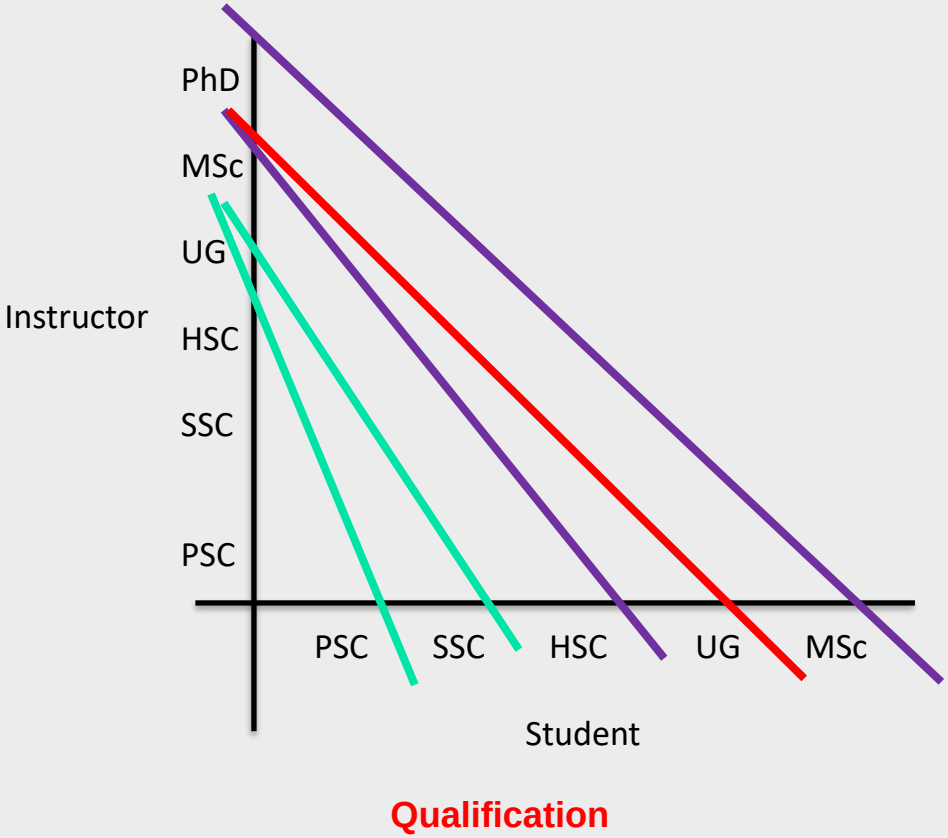
MCSE 666:
Pattern and Speech Recognition
Aspect & Prospect

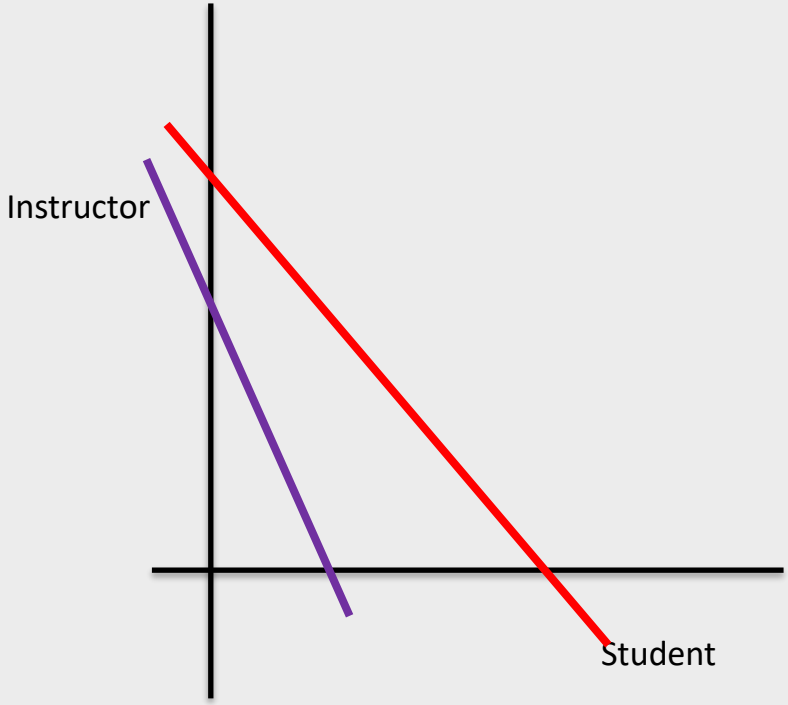
Dr. Md. Aminul Haque Akhand
Dept. of CSE, SUB

2

PG Course: Instructor vs. Student Qualification and Contribution

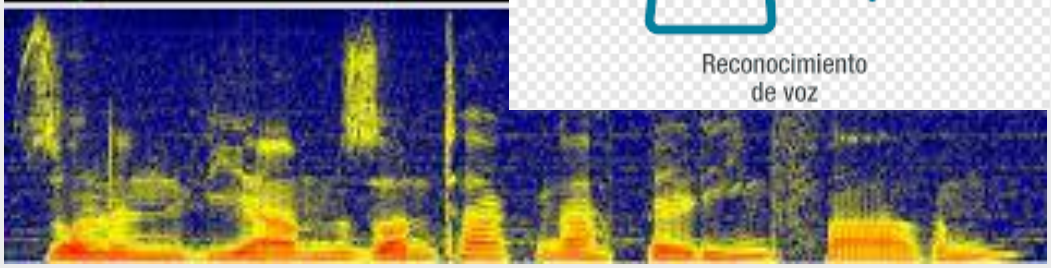


Qualification



Contribution

3 Pattern and Speech Recognition (PSR)



Pattern

Speech

Recognition

4 PSR Individual Terms: Pattern

pattern

noun

1. a repeated decorative design.
   "a neat blue herringbone pattern"

   Similar: design | decoration | motif | marking | ornament | ornamentation

2. a model or design used as a guide in needlework and other crafts.
   "make a pattern for the zigzag edge"

   Similar: sample | specimen | swatch

verb

1. decorate with a repeated design.
   "he was sitting on a soft carpet patterned in rich colours"

   Similar: decorated | ornamented | figured | tessellated | mosaic | goffered

2. give a regular or intelligible form to.
   "the brain not only receives information, but interprets and patterns it"

   Similar: shape | influence | form | model | fashion | mould | style | affect

Translate to Bangla

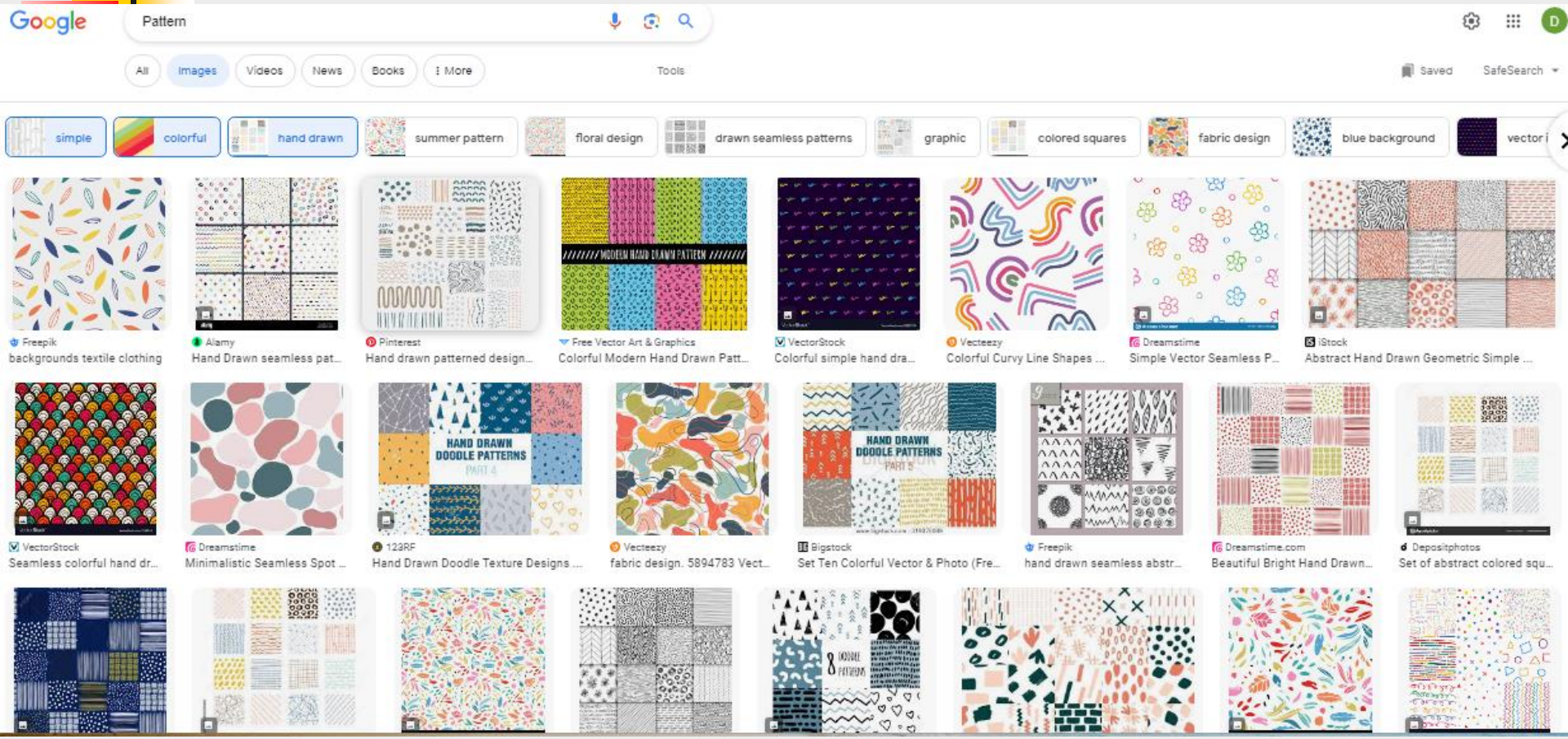
noun

1. প্যাটার্ন
2. আদর্শ

verb

1. আদর্শরূপে গঠন করা
2. আদর্শানুযায়ী গঠন করা

5 PSR Individual Terms: Pattern



A pattern is a regularity in the world, in human-made design, or in abstract ideas. The elements of a pattern repeat in a predictable manner. A geometric pattern is a kind of pattern formed of geometric shapes and typically repeated like a wallpaper design.

6

PSR Individual Terms: Speech

Translate to Bangla

noun

1. বক্তৃতা
2. কথাবার্তা

1. কথা
2. বাকশক্তি

speech

noun

noun: **speech**; plural noun: **speeches**

1. the expression of or the ability to express thoughts and feelings by articulate sounds.
   "he was born deaf and without the power of speech"

   Similar: speaking | talking | verbal communication | verbal expression | articulation

   - a person's style of speaking.
     "she wouldn't accept his correction of her speech"

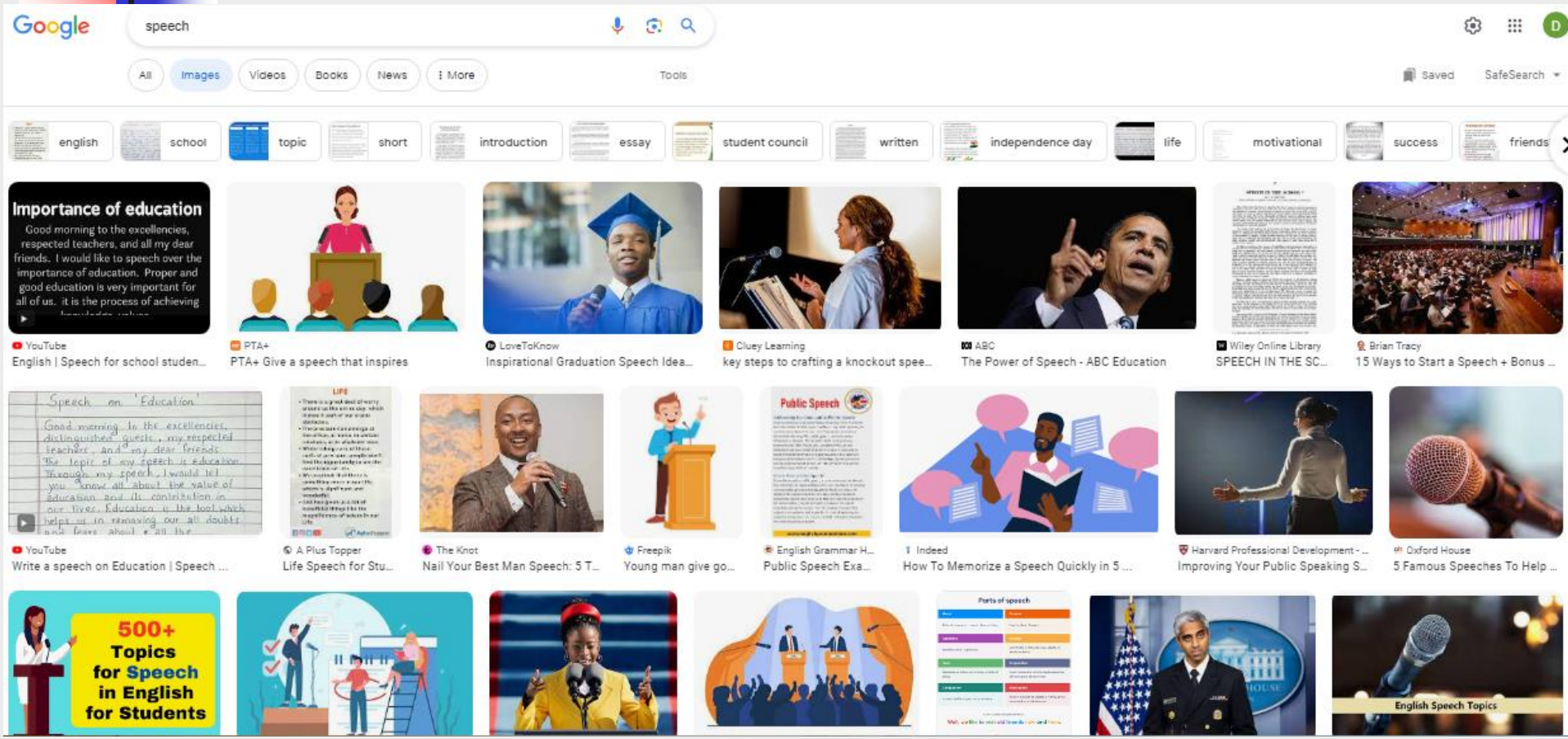
     Similar: diction | elocution | manner of speaking | articulation | enunciation

2. a formal address or discourse delivered to an audience.
   "he gave a speech about the company"

   Similar: talk | address | lecture | discourse | oration | disquisition

   - a sequence of lines written for one character in a play.
     "Antony's speech over Caesar's body"

7 PSR Individual Terms: Speech



Speech, language refer to the means of communication used by people. Speech is the expression of ideas and thoughts by means of articulate vocal sounds, or the faculty of thus expressing ideas and thoughts.

8

PSR Individual Terms: Recognition

Translate to Bangla

noun

1. স্বীকার
2. ঠাহর

1. স্বীকৃতিদান হওয়া
2. স্বীকৃত হওয়া

recognition

noun

noun: **recognition**; plural noun: **recognitions**

1. identification of someone or something or person from previous encounters or knowledge.
   "she saw him pass by without a sign of recognition"

   Similar: identification | recollection | recall | remembrance

2. acknowledgement of the existence, validity, or legality of something.
   "the unions must receive proper recognition"

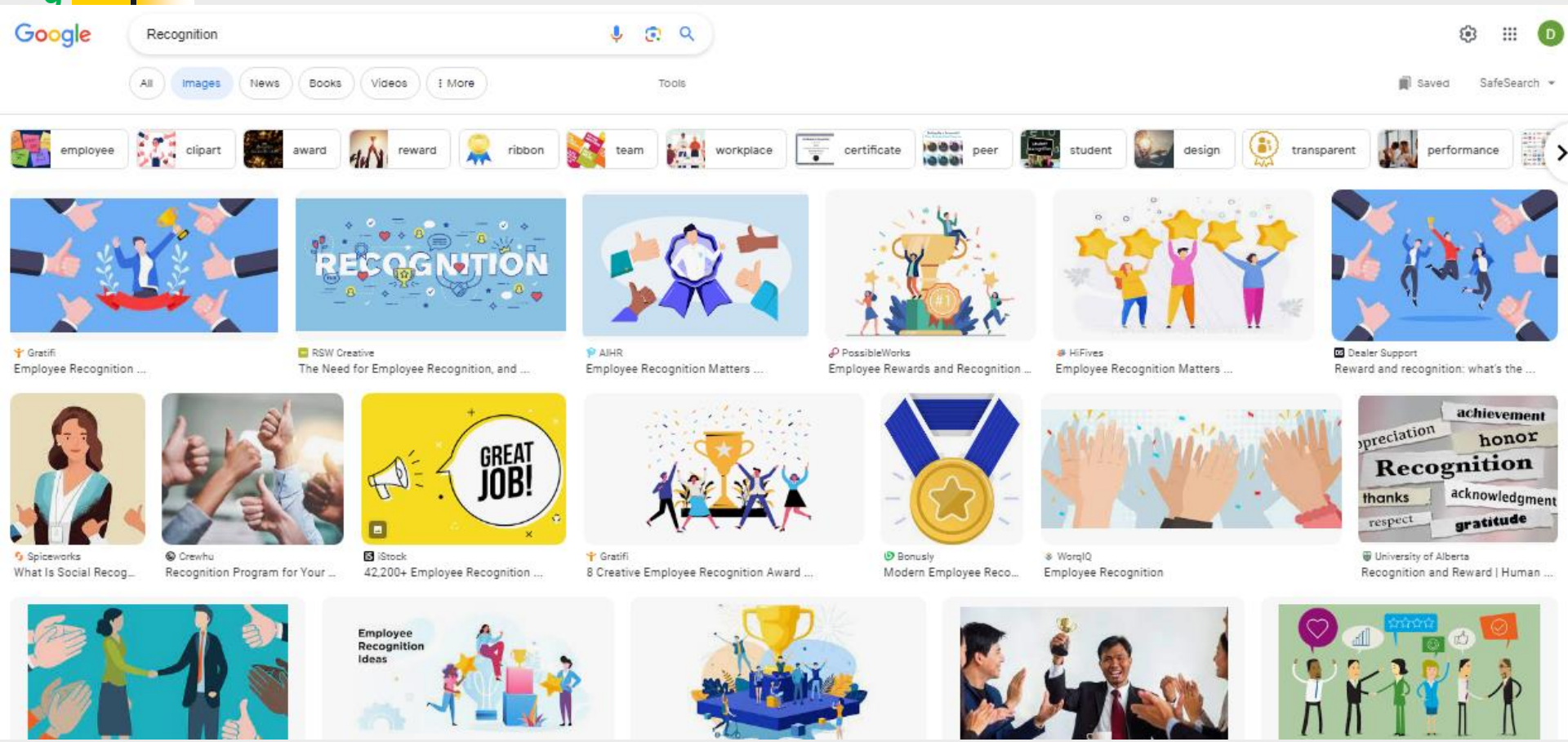
   Similar: acknowledgement | acceptance | admission | conceding | concession

   - appreciation or acclaim for an achievement, service, or ability.
     "his work was slow to gain recognition"

     Similar: appreciation | gratitude | thanks | congratulations | a pat on the back

   - formal acknowledgement by a country that another political entity fulfils the conditions of statehood and is eligible to be dealt with as a member of the international community.
     noun: **diplomatic recognition**; plural noun: **diplomatic recognitions**
     "they are granting full recognition to the republic"

PSR Individual Terms: Recognition



- Recognition in sociology is the public acknowledgment of a person's status or merits (achievements, virtues, service, etc.).
- Another example of recognition is when some person is accorded some special status, such as title or classification.

10

Pattern and Speech Recognition (PSR)

Pattern Speech Recognition

Speech Pattern Recognition

Pattern Recognition and
Speech Recognition

Intelligent Pattern Recognition and
Intelligent Speech Recognition

Examples of Pattern Recognition

Table 2 Examples of Pattern Recognition Applications

Problem Domain	Application	Input Pattern	Pattern Class
Bioinformatics	Sequence Analysis	DNA/Protein sequence	Known type of genes/patterns
Data mining	Searching for meaningful patterns	Points in multi dimension space	Compact and well separated clusters
Document classification	Internet search	Text document	Semantic categories
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters / words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective / non defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres e.g. action, dialogue etc
Biometric recognition	Personal identification	Face, iris & finger print	Authorized user for access control
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories, growth pattern of crops
Speech recognition	Telephone directory enquiry with operator	Speech waveform	Spoken words

Mayank Parasher et al., Anatomy on Pattern Recognition, June 2011, Indian Journal of Computer Science and Engineering 2(3)

Examples of Pattern Recognition

What Are the Real-World Applications of Pattern Recognition?

https://serokell.io/blog/applications-of-pattern-recognition

10 Real Life Examples Of Pattern Recognition

https://numberdyslexia.com/pattern-recognition-real-life-examples/

Pattern recognition and use in real life problem solving

https://suresolv.com/problem-solving-techniques/pattern-recognition-and-use-real-life-problem-solving

What is pattern recognition and why it matters? Definitive guide

https://theappsolutions.com/blog/development/pattern-recognition-guide/

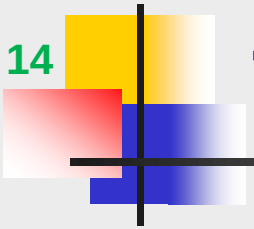
13

MCSE 666: Pattern and Speech Recognition

SUB MCSE 666 Syllabus or Course Outline

Introduction to formal languages and patterns; string languages for pattern description; higher dimensional pattern grammars; syntax analysis as a recognition procedure; stochastic languages; error correcting parsing for string languages; error-correcting tree automata; cluster analysis for syntactic patterns; grammatical inference for syntactic pattern recognition; syntactic approach to texture analysis;

Speech signal: production, perception and characterization; signal processing and analysis; pattern comparison techniques: distortion measures, spectral-distortion measures, time alignment and normalization; recognition system design and implementation: source-coding, template training, and performance analysis; connected word models; continuous speech recognition: sub word units, statistical modeling, and context-dependent units; task oriented models.



Textbooks

15

Open Discussion on Course Conduction

MCSE 666:Pattern and Speech Recognition

Introduction to Pattern Recognition

Dr. Md. Aminul Haque Akhand
Dept. of CSE, SUB

2

What Is Pattern Recognition?

Pattern recognition (PR) is a process by which some input is measured, analyzed, and then classified as belonging to one of a set of classes

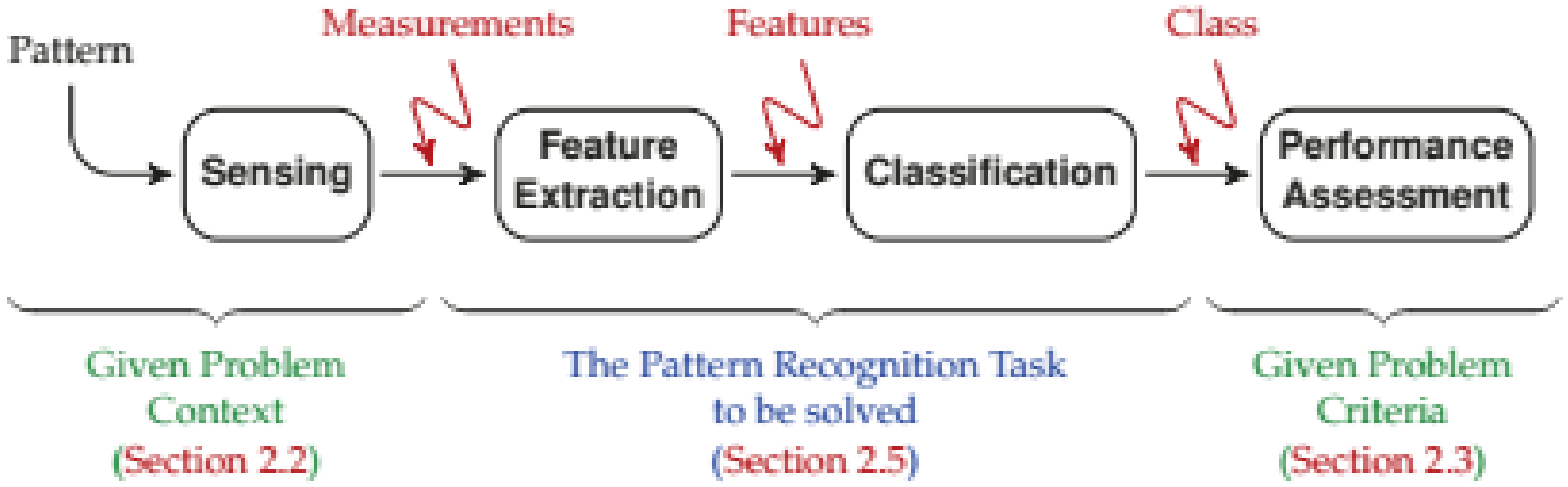
Process of PR and classification is a continual, never ending aspect of every-day human existence:

Pattern recognition task	Possible classes
What is in front of you as you walk?	Door vs. Window Sidewalk vs. Road
What music are you listening to?	Familiar or Unfamiliar Genre (Rock, Classical, . . .) Name of Composer or Group
Is the traffic intersection safe to cross?	Green vs. Red light Pedestrian Walk vs. Stop Car Present vs. Not Present
Reading a page in a textbook	Letters of the Alphabet Text vs. Graphics Languages
You smell something in your	Cookies finished baking? (Yes/No) Is something burning? (Yes/No)

3 What Is Pattern Recognition? Cont.

- PR as a human experience refers to a perceptual process: some form of sensory input is sensed, analyzed, and recognized (classified), either subconsciously (by instinct) or consciously (based on previous experience).
- Patterns may be presented in any sensory modality: vision, hearing, touch, taste, or smell.
- PR as a technical discipline: a process in which an input object is measured, analyzed, and classified by a machine as being more or less similar to some class in a set of classes.

4 Pattern Recognition Framework

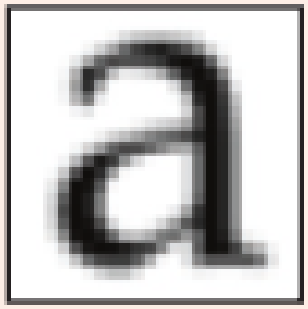
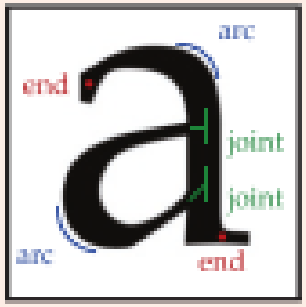


The goal of PR is to provide a machine with a kind of perceptual capability to automatically extract useful information from measured data.

5

PR Illustration with Example

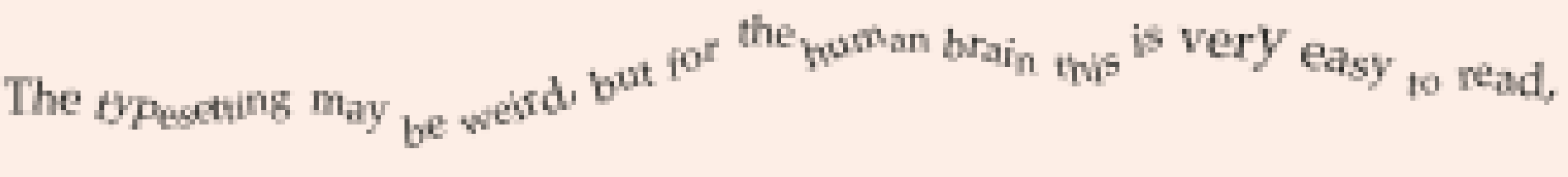
Pattern Recognition of Text

Pattern	Attributes to Measure	Measurements	Strengths and Weaknesses
"a" →		Vector of pixel values	Fast, easy, explicit Sensitive to changes in font style, size, and rotation
"a" →		Vector of shape properties	Robust to changes in size and rotation Complicated features to extract
"a" →	Complex Nonlinear Algorithm	Vector of values, but with no intuition	Possibly very flexible May be very hard to learn Difficult to analyze

6

PR Illustration with Example

Pattern Recognition of the Mind



It deons’t mettar in waht oerdr the lrttees in a wrod are, as lnog as the frist and lsat ltteres rmeain in the rgiht pacle.

- The retina in the eye is densely packed with light-sensitive cells, so that it may be tempting to think
- Our brain effectively sees and perceives the world as a great many pixels.
- There are many simple mind tricks or optical illusions

7

PR Illustration with Example

Pattern Recognition of the Mind

There are no black dots in the white circles.

The two eyes are the same colour. There is no blue pigment, at all, in the left eye.

This is a static image, yet try zooming in and scrolling.

- The eye is not just sending pixellated images to the brain.
- There is a great deal of feature extraction taking place, much already in the retina
- It is very helpful in running through a forest, but perhaps not so useful in staring at deliberately manipulated images on a page.

8

Features from Patterns

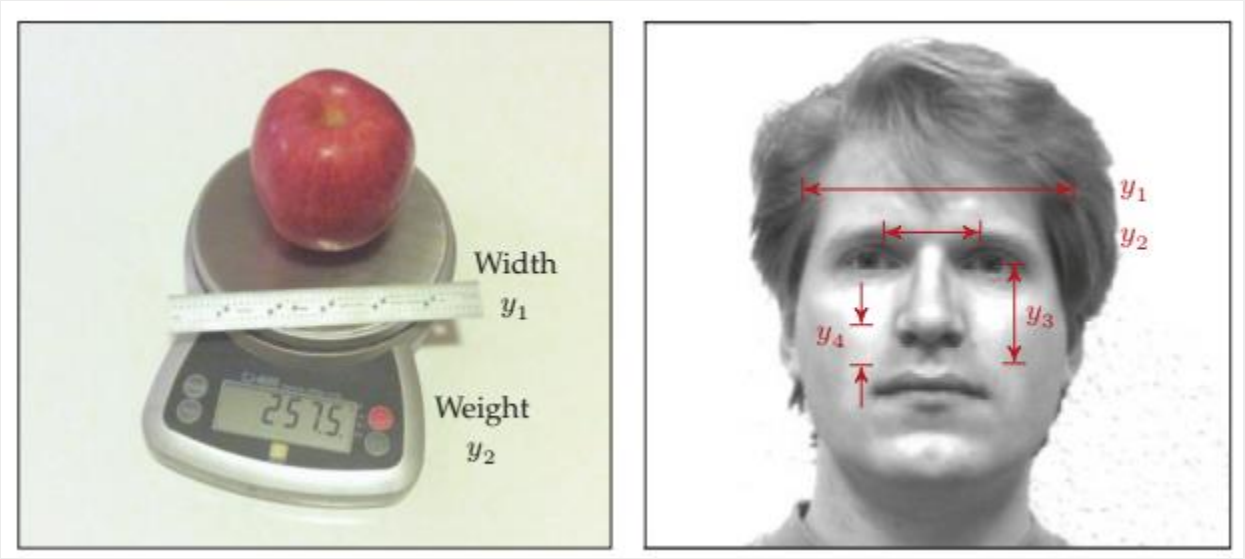
- The word “pattern” may bring to mind texture, fabric, or shape.
- In the context of *pattern recognition*, the notion of pattern is far more broad, and can apply to any *thing* that can be distinguished from another *thing*.
- Identity view point: infer the unknown *identity* of an object
- Type of wildflower, type of songbird, or the name of the person facing a camera — each of these has a certain identity to determine from measurements.

A pattern is assumed to have certain *properties* or *attributes* which distinguishes it from other patterns.

9

Features from Patterns (Cont.)

One or more measurements are taken of a pattern



Size / weight of a piece

Various dimensions from face

Does all measurements from patterns important to distinguish?

10

Features from Patterns (Cont.)

- The selection of appropriate measurements is an essential
- Measurements may cost money and/or time, and
- Poor measurements lead to poor performance of the resulting classifier.

The process of transforming measurements into features facilitate classification, normally in one or both of the following ways:

1. By reducing the dimensionality of the problem: $n < m$
2. By creating features in which patterns are more clearly distinguished.

$$\text{Measurements } \underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \xrightarrow{\underline{x} = f(\underline{y})} \text{Features } \underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

11

Features from Patterns (Cont.)

$$\text{Measurements } \underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \xrightarrow{\underline{x} = f(\underline{y})} \text{Features } \underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- The feature extraction function *f()* can focus the information from x, or it can remove irrelevant information from x, but *f()* never adds information.
- Data Processing Theorem: *x can never have more information than was present in y.*

- An effective feature extraction function *f*() can make the PR problem *easier*,
- However, in principle, the best possible classifier based on the measurements *y* should perform at least as well as the best possible classifier based on the features *x*.

12

Features from Patterns (Cont.)

Features may be intuitive or they may be quite abstract.
Consider, the measurements of an electric motor:

$$\text{Measurements} \quad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} \text{Motor voltage (Volts)} \\ \text{Motor current (Amperes)} \\ \text{Motor speed (RPM)} \\ \text{Motor winding temperature (°C)} \\ \text{Surrounding air temperature (°C)} \end{bmatrix}$$

$$x = f(\underline{y}) = y_1 \cdot y_2$$

Power as Feature: Understandable

$$x = f(\underline{y}) = \sqrt{y_1 - y_2} - \frac{y_3}{y_4}$$

- Uninterpretable feature do not make any physical sense,
- But perhaps be effective as a feature for classification.

13

Classes and Classification

- The whole purpose of PR or classification consists of assigning an object to a class.
- A class is a particular pattern, or possibly a group of patterns which are similar or equivalent in some sense.

In a given problem, the set of classes *C* is defined as

$$C = \{C_1; C_2; -; -; -; -; C_K\};$$

Have to choose one class from *K* different classes.

Members of a class share some common properties or attributes

In PR problems the class set *C* is predefined and has been specified as part of the problem to be solved.

Classes and Classification

So, for example, in a face recognition problem each person is their own class, so the set of classes would be defined as

$$\mathcal{C} = \{\text{“Paul Fieguth”}, \text{“Bob”}, \text{“Jane”}, \text{“Ali”}, \ldots\}$$

so that I would be a member of the “Paul Fieguth” class:

$\in$ “Paul Fieguth”

At a university we might define a different set of classes

$\mathcal{C} = \{$ “Professors”, “Staff”, “Undergraduate Students”, “Graduate Students”$\}$

such that

“Paul Fieguth” $\in$ “Professor”

One *could* use pattern recognition to estimate age, in which case one might have classes like

$$\mathcal{C} = \{\text{“0–10 years”}, \text{“10–20 years”}, \text{“20–30 years”}, \ldots\}$$

such that now I appear in a class as

“Paul Fieguth” $\in$ “50–60 years”

Classes and Classification

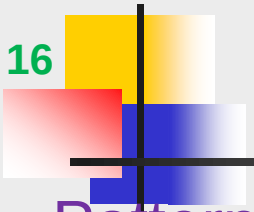
Way of Describing Classes

Via Prototype Idealized representation or notion of the “essence” of the class. Pros: each class is unambiguously defined, Cons: no scope for variability.

Via Parameterized Shape: A generalization of the prototype; the class has a known shape (e.g., rectangular or elliptical), the shape is described in some number of parameters (e.g., ellipse centre, rotation, and axis lengths). Pros. more flexible than that of a single prototype, Cons: still requires the type of shape to be assumed or known.

Via Statistical Distribution: Some description of the likelihood or probability of a class member having a particular set of measurements or features. Pros.: very comprehensive, Cons: There will be circumstances when the statistics are not known and may be difficult to infer.

Via Samples: A set of given samples (many apples, or tigers, or bicycles) directly characterizes the class. Pros: Highly convenient, since nothing further needs to be done to describe the class Cons: Storage and computational challenges, since all of the data need to be saved.



Classes and Classification

Variability in a Class

Patterns do not need to be identical to belong to the same class: not all pictures of a person the same, or of tigers, or of apples.

There are at least two sources of variability present in the measurements associated with a single class:

1. The inherent variability within a class: Every class will consist of members which differ in some way. The degree and nature of the inherent variability will depend greatly on the class definition.

"Fruit" class contains all manners of variability in colour, size, and shape; "Apple" class is much more specific, but apples do come in different colours and patterning; the "Granny Smith Apple" class is even more specific, but still will have apples of different sizes or with more or fewer blemishes.

2. Noise or random variations in measurement: Every measurement involves some sort of physical process which will be subject to error, such as thermal noise in electronics, or quantization noise in converting an analogue signal to a digital representation.

17

Classifier and Classification

In PR tasks the class set *C* is predefined and has been specified as part of the problem to be solved.

A *classifier* is some function *g*(), possibly analytical (i.e., an equation) or a computer algorithm, which assigns a class label to a given feature:

$$g(x) \in C = \{C_1; C_2; : : : ; C_K\}$$

There is a strong relationship between feature extraction and classification, such that

- Good features allow for simpler classifiers, whereas
- Complex classifiers can compensate for weaker features, which are unable to fully separate the pattern classes.

18

Classifier and Classification

Two Fundamental Steps Associated With Classification:
I. Classifier Learning and II. Classifier Testing or Validation

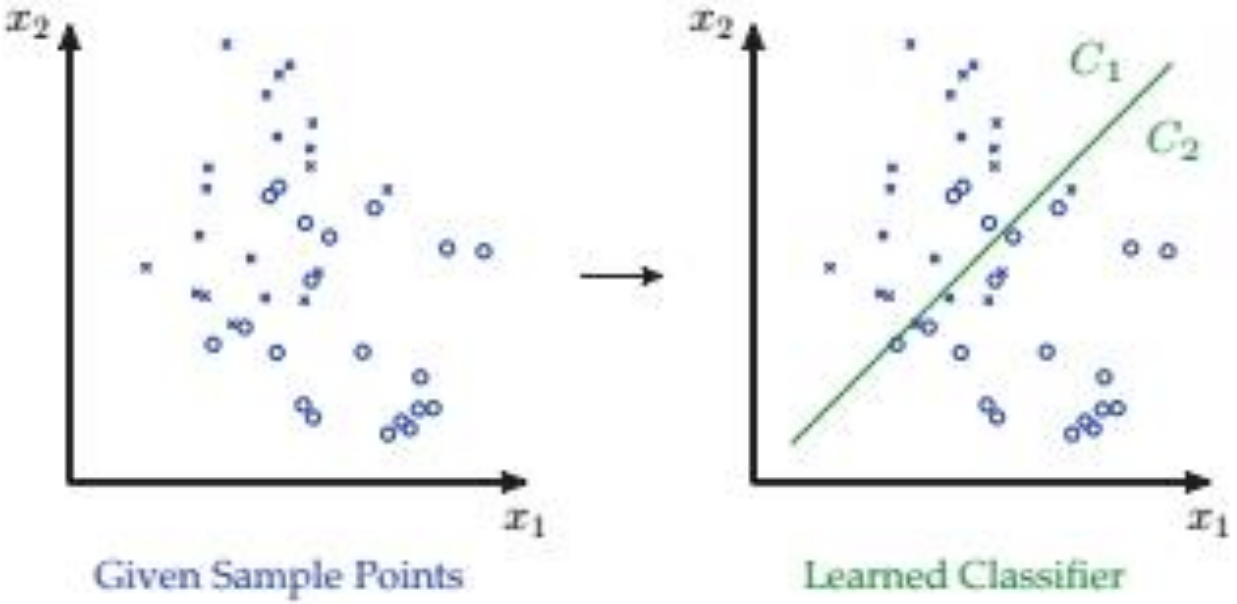


Fig. 2.3. PATTERN RECOGNITION I — CLASSIFIER LEARNING: A classifier, here a straight line (right) dividing a feature space into classifications C_1 and C_2, can be *learned* from a given set of sample points (left).

