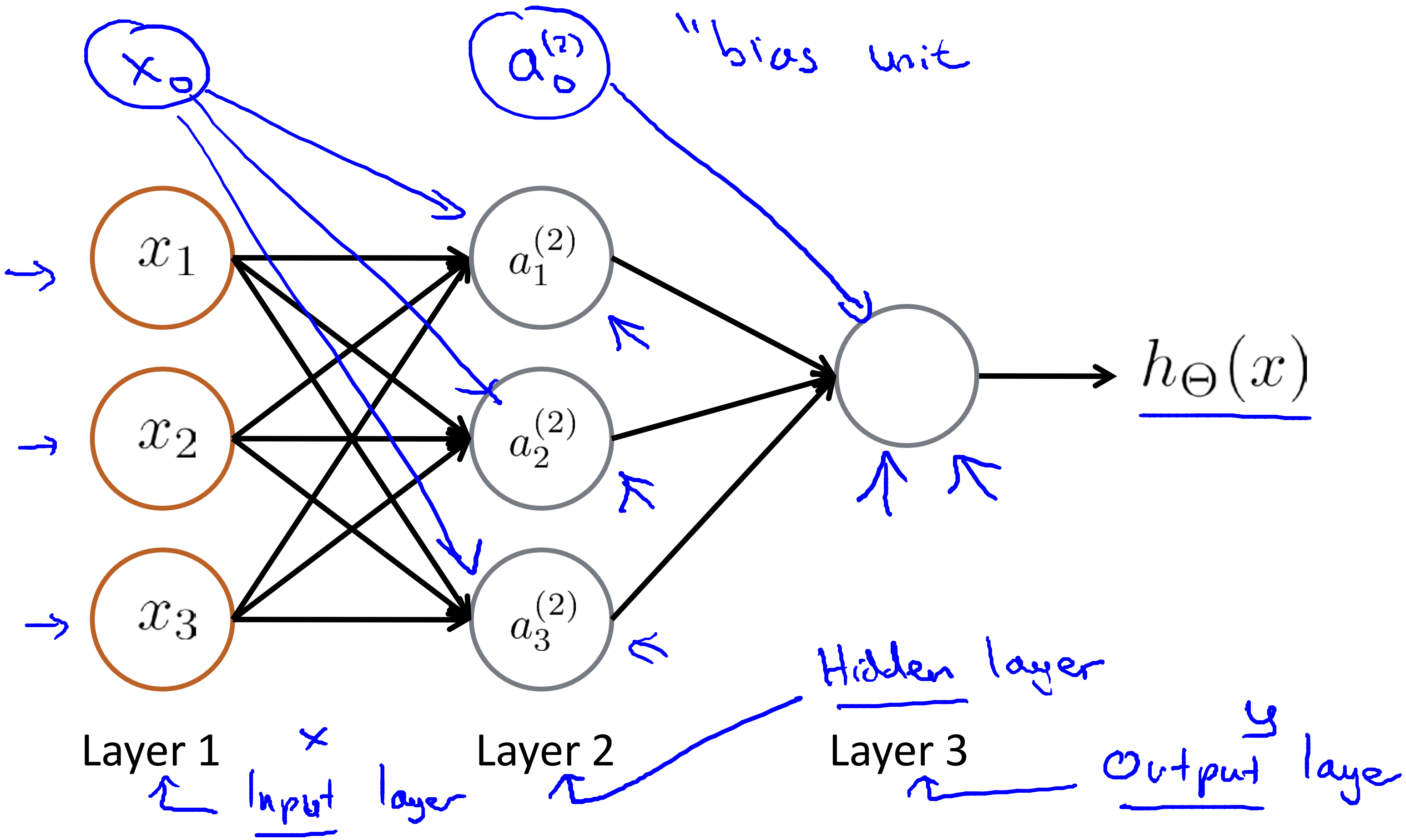
Sigmoid (logistic) activation function.