Classifier and Classification

II. Classifier Testing or Validation

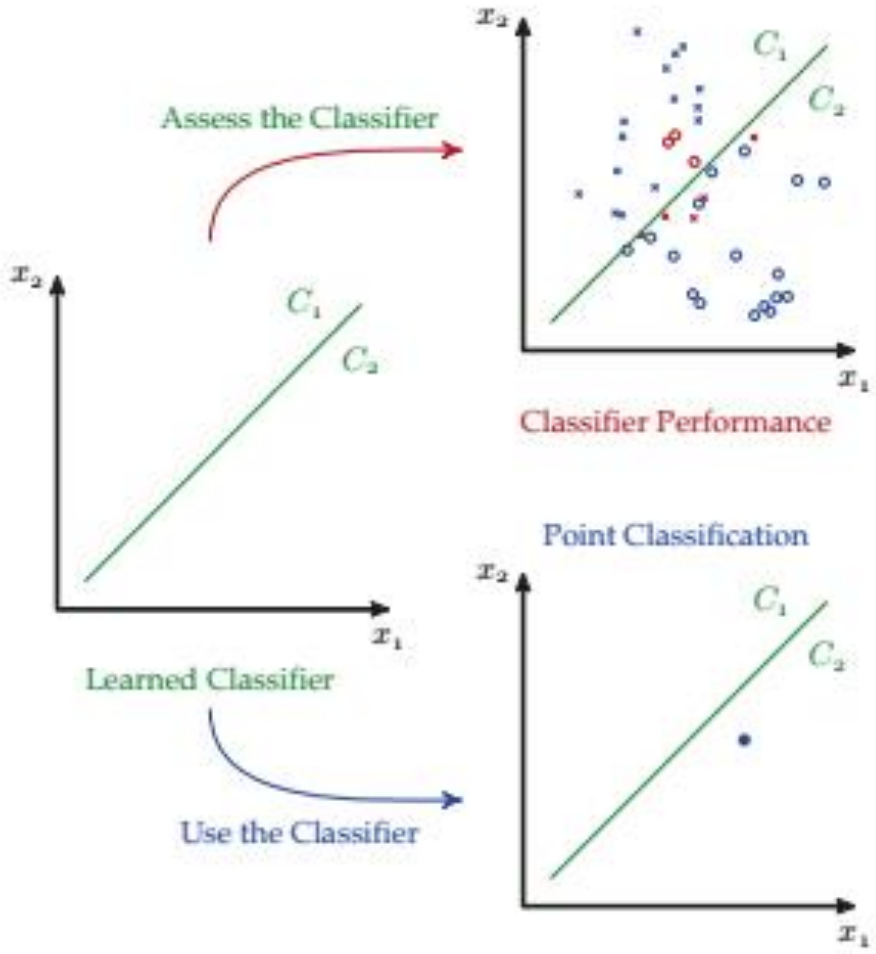


Fig. 2.4. PATTERN RECOGNITION II — CLASSIFIER TESTING: What can we do with the learned classifier, left, from Figure 2.3? We could assess its performance (top), for example by counting how many sample points are classified correctly (blue) and incorrectly (red). Or we could apply the classifier (bottom) to a new, unknown point and then classify it.

20

Classifier and Classification

Illustrative Examples of Classification

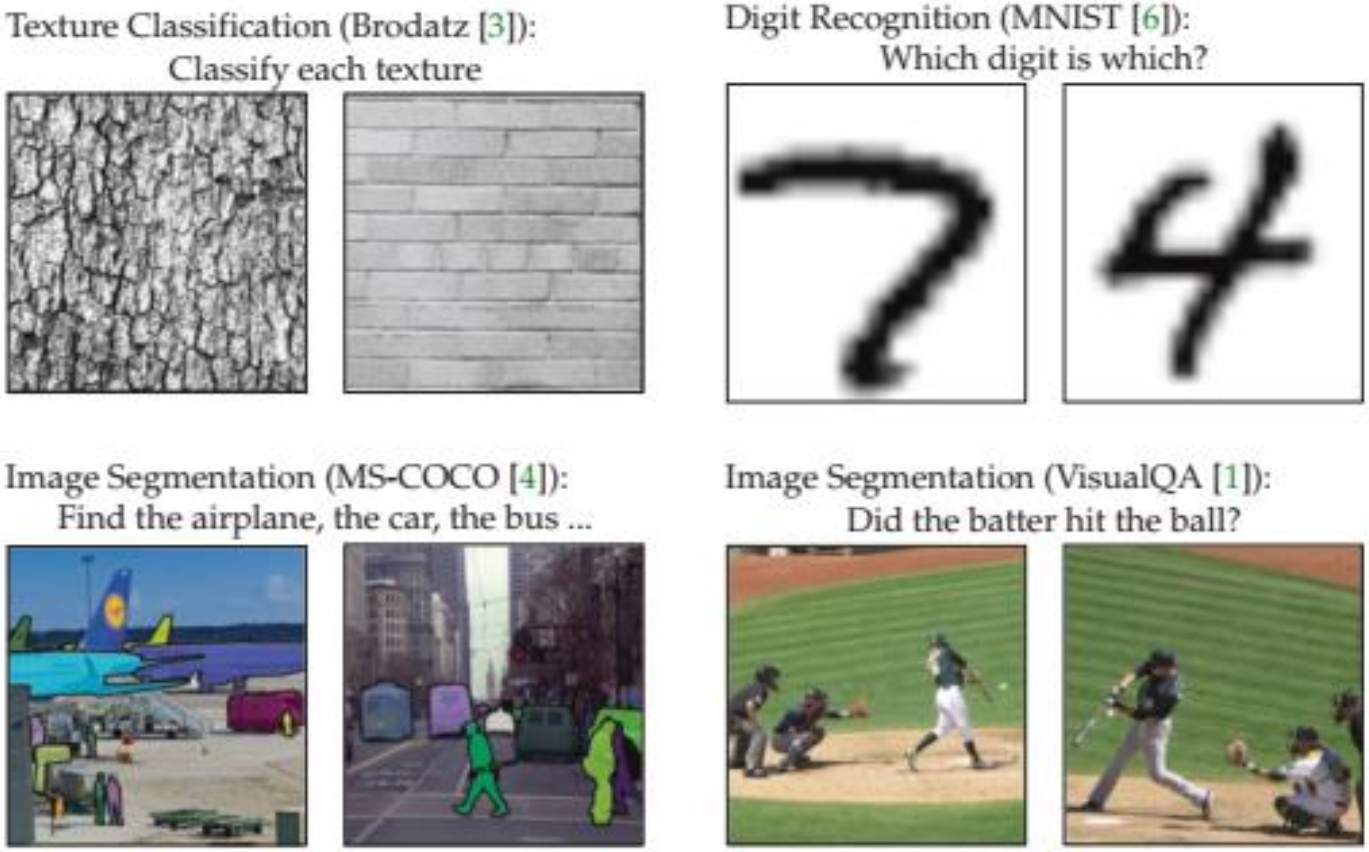


Fig. 2.5. PATTERN RECOGNITION ON IMAGES: Since the human visual system is so dominant in human perception, a great deal of pattern recognition focuses on image-related problems. Here four examples are shown, from comparatively straightforward (top), the classification or recognition of whole images, to rather advanced (bottom), such as recognizing the objects within an image or being able to answer complex high-level questions.

21

Classifier and Classification

Pattern Recognition Problems

Four Fundament Types PR Problems:

1. Model-Based
2. Supervised
3. Semi-Supervised
4. Unsupervised

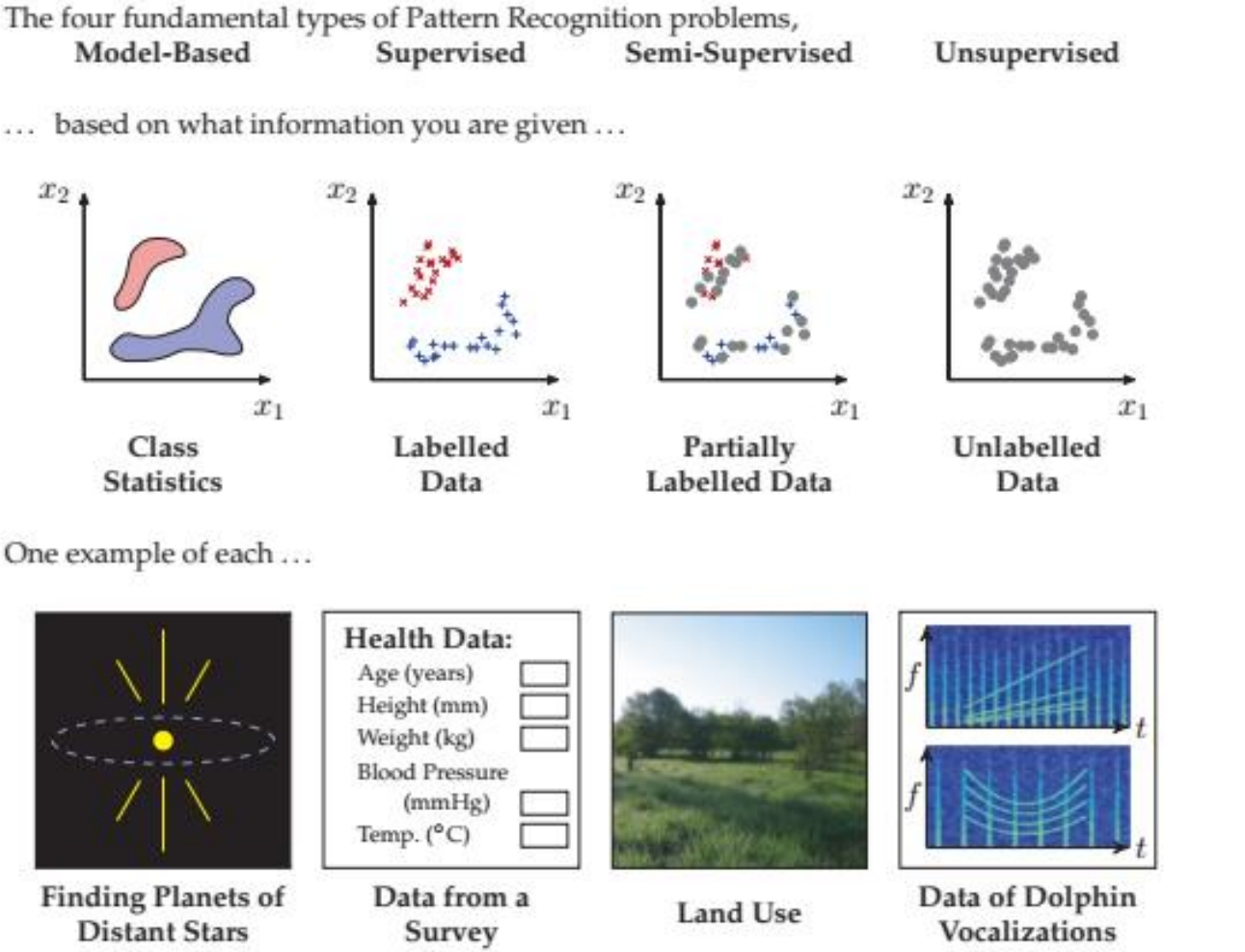


Fig. 2.6. THE FUNDAMENTAL PROBLEMS: There are four fundamental pattern recognition problems, ordered from the most detailed problem description (left) to the most ambiguous (right).

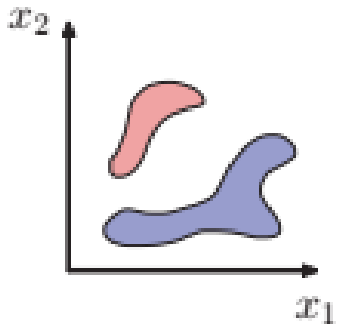
22

Classifier and Classification

Pattern Recognition Problem: Model-Based

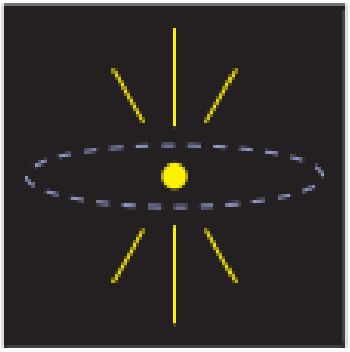
Scenario 1 (Model-Based): Model is known or given

- Hope to have most information regarding a PR problem the behaviour of the measurements for each of the pattern classes.
- Normally characterized in a statistical fashion, such as $p(y|C)$ = The distribution of measurement vector y given class C



Class Statistics

One example of each …



Finding Planets of Distant Stars

- Such detailed information will be available only in those contexts where the physical process is known by which a given pattern class gives rise to measurements.
- Very convenient to have detailed information, since statistical decision theory allow to explicitly define the optimal classifier, in the sense of minimizing the probability of classification error

23

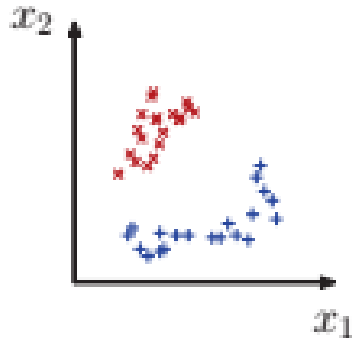
Classifier and Classification

Pattern Recognition Problem: Supervised

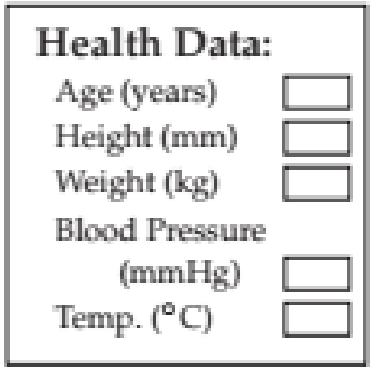
Scenario 2 (Supervised): Model is not known, labelled data are available

- Do not have an exact description of the problem, as in Scenario 1
- Given labelled data, meaning data pairs of the form $\{y_i; C_k\}$ -> The ith measurement vector y_i is in class C_k

- Labelled data do not just *magically* appear; labelled or tagged by a human observer, so refer scenario as *supervised*
- Exceptionally expensive or labour-intensive with larger datasets.
- May derive a classifier directly from the given labelled or learn an empirical probability model as of Scenario 1



Labelled Data



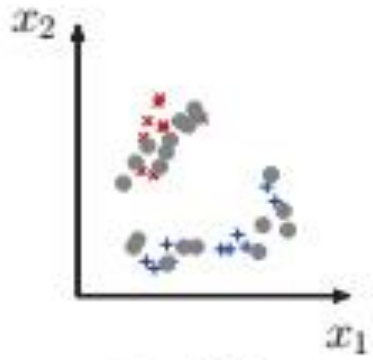
Data from a Survey

Classifier and Classification

Pattern Recognition Problem: Semi-Supervised

Scenario 3 (Semi-Supervised): Model is not known, some are labelled data, some are not

- Labelled data can be expensive, requiring manual labelling. A huge data be unlabeled and Scenario 2 ignores all unlabeled data.
- Problem refers to as semi-supervised when some degree of human input is required
- In semi-supervised case, a small set of samples have been manually labelled for classification (e.g., face images tagged by a human observer), but then also to leverage a very large set of unlabeled data



Partially Labelled Data

Land Use

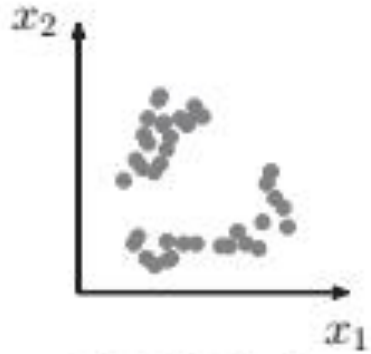
25

Classifier and Classification

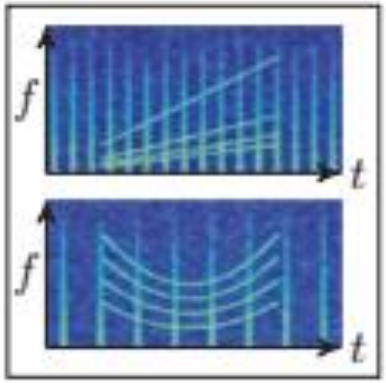
Pattern Recognition Problem: Unsupervised

Scenario 4 (Unsupervised): Model is not known, no labelled data available

- Pattern measurements are available, however the points have no associated class information; this is known as an unsupervised problem.
- The range of problems here is still very broad, depending on whether we are told the number of classes, or their typical size or separation, or perhaps nothing at all.
- Clustering problems are in Unsupervised category



Unlabelled Data



Data of Dolphin Vocalizations

26

PR Case Study: Biometric Recognition

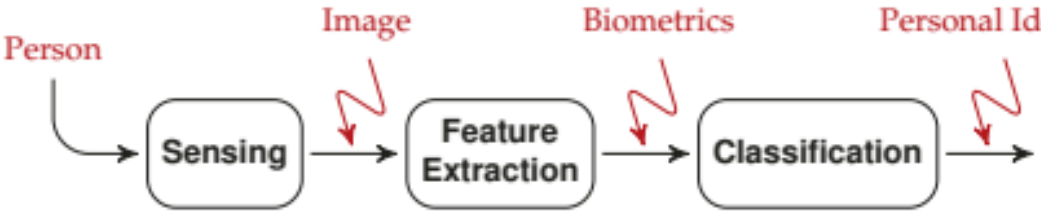


Fig. 2.7. BIOMETRIC RECOGNITION: In order to recognize a specific person on the basis of remotely-sensed biometrics, we need to acquire (sense) an image of some part of the body which has a unique signature (fingerprint, retina etc), extract biometric features from this image, and then develop a classifier that reliably recognizes the individual.

Biometrics

- Face
- Fingerprint
- Iris (the colored region in your eye around the pupil)
- Retina (the pattern of arteries in the back of your eye)
- Veins (vein structure in hand or arm)
- DNA

Basic Components of Biometric Recognition:

1. Image Acquisition
2. Feature Extraction
3. Classification

The classifier returns one member of the class set

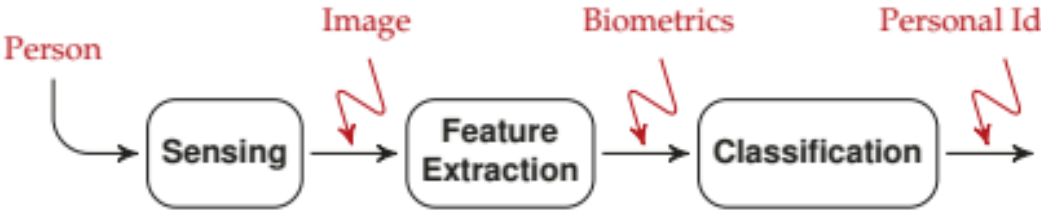
$$C = \{NoMatch; Person_1; Person_2; : : : ; Person_K\}$$

Classifying *x* as *NoMatch* where *Person*1 is correct: Frustration

Classifying *x* as *Person*2 where *Person*1 is correct: Security Breach

27

PR Case Study: Biometric Recognition



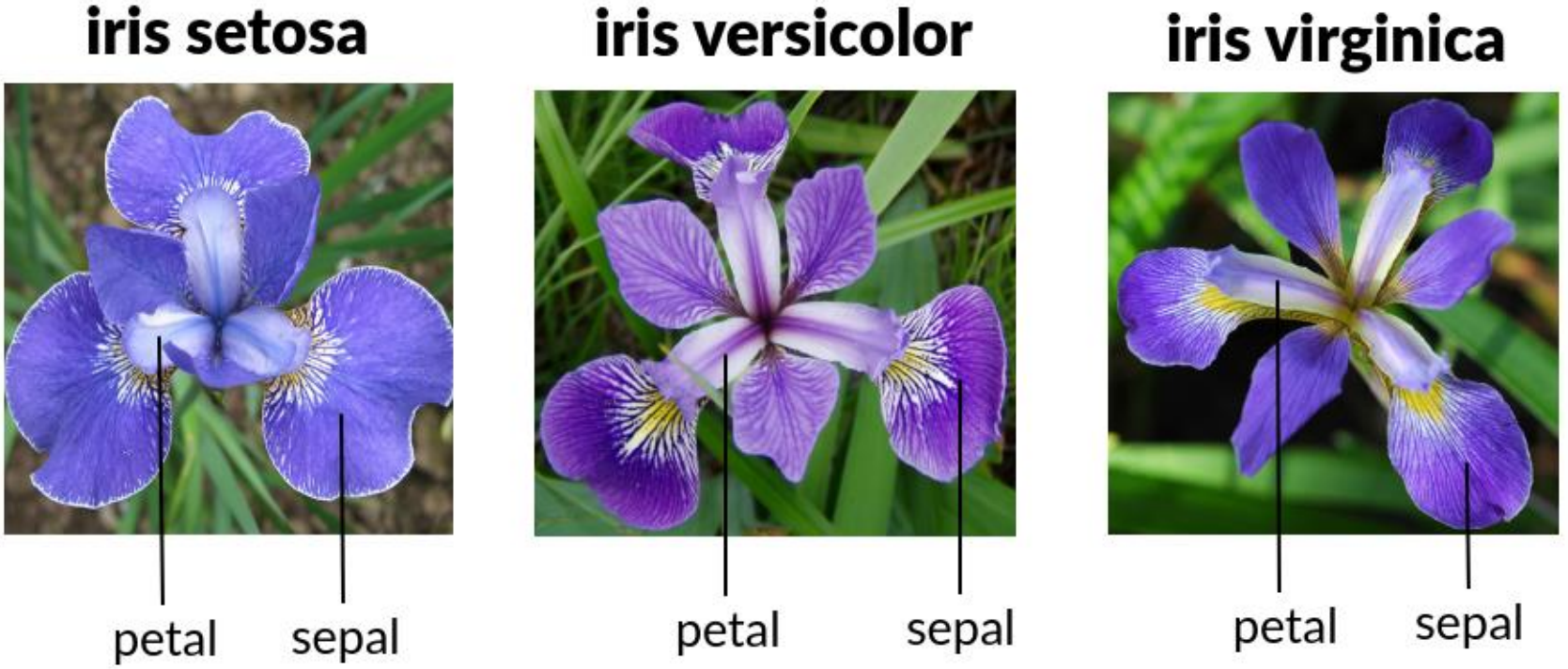
Classifying *x* as *NoMatch* where *Person*1 is correct: Frustration
Classifying *x* as *Person*2 where *Person*1 is correct: Security Breach

Avoiding frustration and security breaches, a successful biometric strategy must also satisfy

- Universality: Every person should be measurable, regardless of age and health
- Uniqueness: The feature vectors extracted for a given person should be robustly unique
- Consistency: For a given person the feature vector should be highly repeatable from one try to the next, and should be slowly (or not at all) varying over time.

Other criteria: non-invasiveness, social acceptability, or how easily the system would be to defeat via nefarious means and so on

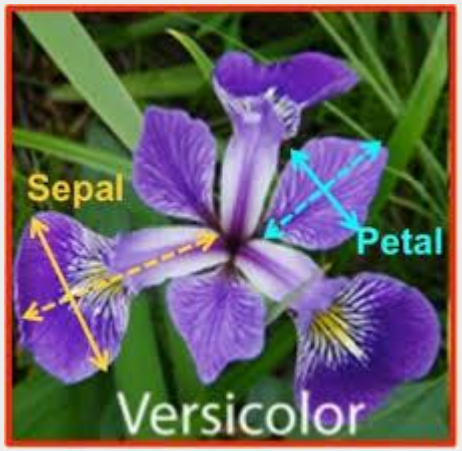
PR Hands-on: Iris Flower Recognition



$C = \{C_1; C_2; C_3\}$ = {"Iris Setosa", "Iris Versicolor", "Iris Virginica"}

Each plant four measurements were taken:

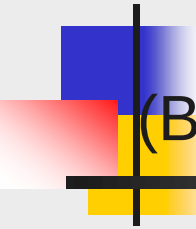
y = (Sepal Length, Sepal Width, Petal Length, Petal Width)



29

Open Discussion

MCSE 666:Pattern and Speech Recognition



Learning

(Biological, Machine Learning, Regression, and Classification)

Dr. Md. Aminul Haque Akhand
Dept. of CSE, SUB

2

Learning

Learning is the process of acquiring new or existing modifying knowledge, behaviors, skills, values, or preferences

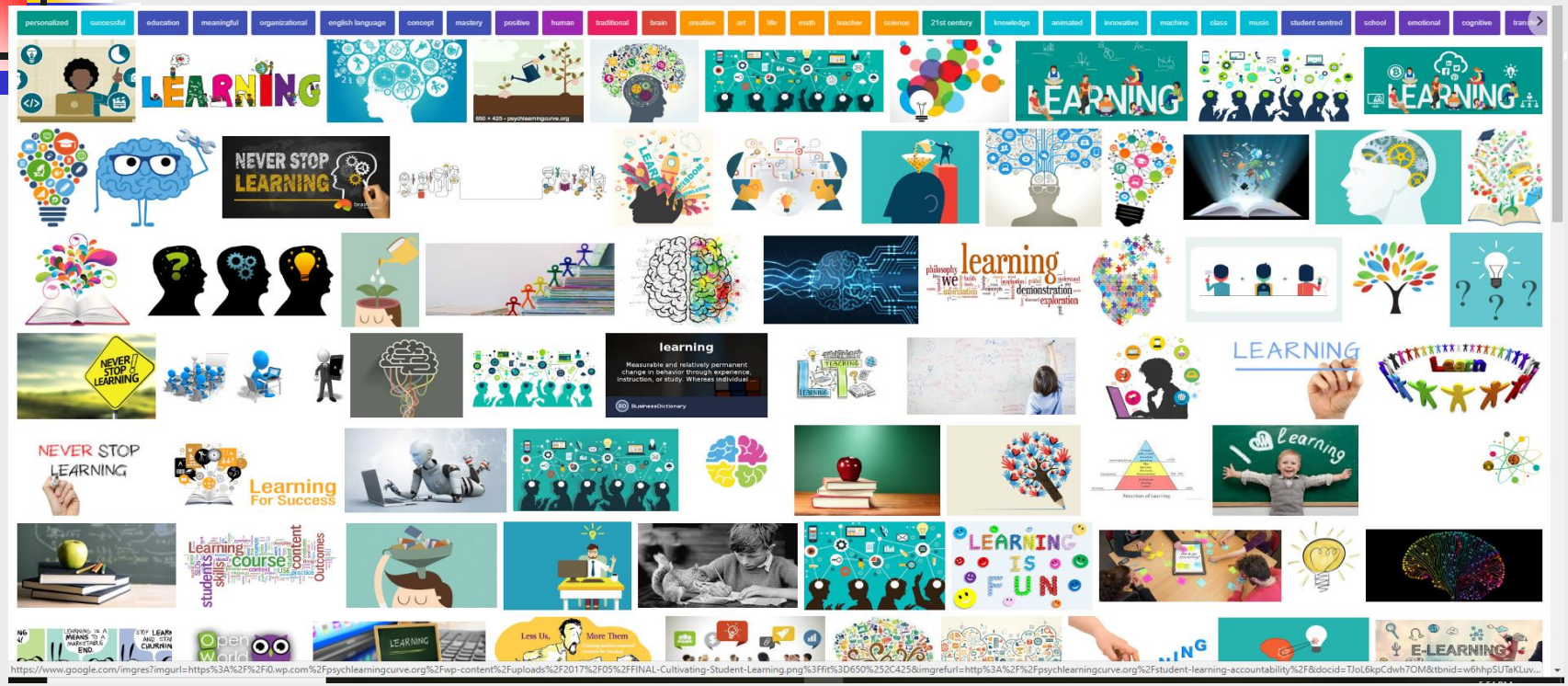
https://en.wikipedia.org/wiki/Learning

Learning is “a process that leads to change, which occurs as a result of experience and increases the potential for improved performance and future learning”
---(Ambrose et al, 2010, p.3).

“A change in human disposition or capability that persists over a period of time and is not simply ascribable to processes of growth.”

— From The Conditions of Learning by Robert Gagne

3 Learning



The change in the learner may happen at the level of knowledge, attitude, or behaviour. As a result of learning, learners come to see concepts, ideas, and/or the world differently.

Learning is not something done to students, but rather something students themselves do. It is the direct result of how students interpret and respond to their experiences.

https://www.teachwithmrst.com/post/what-is-learning

Human brain is the main element of learning

4

Machine

A machine is a physical system using power to apply forces and control movement to perform an action. … Machines can be driven by animals and people, by natural forces such as wind and water, and by chemical, thermal, or electrical power ...They can also include computers and sensors that monitor performance and plan movement, often called mechanical systems. **https://en.wikipedia.org/wiki/Machine**

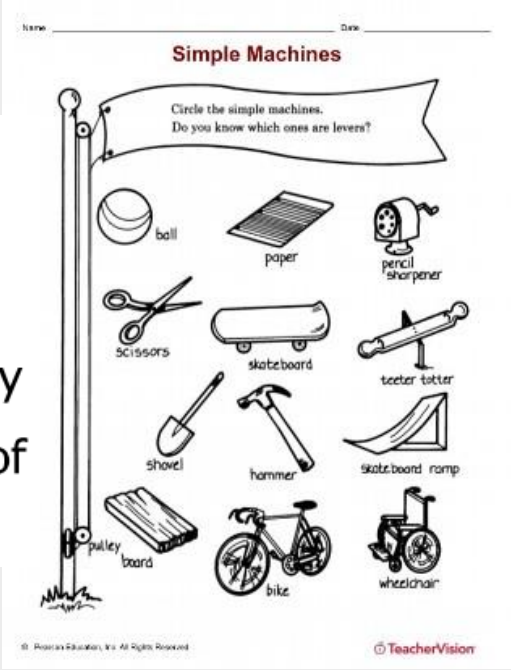
An apparatus using or applying mechanical power and having several parts, each with a definite function and together performing a particular task. *'a fax machine'*

Oxford Dictionary

1.General: Semi or fully automated device that magnifies human physical and/or mental capabilities in performing one or more operations.

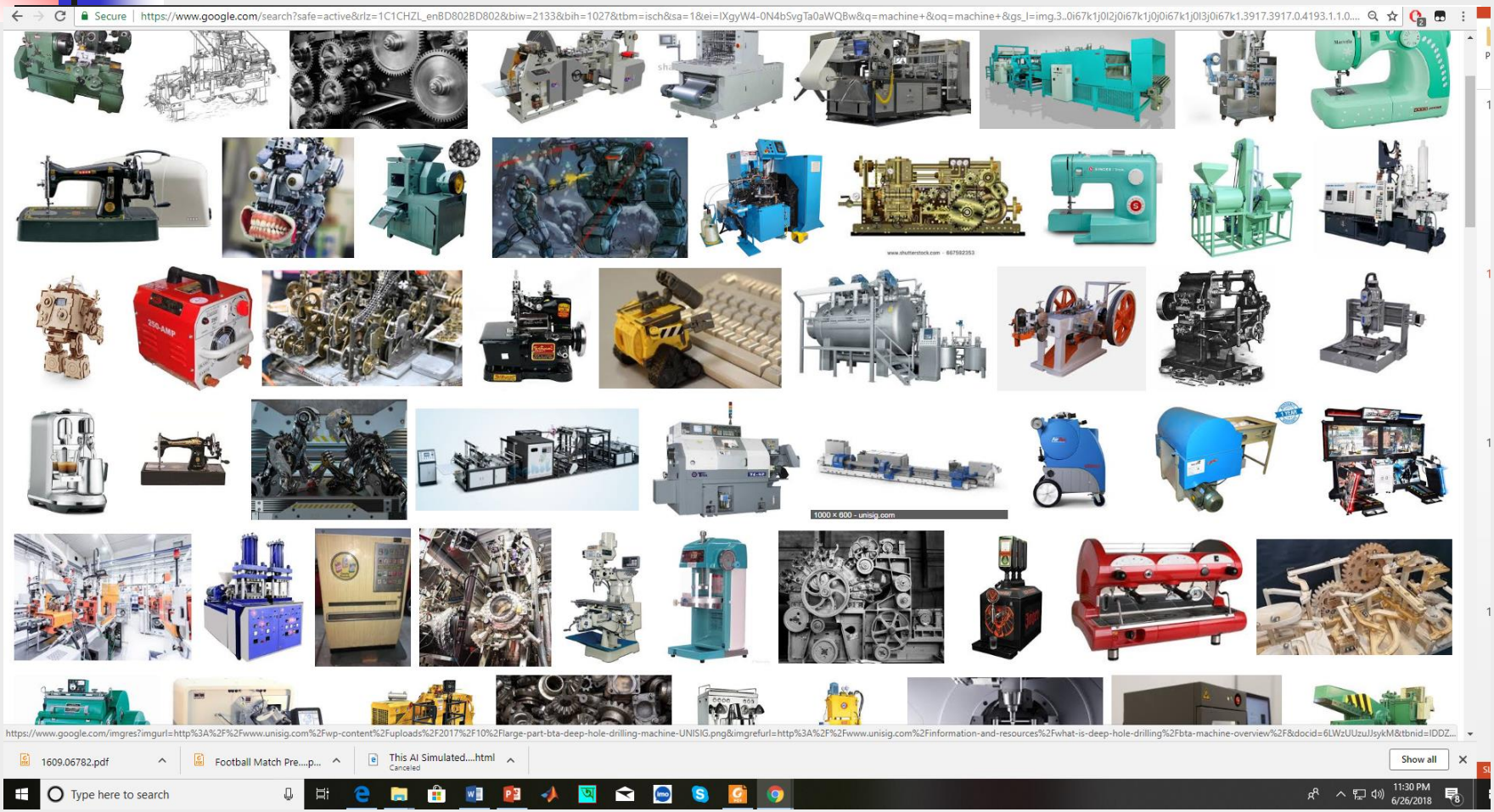
2.Mechanics: Device that makes mechanical work easier by overcoming a resistance (load) at one end by application of effort (force) at the other end.

BusinessDictionary.com



5

Machine



Machines include a system of mechanisms that shape the actuator input to achieve a specific application of output forces and movement.

How make decision to perform task(s)?

6

Machine Learning

In 1959, Arthur Samuel, a pioneer in the field of machine learning (ML) defined it as the "field of study that gives computers the ability to learn without being explicitly programmed".

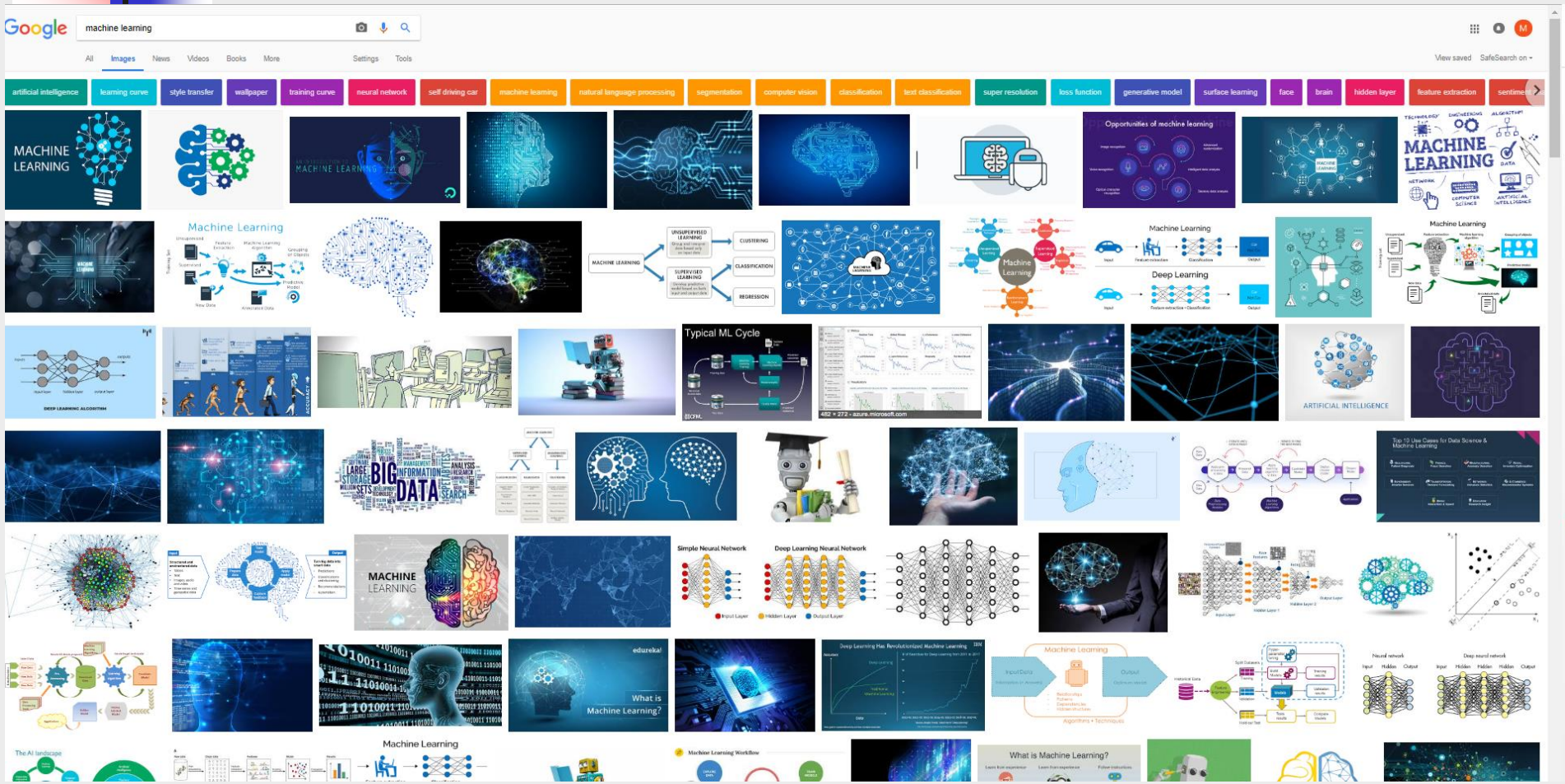
https://theconversation.com/what-is-machine-learning-76759

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data.[1] It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.[2] Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.[3]

https://en.wikipedia.org/wiki/Machine_learning

7

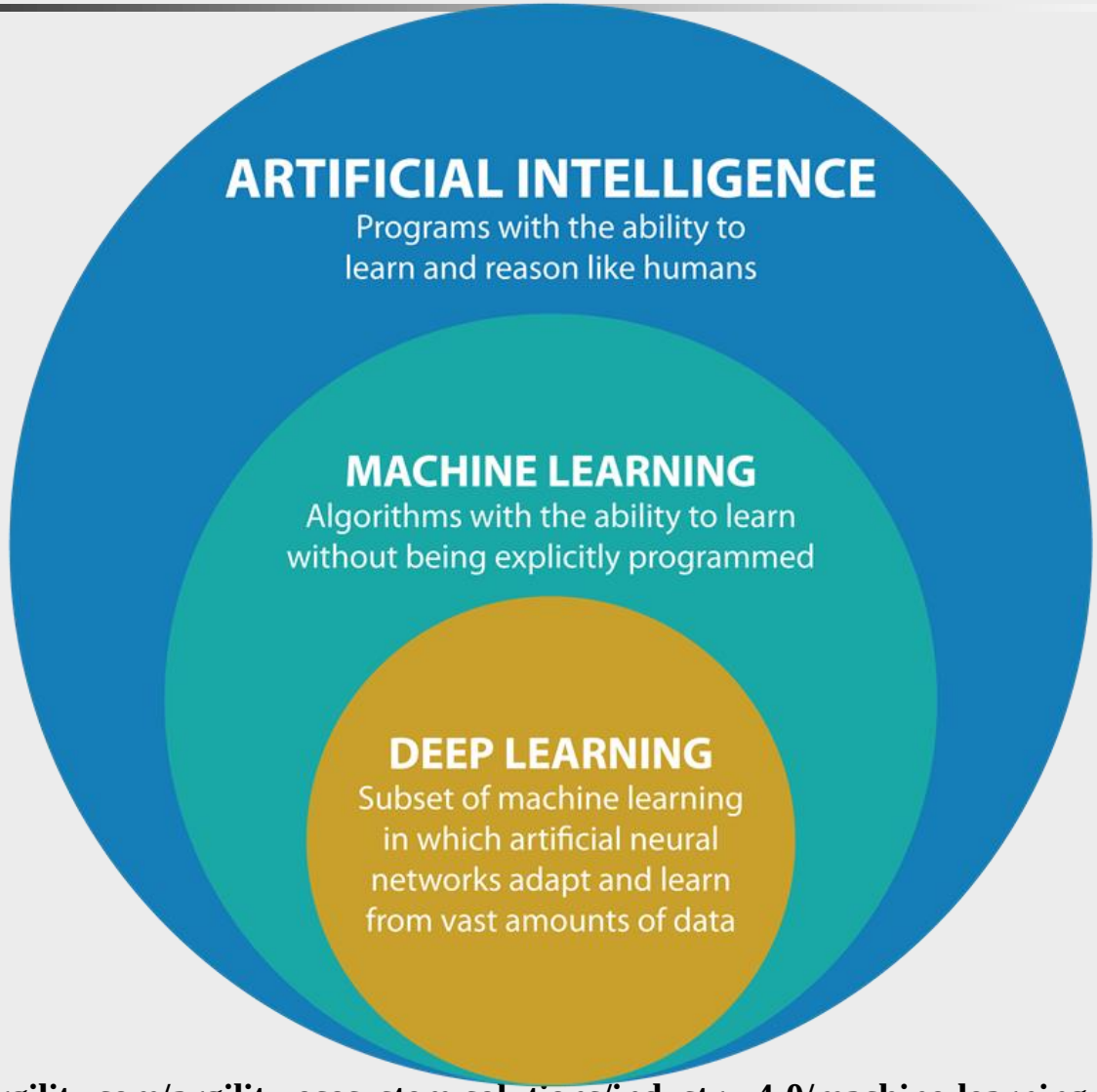
Machine Learning



Technique to give computer brain like learning ability through progressively update with data, without being explicitly programmed.

8

AI ->Machine Learning->Deep Learning



https://www.argility.com/argility-ecosystem-solutions/industry-4-0/machine-learning-deep-learning/

9

AI ->Machine Learning->Deep Learning

Artificial Intelligence (AI)

AI is the broadest term, applying to any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning).

Machine Learning

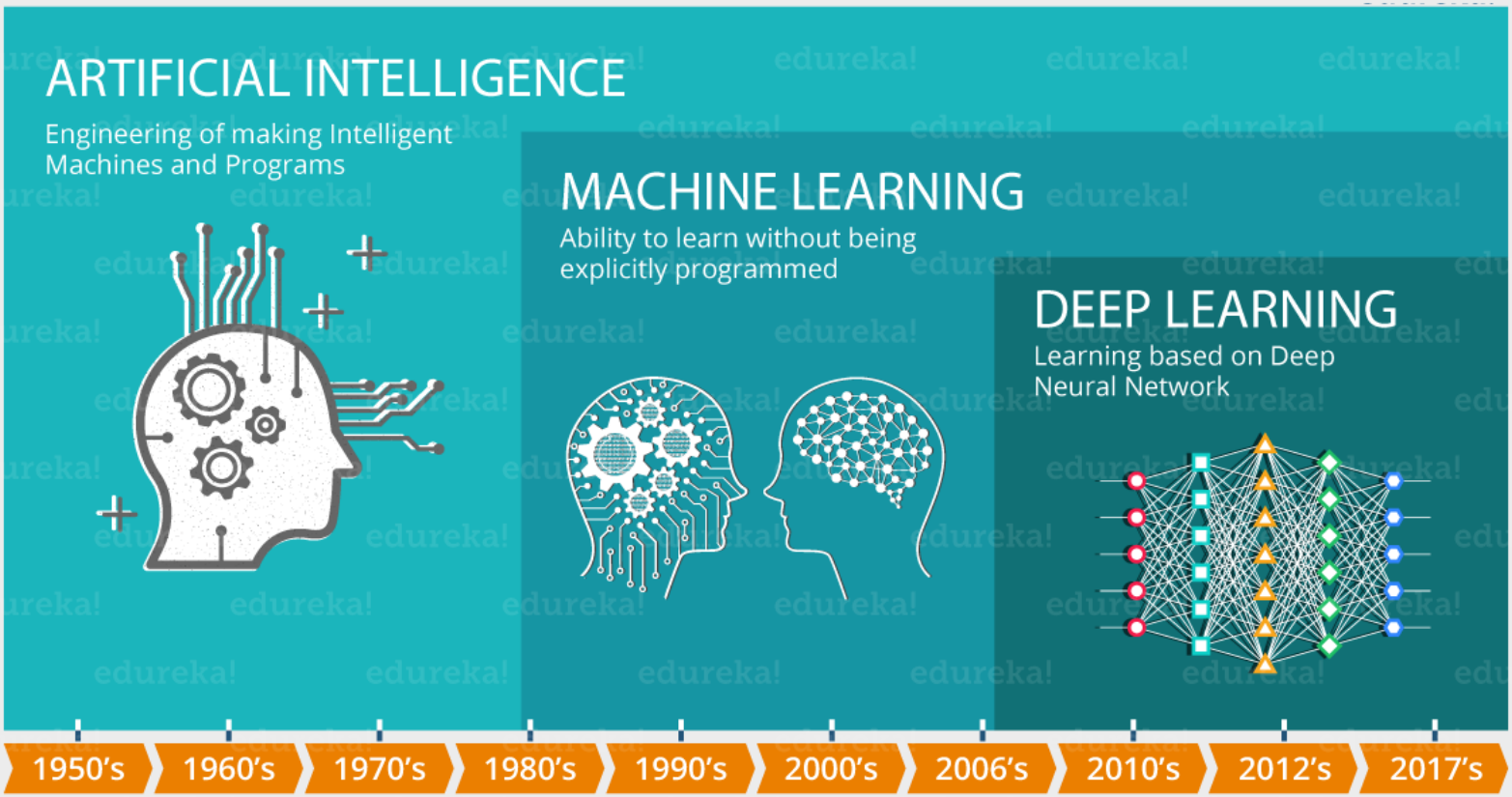
The subset of AI that includes that enable machines to improve at tasks with experience. The category includes deep learning.

Deep Learning

The subset of machine learning composed of algorithms to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data.

11

AI ->Machine Learning->Deep Learning



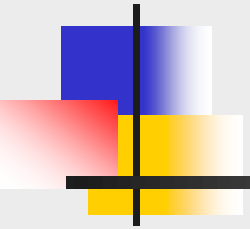
12

History of ML and Deep Learning

1906
Neuron/
Nerve Cell

Ramon y Cajal

https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html



Learning Related to Pattern Recognition

14

Learning in Pattern Recognition

- *What Does It Mean to Learn?*
- *What does it mean to have learned?*
- How do we recognize that an algorithm or a method, regardless of how simple or complex, has succeeded in learning?

The object being learned is a classifier *g*(*x*), some sort of black-box / function / algorithm / method / concept / system which takes a given feature *x* and returns the associated class *C*:

$$g(\underline{x}) \in \mathcal{C} = \{C_1, C_2, \ldots, C_K\}$$

We suppose that we are given a dataset $\mathcal{D}$ of N feature-class[1] pairs

$$\mathcal{D} = \{(\underline{x}_i, c_i)\} \qquad \text{where} \quad \underline{x}_i \in \mathbb{R}^n,\ c_i \in \mathcal{C},\ 1 \leq i \leq N$$

15

Learning in Pattern Recognition

The most primitive type of learning is just memorization, in which case we would consider *g*() to have learned from the given dataset based on the number of feature-class pairs it successfully reproduces:

$$\text{Correct Count} = \sum_i \delta\left(c_i, g(\underline{x}_i)\right) \qquad \delta(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$$

In principle, memorization is actually a credible approach to developing a classifier, however in general there are two significant limitations:

1. Memorizing is hard : memory concern

2. Memorizing is not enough: learning is not just to remember

#Generalize is more important to learning, to be able to reach correct conclusions about instances which you have *not* seen.

$$\underline{\hat{\theta}} = \arg_{\underline{\theta}} \max \sum_i \delta\left(c_i, g(\underline{x}_i, \underline{\theta})\right)$$

Learning in Pattern Recognition

True Class			Classification		
c_1	c_2	c_3	$g(\underline{x}_1) = \text{Dog}$	$g(\underline{x}_2) = \text{Dog}$	$g(\underline{x}_3) = \text{Cat}$
Dog	Dog	Dog	✓	✓	✗
Dog	Dog	Cat	✓	✓	✓
Dog	Cat	Dog	✓	✗	✗
Dog	Cat	Cat	✓	✗	✓
Cat	Dog	Dog	✗	✓	✗
Cat	Dog	Cat	✗	✓	✓
Cat	Cat	Dog	✗	✗	✗
Cat	Cat	Cat	✗	✗	✓
			50%	50%	50%

← Average Performance

Fig. 3.1. NO FREE LUNCH: There is no objectively best classifier, averaged over all possible outcomes all classifiers perform the same as random guessing. A given classifier $g()$ is asked to classify three previously-unseen features $\underline{x}_1, \underline{x}_2, \underline{x}_3$ having true associated classes c_1, c_2, c_3, where the features are classified into two possible classes of *Cat* and *Dog*. Averaged over all possible truths (left), the classifier $g()$ does no better than random guessing. Indeed, averaged over all truths, *every* classifier will perform the same.

Averaged over all possibilities for the unseen data, no classifier generalizes better than any other!

17

Robustness in Learning

$$g(x, \underline{\theta}) = \theta_p x^p + \theta_{p-1} x^{p-1} + \ldots + \theta_1 x^1 + \theta_0 x^0.$$

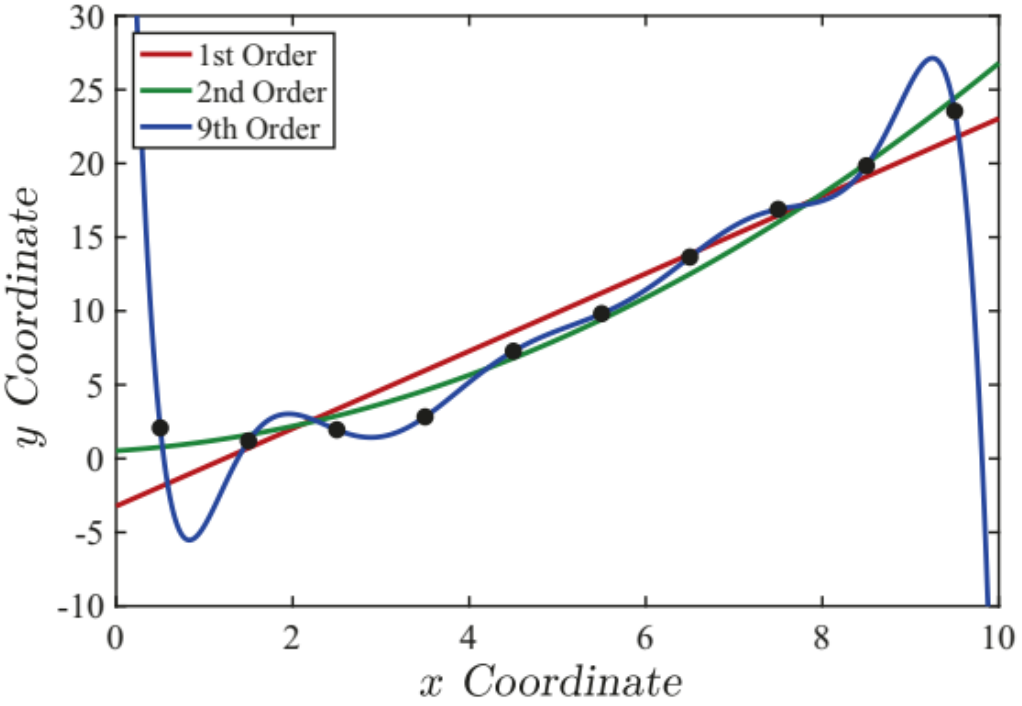
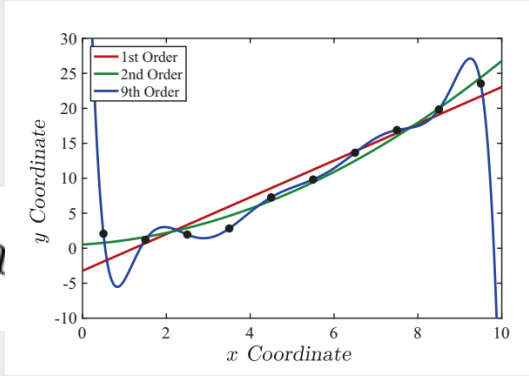


Fig. 3.2. What does it mean to *fit* a model to data? Here we have ten data points (black dots), which come from some unknown model with added random noise. We fit polynomials to the points, for which the first-order (linear regression), second-order (parabolic regression), and ninth-order fits are shown. A *p*th-order polynomial can always be found that passes through *p* + 1 given points, so here the ninth-order polynomial fits the points exactly, however it seems like a very unlikely generalization of the data.

Robustness in Learning

18

$$g(x, \underline{\theta}) = \theta_p x^p + \theta_{p-1} x^{p-1} + \ldots + \theta_1 x$$



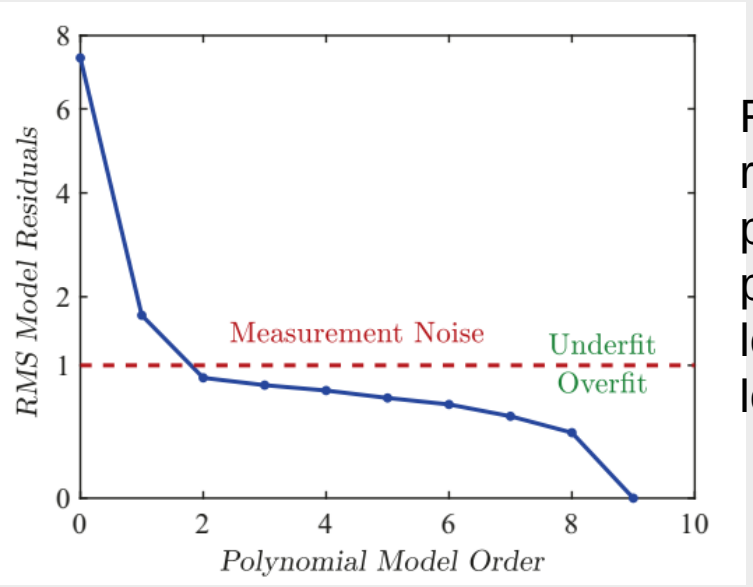


Fig. 3.3. Following up on Figure 3.2, we can plot the root-mean-square (RMS) difference between the polynomial fit and the data points, as a function of the polynomial order. In this case the measurement noise level was assumed to have been provided, it was not learned.

$$\mathrm{RMS}(\underline{\theta}) = \left(\frac{1}{N} \sum_{i=1}^{N} \left(y_i - g(x_i, \underline{\theta}) \right)^2 \right)^{1/2},$$

In this example the added noise has a standard deviation of σ = 1, meaning that for the correct model, RMS(θexact) = 1. As a result,

- Any RMS difference below σ must be overfitting, meaning that g(x; θ) is partly fitting to noise, by taking some of the noise into account when learning θ;
- Any RMS difference above σ suggests that the learned model has not adequately generalized, or has not been given adequate flexibility in θ (enough degrees of freedom q) to capture the variations that need to be captured.

19

Robustness in Learning

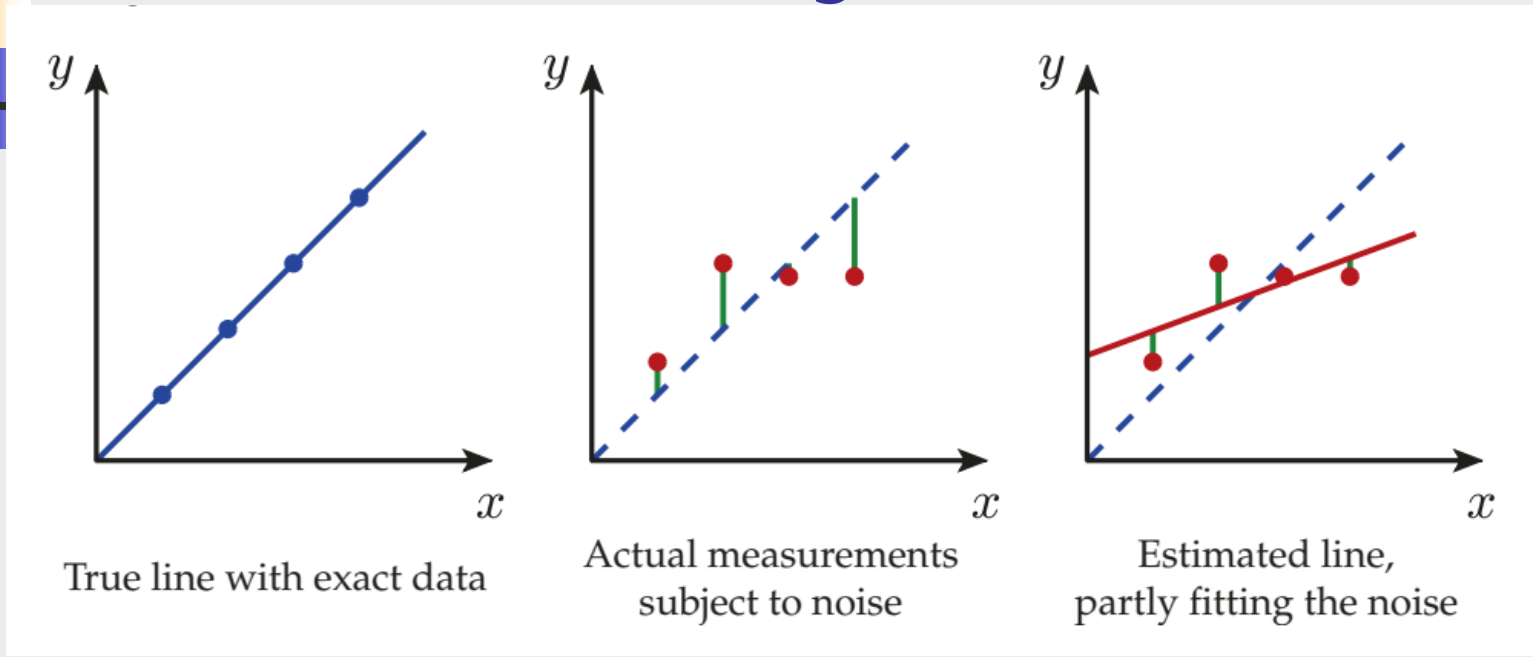
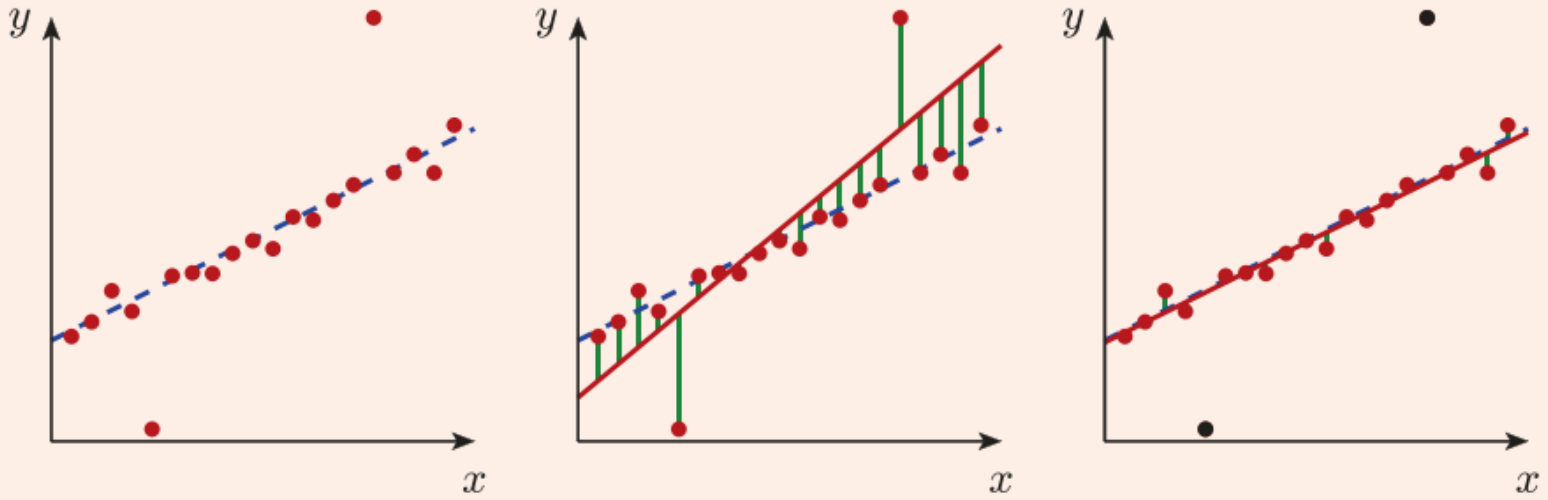


Fig. 3.4. Overfitting: Any learning, whether of linear regression (here) or a pattern recognition classifier (Figure 3.5), is said to be overfitting if it begins to tune its parameters to the behaviour of the noise, rather than of the underlying phenomenon we wish to learn. The estimated line (red) is quite plausible, given the four data points (red dots), however it is clear how the line has accommodated (fit) the noise, to make the residuals (green) smaller

20

Robustness in Learning



Actual measurements subject to noise having two outliers

Estimated line based on the RMS criterion of (3.8) (Not Outlier Robust)

Estimated line ignoring the two outliers (Outlier Robust)

So how do we make learning *robust* to outliers? Really this is a vast topic, which we can only begin to touch here. A variety of approaches is possible:

1. Detect and remove the outliers (as was done in the right-most panel, above),
2. Choose parameters in $\underline{\theta}$ insensitive to outliers, or
3. Choose an optimization metric insensitive to outliers.

21

Robustness in Learning

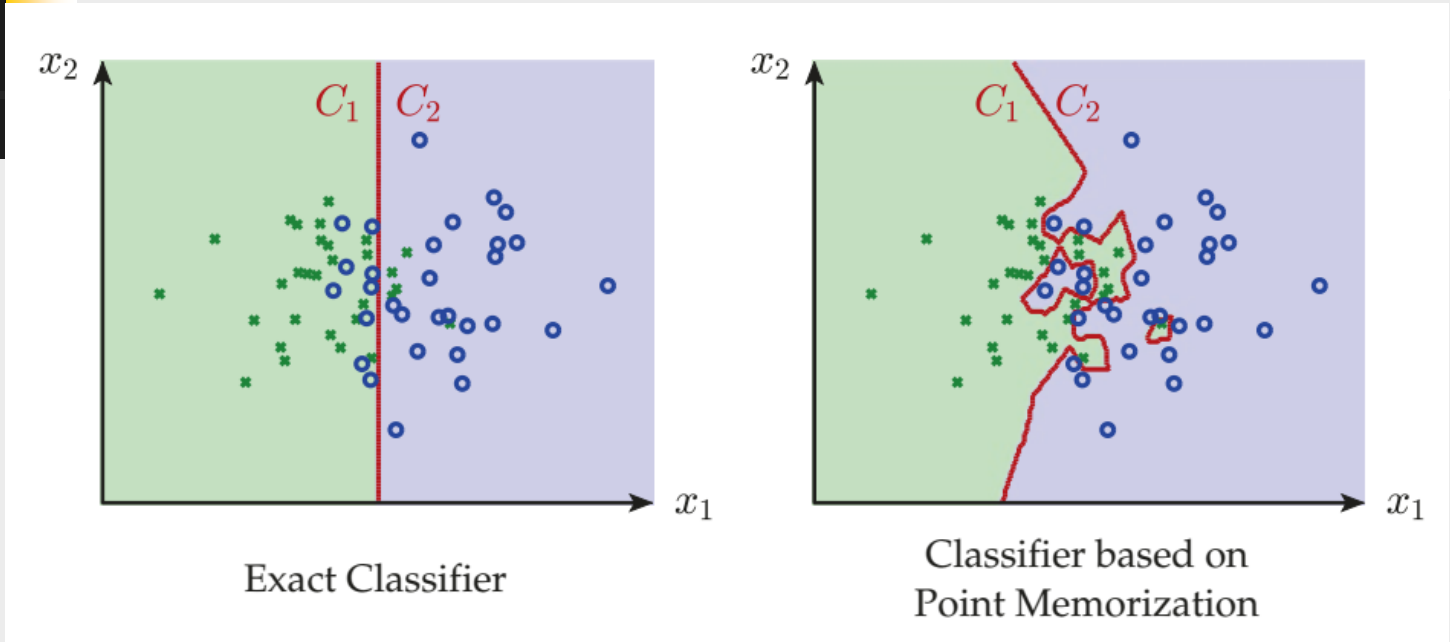
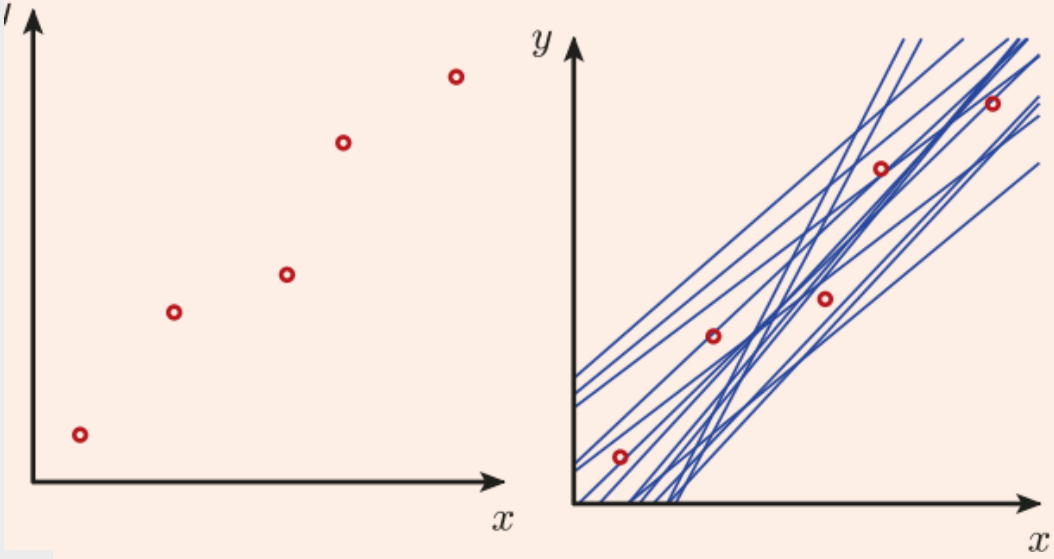


Fig. 3.5. OVERFITTING: As in Figure 3.4, but here for a pattern recognition classifier. We will have to wait for Chapter 6 for the details of the classifier to be discussed, however the principle is the same as in regression: Any learning is overfitting if it tunes its parameters to the behaviour of the noise. That tuning is obvious here, in that the memorized classifier (right) is tuning its decision (coloured background) on the basis of individual training points, significantly distracted from the correct or ideal classification (left).

22

Regression and Classification

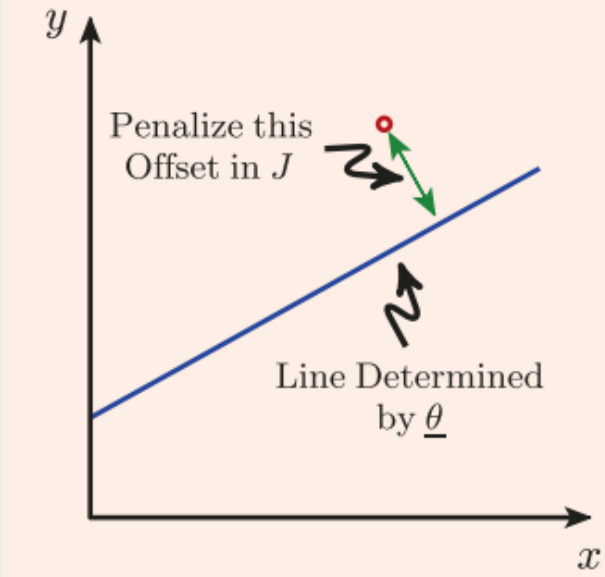


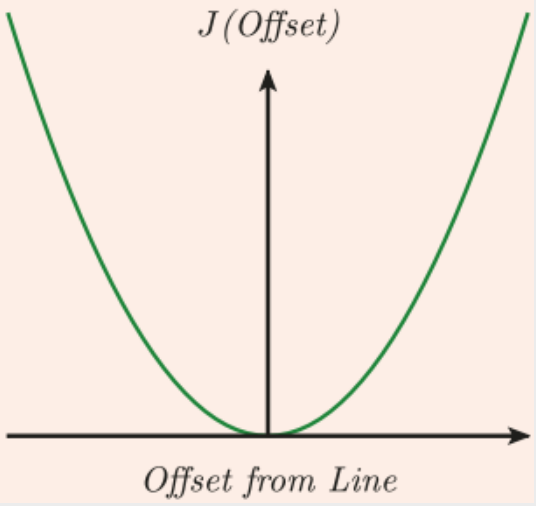
For each possible line, described by

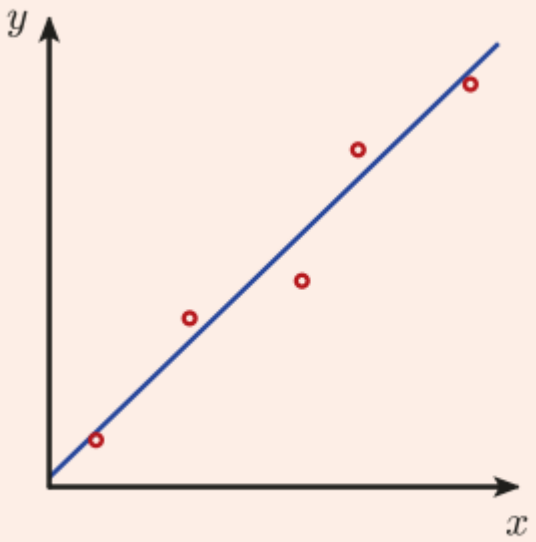
$$\underline{\theta} = \begin{bmatrix} \text{Angle of Line} \\ \text{Y-Intercept of Line} \end{bmatrix}$$

we can assess the penalty associated with the
as

$$\text{Penalty}(\underline{\theta}) = \sum_i J\big(\text{Offset from line } \underline{\theta} \text{ to } (x_i,$$



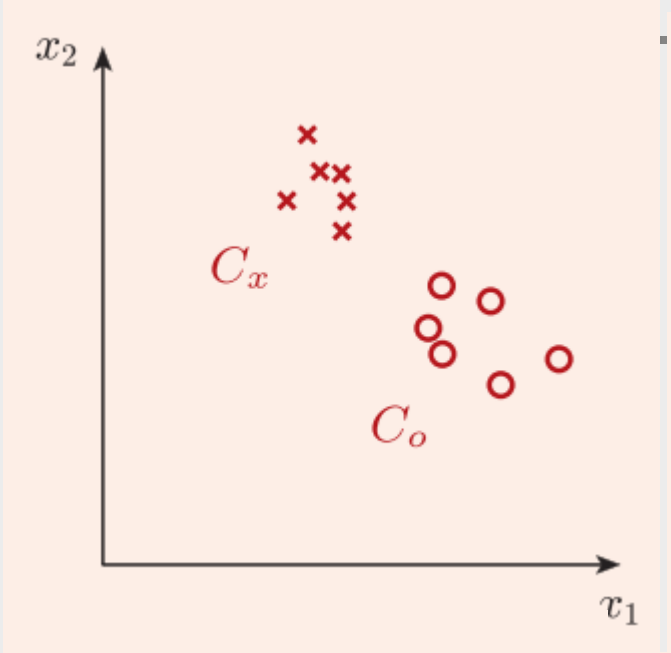


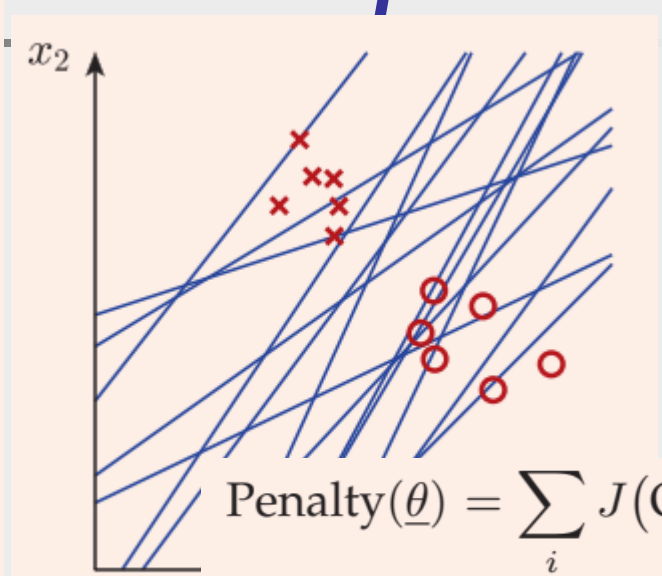


Optimal Line

23

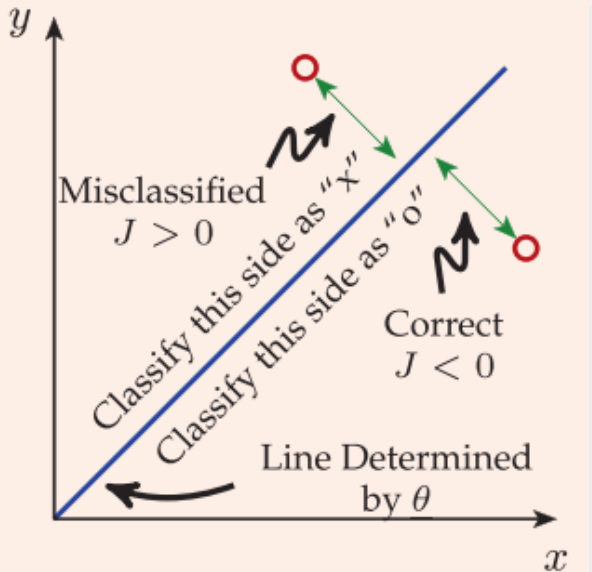
Rearession and Classification

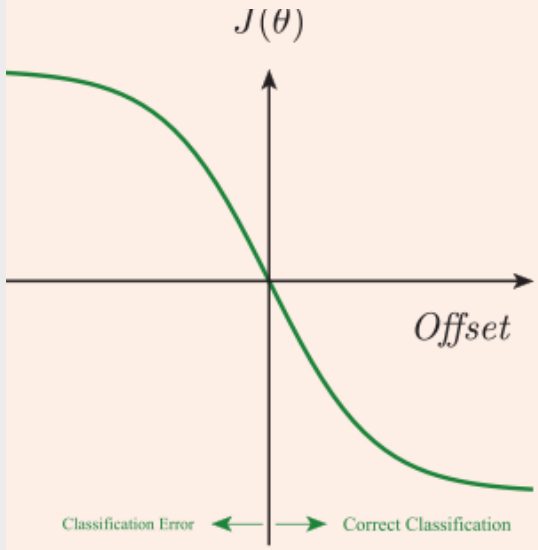


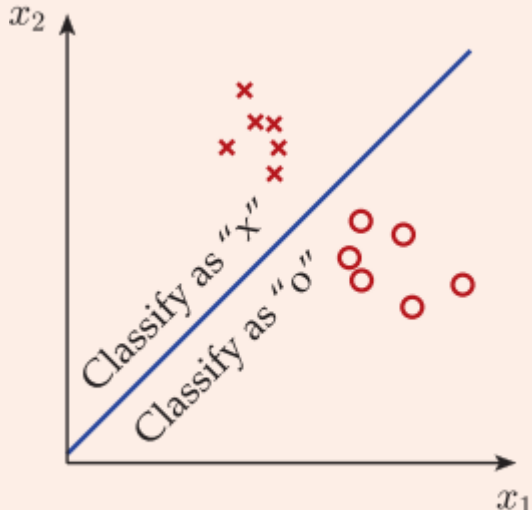


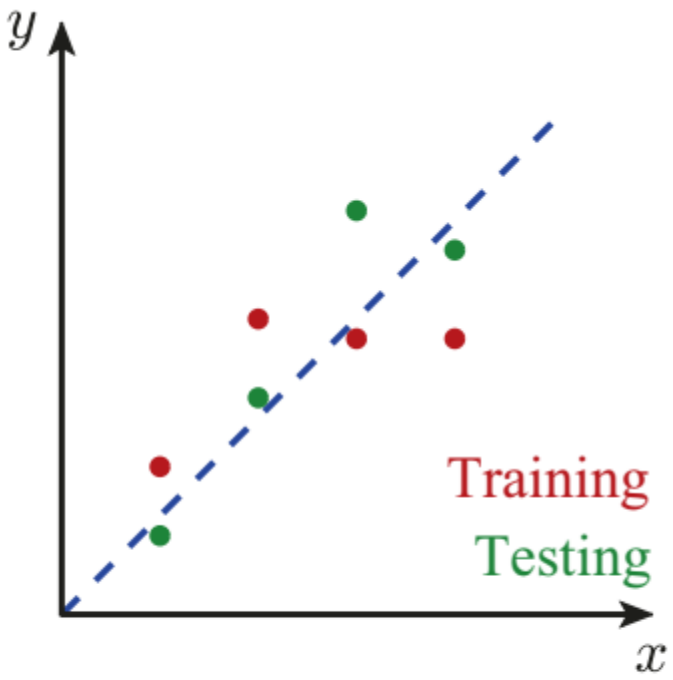
$$\text{Penalty}(\underline{\theta}) = \sum_i J\left(\text{Offset from line } \underline{\theta} \text{ to } \underline{x}_i \in C_x\right)$$

$$+ \sum_j J\left(\text{Offset from line } \underline{\theta} \text{ to } \underline{x}_j \in C_o\right)$$

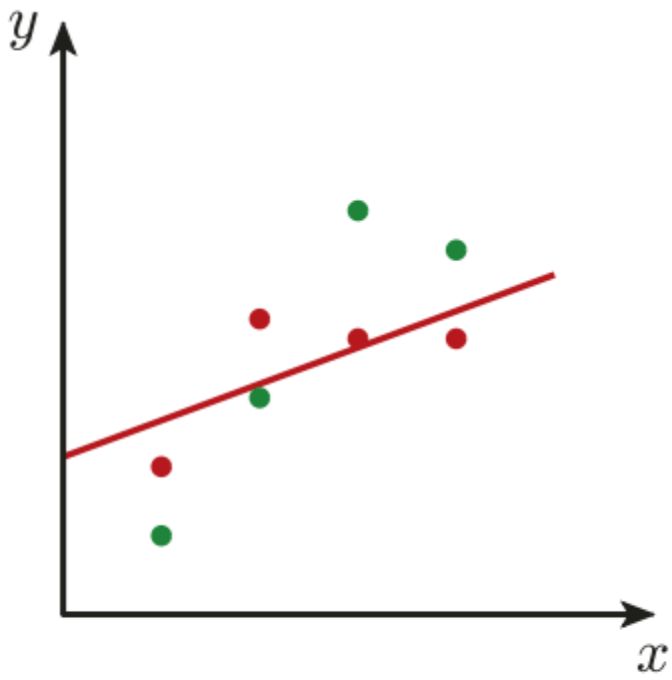








Training and testing data



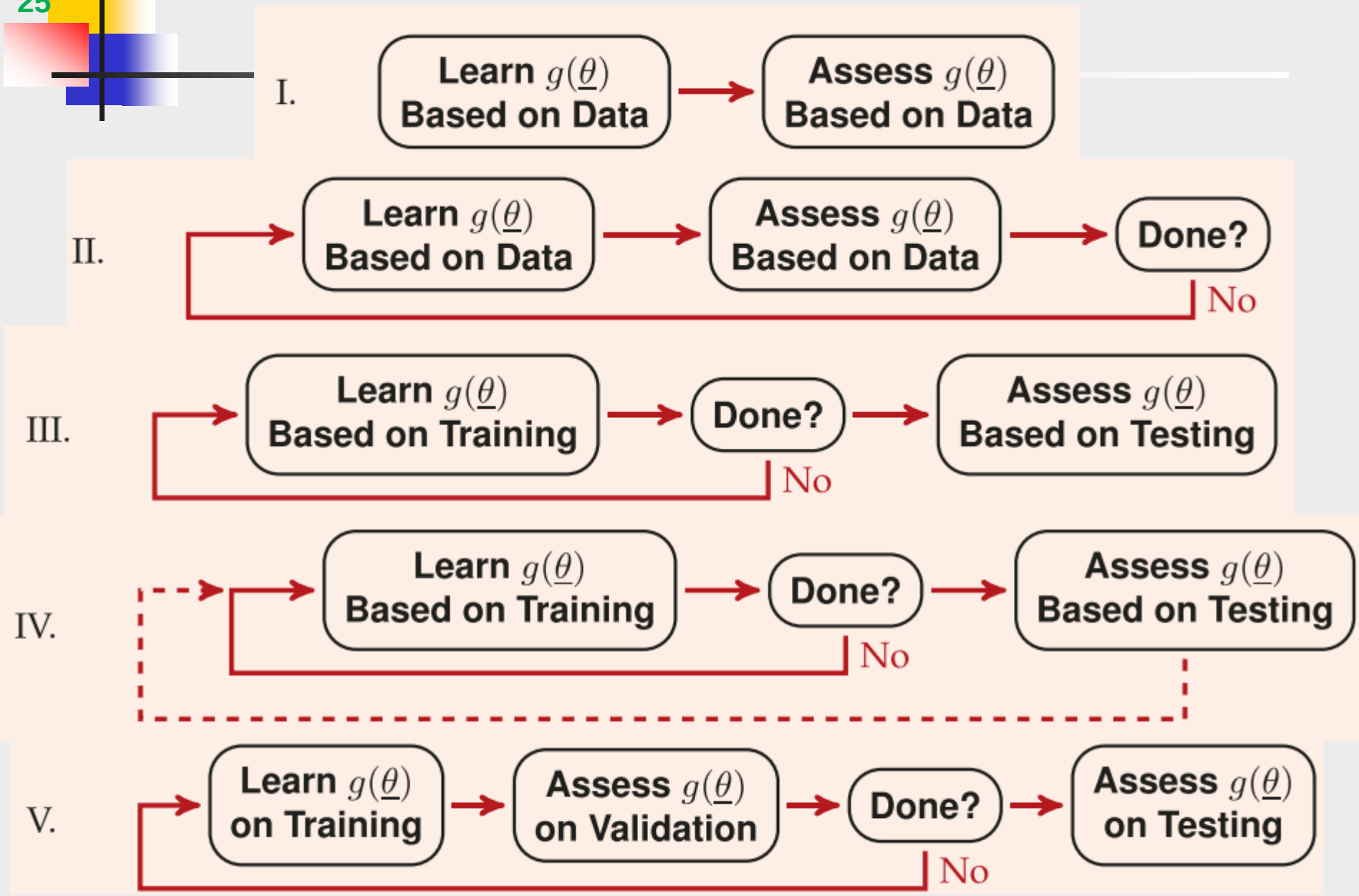
Estimated line

$\text{StdDev}(\text{Training Residual}) = 0.7$

$\text{StdDev}(\text{Testing Residual}) = 1.2$

Fig. 3.7. We have separate training and testing data (left), such that the learned model (red line) is deduced from the training data, but assessed against the testing data. Observe the degree to which the estimated line fits to the noise, based on the difference between the fit to training data (overfit, under-reporting model inconsistency) and the fit to testing data (which is an accurate, objective assessment).