$$g(z) = \frac{1}{1+e^{-z}}$$

Source: Andrew Ng

5. Classification with Neural network

Neural Network

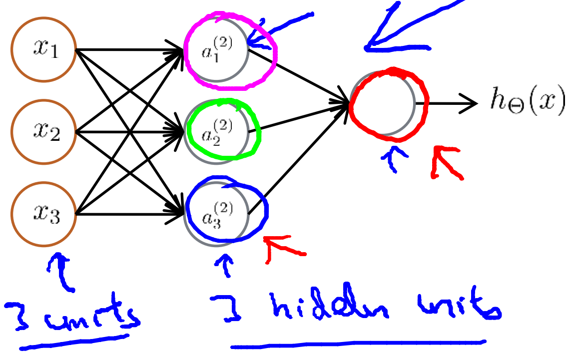


Source: Andrew Ng

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

5. CLASSIFICATION WITH NEURAL NETWORK

Neural Network



→ $a_i^{(j)} =$ "activation" of unit *i* in layer *j*

→ $\Theta^{(j)} =$ matrix of weights controlling function mapping from layer *j* to layer *j+1*

$\Theta^{(1)} \in \mathbb{R}^{3\times4}$ $h_{\Theta}(x)$

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$

$\Theta^{(2)}$

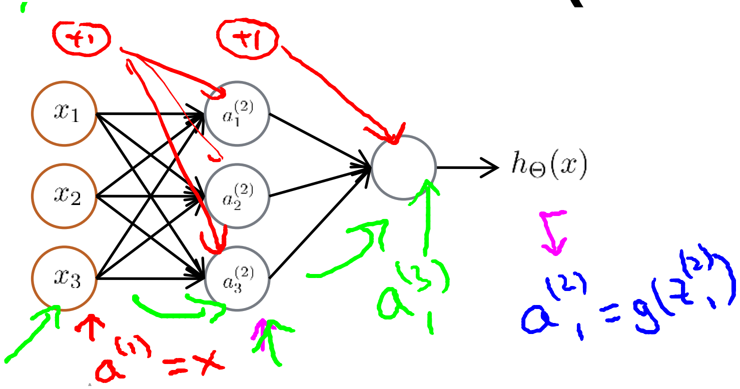
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$$

If the network has s_j nodes (units) at level *j* and s_{j+1} nodes at level *j+1*, then the size/dimension of $\Theta^{(j)}$ is s_{j+1}x(s_j+1)

Source: Andrew Ng

5. CLASSIFICATION WITH NEURAL NETWORK

ANN: Feed forward (Forward propagation)



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$a_1^{(2)} = g(\underline{\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3}) \quad z_1^{(2)}$$
$$a_2^{(2)} = g(\underline{\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3}) \quad z_2^{(2)}$$
$$a_3^{(2)} = g(\underline{\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3}) \quad z_3^{(2)}$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

$a^{(3)} = g(z^{(3)})$

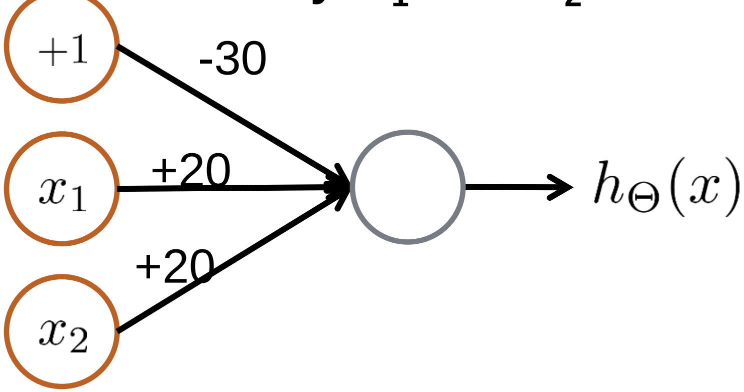
$z^{(2)} = \Theta^{(1)} a^{(1)}$
$a^{(2)} = g(z^{(2)})$
Thêm $a_0^{(2)} = 1$ vào $a^{(2)}$
$z^{(3)} = \Theta^{(2)} a^{(2)}$
$h_\theta(x) = a^{(3)} = g(z^{(3)})$