Use of Data in Learning



26

Classifier Evaluation / Performance Measure

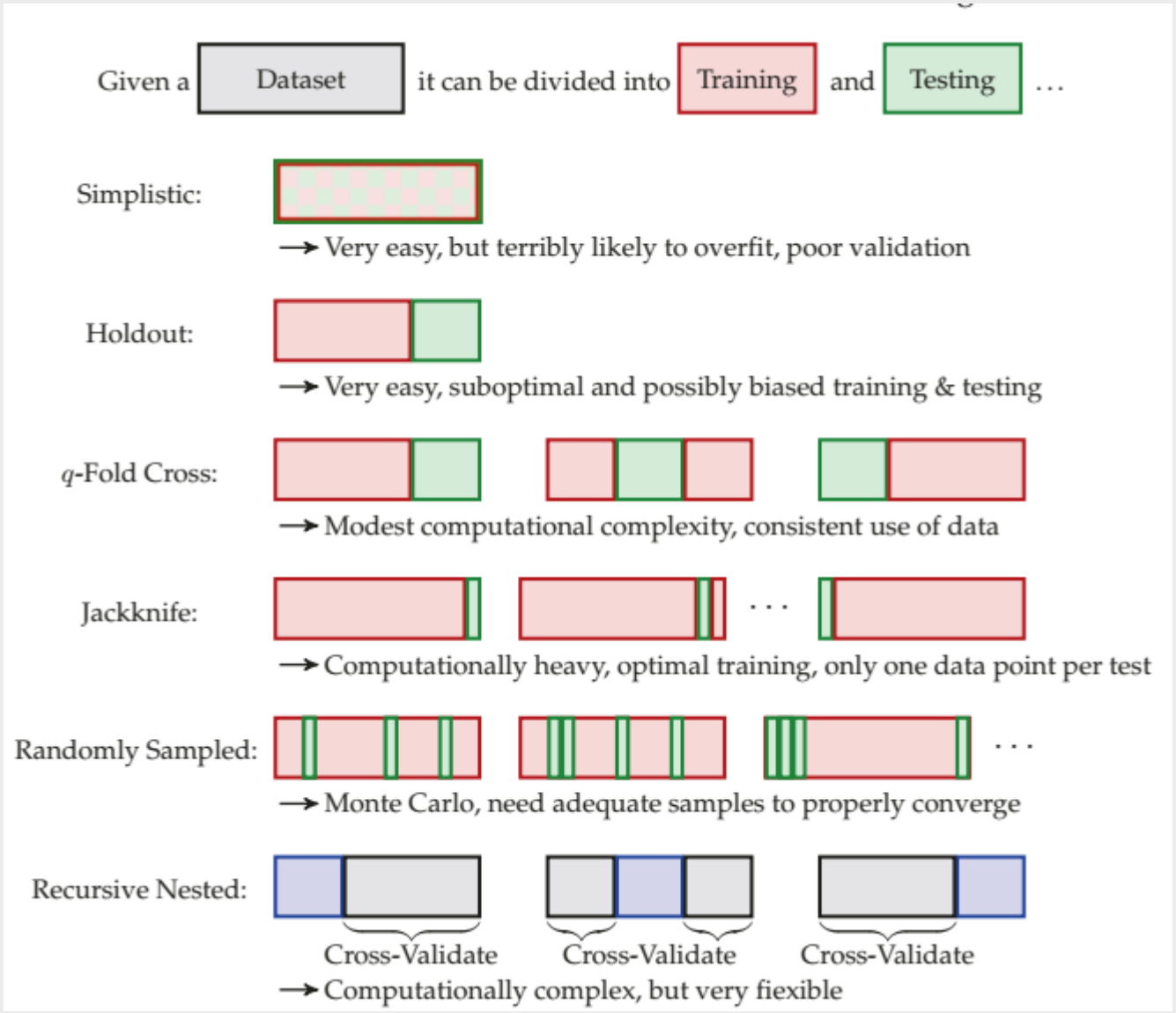
1. **Test Set Accuracy**
2. **Test Set Error Rate**
3. **Confusion Matrix**

Given a classifier g, the corresponding confusion matrix from (3.19),

True Class \ Is classified as …	C_1	C_2	$\cdots$	C_K
C_1	$S_g(C_1\|C_1)$	$S_g(C_2\|C_1)$	$\cdots$	$S_g(C_K\|C_1)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
C_K	$S_g(C_1\|C_K)$	$S_g(C_2\|C_K)$	$\cdots$	$S_g(C_K\|C_K)$

Classifier Validation

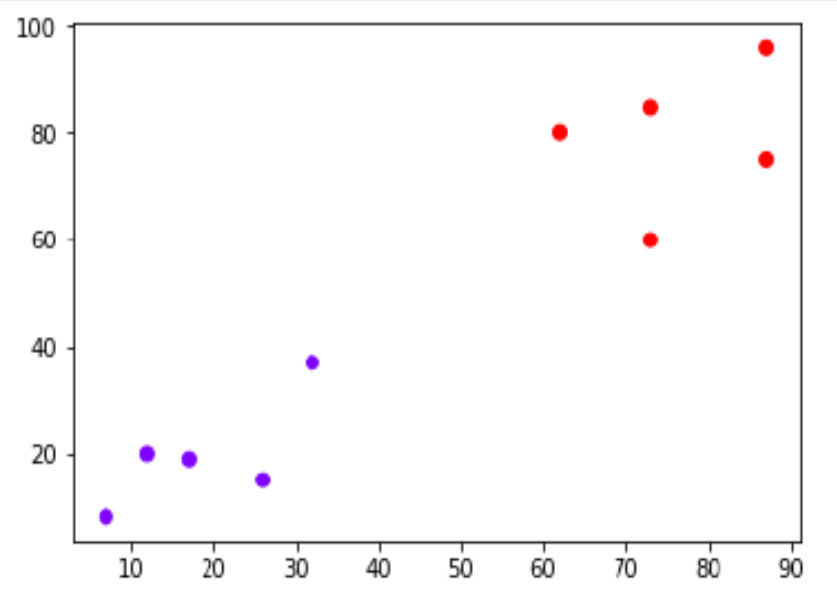
27

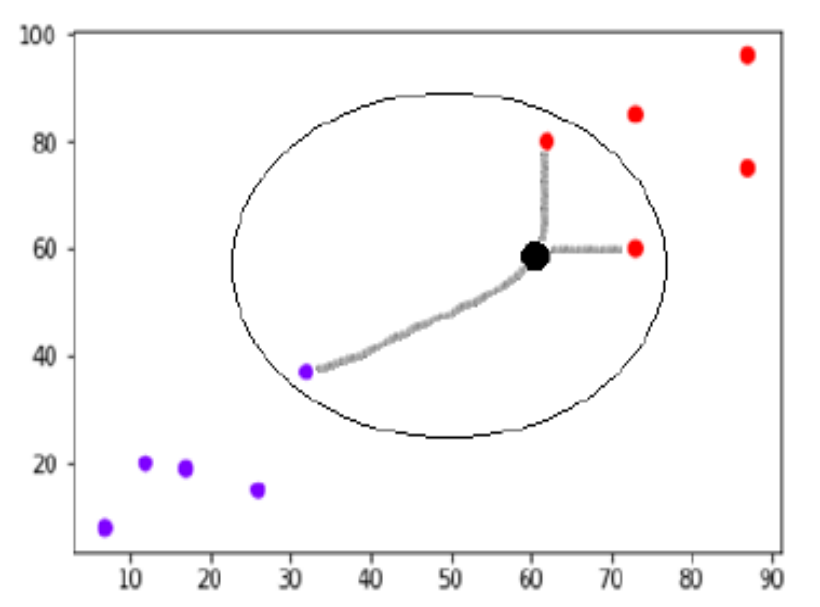


28

K-nearest neighbors (KNN) Classifier

- **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.



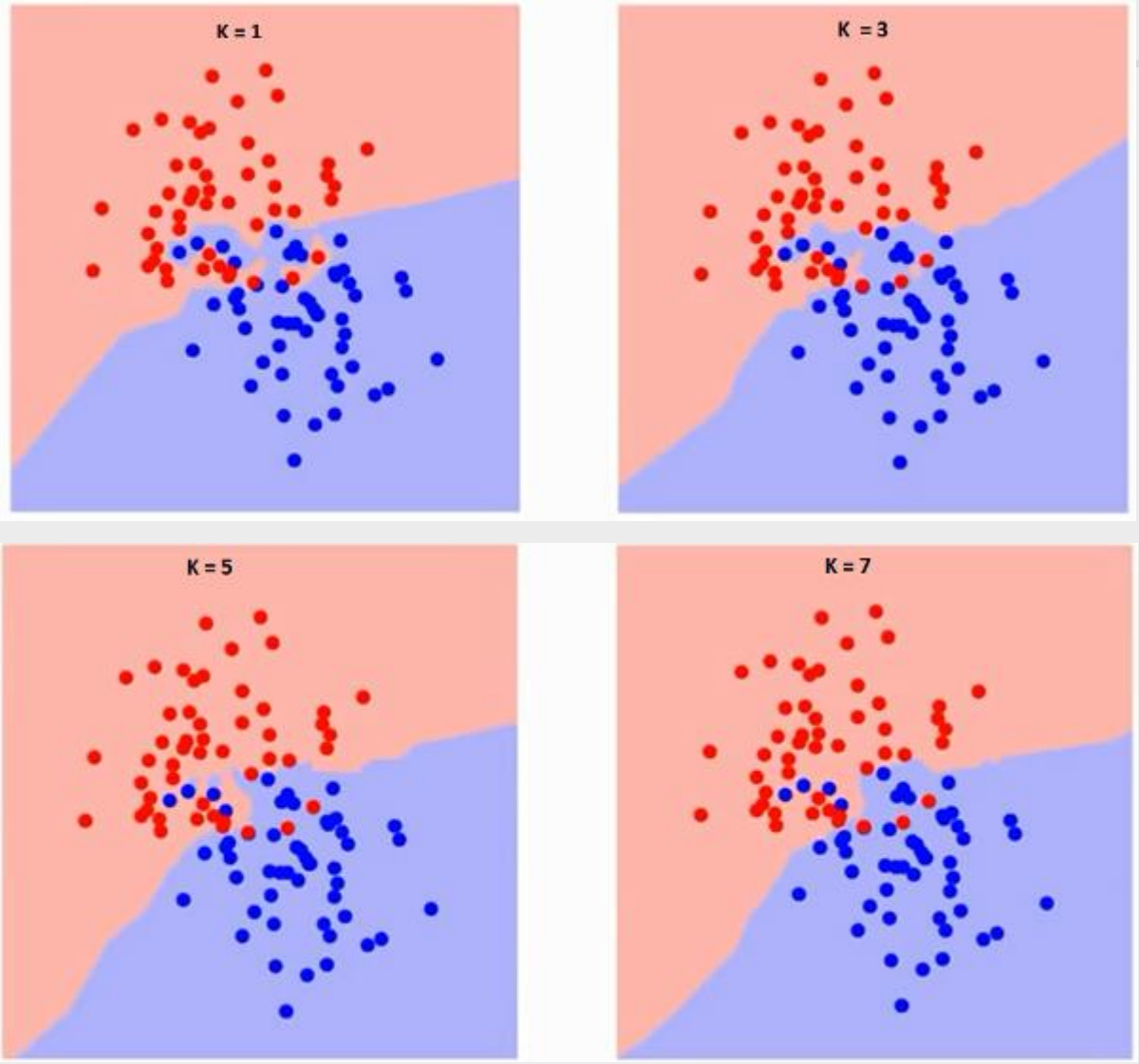


https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm

29

K-nearest neighbors (KNN) Classifier - The Simplest Classifier

30 K-nearest neighbors (KNN) Classifier



The boundary becomes smoother with increasing value of K.

With K increasing to infinity it finally becomes all blue or all red depending on the total majority.

https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/#h-what-is-knn-k-nearest-neighbor-algorithm

31 K-nearest neighbors (KNN) Classifier

Pros of KNN

- Very simple algorithm to understand and interpret.
- Very useful for nonlinear data because there is no assumption about data in this algorithm.
- Versatile algorithm as we can use it for classification as well as regression.
- High accuracy but there are much better supervised learning models than KNN.

Cons of KNN

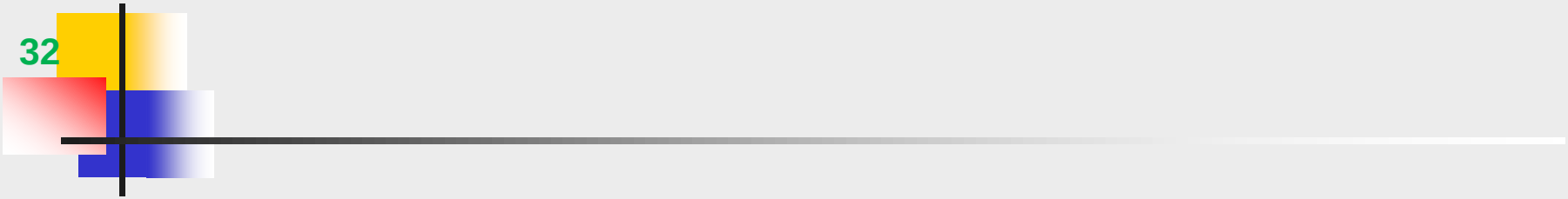
- Computationally a bit expensive algorithm because it stores all the training data.
- High memory storage required as compared to other supervised learning algorithms.
- Prediction is slow in case of big N.
- Very sensitive to the scale of data as well as irrelevant features.

Applications of KNN

Banking System: to predict weather an individual is fit for loan approval? Does that individual have the characteristics similar to the defaulters one?

Calculating Credit Ratings: can be used to find an individual's credit rating by comparing with the persons having similar traits.

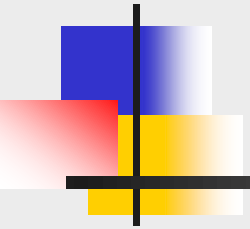
Other areas in which KNN algorithm can be used are Speech Recognition, Handwriting Detection, Image Recognition and Video Recognition.



Thanks for your attention

Question and Answer

MCSE 666:Pattern and Speech Recognition

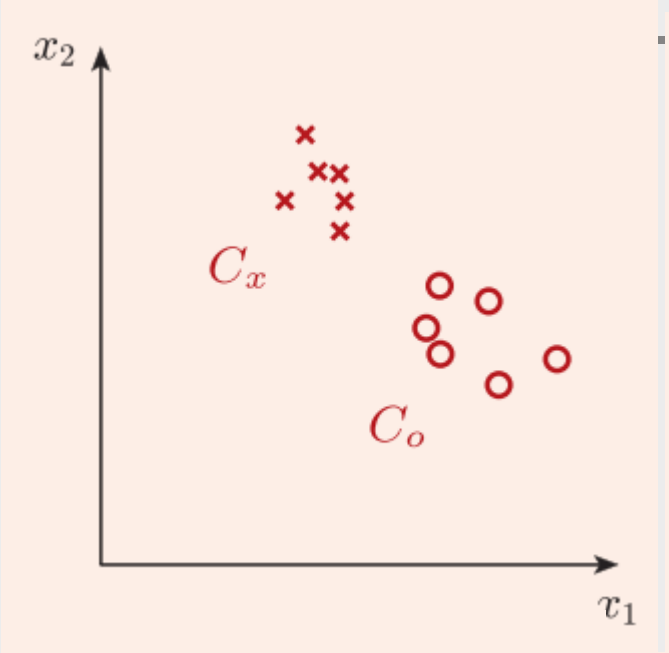


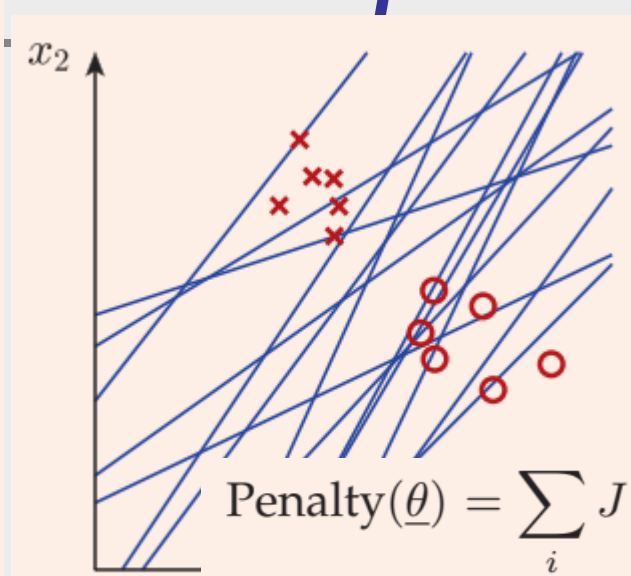
Classifier Learning

(K-Nearest Neighbors and Naive Bayes)

Dr. Md. Aminul Haque Akhand
Dept. of CSE, SUB

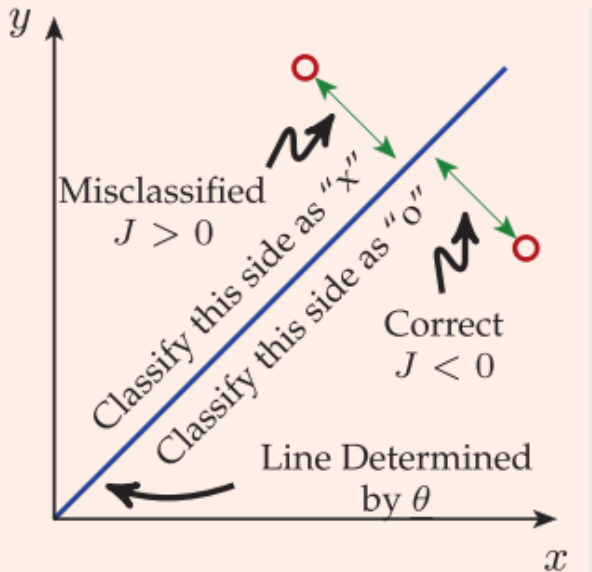
2 Rearession and Classification

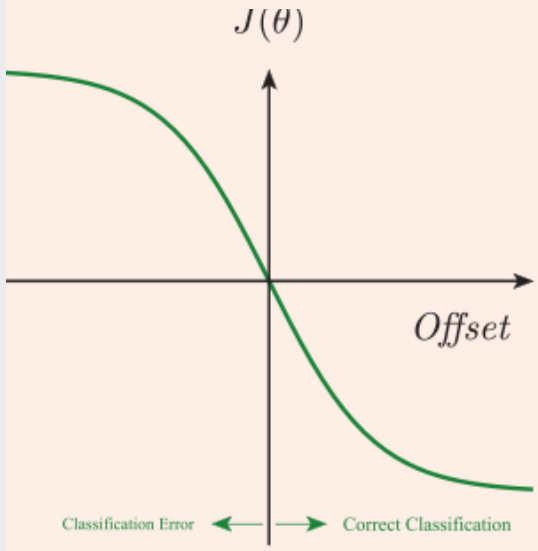


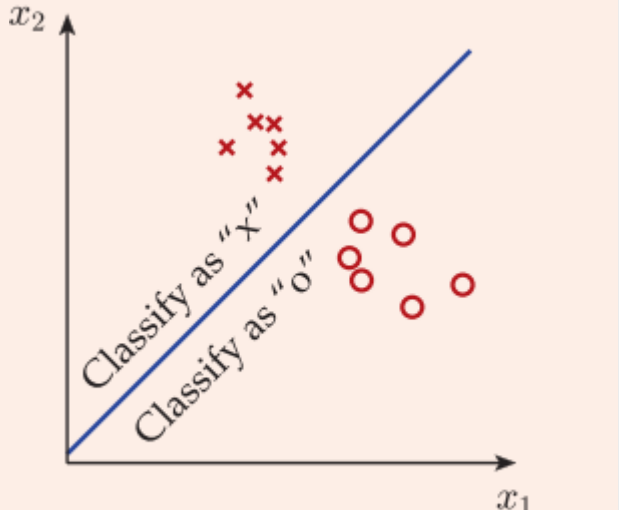


$$\text{Penalty}(\underline{\theta}) = \sum_i J(\text{Offset from line } \underline{\theta} \text{ to } \underline{x}_i \in C_x)$$

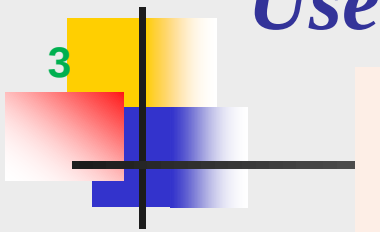
$$+ \sum_j J(\text{Offset from line } \underline{\theta} \text{ to } \underline{x}_j \in C_o)$$







Use of Data in Learning



Classifier Evaluation / Performance Measure

4

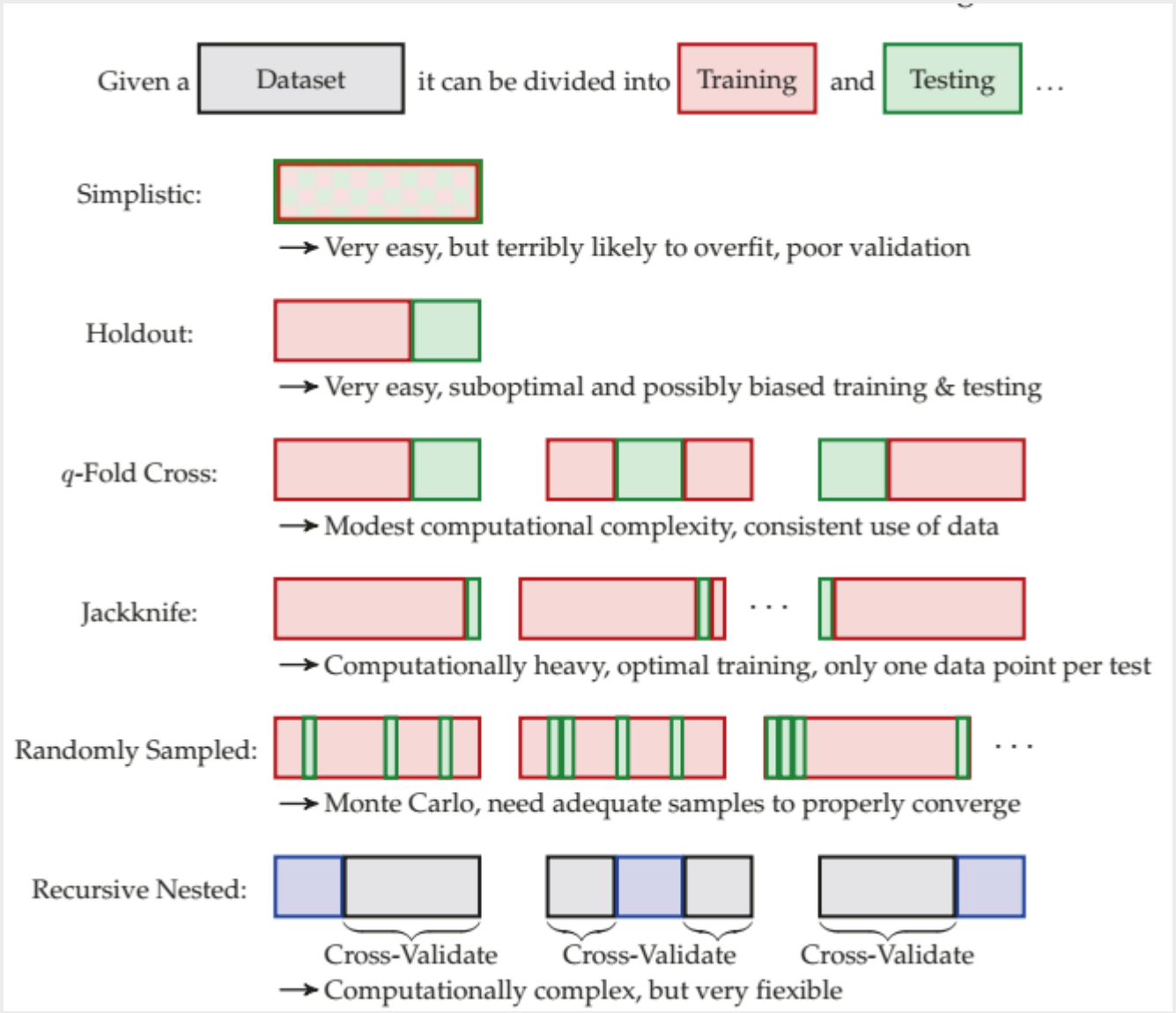
1. Test Set Accuracy
2. Test Set Error Rate
3. Confusion Matrix

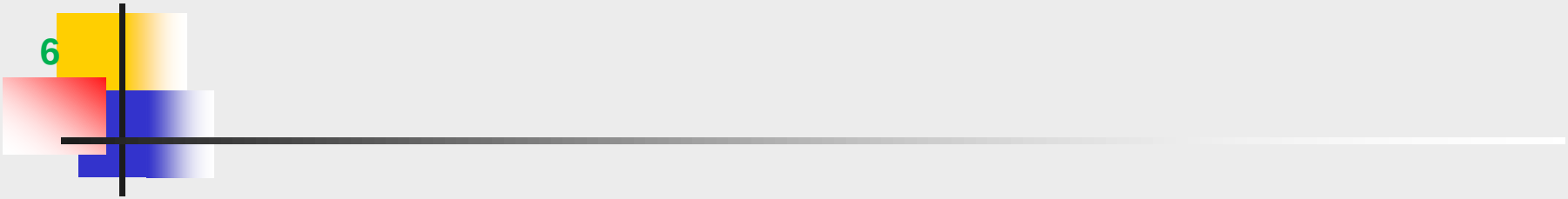
Given a classifier g, the corresponding confusion matrix from (3.19),

True Class \ Is classified as …	C_1	C_2	$\cdots$	C_K
C_1	$S_g(C_1\|C_1)$	$S_g(C_2\|C_1)$	$\cdots$	$S_g(C_K\|C_1)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
C_K	$S_g(C_1\|C_K)$	$S_g(C_2\|C_K)$	$\cdots$	$S_g(C_K\|C_K)$

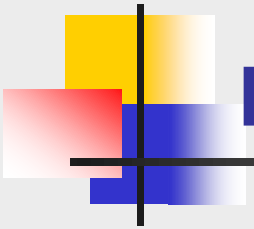
Classifier Validation

5





K-Nearest Neighbor (KNN) Classifier



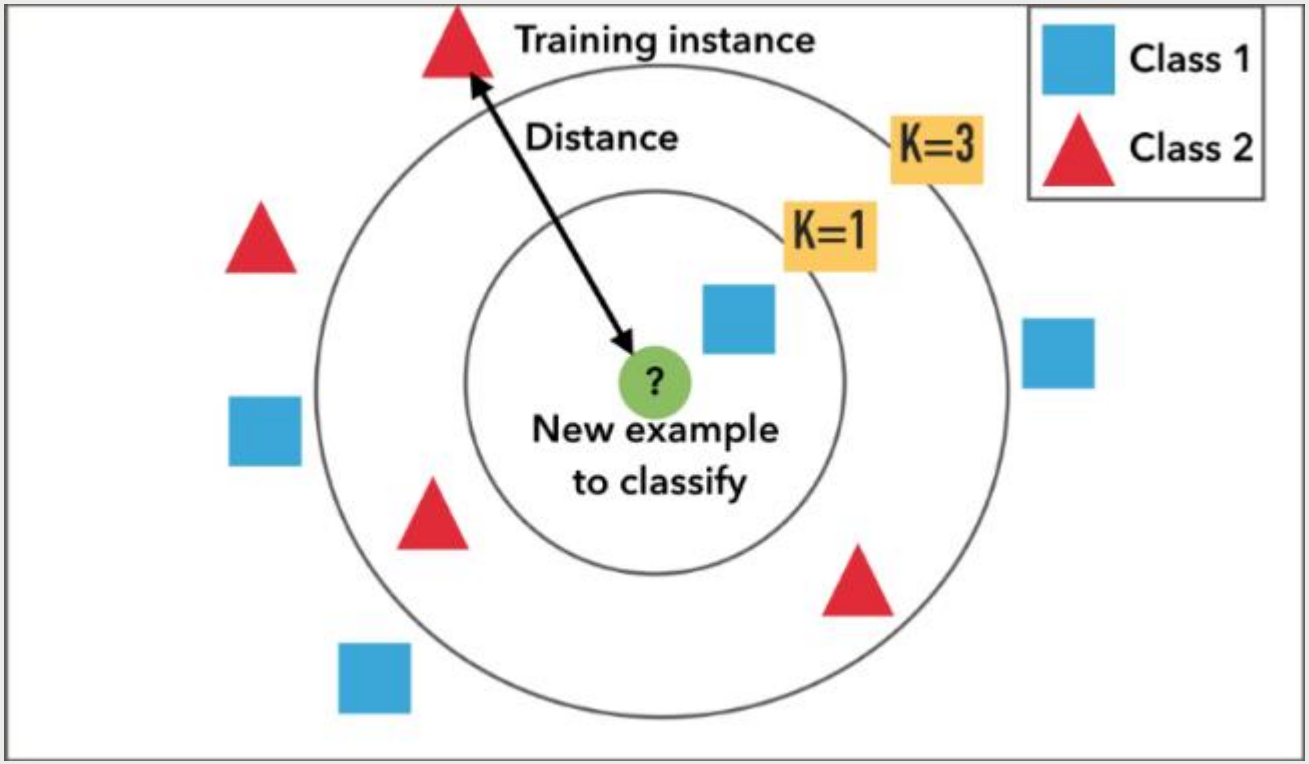
KNN Overview

- *KNN algorithm is one of the simplest classification algorithm*
- *Non-parametric*

  It does not make any assumptions on the underlying data distribution
- *Lazy learning algorithm.*
  - *there is no explicit training phase or it is very minimal.*
  - *also means that the training phase is pretty fast .*
  - *Lack of generalization means that KNN keeps all the training data.*
- *Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.*

KNN

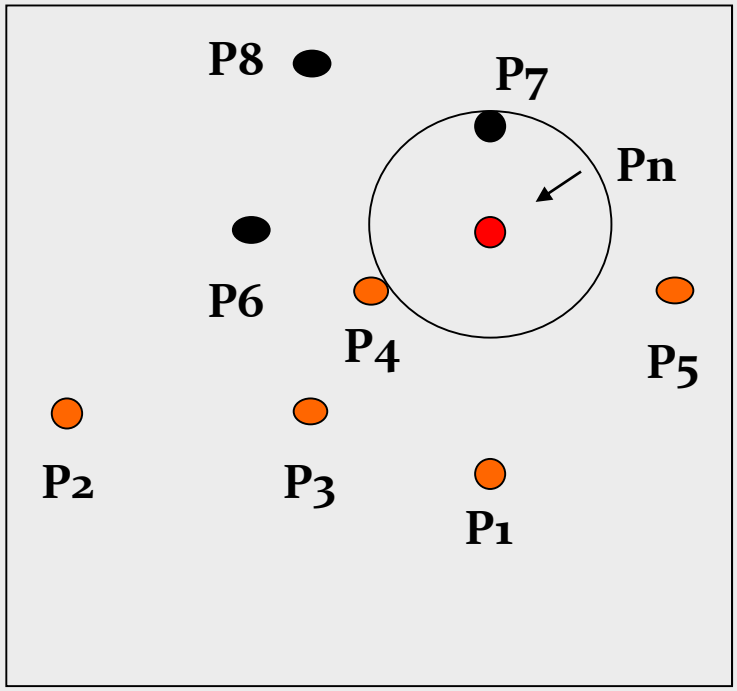
- *KNN Algorithm is based on feature similarity*
- *How closely out-of-sample features resemble our training set determines how we classify a given data point*