Source: Andrew Ng

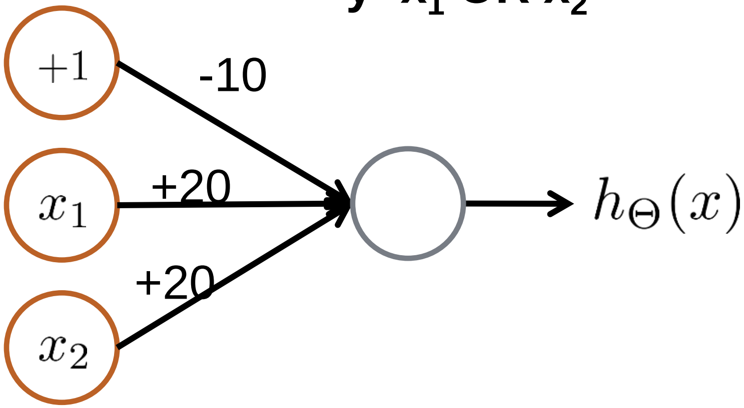
5. Classification with Neural network

- Ex., presenting ANNs for basic logic operators

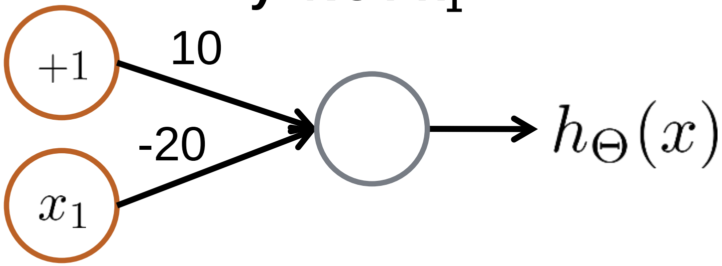
$y=x_1$ AND x_2



$y=x_1$ OR x_2



$y=$NOT x_1



Validate using logic tables for the above ANNs!

Source: Andrew Ng

5. Classification with Neural network

- Ex., use ANN to present a more complicated logic operation: x_1 NOR x_2
- x_1 NOR x_2= NOT x_1 XOR x_2:

x_1	x_2	x_1 XOR x_2	x_1 NOR x_2
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

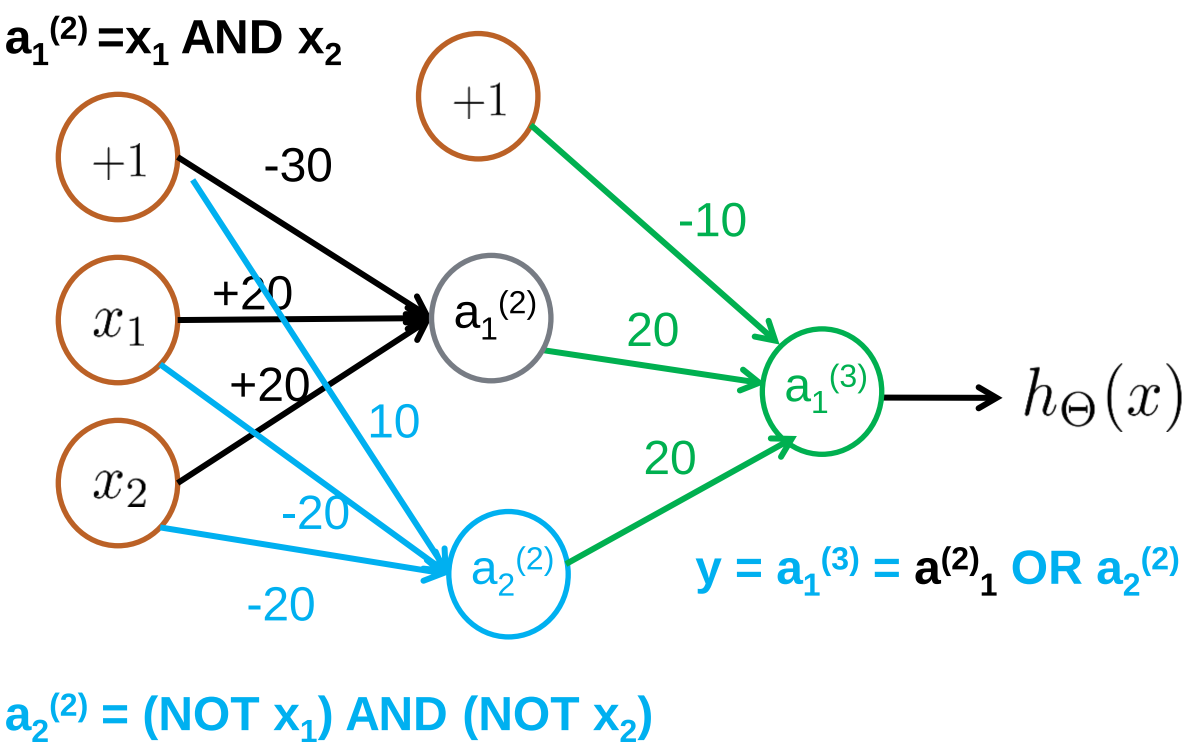
=>x_1 NOR x_2=(x_1 AND x_2)OR(NOT x_1 AND NOT x_2)

=> Integrate basic ANNs in the previous slide for presenting this expression!

Source: Andrew Ng

5. Classification with Neural network

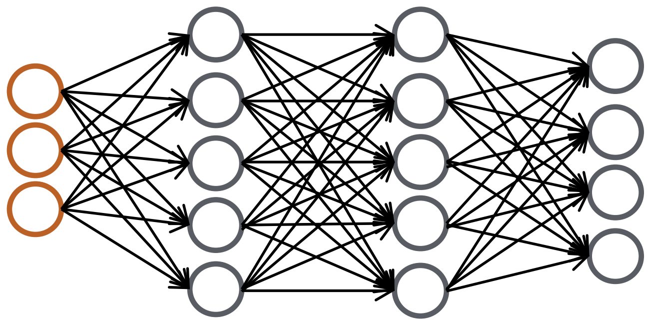
- Present: x_1 NOR x_2 = (x_1 AND x_2) OR (NOT x_1 AND NOT x_2)



Source: Andrew Ng

5. Classification with Neural network

- Cost function in ANN



$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$$

$L =$ Total number of layers in the network

$s_l =$ No. of nodes (not included the bias node) in level *l*

Binary classification

$$y = 0 \text{ or } 1$$

1 output unit

Multi-class classification (K classes)

$$y \in \mathbb{R}^K$$

E.g. $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

pedestrian car motorcycle truck

K output units

Source: Andrew Ng

5. Classification with Neural network

- Cost function in ANN

Logistic regression:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Neural network:

$$h_\Theta(x) \in \mathbb{R}^K \quad (h_\Theta(x))_i = i^{th} \text{ output}$$

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1-y_k^{(i)})\log(1-(h_\Theta(x^{(i)}))_k)\right]$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

Source: Andrew Ng

5. Classification with Neural network

- Minimizing the Cost in ANN: Backpropagation method

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log h_\theta(x^{(i)})_k + (1 - y_k^{(i)}) \log(1 - h_\theta(x^{(i)})_k)\right]$$
$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_j^{(l)})^2$$

$$\min_{\Theta} J(\Theta)$$

Need to calculate:

- $J(\Theta)$
- $\frac{\partial}{\partial\Theta_{ij}^{(l)}}J(\Theta)$

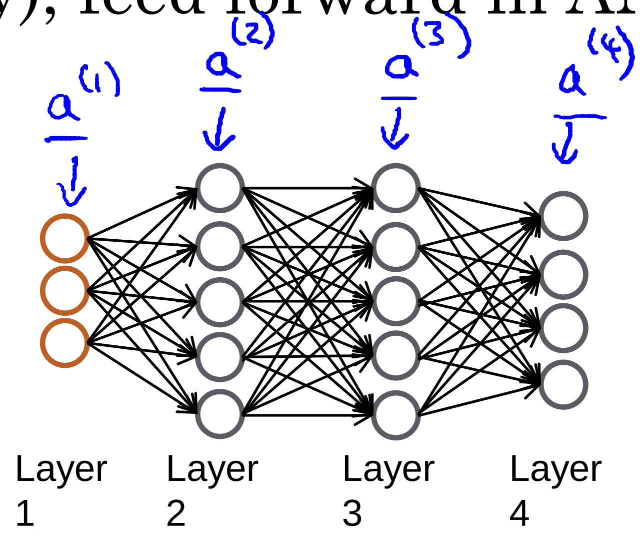
$\Theta_{ij}^{(l)} \in \mathbb{R}$

Source: Andrew Ng

5. Classification with Neural network

- Calculate the change of the derivation of the Cost function when changing parameters (gradient computation)
  - Given a training dataset (x,y), feed forward in ANN

$$a^{(1)} = x$$
$$z^{(2)} = \Theta^{(1)} a^{(1)}$$
$$a^{(2)} = g(z^{(2)}) \quad (\text{add } a_0^{(2)})$$
$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$a^{(3)} = g(z^{(3)}) \quad (\text{add } a_0^{(3)})$$
$$z^{(4)} = \Theta^{(3)} a^{(3)}$$
$$a^{(4)} = h_\Theta(x) = g(z^{(4)})$$