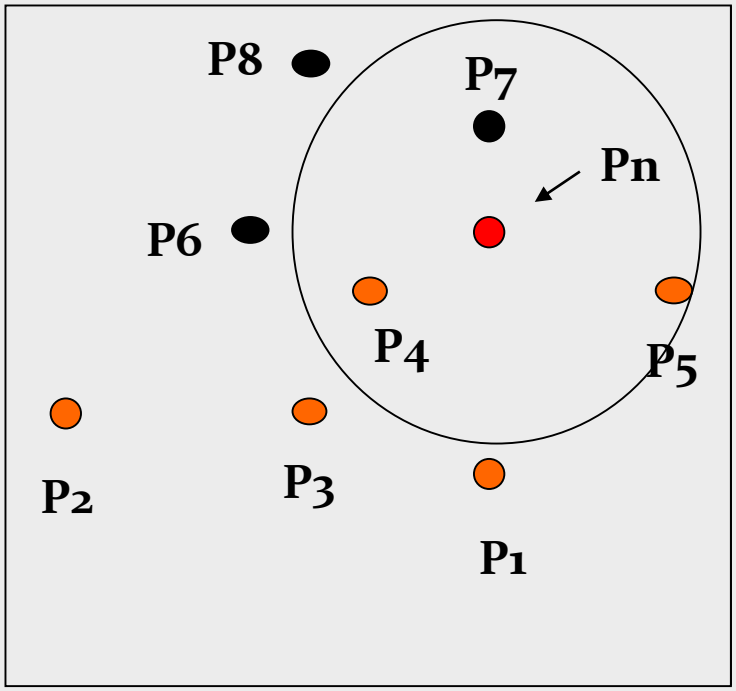
9

KNN for K = 1 or K = 3

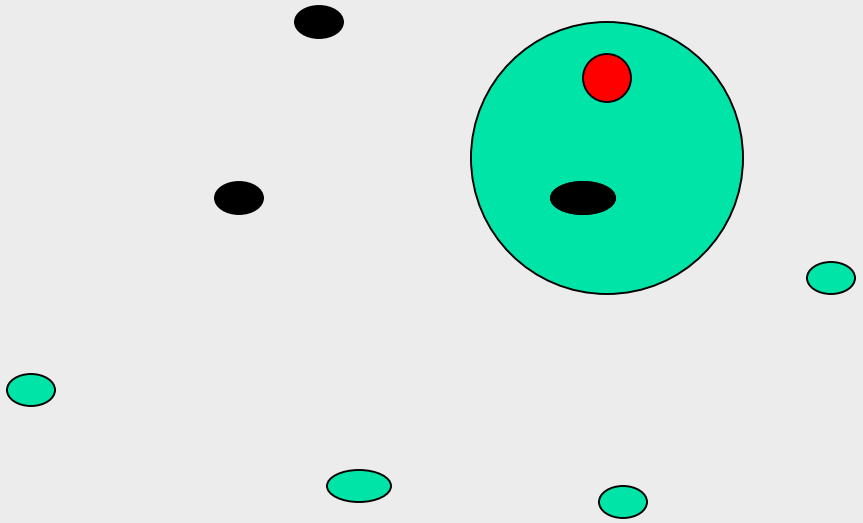
k = 1



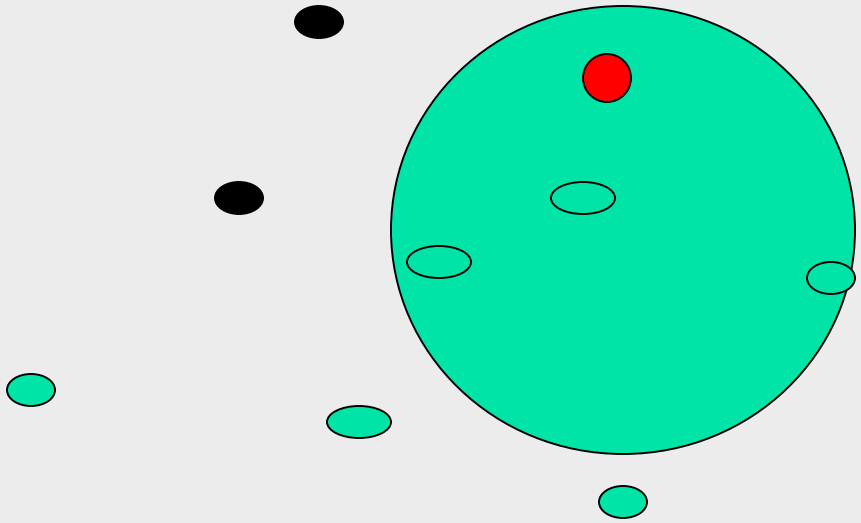
k = 3



1-Nearest Neighbor



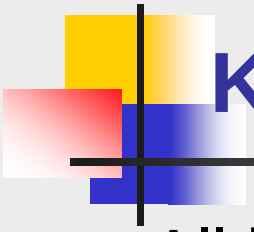
3-Nearest Neighbor



Classification Steps

- **Training phase:** a model is constructed from the training instances.
  - classification algorithm finds relationships between predictors and targets
  - relationships are summarised in a model
- **Testing phase:** test the model on a test sample whose class labels are known but not used for training the model
- **Usage phase:** use the model for classification on new data whose class labels are unknown

https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm



KNN Features

- All instances correspond to points in an n-dimensional Euclidean space
- Classification is delayed till a new instance arrives
- Classification done by comparing feature vectors of the different points
- Target function may be discrete or real-valued

➢ It uses the local neighborhood to obtain a prediction

➢ The K memorized examples more similar to the one that is being classified are retrieved

KNN Features

- A distance function is needed to compare the examples similarity

Euclidean :

$$d(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

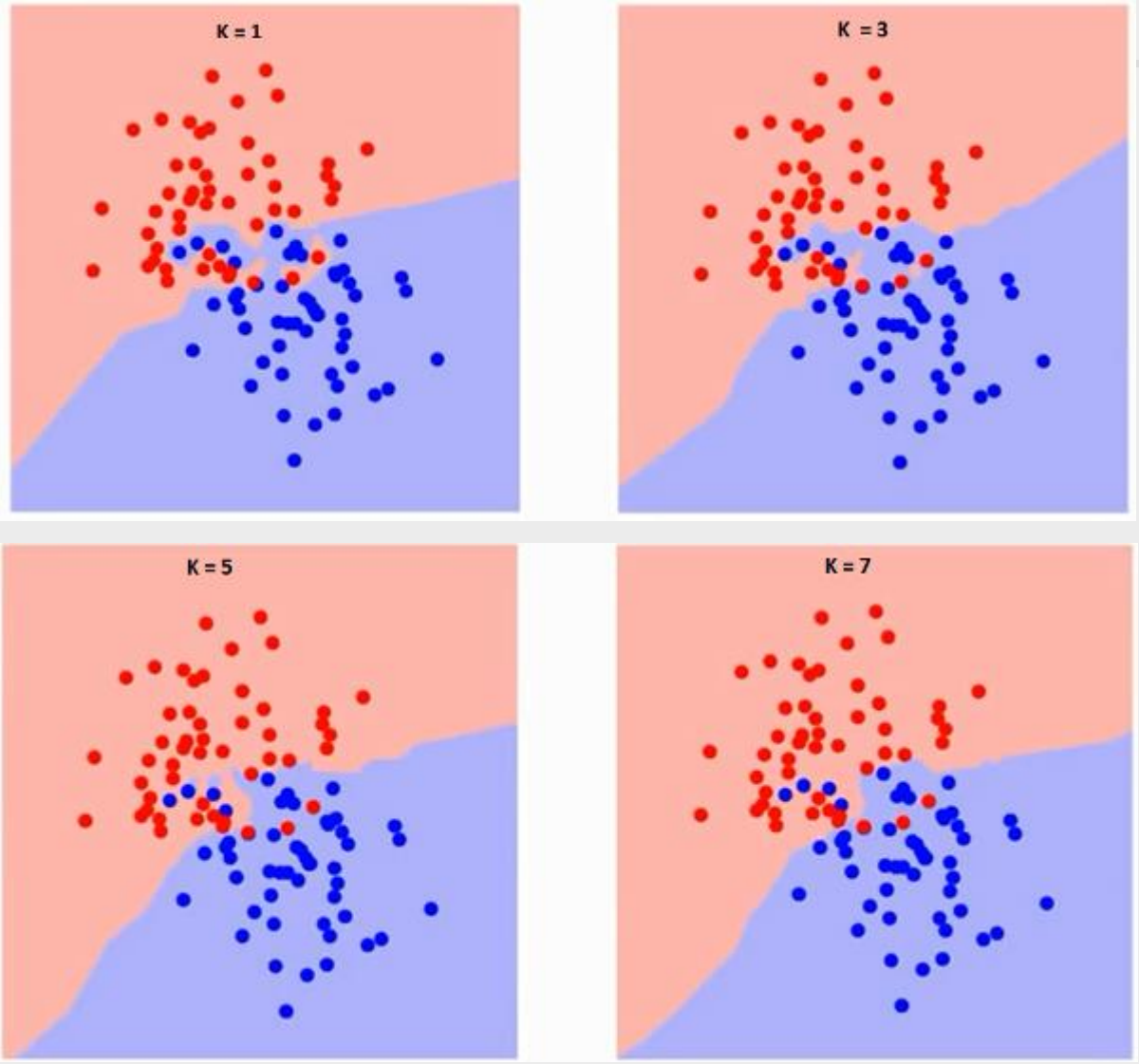
Manhattan / city - block :

$$d(x,y) = \sum_{i=1}^{m}|x_i - y_i|$$

- This means that if we change the distance function, we change how examples are classified

Learning is very simple but Classification is time consuming

15 KNN Classifier : Effect of K Value



The boundary becomes smoother with increasing value of K.

With K increasing to infinity it finally becomes all blue or all red depending on the total majority.

https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/#h-what-is-knn-k-nearest-neighbor-algorithm

16

KNN Classifier: Conclusions

Pros of KNN

- Very simple algorithm to understand and interpret.
- Very useful for nonlinear data because there is no assumption about data in KNN.
- Versatile algorithm as we can use it for classification as well as regression.
- High accuracy but there are much better supervised learning models than KNN.

Cons of KNN

- Computationally a bit expensive algorithm because it stores all the training data.
- High memory storage required as compared to other supervised learning algorithms.
- Prediction is slow in case of big N.
- Very sensitive to the scale of data as well as irrelevant features.

Applications of KNN

Banking System: to predict weather an individual is fit for loan approval? Does that individual have the characteristics similar to the defaulters one?

Calculating Credit Ratings: can be used to find an individual's credit rating by comparing with the persons having similar traits.

Other areas: can be used are Speech Recognition, Handwriting Detection, Image Recognition and Video Recognition.

17

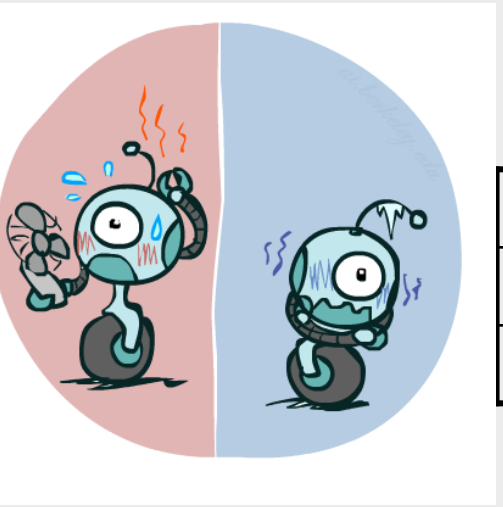
Naïve Bayes (NB) Classifier

18

Probability Distributions

Associate a probability with each value

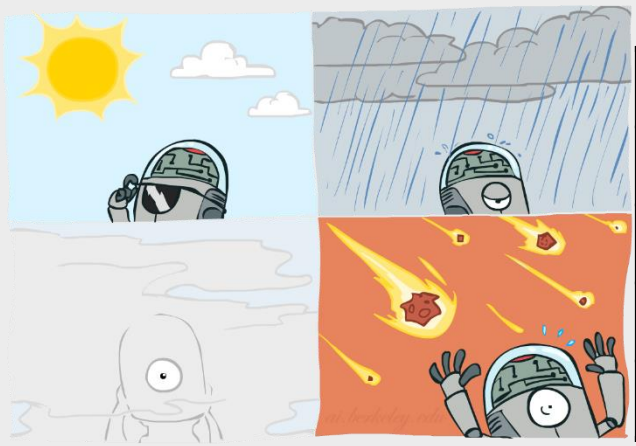
Temperature:



$P(T)$

T	P
hot	0.5
cold	0.5

Weather:



$P(W)$

W	P
sun	0.6
rain	0.1
fog	0.3
meteor	0.0

19

Probabilistic Models

- A probabilistic model is a joint distribution over a set of random variables

- Probabilistic models:
  - (Random) variables with domains
  - Assignments are called *outcomes*
  - Joint distributions: say whether assignments (outcomes) are likely
  - *Normalized:* sum to 1.0
  - Ideally: only certain variables directly interact

Distribution over T, W

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

20

Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_s P(t, s)$$

$P(T)$

T	P
hot	0.5
cold	0.5

$$P(s) = \sum_t P(t, s)$$

$P(W)$

W	P
sun	0.6
rain	0.4

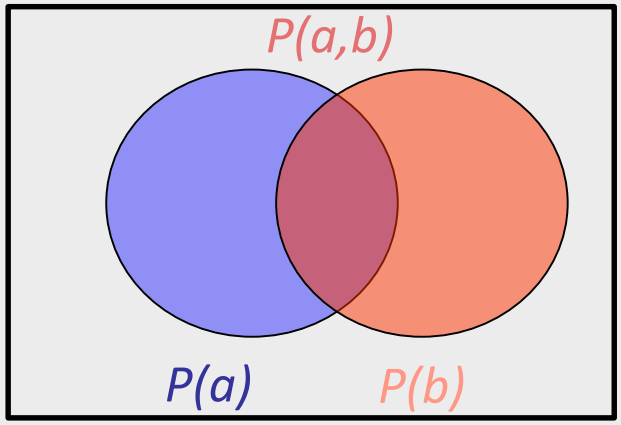
$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

21

Conditional Probabilities

- A simple relation between joint and conditional probabilities
  - In fact, this is taken as the *definition* of a conditional probability

$$P(a|b) = \frac{P(a,b)}{P(b)}$$



$P(T,W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(W=s|T=c) = \frac{P(W=s,T=c)}{P(T=c)} = \frac{0.2}{0.5} = 0.4$$

$$= P(W=s,T=c) + P(W=r,T=c)$$

$$= 0.2 + 0.3 \quad = 0.5$$

22

Normalization Trick

$$P(W = s|T = c) = \frac{P(W = s, T = c)}{P(T = c)}$$

$$= \frac{P(W = s, T = c)}{P(W = s, T = c) + P(W = r, T = c)}$$

$$= \frac{0.2}{0.2 + 0.3} = 0.4$$

$$P(a|b) = \frac{P(a, b)}{P(b)}$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(W|T = c)$

W	P
sun	0.4
rain	0.6

$$P(W = r|T = c) = \frac{P(W = r, T = c)}{P(T = c)}$$

$$= \frac{P(W = r, T = c)}{P(W = s, T = c) + P(W = r, T = c)}$$

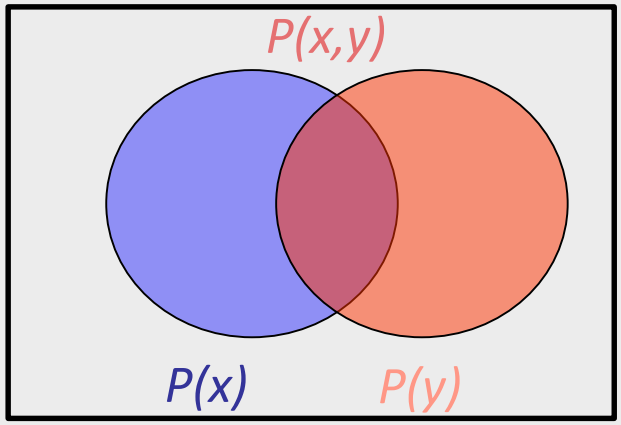
$$= \frac{0.3}{0.2 + 0.3} = 0.6$$

23

Bayes' Rule

$$P(a|b) = \frac{P(a,b)}{P(b)}$$

$$P(x|y) = \frac{P(x,y)}{P(y)}$$



$$P(y|x) = \frac{P(x,y)}{P(x)}$$

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

24

Bayes' Rule

Prior, conditional and joint probability

- Prior probability: $P(X)$
- Conditional probability: $P(X_1 \mid X_2), P(X_2 \mid X_1)$
- Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
- Relationship: $P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) = P(X_1 \mid X_2)P(X_2)$
- Independence: $P(X_2 \mid X_1) = P(X_2), P(X_1 \mid X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$

Bayesian Rule

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \qquad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

25

Naïve Bayes (NB) Classifier

- Establishing a probabilistic model for classification
  - Discriminative model $P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$
- MAP classification rule
  - MAP: Maximum A Posterior
  - Assign x to c^* if

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*, \; c = c_1, \cdots, c_L$$

26

Naïve Bayes (NB) Classifier

Naïve Bayes Algorithm (for discrete input attributes)

– Learning Phase: Given a training set **S**,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in **S**;

For every attribute value a_{jk} of each attribute x_j $(j = 1, \cdots, n; k = 1, \cdots, N_j)$

$\hat{P}(X_j = a_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = a_{jk} \mid C = c_i)$ with examples in **S**;

Output: conditional probability tables; for x_j, $N_j \times L$ elements

– Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$,

Look up tables to assign the label c^* to **X'** if

$$[\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 \mid c) \cdots \hat{P}(a'_n \mid c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

27

Play Tennis: Classification with NB

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

28

NB: Learning Phase

P(Outlook=*o* | Play=*b*)

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

P(Temperature=*t* | Play=*b*)

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

P(Humidity=*h* | Play=*b*)

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

P(Wind=*w* | Play=*b*)

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

P(Play=*Yes*) = 9/14

P(Play=*No*) = 5/14

29

NB: Test Phase

- Given a new instance,

  $\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

- Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9

P(Temperature=*Cool*|Play=*Yes*) = 3/9

P(Huminity=*High*|Play=*Yes*) = 3/9

P(Wind=*Strong*|Play=*Yes*) = 3/9

P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5

P(Temperature=*Cool*|Play==*No*) = 1/5

P(Huminity=*High*|Play=*No*) = 4/5

P(Wind=*Strong*|Play=*No*) = 3/5

P(Play=*No*) = 5/14

- MAP rule

P(*Yes*|$\mathbf{x}'$): [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

P(*No*|$\mathbf{x}'$): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|$\mathbf{x}'$) < P(*No*|$\mathbf{x}'$), we label $\mathbf{x}'$ to be "*No*".

30

NB for Continuous-values Input Attributes

- Numberless values for an attribute
- Conditional probability modeled with the normal distribution

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

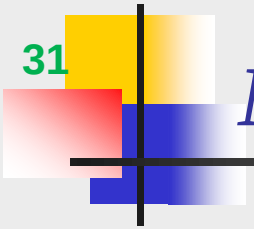
μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

- Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_n)$, $C = c_1, \cdots, c_L$

  Output: $n \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

- Test Phase: for $\mathbf{X}' = (X_1', \cdots, X_n')$
  - Calculate conditional probabilities with all the normal distributions
  - Apply the MAP rule to make a decision



31

NB Remarks

Naïve Bayes based on the independence assumption

- Training is very easy and fast; just requiring considering each attribute in each class separately
- Test is straightforward; just looking up tables or calculating conditional probabilities with normal distributions

32

NB Classifier Online Resource

Naive Bayes Classifiers
https://www.geeksforgeeks.org/naive-bayes-classifiers/

Naive Bayes Classifier (Explained in Bangla) || Algorithm in Data Mining || Data Mining Tutorial

https://www.youtube.com/watch?v=cxWPanlW-L0

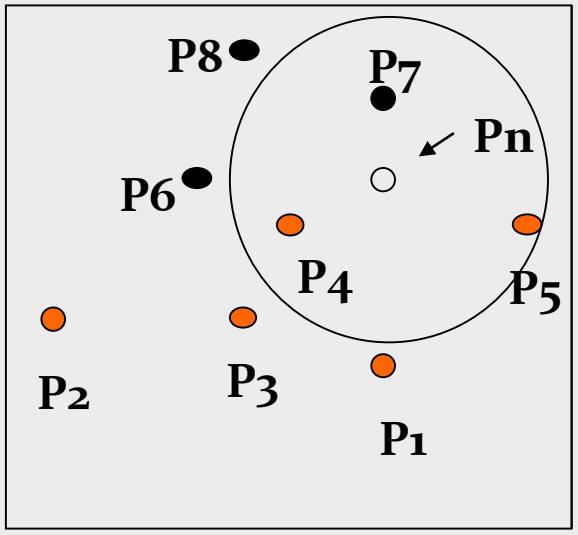
Solved Example Naive Bayes Classifier to classify New Instance PlayTennis Example Mahesh Huddar

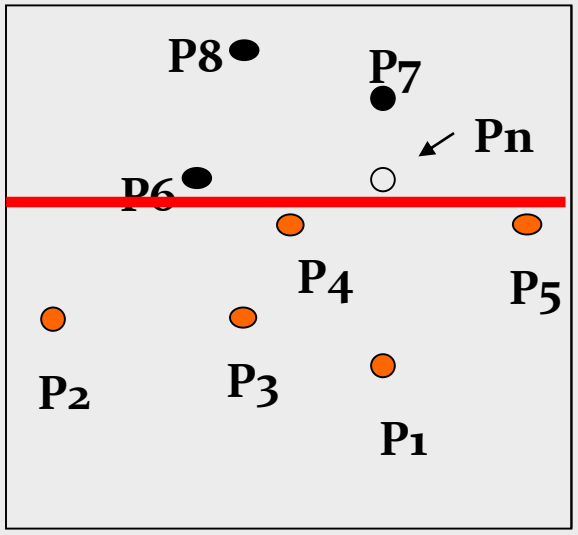
https://www.youtube.com/watch?v=XzSlEA4ck2I&list=WL

33

Lazy & Eager Learning

- Two Types of Learning Methodologies
  - Lazy Learning
    - Instance-based learning. (k-NN)
  - Eager Learning
    - Decision-tree and Bayesian classification.
    - ANN & SVM





34

Lazy & Eager Learning : Key Differences

- Lazy Learning
  - Do not require model building
  - Less time training but more time predicting
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
- Eager Learning
  - Require model building
  - More time training but less time predicting
  - must commit to a single hypothesis that covers the entire instance space

35

Thanks for your attention

Question and Answer

Biological Neuron to Artificial Neural Network

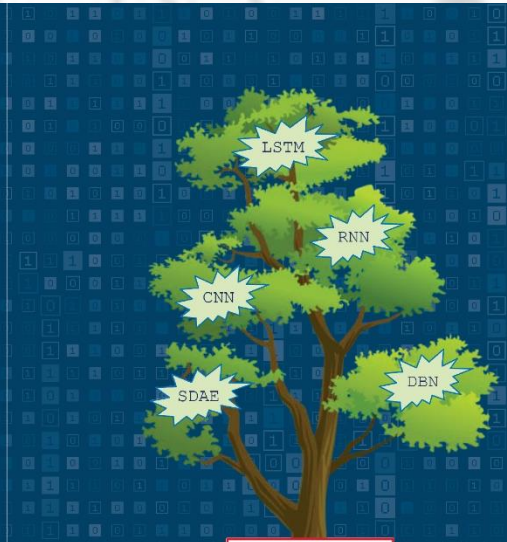
Ref.: Chapter 1: Preliminaries, Deep Learning Fundamentals - A Practical Approach to Understanding Deep Learning Methods ISBN:978-984-35-0812-6, University Grants Commission (UGC), Bangladesh, 2021

Prof. Dr. Md. Aminul Haque Akhand
Dept. of Computer Science and Engineering (CSE)
Khulna University of Engineering & Technology (KUET)
www.kuet.ac.bd/cse/akhand

2 Deep Learning Fundamentals
- A Practical Approach to Understanding Deep Learning Methods

Dr. Muhammad Aminul Haque Akhand
Professor
Department of Computer Science and Engineering (CSE)
Khulna University of Engineering & Technology (KUET)
Website: www.kuet.ac.bd/cse/akhand

Dr. M. A. H. Akhand is currently a Professor of CSE department at KUET, Bangladesh. He is also head of the Computational Intelligence Research Group of this department. His primary research interests are in machine learning, artificial neural network, deep learning, swarm intelligence, optimization, and pattern recognition. He received his B.Sc. degree in Electrical and Electronic Engineering from KUET in 1999, the M.E. degree in Human and Artificial Intelligent Systems in 2006, and the Doctoral degree in System Design Engineering in 2009 from University of Fukui, Japan. He has more than 100 publications in International Journals and Conferences. Dr. Akhand received several best paper Prizes in International Conferences. He is also an editorial board member of several International Journals.

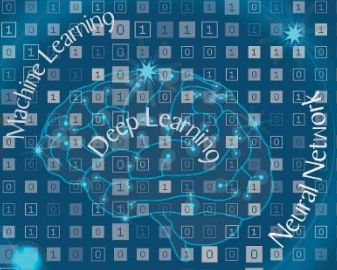


Deep Learning
Neural Network
Machine Learning

বাংলাদেশ বিশ্ববিদ্যালয় মঞ্জুরী কমিশন
University Grants Commission of Bangladesh

Deep Learning Fundamentals - A Practical Approach to Understanding Deep Learning Methods
Dr. Muhammad Aminul Haque Akhand

DEEP LEARNING FUNDAMENTALS
A Practical Approach to Understanding Deep Learning Methods



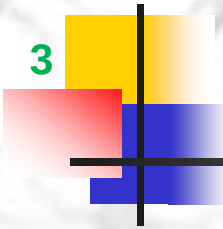
Dr. Muhammad Aminul Haque Akhand

About the Book

Machine learning (ML) is the virtue of modern science and its outcomes are intelligent systems that became integral parts of daily lives. Artificial neural network (NN) is the most prominent ML method and deep learning (DL) is its latest form mimicking the learning mechanism of human brain. At present, DL is the most prominent research area and its different methods are employed in different research studies and development of significant systems. This book contains a comprehensive study of prominent DL methods from NN foundation. The book is written based on the experience of many years following pedagogical features with illustrations, step-by-step algorithms, worked examples and MATLAB code for real-world problems. This book might be a textbook for DL related courses in PG level of Computer Science, Engineering and related disciplines. This book will also be useful for young researchers to work with DL related methods.

Cover Concept & Design: N. K. Kaikobad Rana

https://kuet.ac.bd/cse/akhand/



Contents

4

Why Artificial Neural Networks?

Human brain has many desirable characteristics not presented in modern computers:

massive parallelism,
distributed representation and computation,
learning ability,
generalization ability,
adaptability,
inherent contextual information processing,
fault tolerance, and
low energy consumption.

It is hope that devices based on biological neural networks will process some of these characteristics.

5

Why Artificial Neural Networks?

- Modern digital computers outperform humans in the domain of numeric computation and the related symbolic manipulation.

- However, humans can solve complex perceptual problems effortlessly, and more efficiently than the world's fastest computer.

- Human performs these remarkable benefits due to the characteristics of his/her neural system whose (neural system) working procedure is completely different from a conventional computer system.

Comparison of Brains and Traditional Computers

- 200 billion neurons, 32 trillion synapses
- Element size: 10^{-6} m
- Energy use: 25W
- Processing speed: 100 Hz
- Parallel, Distributed
- Fault Tolerant
- Learns: Yes
- Intelligent/Conscious: Usually

- 1 billion bytes RAM but trillions of bytes on disk
- Element size: 10^{-9} m
- Energy watt: 30-90W (CPU)
- Processing speed: 10^{9} Hz
- Serial, Centralized
- Generally not Fault Tolerant
- Learns: Some
- Intelligent/Conscious: Generally No

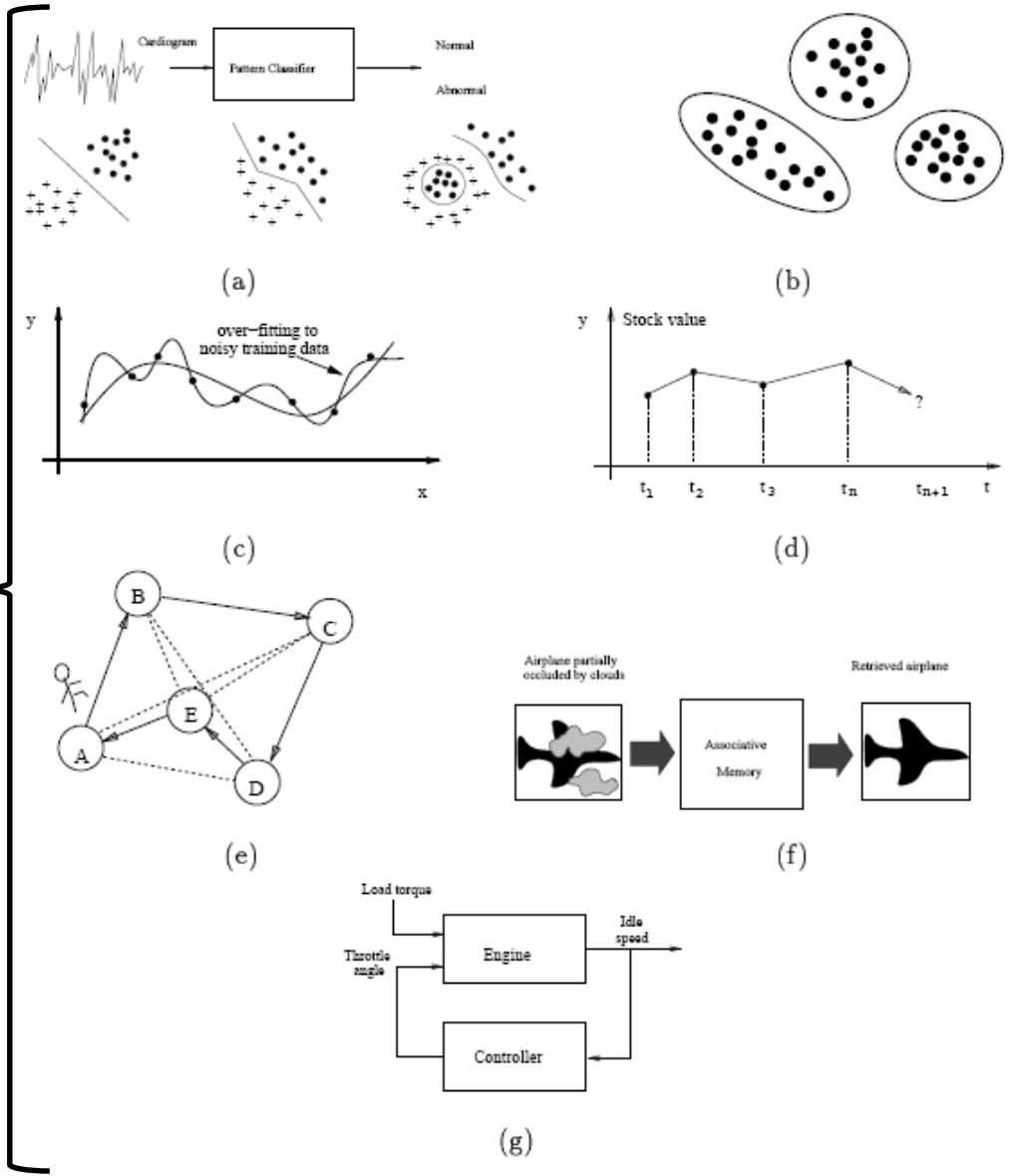
7 von Neumann Computer vs. Biological Neural System

	von Neumann Computer	Biological Neural System
processor	complex high-speed one (or a few)	simple low-speed a large number
memory	separate localized noncontent addressable	integrated in the neuron distributed content addressable
computing	centralized sequential strored-programs	distributed parallel self-learning
reliability	vulnerable	robust
expertise	numerical symbolic	perceptual problems manipulations
environment	well-defined	poorly defined unconstrained

8 The Goal

ANN is designed with the goal of building **intelligent machines to solve complex perceptual problems**, such as pattern recognition and optimization, by mimicking the networks of real neurons in the human brain.

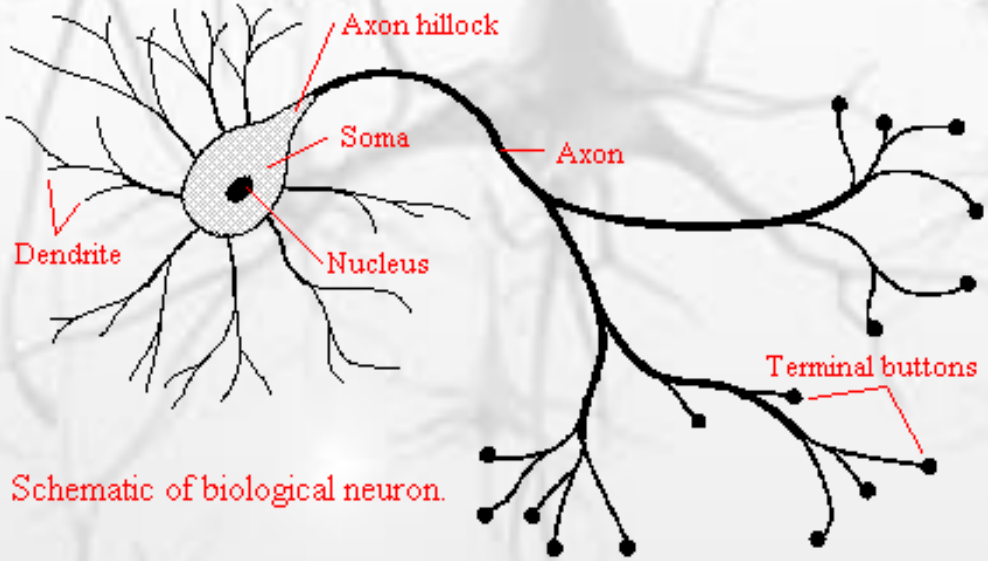
Tasks handled by ANNs



9

Artificial Neural Networks(ANNs) Motivations

❖ANNs are inspired by biological neural networks



Schematic of biological neuron.

A *neuron* (or nerve cell) is a special biological cell, the essence of life, with information processing ability. The introduction of neurons as basic structural constituents of the brain was credited to Ramon y Cajal who won the 1906 Nobel prize for physiology and medicine (shared with Camillo Golgi) for the crucial discovery of the extensive interconnections within the *cerebral cortex*, the portion of the brain where approximately 90% of the neurons in the human are located.

Biological Neuron

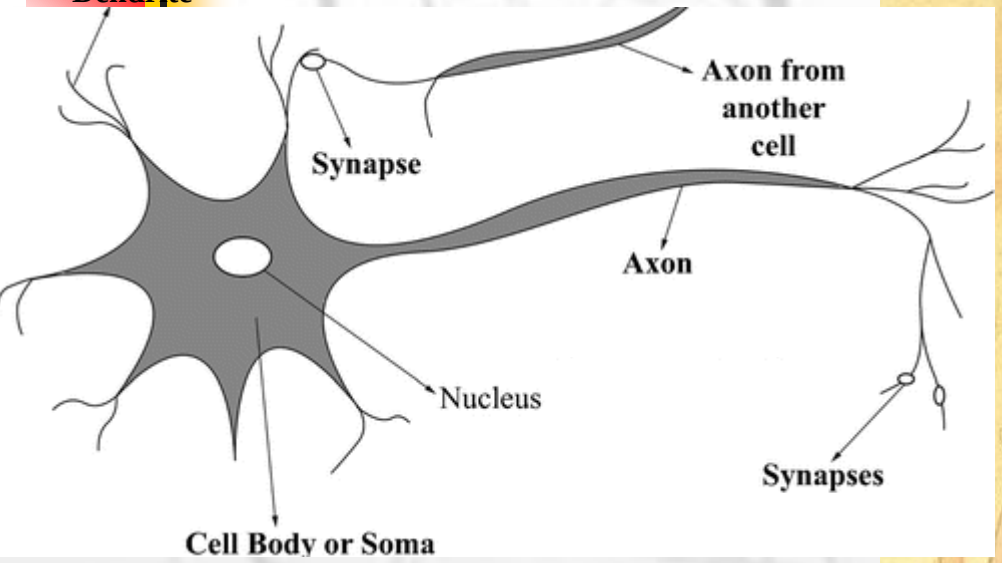
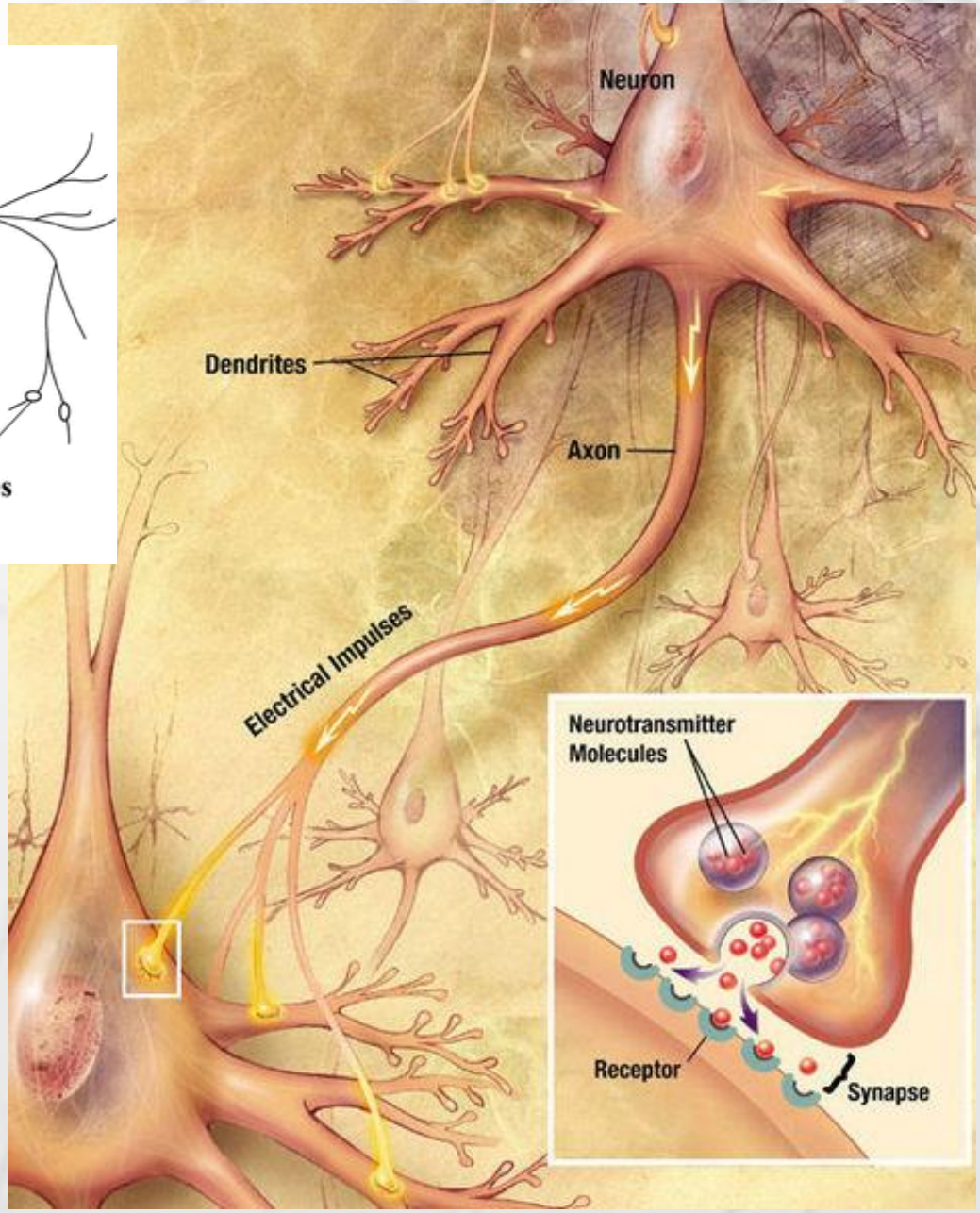
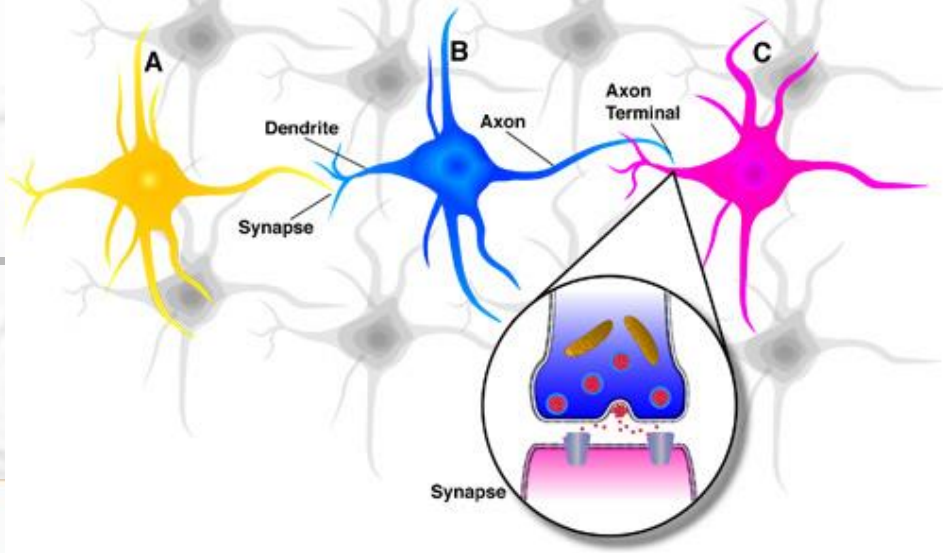


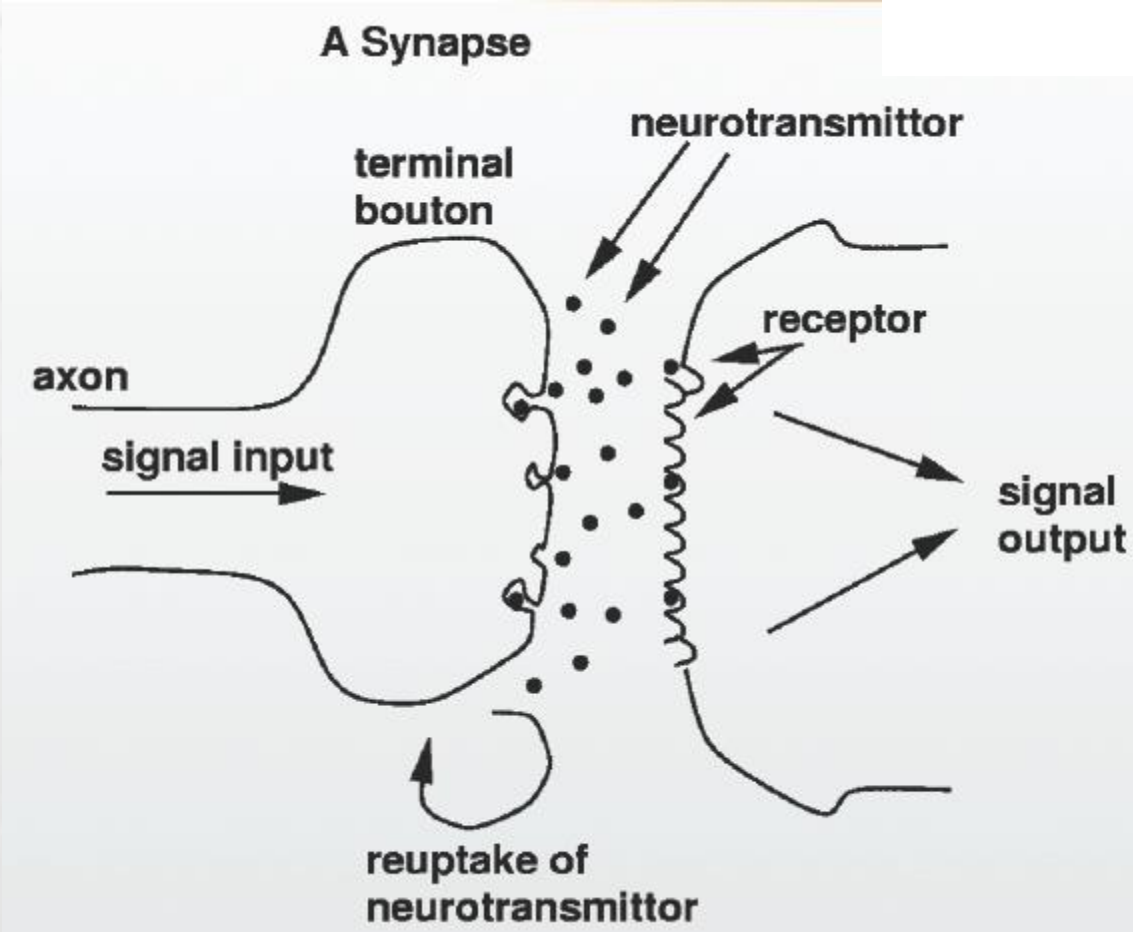
Figure 1.1: A sketch of a biological neuron.



11

Synapses



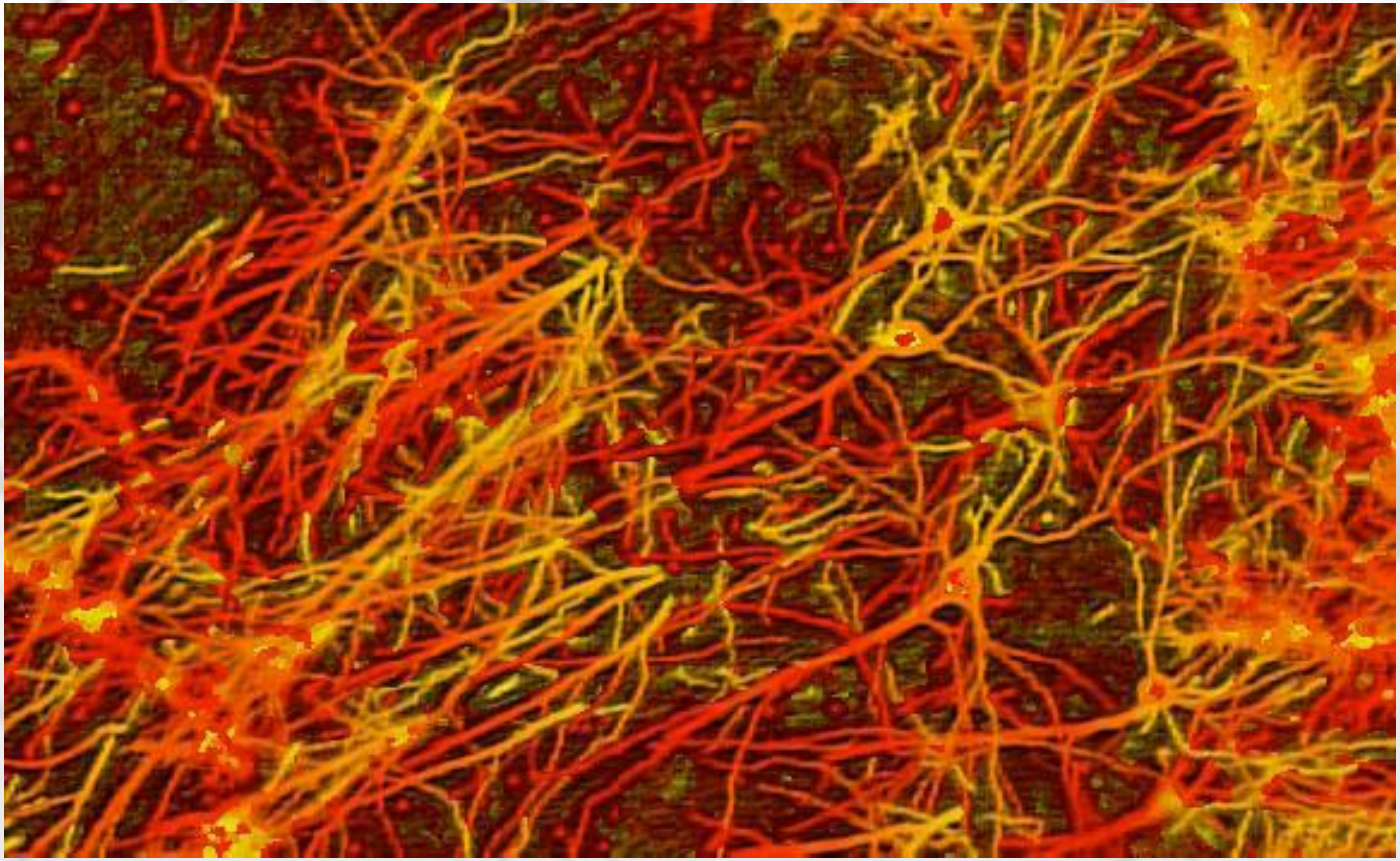


Synapses

- transmit signal to next neuron
- vary in strength
- *change* strength in response to use (learning!)

12

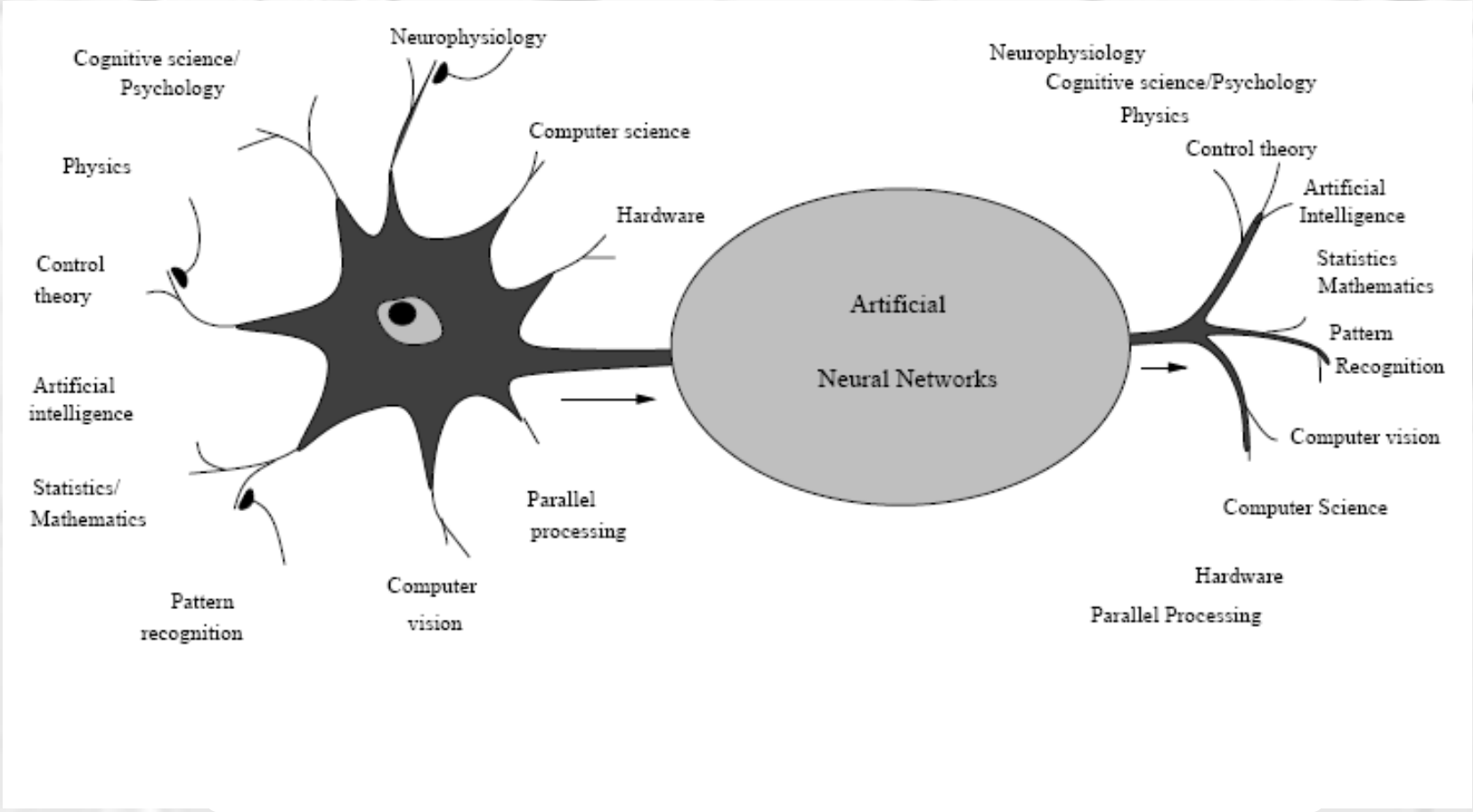
A Real Neural Network



Neurons are massively connected, much more complex and denser than today's telephone networks. Each neuron is connected to $10^3 - 10^4$ other neurons. The number of interconnections depends on the location of the neuron in the brain and the type of the neuron. In total, the human brain contains approximately $10^{14} - 10^{15}$ interconnections.

13

ANN's Relationship with Other Disciplines



14

Computational Model

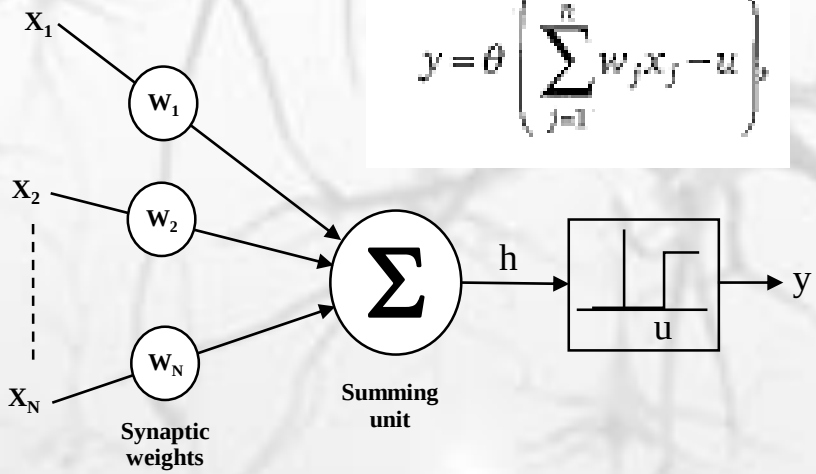


Figure 1.2: McCulloch-Pitts model of a neuron.

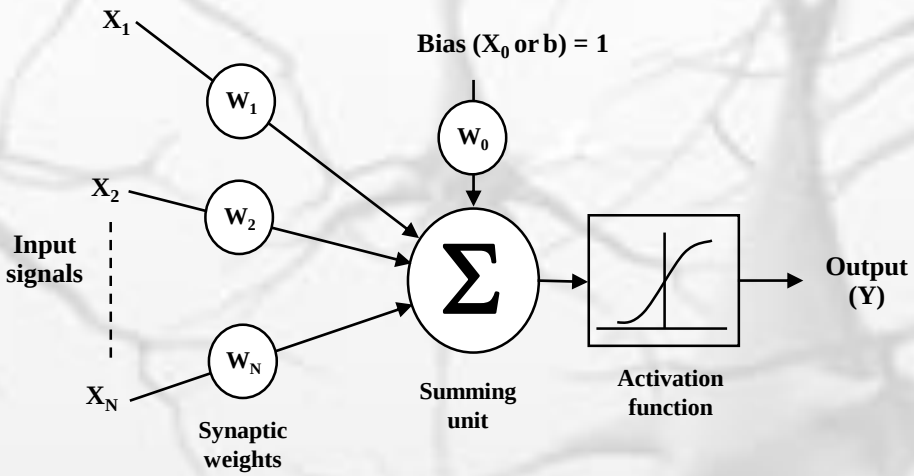


Figure 1.3: Generalized model of an artificial neuron.

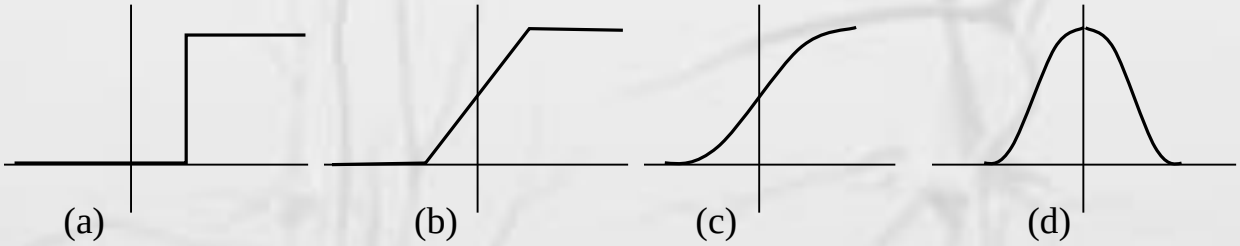


Figure 1.4: Different types of activation functions: (a) threshold, (b) piecewise linear, (c) sigmoid, and (d) Gaussian.

15

Sigmoid Activation Function

$$y=1/(1+exp(-ax))$$

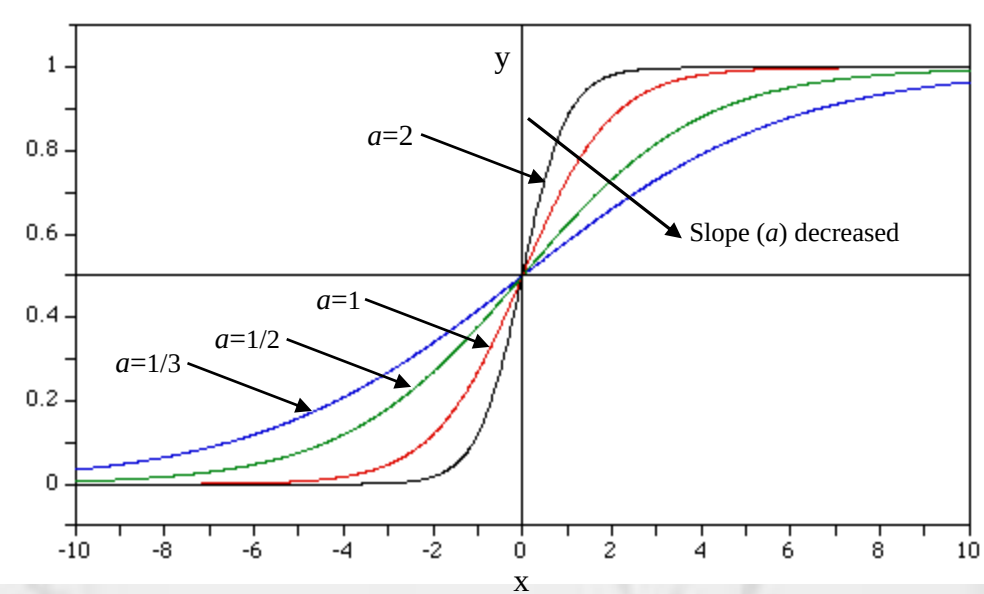
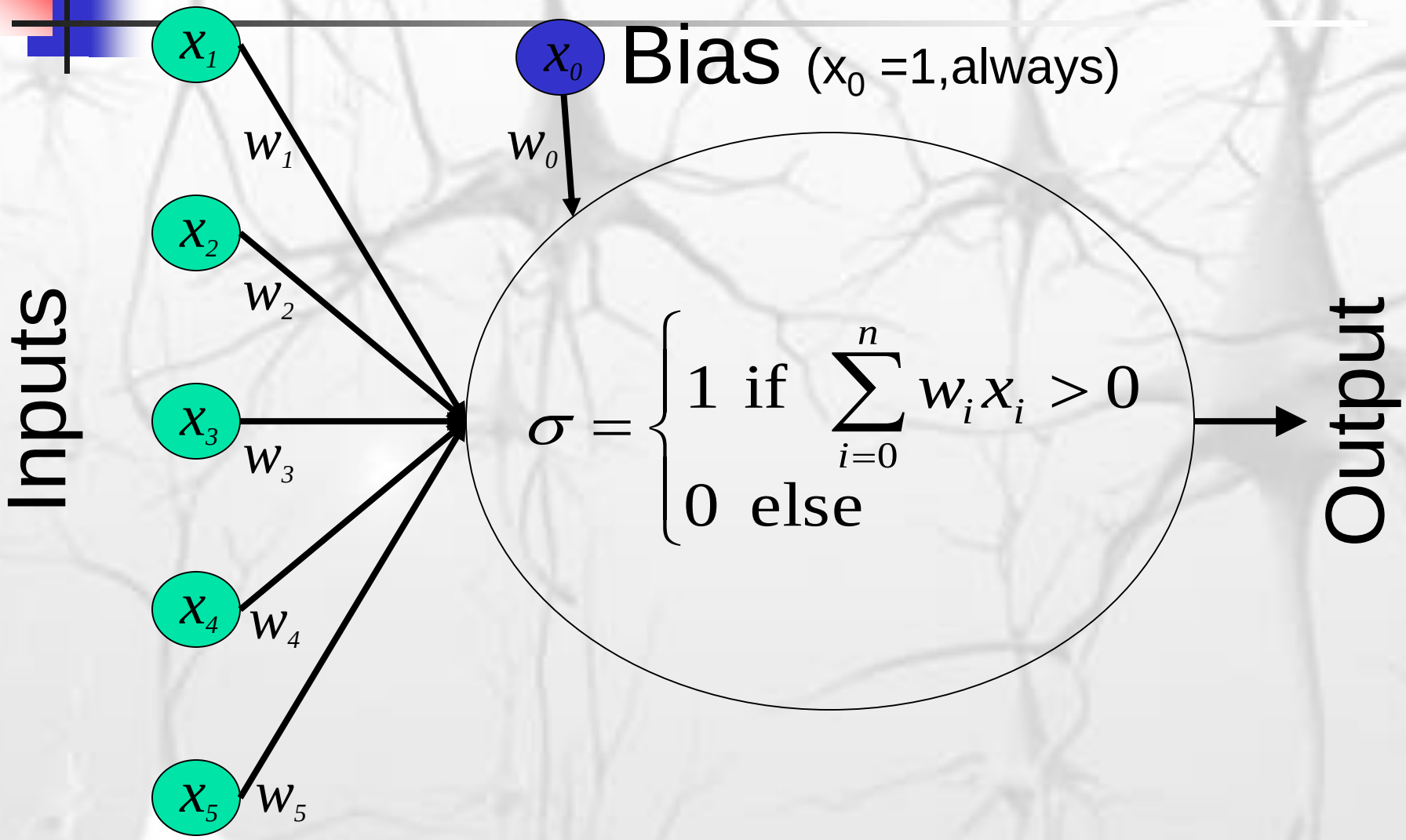
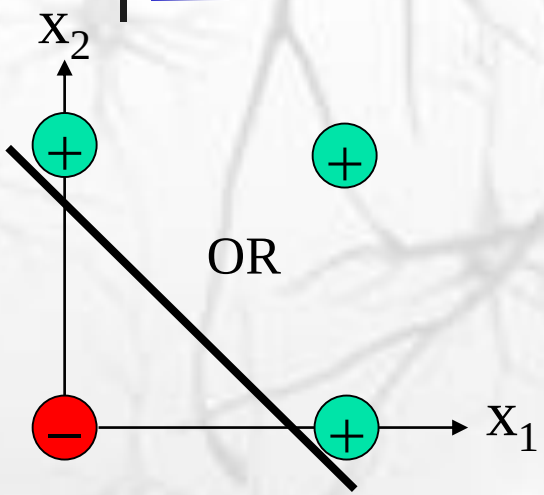


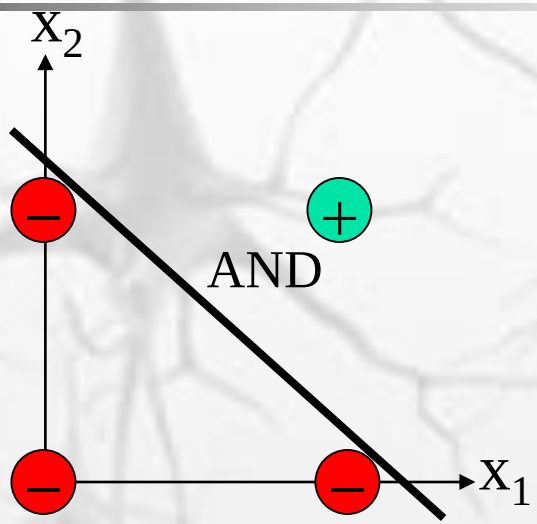
Figure 1.5: Sigmoid function with various slope (a) values.

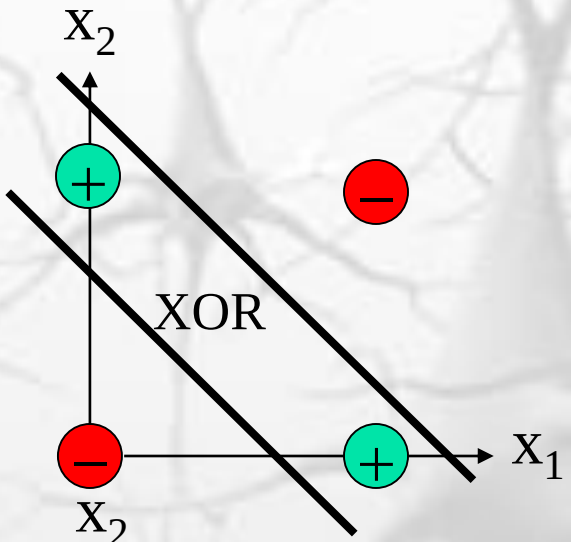
Ability of Single Neuron

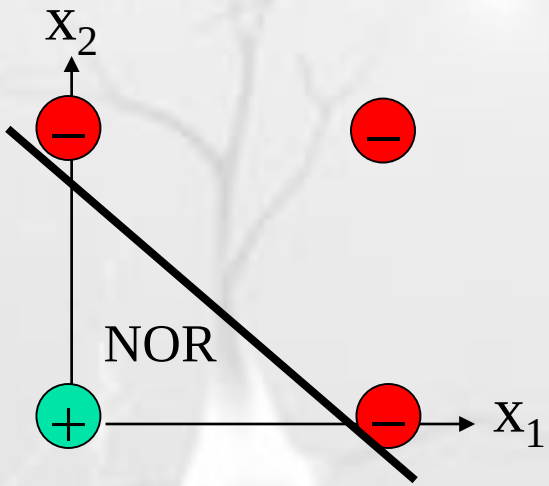


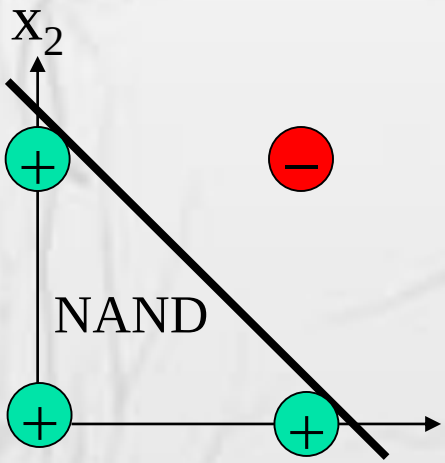
Logical Operators

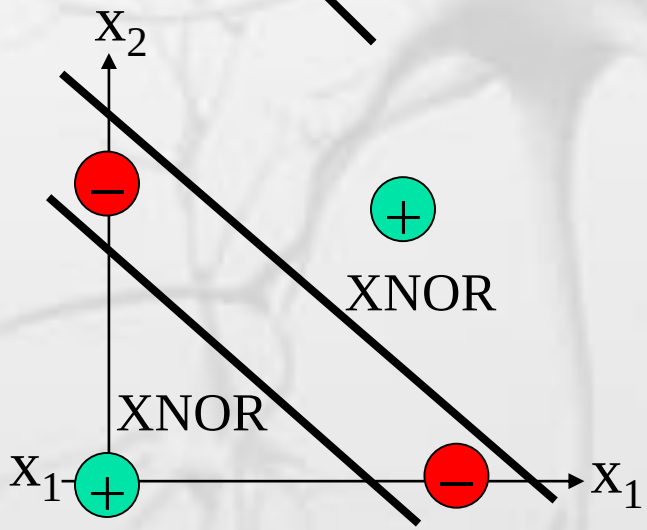




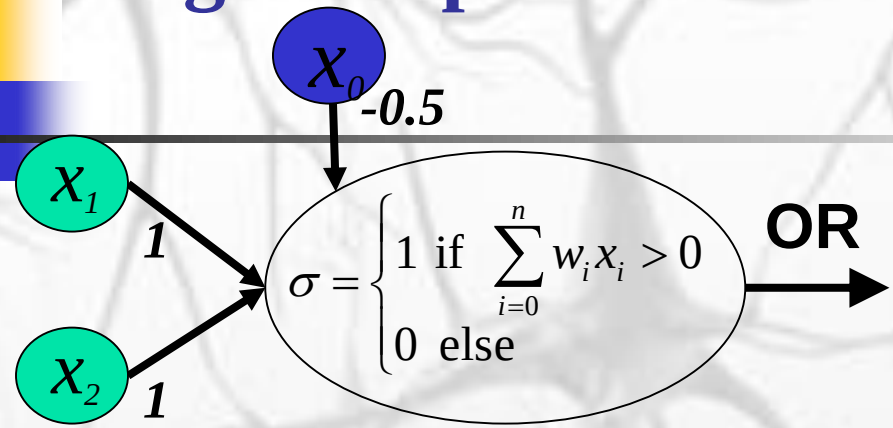


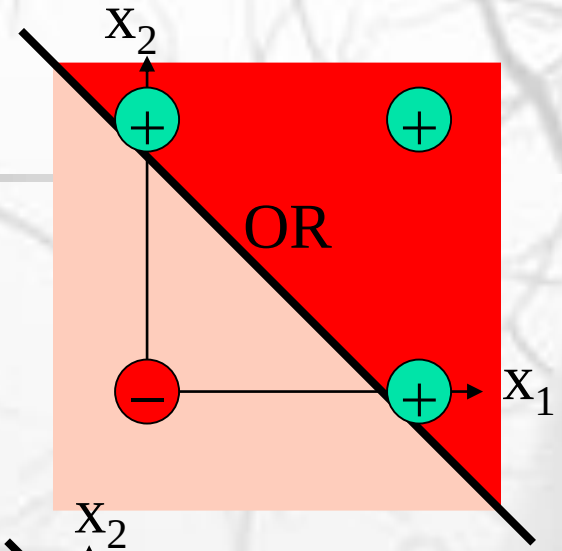


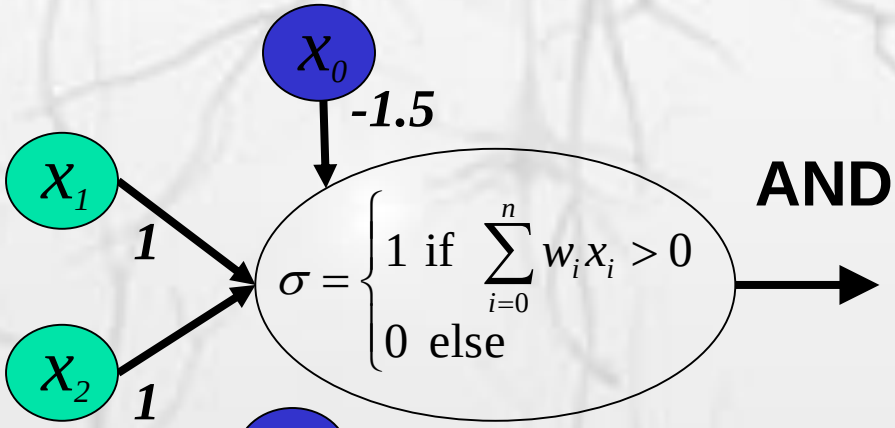


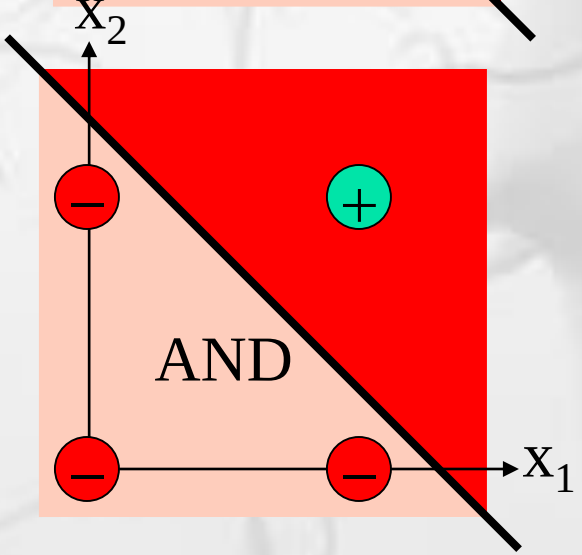


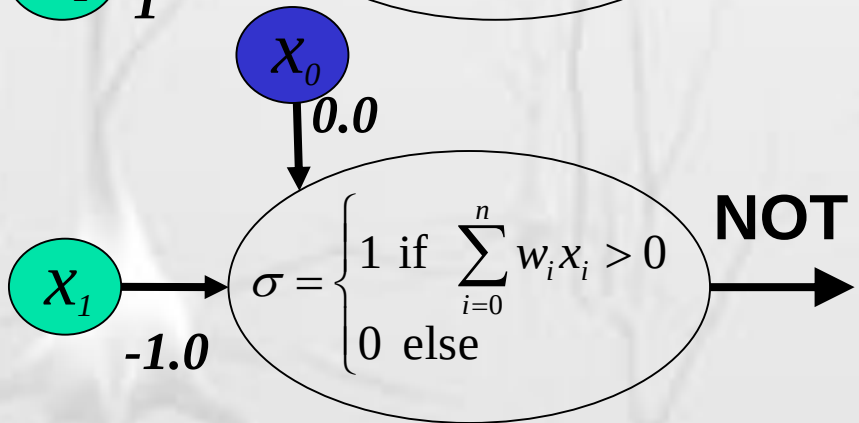
Logical Operators



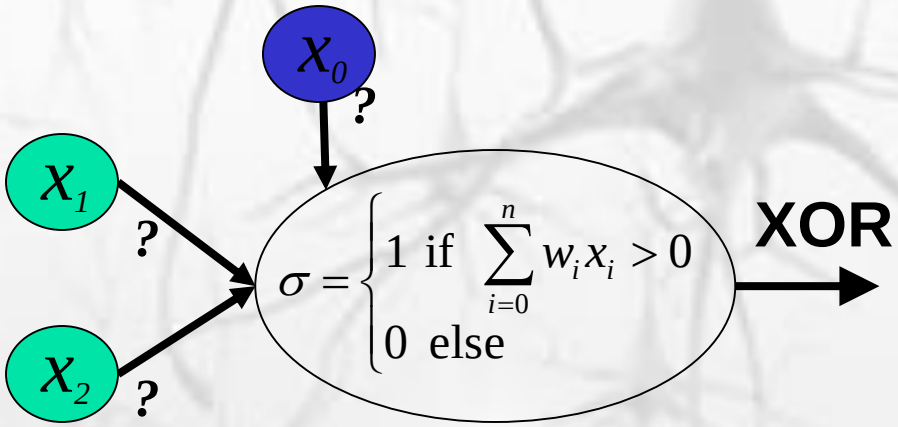


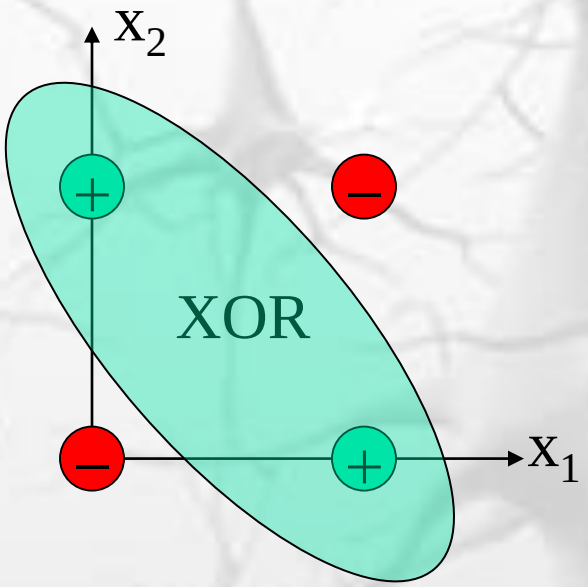






Nonlinear Problem

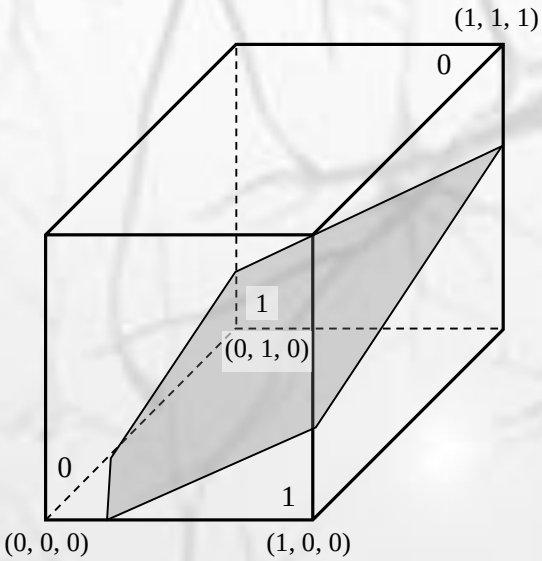




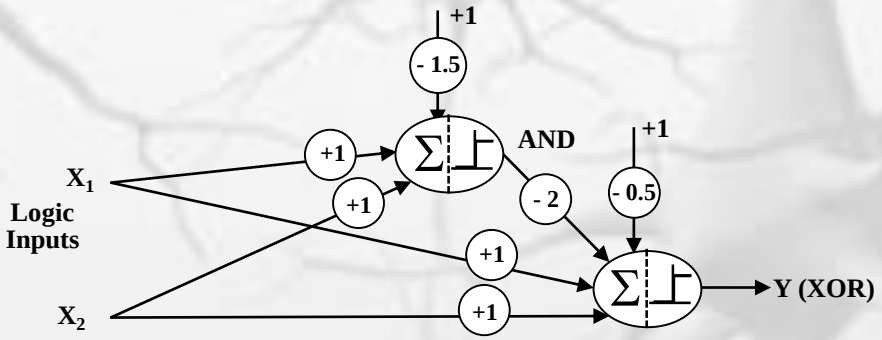
No arrangement work, not linearly separable. Require two boundary lines.

20

XOR solution in Higher Dimension



a) Third dimension uplift XOR (1, 1) as (1, 1, 1) in three dimension.



b) XOR solution with additional input from AND gate.

Figure 1.8: Solution of XOR in three dimension.

An additional dimension (input) converts the problem to a linearly separable.

21

XOR solution with HN

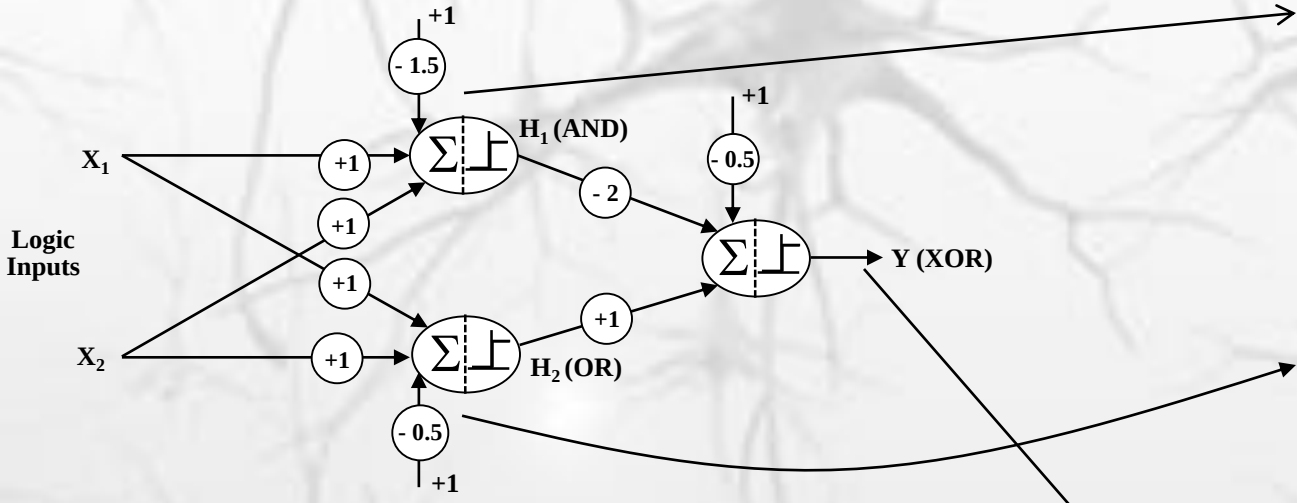
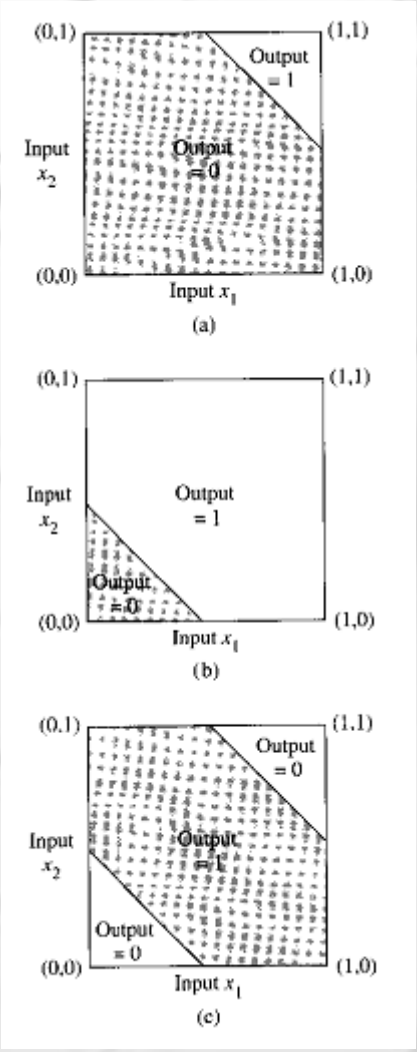


Figure 1.9: XOR logic combining AND and OR logic operations.