Source: Andrew Ng

Dr. Tran Minh Quang – quangtran@hcmut.edu.vn

5. CLASSIFICATION WITH NEURAL NETWORK

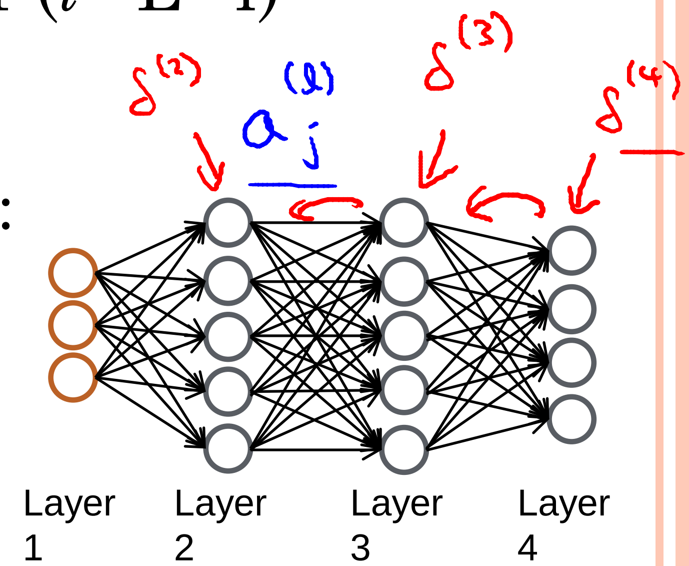
- Let $\delta^{(l)}_j$ be the "error" created by node *j* at *l* layer
  - At each node in the output layer (l =L=4)

$\delta^{(4)}_j = a^{(4)}_j - y_j \qquad (\delta^{(4)} = a^{(4)} - y)$

Calculate "errors" of inner nodes:

$\delta^{(3)} = (\Theta^{(3)})^T \delta^{(4)} . g'(z^{(3)})$

$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} . g'(z^{(2)})$



Note: Do not calculate $\delta^{(1)}$ and

$$\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = a_j^{(l)} \delta_i^{(l+1)}$$

Source: Andrew Ng

5. Classification with Neural network

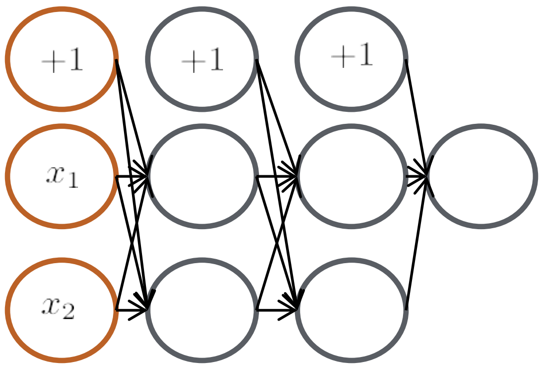
- Backpropagation algorithm
  - Given a training dataset $\{(x^{(1)}, y^{(1)}),\ldots, (x^{(N)}, y^{(N)})\}$
  - Assign $\Delta^{(l)}_{ij} = 0$ (for all i, j, l)
  - For i=1 to N (N = |D|)
    - $a^{(i)} := x^{i}$ conduct feed forward to calculate $a^{(l)}$ (l=1,2,3,..L)
    - Use $y^{(i)}$ to calculate $\delta^{(L)} = a^{(L)} - y$
    - Calculate $\delta^{(L-1)}, \delta^{(L-2)},\ldots,\delta^{(2)}$
    - Calculate $\Delta^{(l)}_{ij} := \Delta^{(l)}_{ij} + a_{(j)}^{(l)}\, \delta_{(i)}^{(l+1)}$
  - Assign $$\begin{cases} D_{ij}^{(l)} := \dfrac{1}{N}\Delta_{ij}^{(l)} + \lambda\theta_{ij}^{(l)},\ j \neq 0 \\ D_{ij}^{(l)} := \dfrac{1}{N}\Delta_{ij}^{(l)},\ j = 0 \end{cases}$$