AND

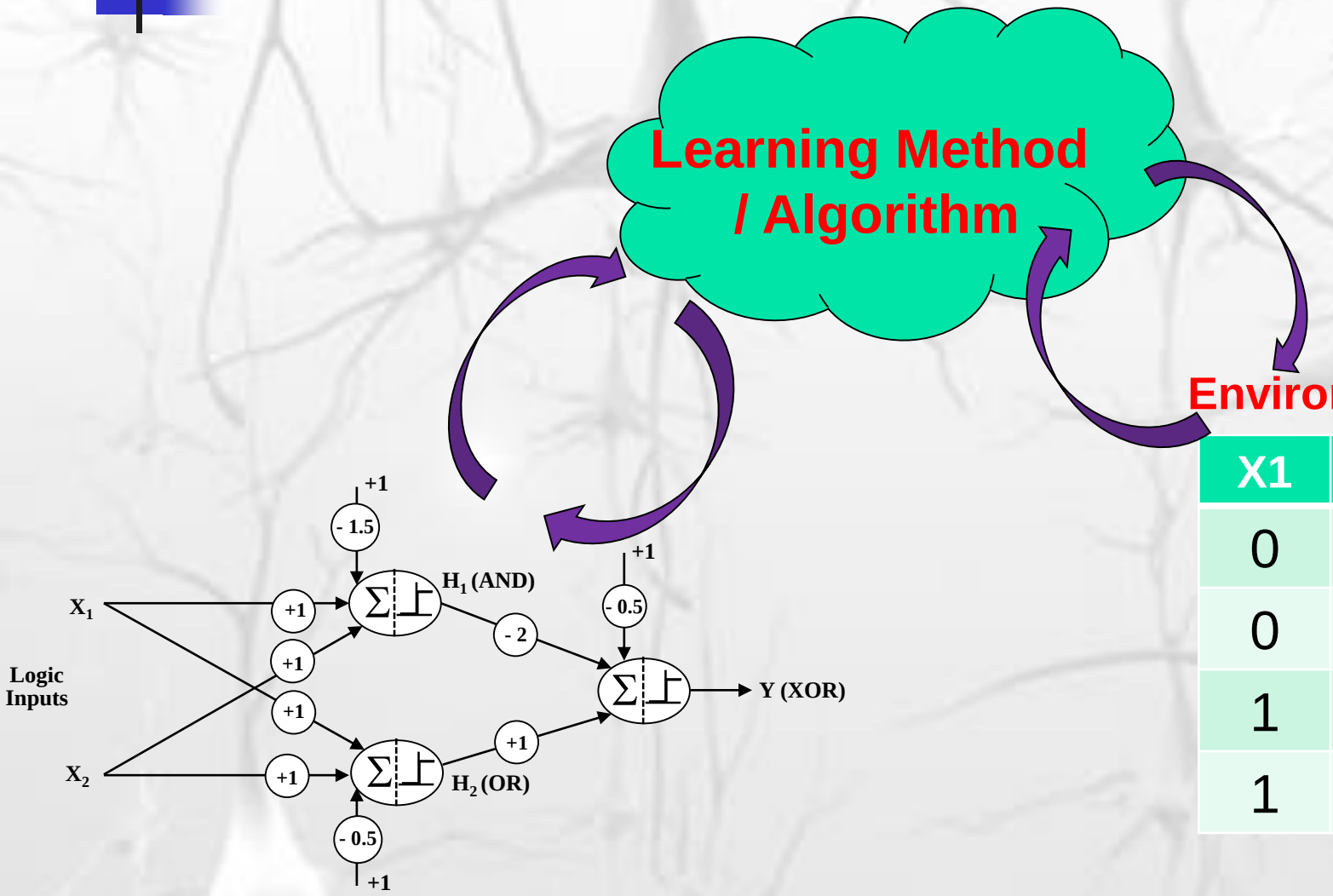
OR

XOR

XOR logic through manipulation by hidden neurons

22

Learning



X1	X2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

23

Learning

- The ability to learn is a fundamental trait (characteristic) of intelligence.

- ANN-> Updating NN architecture and connection weights to perform a specific task efficiently.

- ANN ability to automatically learning from examples makes them attractive.

- To understand or design a learning process need: a model of environment or information is available to NN and learning rules.

- A learning algorithm -> procedure for adjusting weights

- Three main learning paradigms-> Supervised (learning with a teacher), unsupervised and hybrid

- Reinforcement learning is variant of supervised learning receives critique on the correctness of network output instead of correct answer.

24 Learning

- Learning theory must address three fundamental and practical issues:

  1. Capacity – how many patterns can be stored , what functions and decision boundaries a NN can perform

  2. Sample Complexity – Number training patterns needed to train

  3. Computational Complexity – Time required for a learning algorithm to estimate a solution

- Four basic types of Learning rules: Error-Correction, Boltzman, Hebbian, and Competitive.

X1	X2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

25

Network Architecture / Topology

Based on Connection pattern (architecture): feed forward and recurrent (feedback)

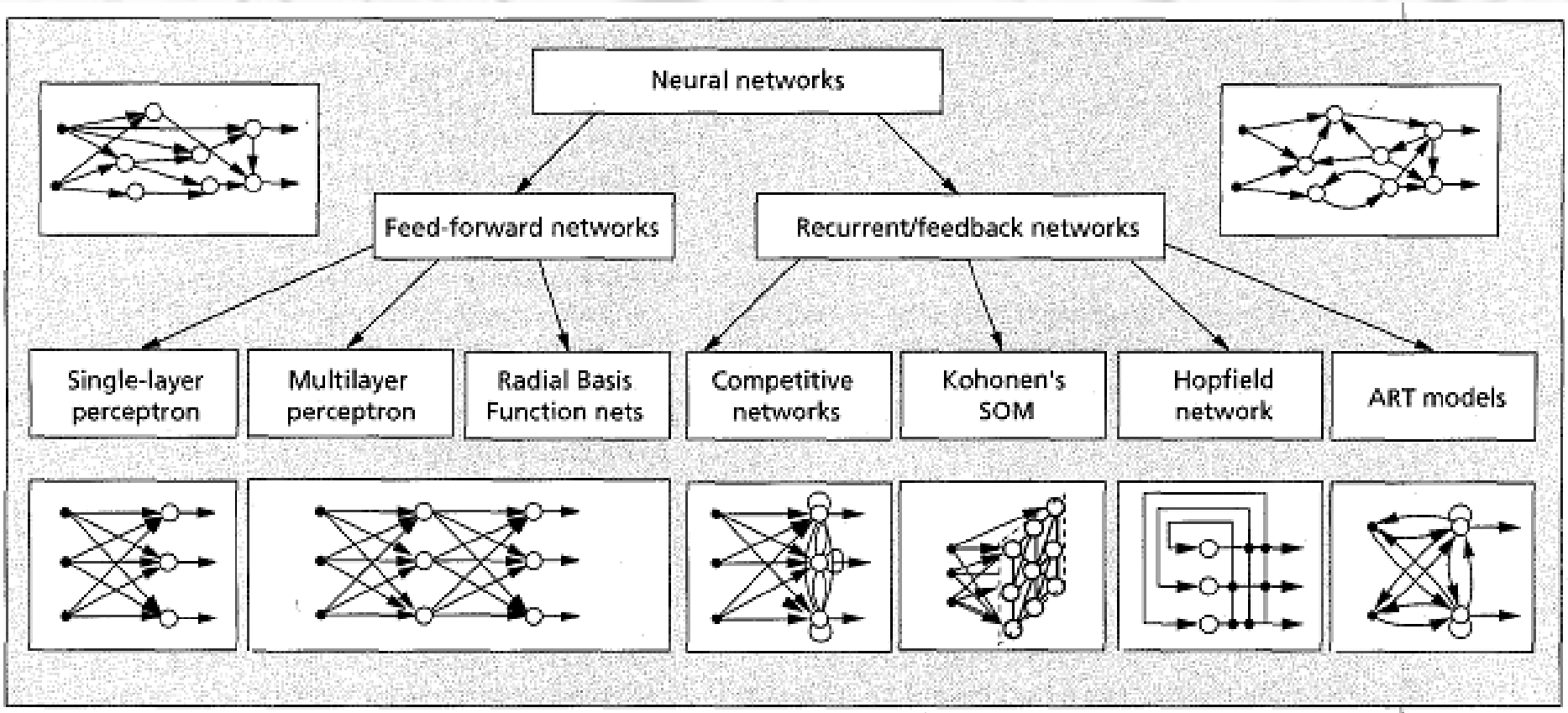


Figure 4. A taxonomy of feed-forward and recurrent/feedback network architectures.

Different architectures require appropriate learning algorithms.

27

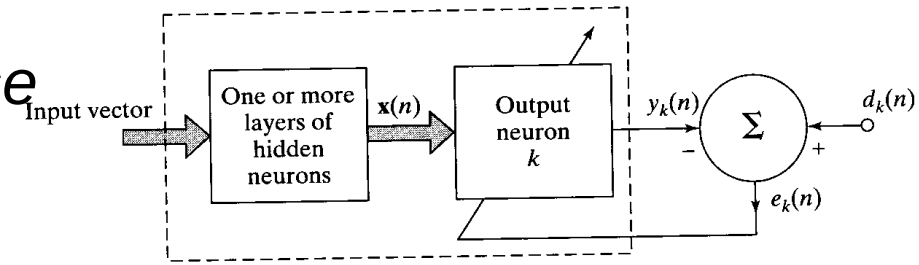
Well Known Learning Algorithms

Paradigm	Learning rule	Architecture	Learning algorithm	Task
Supervised	Error-correction	Single- or multilayer perceptron	Perceptron learning algorithms Back-propagation Adaline and Madaline	Pattern classification Function approximation Prediction, control
	Boltzmann	Recurrent	Boltzmann learning algorithm	Pattern classification
	Hebbian	Multilayer feed-forward	Linear discriminant analysis	Data analysis Pattern classification
	Competitive	Competitive	Learning vector quantization	Within-class categorization Data compression
		ART network	ARTMap	Pattern classification Within-class categorization
Unsupervised	Error-correction	Multilayer feed-forward	Sammon's projection	Data analysis
	Hebbian	Feed-forward or competitive	Principal component analysis	Data analysis Data compression
		Hopfield Network	Associative memory learning	Associative memory
	Competitive	Competitive	Vector quantization	Categorization Data compression
		Kohonen's SOM	Kohonen's SOM	Categorization Data analysis
		ART networks	ART1, ART2	Categorization
Hybrid	Error-correction and competitive	RBF network	RBF learning algorithm	Pattern classification Function approximation Prediction, control

28

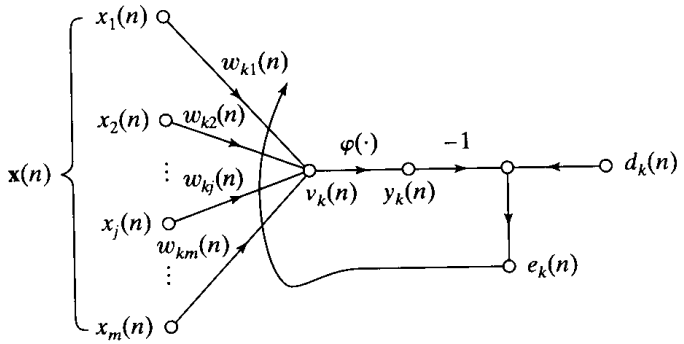
Learning Rule: Error-Correction

- *error signal = desired response – output signal*
- $e_k(n) = d_k(n) - y_k(n)$
- *$e_k(n)$ actuates a control mechanism to make the output signal $y_k(n)$ come closer to the desired response $d_k(n)$ in step by step manner*



Multilayer feedforward network

(a) Block diagram of a neural network, highlighting the only neuron in the output layer

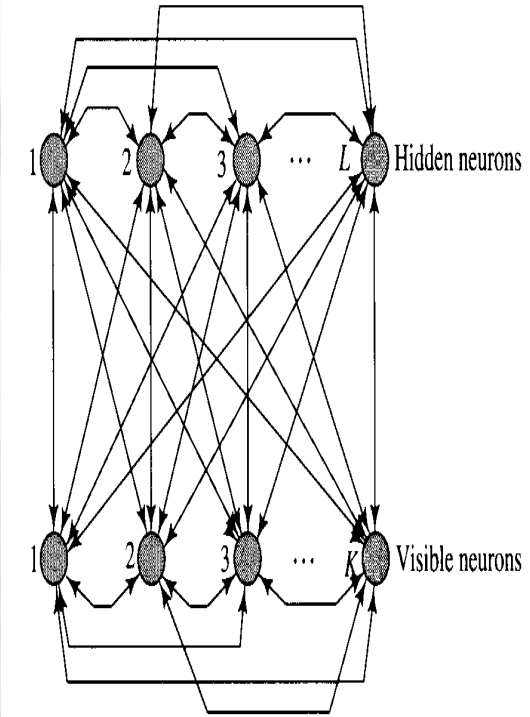


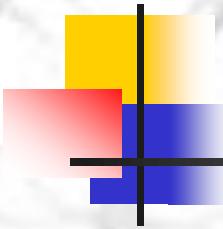
(b) Signal-flow graph of output neuron

FIGURE 2.1 Illustrating error-correction learning.

Learning Rule: Boltzmann

- *The primary goal of Boltzmann learning is to produce a neural network that correctly models input patterns according to a Boltzmann distribution.*
- *The Boltzmann machine consists of stochastic neurons. A stochastic neuron resides in one of two possible states (± 1) in a probabilistic manner.*
- *The use of symmetric synaptic connections between neurons.*
- *The stochastic neurons partition into two functional groups: visible and hidden.*
- *During the training phase of the network , the visible neurons are all clamped onto specific states determined by the environment.*
- *The hidden neurons always operate freely; they are used to explain underlying constraints contained in the environmental input vectors.*





Learning Rule: Hebbian

1. *If two neurons on either side of synapse (connection) are activated simultaneously, then the strength of that synapse is selectively increased.*

2. *If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.*

A Hebbian synapse increases its strength with positively correlated presynaptic and postsynaptic signals, and decreases its strength when signals are either uncorrelated or negatively correlated.

Learning Rule: Competitive

- *The output neurons of a neural network compete among themselves to become active.*
- *- a set of neurons that are all the same (excepts for synaptic weights)*
- *- a limit imposed on the strength of each neuron*
- *- a mechanism that permits the neurons to compete -> a winner-takes-all*

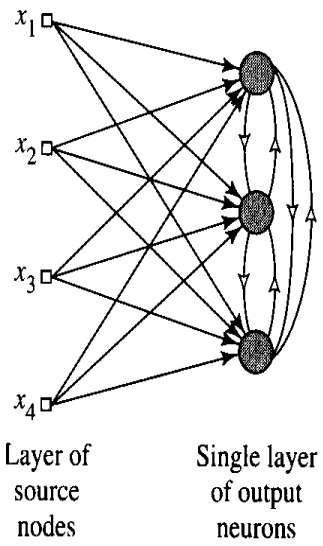


FIGURE 2.4 Architectural graph of a simple competitive learning network with feedforward (excitatory) connections from the source nodes to the neurons, and lateral (inhibitory) connections among the neurons; the lateral connections are signified by open arrows.

- *The standard competitive learning rule*
- $\Delta w_{kj} = \eta(x_j - w_{kj})$ *if neuron k wins the competition*
  $= 0$ *if neuron k loses the competition*

Learning Paradigms : Learning with a Teacher

Learning with a Teacher (=supervised learning)

The teacher has knowledge of the environment

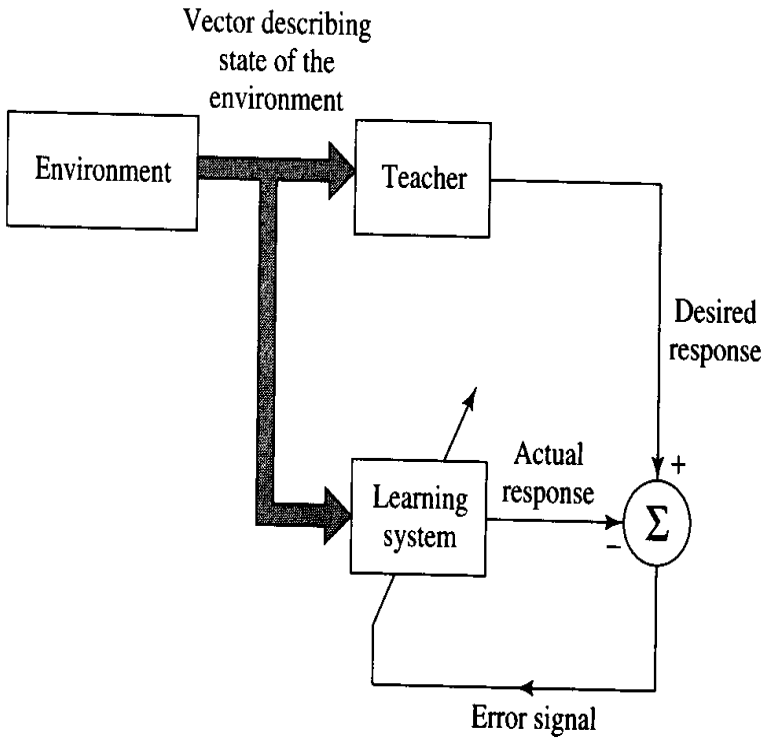


FIGURE 2.6 Block diagram of learning with a teacher.

Learning Paradigms: Learning without a Teacher

Learning without a Teacher: no labeled examples available of the function to be learned.

1) Reinforcement learning

2) Unsupervised learning

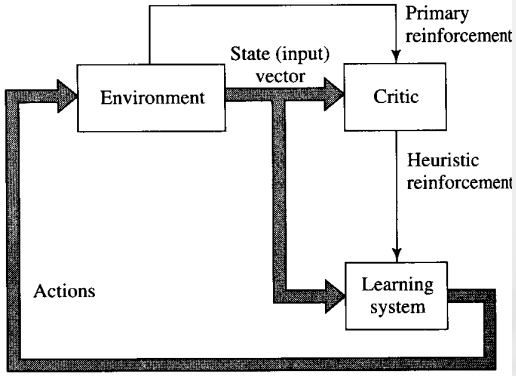


FIGURE 2.7 Block diagram of reinforcement learning.

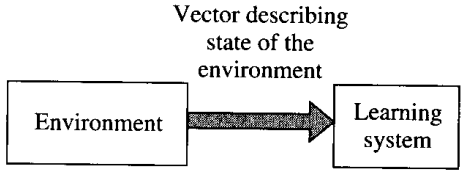


FIGURE 2.8 Block diagram of unsupervised learning.

Learning Paradigms: Reinforcement

Reinforcement learning:

The learning of input-output mapping is performed through continued interaction with the environment in oder to minimize a scalar index of performance.

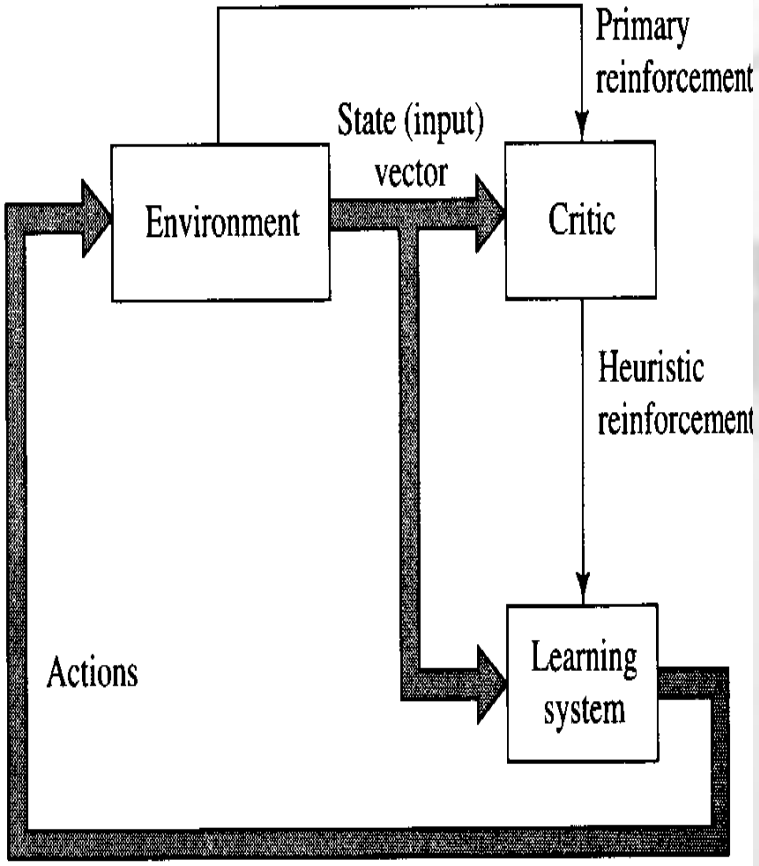


FIGURE 2.7 Block diagram of reinforcement learning.

Learning Paradigms: Unsupervised

- *Unsupervised Learning: There is no external teacher or critic to oversee the learning process.*

- *The provision is made for a task independent measure of the quality of representation that the network is required to learn.*

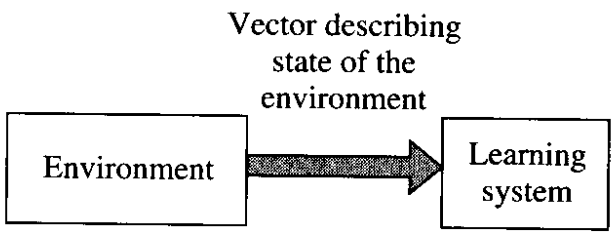


FIGURE 2.8 Block diagram of unsupervised learning.

36

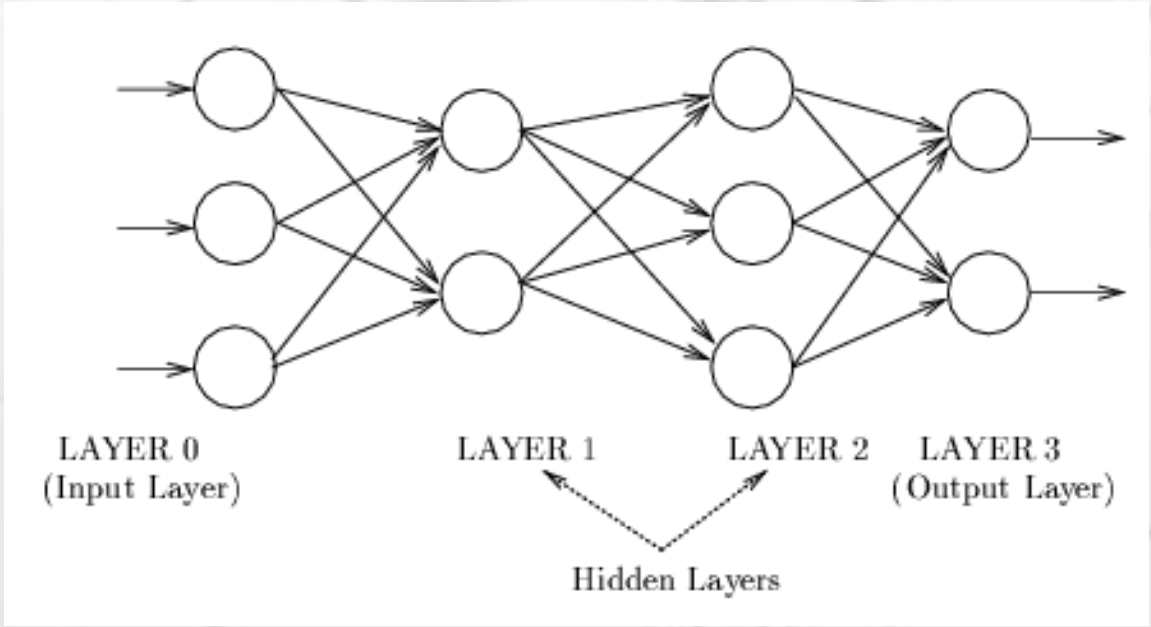
Training of Multilayer Feed-Forward Neural Networks : Back-propagtaion

37

Multilayer Feed-Forward Neural Networks

Feed -forward Networks

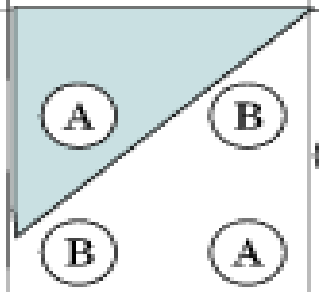
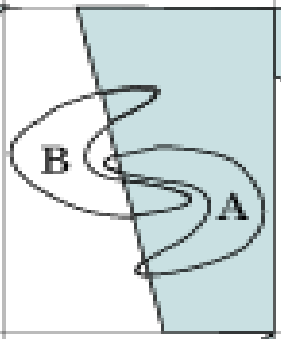
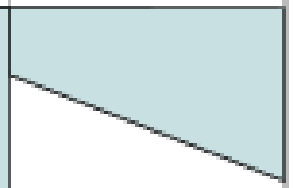
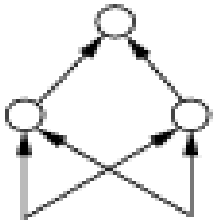
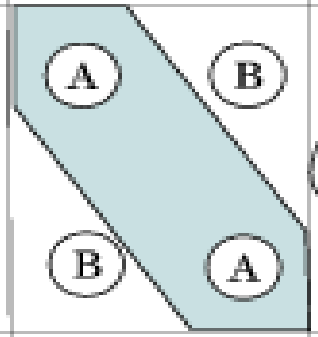
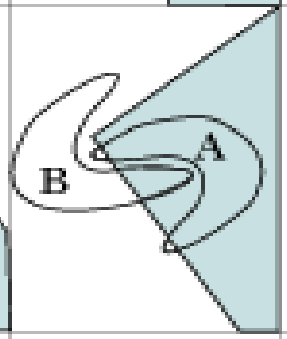
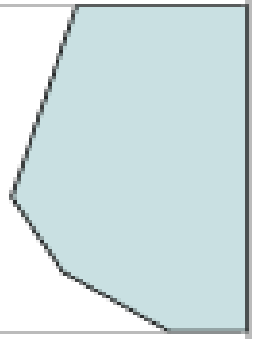
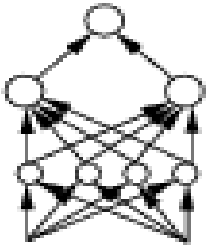
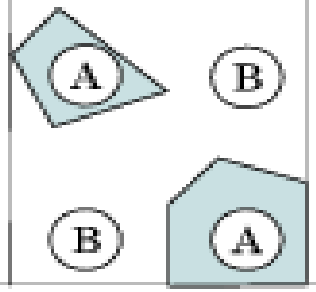
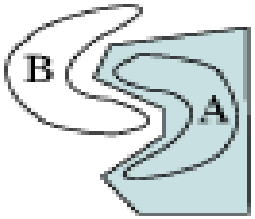
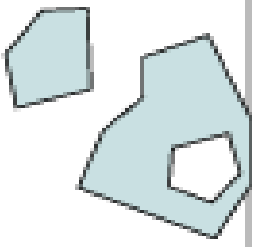
- ***A connection is allowed from a node in layer* i *only to nodes in layer* i + *1.***
- ***Most widely used architecture.***



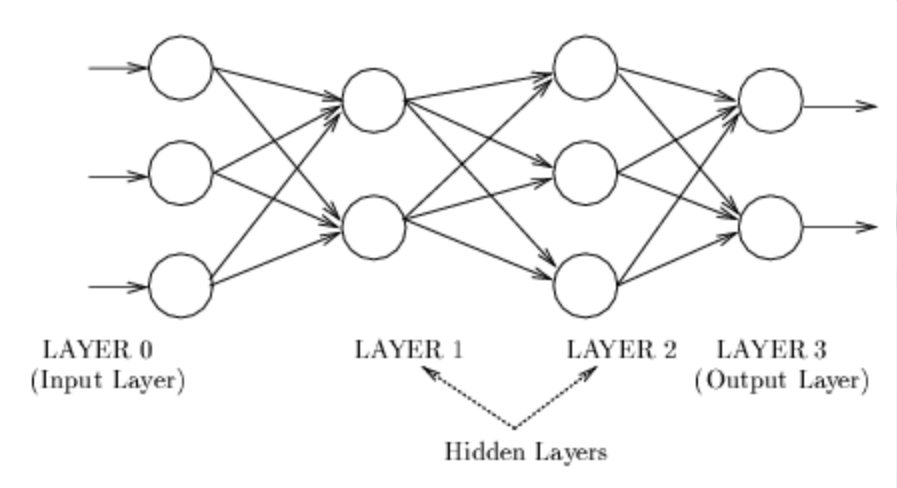
Conceptually, nodes at higher levels successively abstract features from preceding layers

38

Geometric interpolation of the role of HN

Structure	Types of decision regions	Exclusive OR problem	Classes with meshed regions	Most general region shapes
Single-layer 	Half Plane (Bounded by hyperplane)			
Two-layer 	Convex (Open or closed regions)			
Three-layer 	Arbitrary (Complexity limited by number of neurons)			

How to get appropriate weight set?



Back-propagation is the famous algorithm for training feed-forward networks

http://en.wikipedia.org/wiki/Backpropagation

Backpropagation, or propagation of error, is a common method of teaching artificial neural networks how to perform a given task.
It was first described by Paul Werbos in 1974, but it wasn't until 1986, through the work of David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, that it gained recognition, and it led to a **"renaissance"** in the field of artificial neural network research.

letters to nature

Nature **323**, 533 - 536 (09 October 1986); doi:10.1038/323533a0

Learning representations by back-propagating errors

DAVID E. RUMELHART*, GEOFFREY E. HINTON† & RONALD J. WILLIAMS*

*Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA

†Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

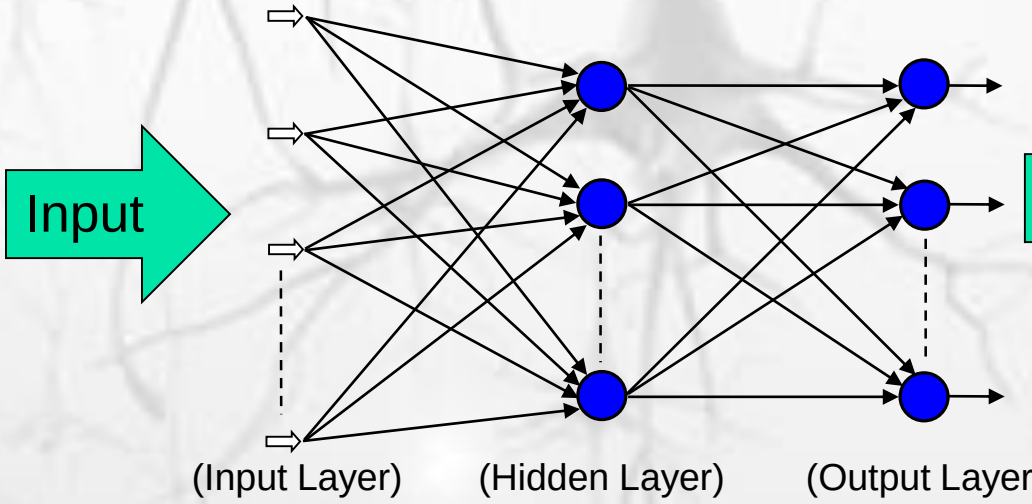
†To whom correspondence should be addressed.

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

References

1. Rosenblatt, F. *Principles of Neurodynamics* (Spartan, Washington, DC, 1961).
2. Minsky, M. L. & Papert, S. *Perceptrons* (MIT, Cambridge, 1969).
3. Le Cun, Y. *Proc. Cognitiva* **85**, 599–604 (1985).

Back-propagation(BP)

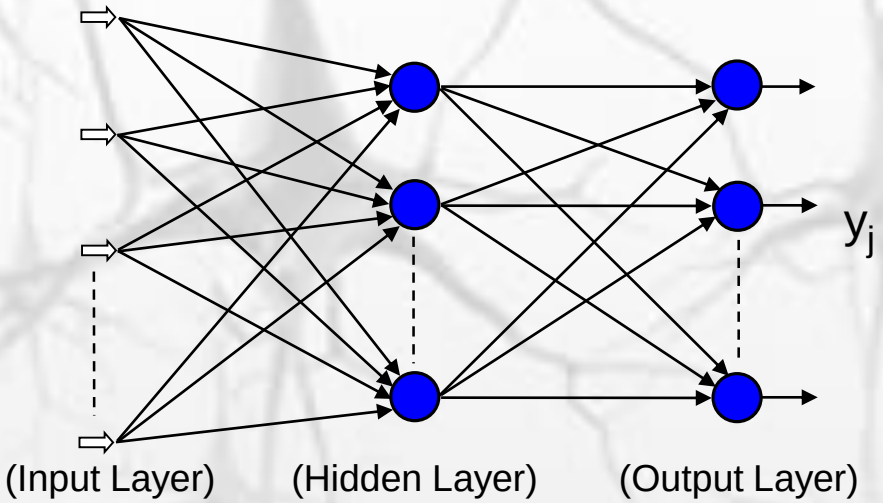


Error =
Desired Output
- Actual Output

Forward Pass: Error is calculated based on desired and actual output.

Backward Pass: Synaptic weights are adjusted based on calculated error.

Back-propagation



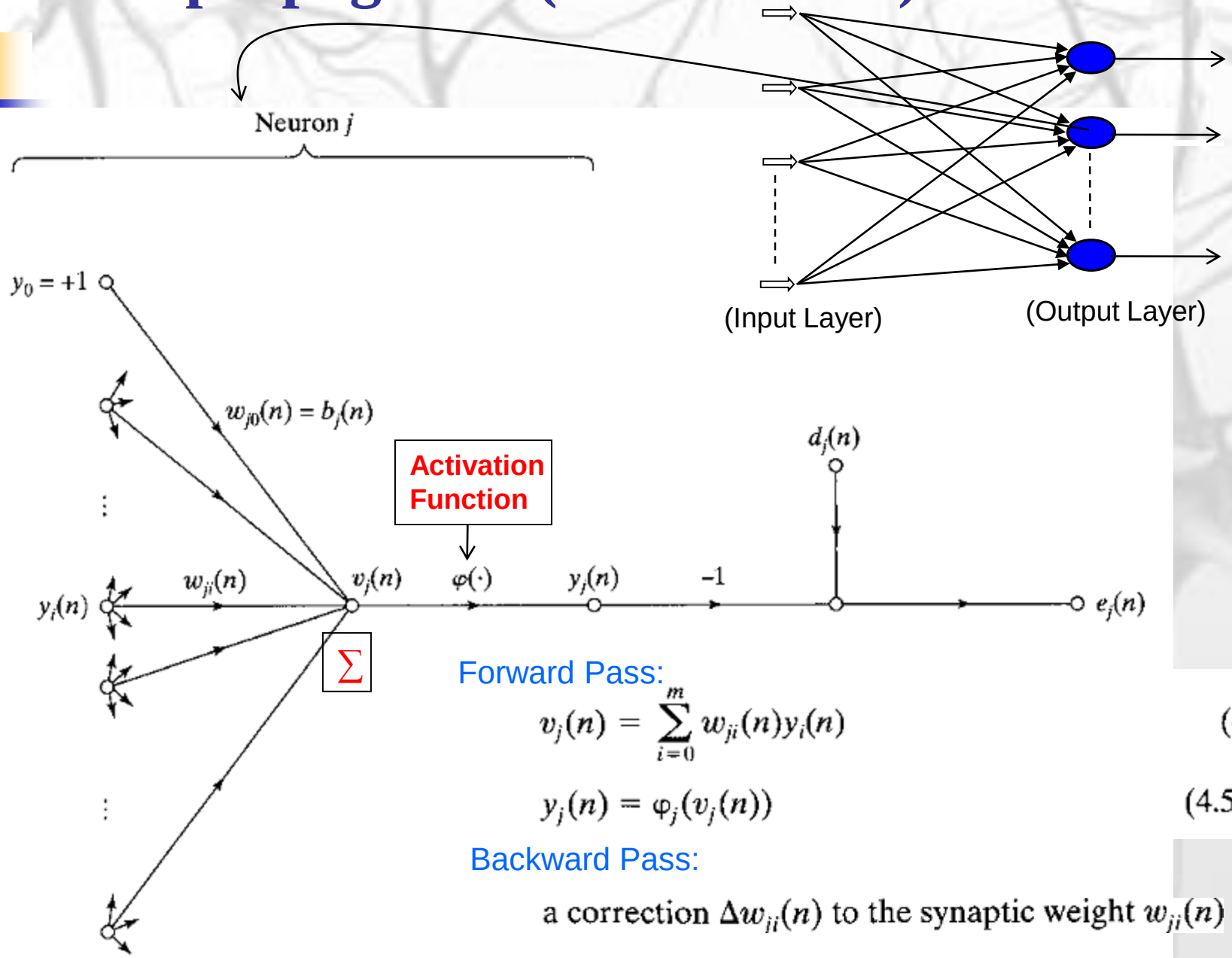
$$e_j(n) = d_j(n) - y_j(n), \qquad \text{neuron } j \text{ is an output node} \tag{4.1}$$

$$\mathscr{E}(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{4.2}$$

$$\mathscr{E}_{\text{av}} = \frac{1}{N}\sum_{n=1}^{N} \mathscr{E}(n) \tag{4.3}$$

performance. The objective of the learning process is to adjust the free parameters of the network to minimize $\mathscr{E}_{\text{av}}$. ~~To do this minimization, we use an approximation similar~~

Back-propagation (Without HN)



Forward Pass:

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \qquad (4.4)$$

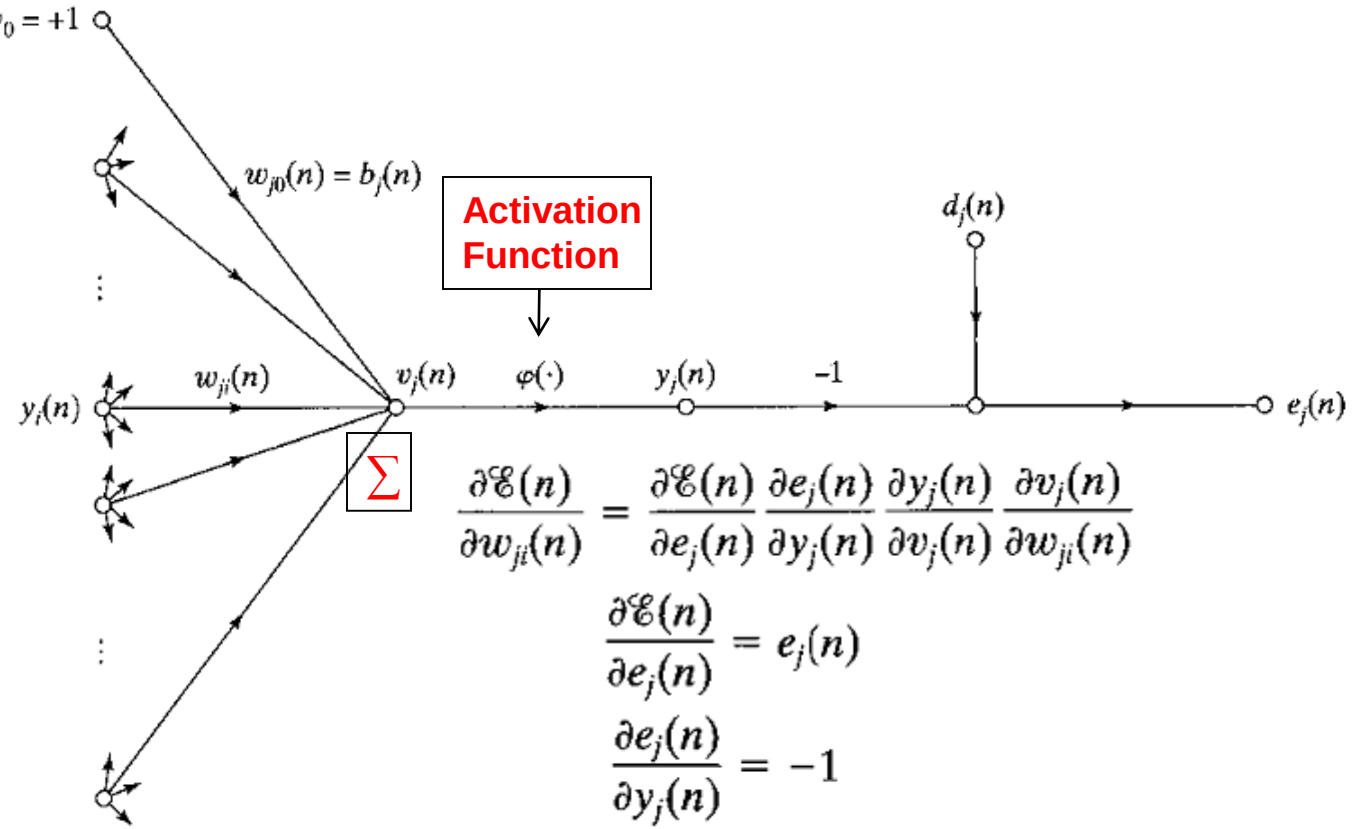
$$y_j(n) = \varphi_j(v_j(n)) \qquad (4.5)$$

Backward Pass:

a correction $\Delta w_{ji}(n)$ to the synaptic weight $w_{ji}(n)$

FIGURE 4.3 Signal-flow graph highlighting the details of output neuron j.