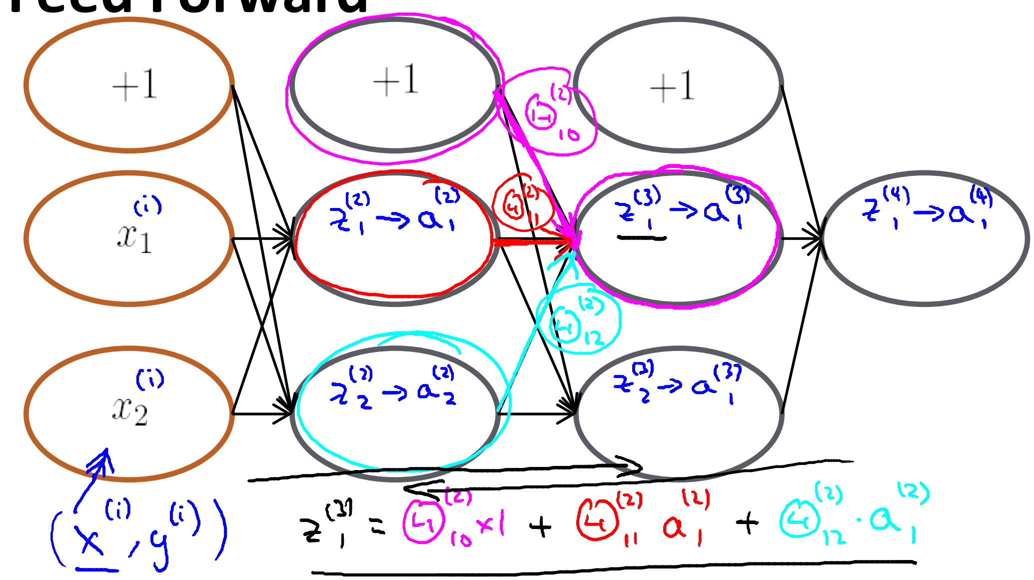
$$\frac{\partial}{\partial\theta_{ij}^{(l)}} J(\theta) = D_{ij}^{(l)}$$

Source: Andrew Ng

5. Classification with Neural network

- Ex., about backpropagation:





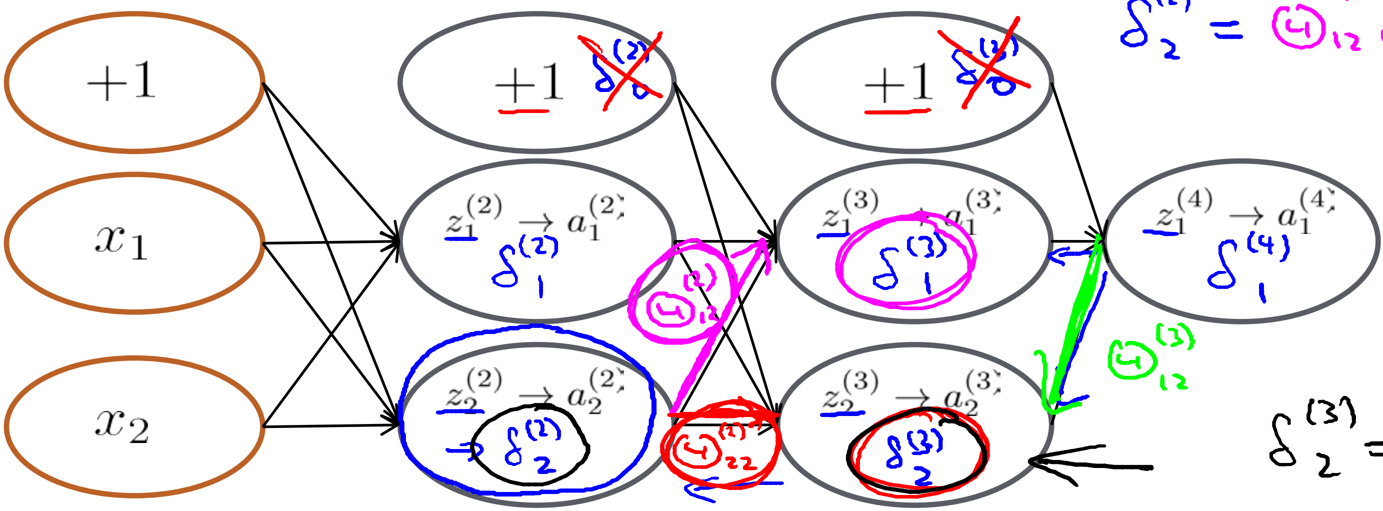
Source: Andrew Ng

5. CLASSIFICATION WITH NEURAL NETWORK

- Ex., about backpropagation:

$$\delta_1^{(4)} = y^{(i)} - a_1^{(4)}$$

$$\delta_2^{(2)} = \Theta_{12}^{(2)} \delta_1^{(3)} + \Theta_{22}^{(2)} \cdot \delta_2^{(3)}$$



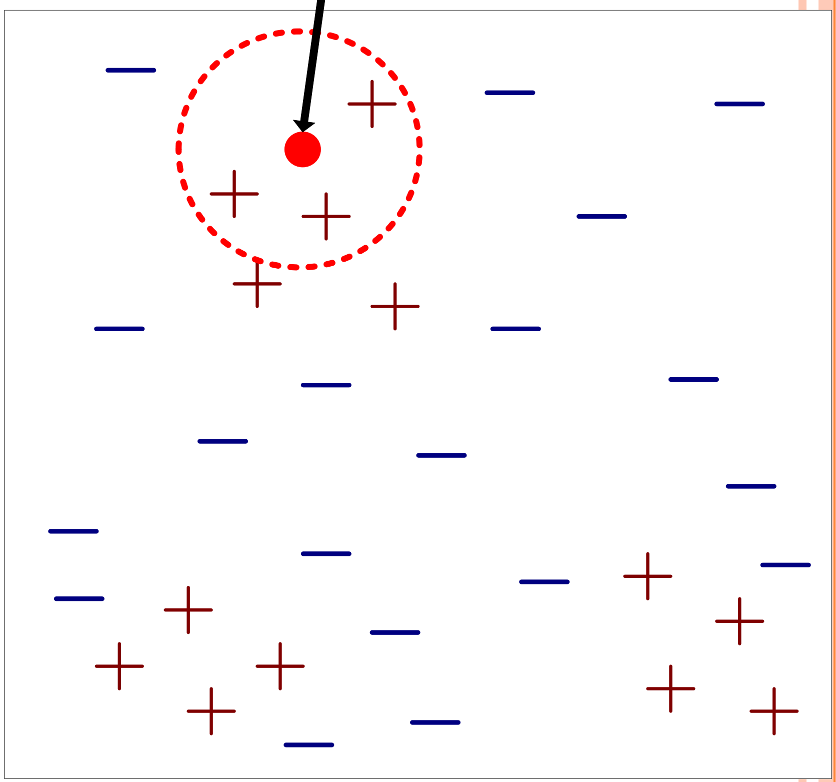
$$\delta_2^{(3)} = \Theta_{12}^{(3)} \cdot \delta_1^{(4)}.$$

Source: Andrew Ng

6. Other Classification Methods

- k-nn (k-nearest neighbor)
  - Given a training dataset D (with labels), classify record/object X to a particular class based on *k* objects that are the most similar to X (majority vote)
  - Issues
    - What kind of similarity measure to be used ?
    - How to identify k ?

      → k <= $|D|^{1/2}$



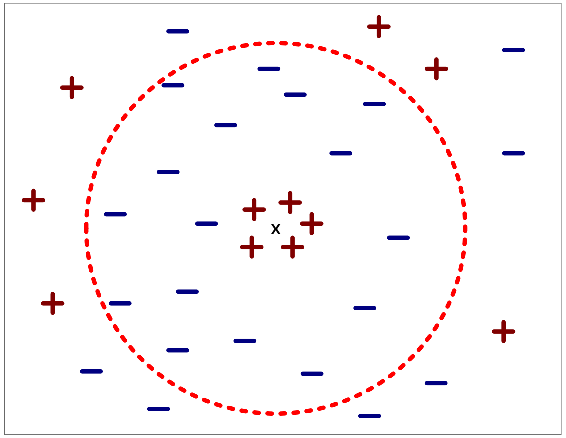
6. Other Classification Methods

- Select a measure
  - Euclidean

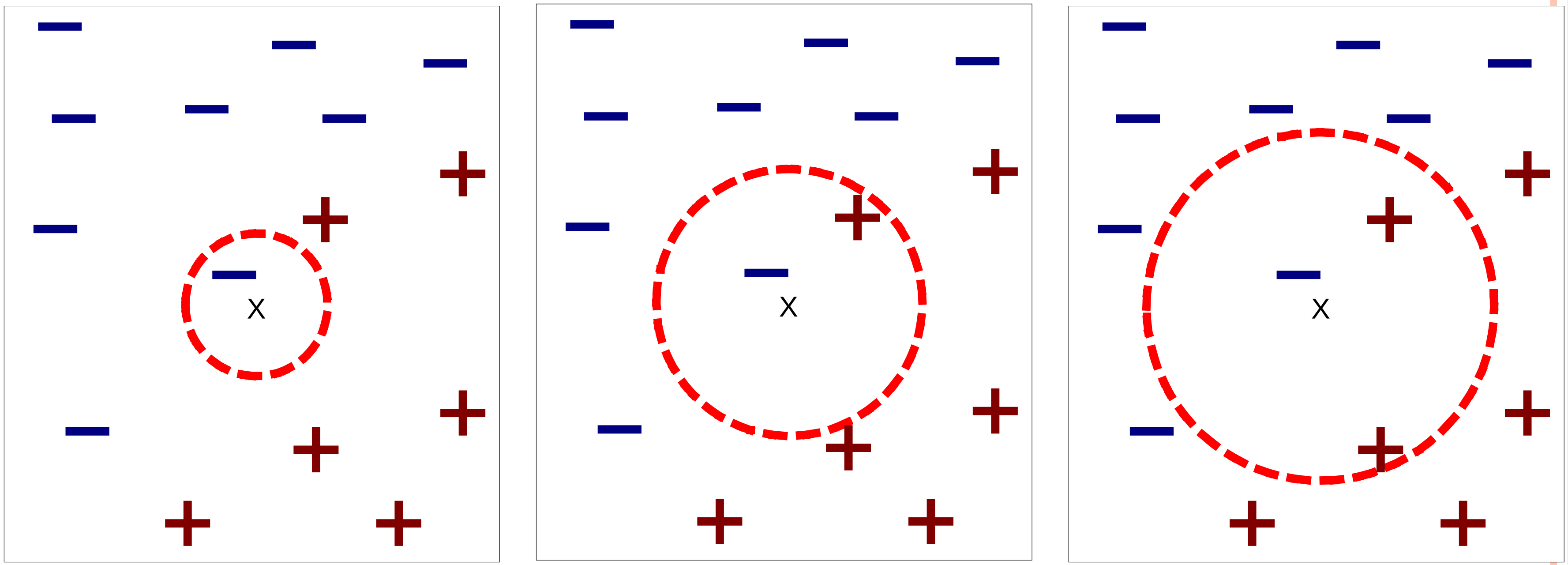
$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Select value of k
  - If k is too small -> affected by noise
  - If k is too large -> selected objects may come from different classes.

k is large!



6. Other Classification Methods



(a) 1-nearest neighbor

(b) 2-nearest neighbor

(c) 3-nearest neighbor

X $\in$ MINUS

X $\in$ MINUS
hay
X $\in$ PLUS ?

X $\in$ PLUS

7. Evaluate and Select a Classification Model

- Evaluation criteria
  - Accuracy
    - Describes how good a classifier can recognize different objects in the dataset
  - Speed
    - The computation cost for training and using the classifier
  - Robustness
    - The capability of the classifier to work with datasets that contain noise or missing data
  - Scalability
    - Possibility to build a classifier with a very large datasets
    - The capability to update/retrain the classifier with new dataset
  - Interpretability
    - The ability to understand the way a classifier work

7. Evaluate and Select a Classification Model

- Criteria: High Precision (P) and high Recall (R)

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F - Score = \frac{2 * P * R}{P + R}$$

TP: True positive; FP: False positive; FN: False negative

Fact / Classified	X	!X
X	TP	FP
!X	FN	TN

Precision

Recall

- E.x: Dataset has 9 BG and 4 FG flows (total: 13 flows)

  The classifier picks up 7 (4 BG and 3 FG) flows as BG flows. => P=4/(3+4)=4/7; R=4/(4+5)=4/9

7. Evaluate and Select a Classification Model

- Evaluate the accuracy/effectiveness
  - Holdout method: Randomly divide D to 2 different sets
    - Training set: (e.g., 2/3)
    - Test set (e.g., 1/3)
  - Cross validation
    - Divide D to k (k=10) portions with the same size
    - Iteration i, use D_i for testing and the rest for training
    - Calculate the average of evaluation measures from k rounds of execution

8. SUMMARY

- Classification with Decision trees: ID3, C4.5, CART
  - Slitting attribute selection
- Classification with Bayesian
  - Based on Bayes's theorem
- Classification with artificial neural network (ANN)
- K-nn classification
  - Based on the distance (or similarity)
- Evaluation and selection of classifier
  - Criteria, measures, and methods

Q&A

quangtran@hcmut.edu.vn