44



$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{4.6}$$

$$\frac{\partial \mathscr{E}(n)}{\partial e_j(n)} = e_j(n) \tag{4.7}$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{4.8}$$

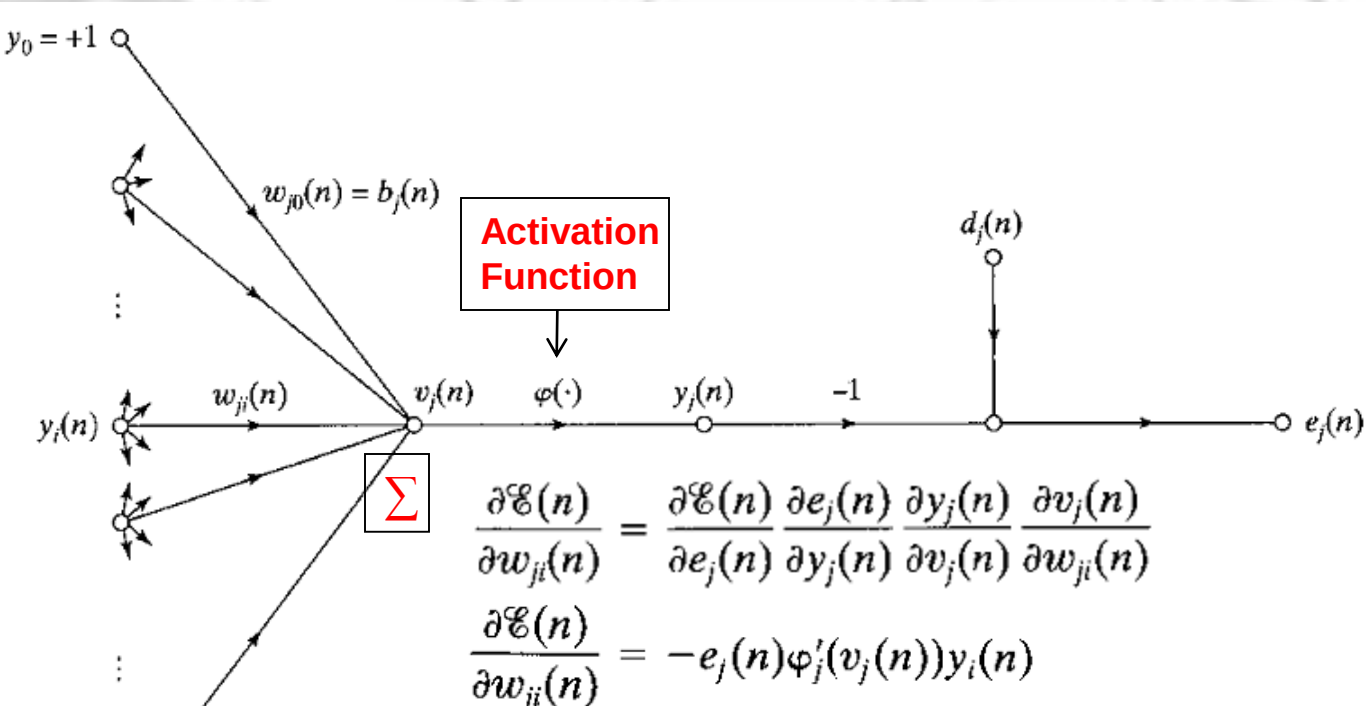
$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)) \tag{4.9}$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \tag{4.10}$$

$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n) \tag{4.11}$$

Depends on Activation Function

45



$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{4.6}$$

$$\frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n) \tag{4.11}$$

The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the *delta rule*:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathscr{E}(n)}{\partial w_{ji}(n)} \tag{4.12}$$

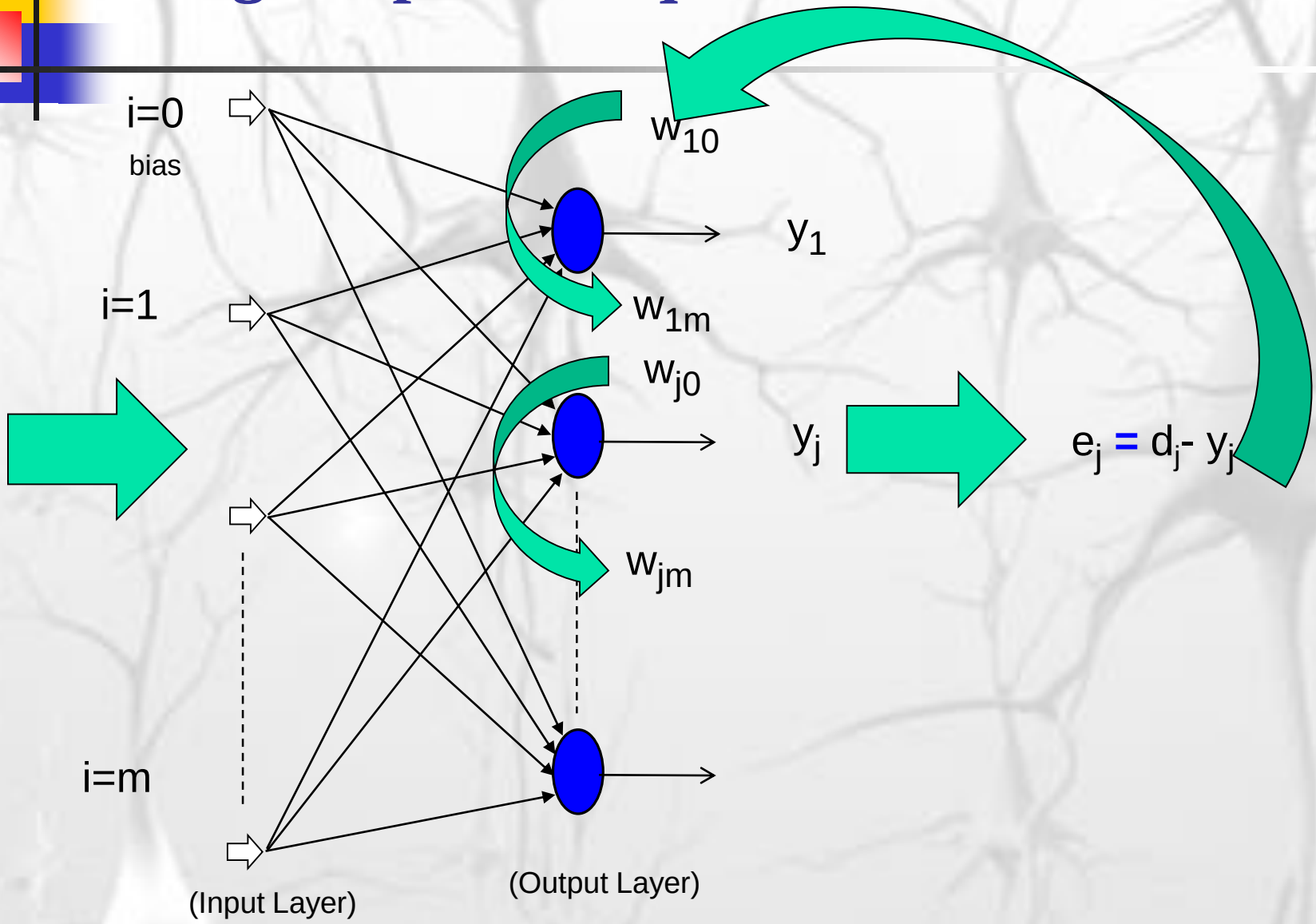
where η is the *learning-rate parameter* of the back-propagation algorithm. The use of the minus sign in Eq. (4.12) accounts for *gradient descent* in weight space (i.e., seeking a direction for weight change that reduces the value of $\mathscr{E}(n)$). Accordingly, the use of Eq. (4.11) in (4.12) yields

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{4.13}$$

where the *local gradient* $\delta_j(n)$ is defined by

$$\delta_j(n) = -\frac{\partial \mathscr{E}(n)}{\partial v_j(n)} = -\frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n)\varphi_j'(v_j(n)) \tag{4.14}$$

Weight Update Sequence in BP



BP for Multilayer (With HN)

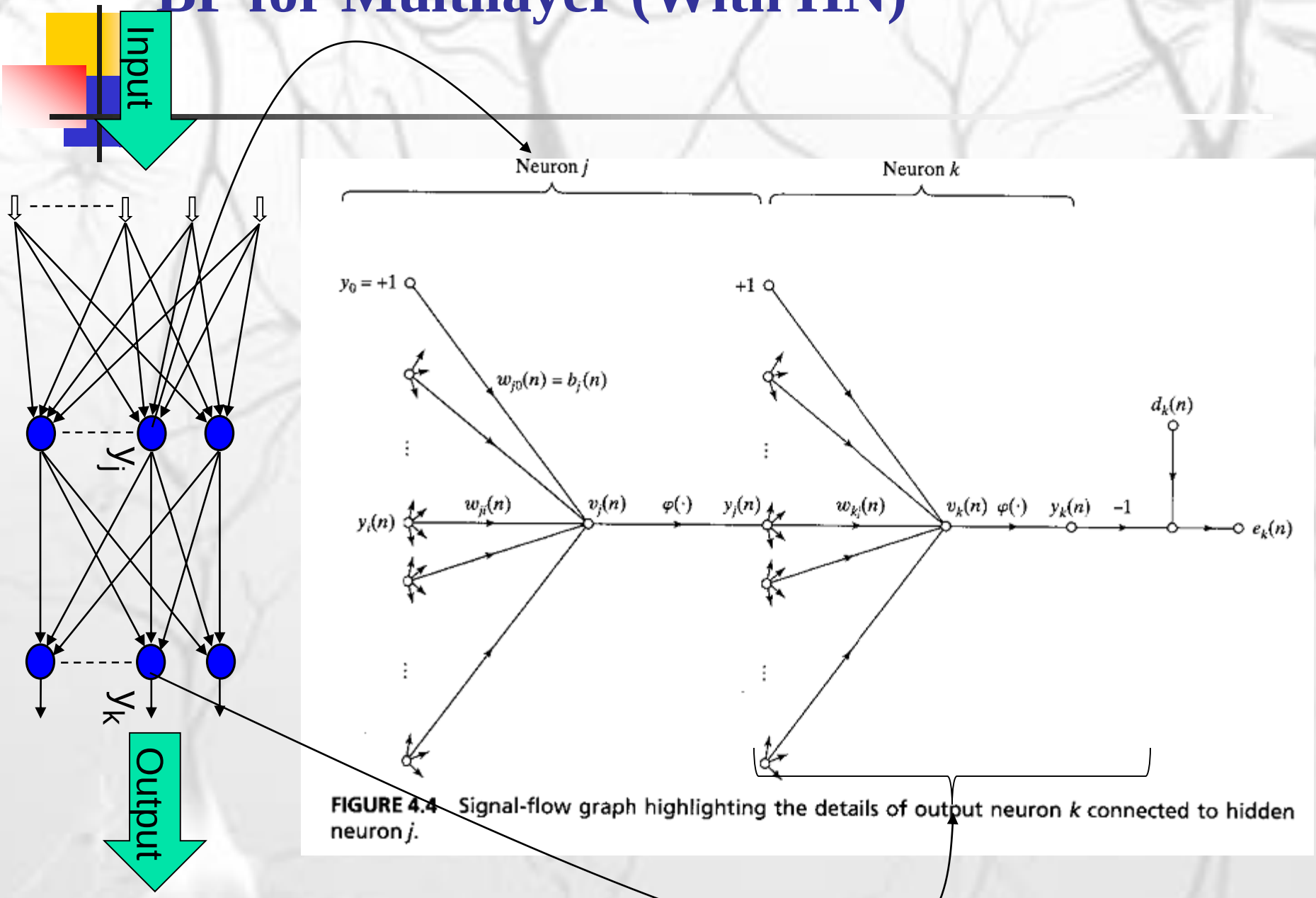
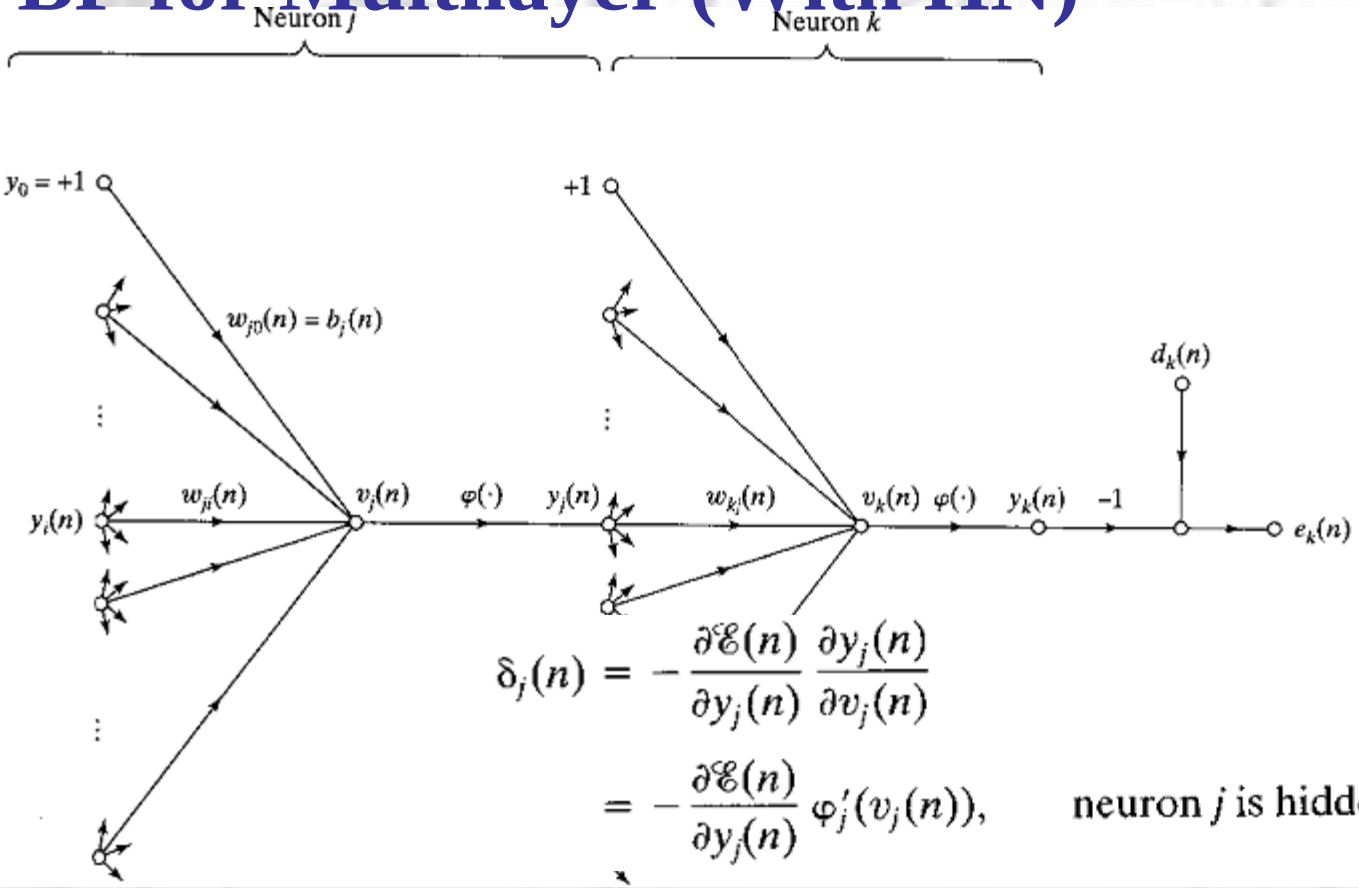


FIGURE 4.4 Signal-flow graph highlighting the details of output neuron k connected to hidden neuron j.

BP for Multilayer (With HN)



$$\delta_j(n) = -\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} \varphi_j'(v_j(n)), \qquad \text{neuron } j \text{ is hidden} \tag{4.15}$$

$$\mathscr{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n), \qquad \text{neuron } k \text{ is an output node} \tag{4.16}$$

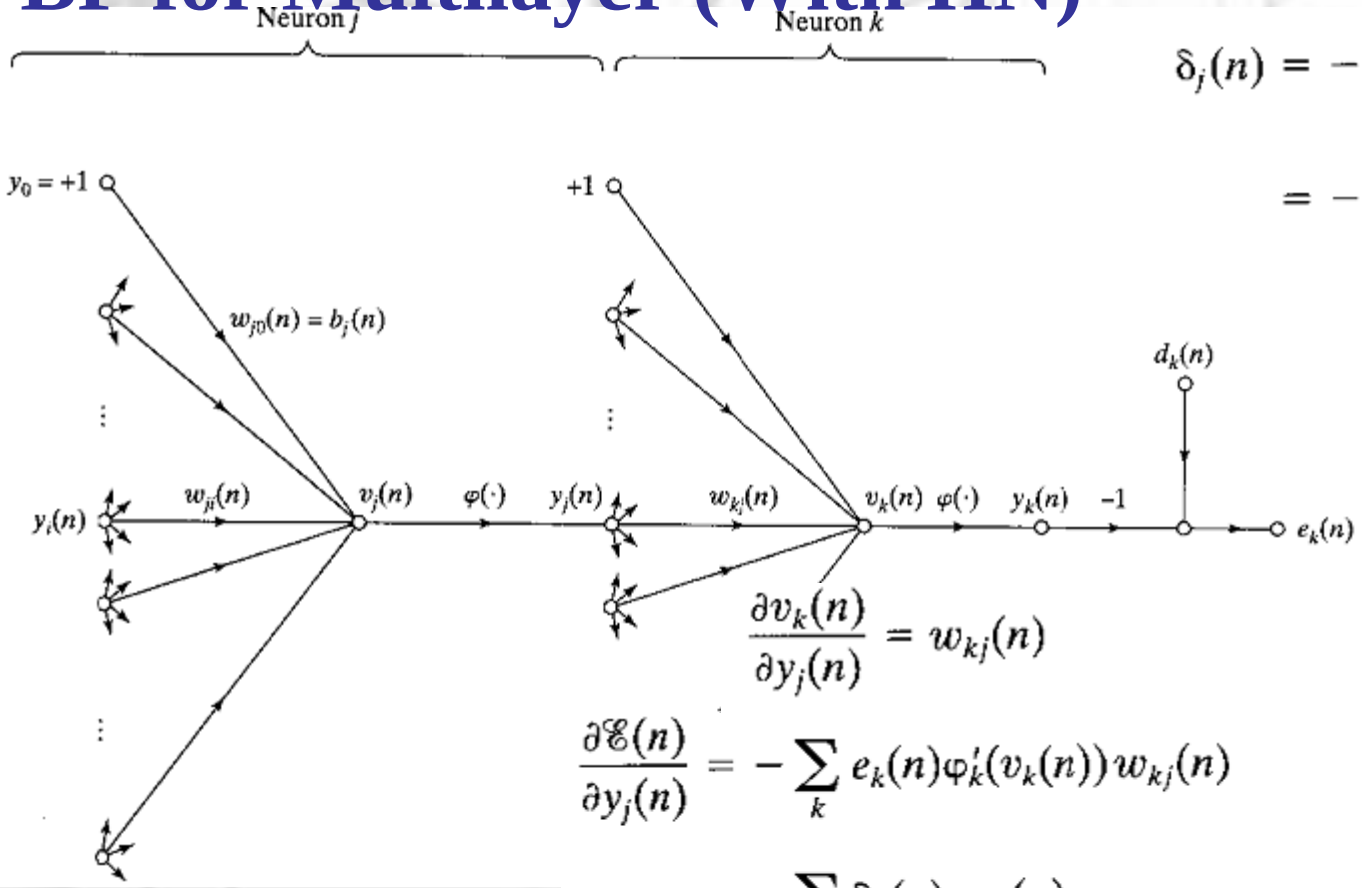
$$\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \tag{4.17}$$

$$\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \tag{4.18}$$

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n)), \tag{4.19}$$

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n) y_j(n) \tag{4.21}$$

BP for Multilayer (With HN)



$$\delta_j(n) = -\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}$$

$$= -\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} \varphi_j'(v_j(n)),$$

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \tag{4.22}$$

$$\frac{\partial \mathscr{E}(n)}{\partial y_j(n)} = -\sum_k e_k(n)\varphi_k'(v_k(n))w_{kj}(n) \tag{4.23}$$

$$= -\sum_k \delta_k(n)w_{kj}(n)$$

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n)w_{kj}(n), \qquad \text{neuron } j \text{ is hidden} \tag{4.24}$$

$$\begin{pmatrix} \textit{Weight} \\ \textit{correction} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \textit{learning-} \\ \textit{rate parameter} \\ \eta \end{pmatrix} \cdot \begin{pmatrix} \textit{local} \\ \textit{gradient} \\ \delta_j(n) \end{pmatrix} \cdot \begin{pmatrix} \textit{input signal} \\ \textit{of neuron } j \\ y_i(n) \end{pmatrix} \tag{4.25}$$

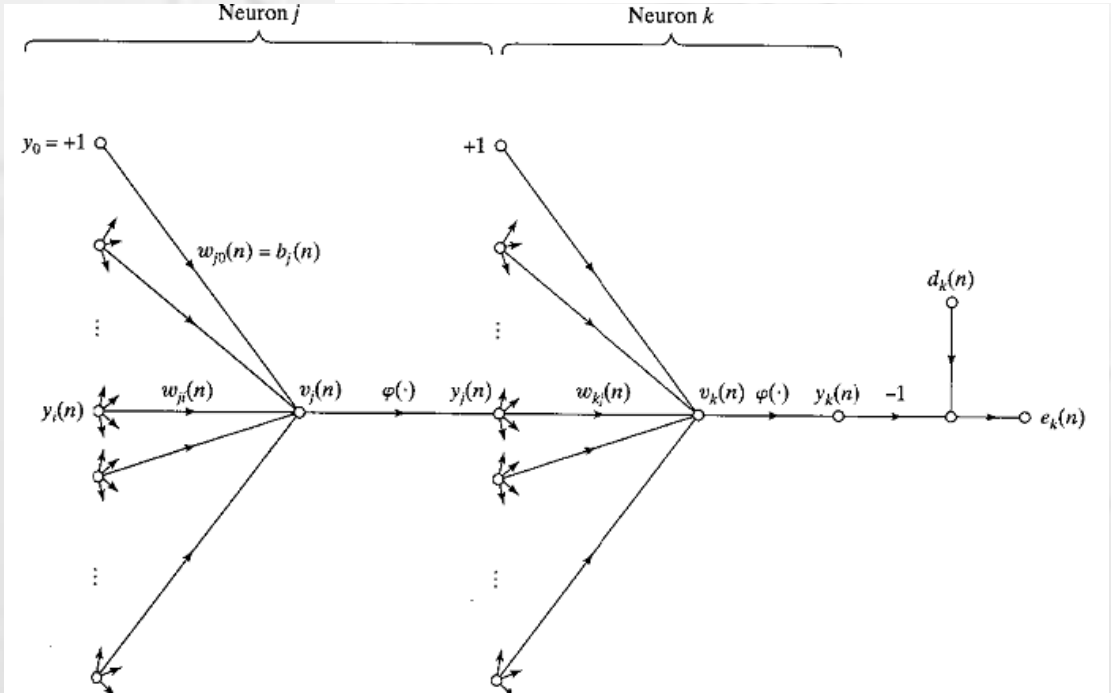
BP for Multilayer (With HN)

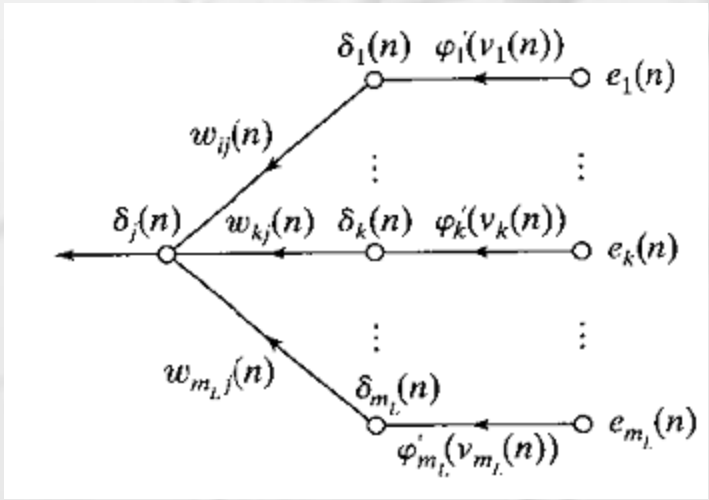
$$w_{ji}(t+1)=w_{ji}(t)+\Delta w_{ji}$$

$$\begin{pmatrix}\text{Weight}\\ \text{correction}\\ \Delta w_{ji}(n)\end{pmatrix}=\begin{pmatrix}\text{learning-}\\ \text{rate parameter}\\ \eta\end{pmatrix}\cdot\begin{pmatrix}\text{local}\\ \text{gradient}\\ \delta_j(n)\end{pmatrix}\cdot\begin{pmatrix}\text{input signal}\\ \text{of neuron } j\\ y_i(n)\end{pmatrix}$$

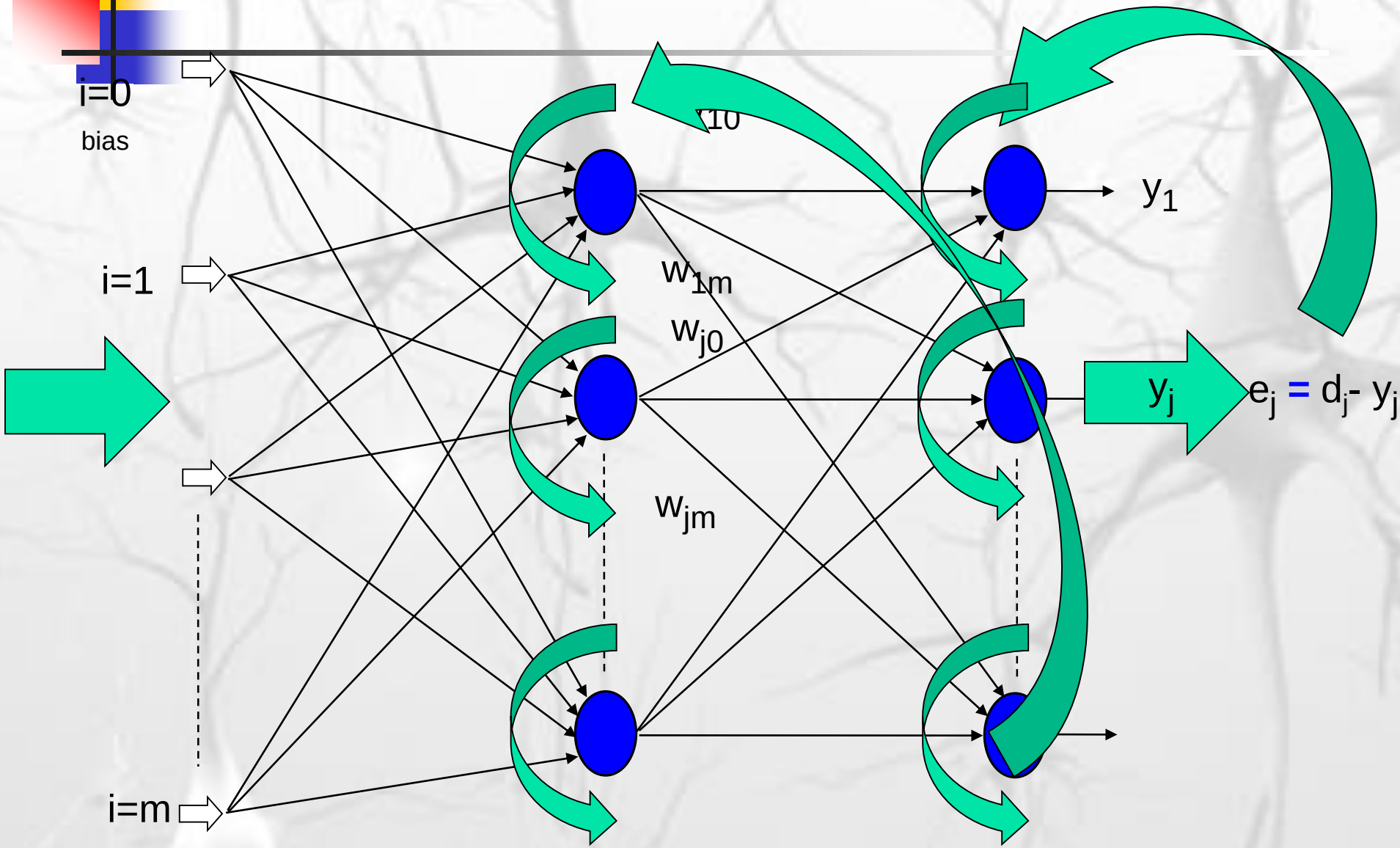
$\delta_k(n)=e_k(n)\varphi'(v_k(n))$ for k output neuron

$$\delta_j(n) = \varphi_j'(v_j(n))\sum_k \delta_k(n)w_{kj}(n), \qquad \text{neuron } j \text{ is hidden}$$

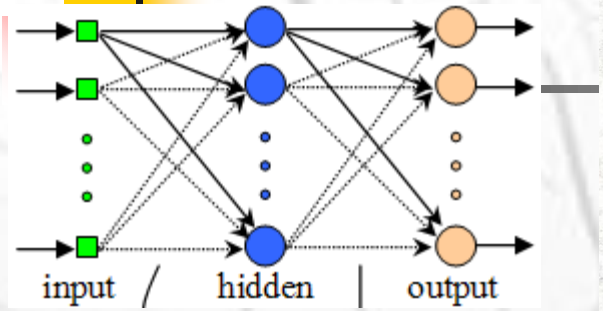




Weight Update Sequence in BP



Matter of Activation Function(AF)



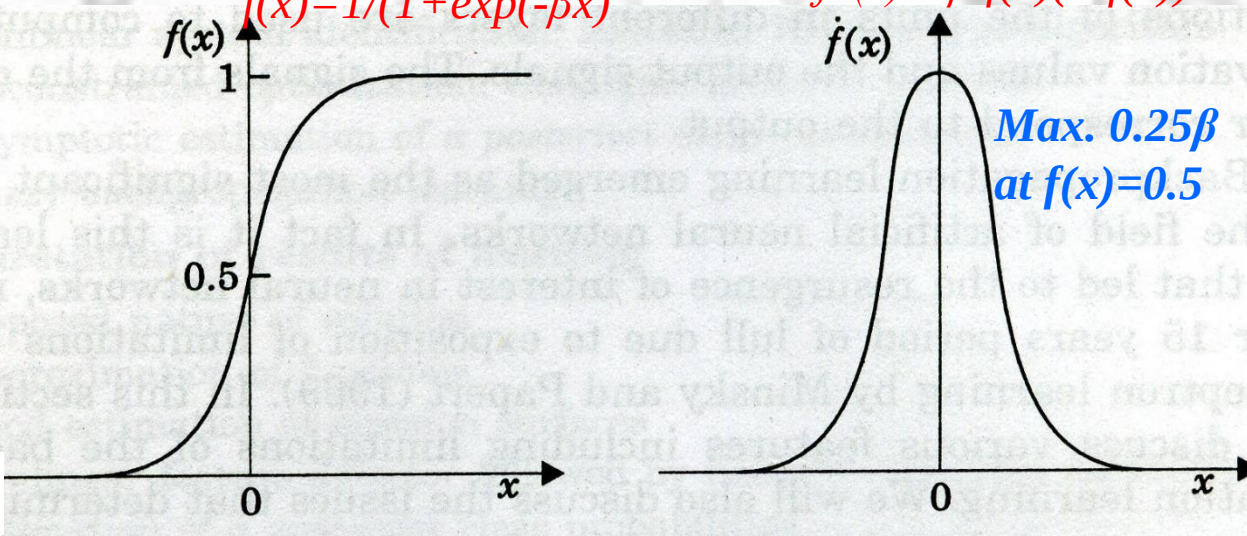
$$\delta_k(n) = e_k(n)\varphi'(v_k(n))$$

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

Differentiability is the only condition for AF

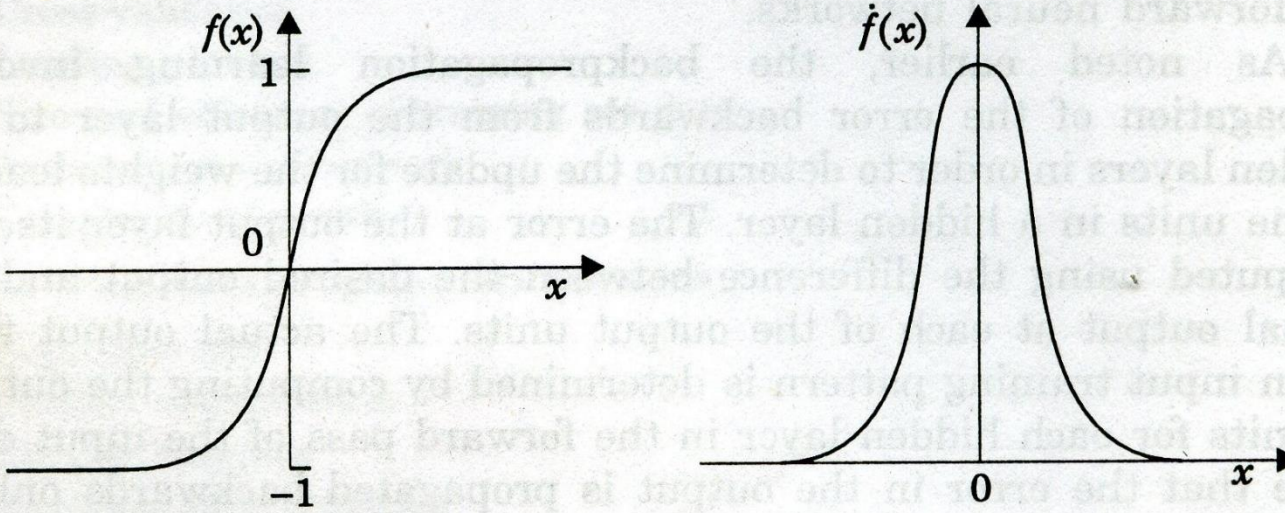
an example of continuously differentiable nonlinear AF is sigmoidal nonlinearity; its two forms are :

$f(x)=1/(1+\exp(-\beta x))$ $\quad f'(x)=\beta f(x)(1-f(x))$



(a) Logistic function and its derivative

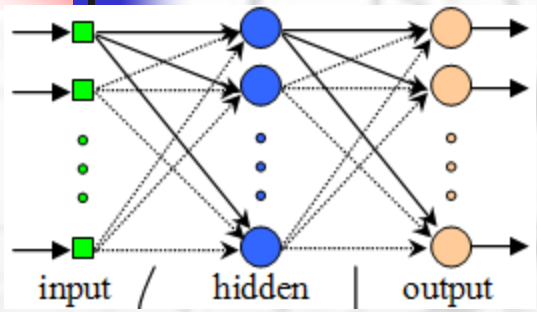
$f(x) = \tanh \beta x$ $\quad \dot{f}(x) = \beta(1 - f^2(x))$



(b) Hyperbolic tangent function and its derivative

53

BP at a Glance



$$\left(\begin{array}{l}\text{Weight}\\ \text{correction}\end{array}\right) = \left(\begin{array}{l}\text{learning-}\\ \text{rate parameter}\end{array}\right) \cdot \left(\begin{array}{l}\text{local}\\ \text{gradient}\end{array}\right) \cdot \left(\begin{array}{l}\text{input signal}\\ \text{of neuron } j\end{array}\right)$$

$$\Delta w = \eta\ \delta\ x$$

The local gradient of output unit (δ_o) and hidden unit (δ_h) are defined by:

$$\delta_o = -\frac{\partial e}{\partial f_o}\frac{\partial f_o}{\partial x_o} \qquad \delta_h = \sum_o \delta_o w_o \frac{\partial f_h}{\partial x_h}$$

$$e(n) = \frac{1}{2}(d(n) - f_o(n))^2, \qquad \frac{\partial e_o(n)}{\partial f_o(n)} = -(d(n) - f_o(n))$$

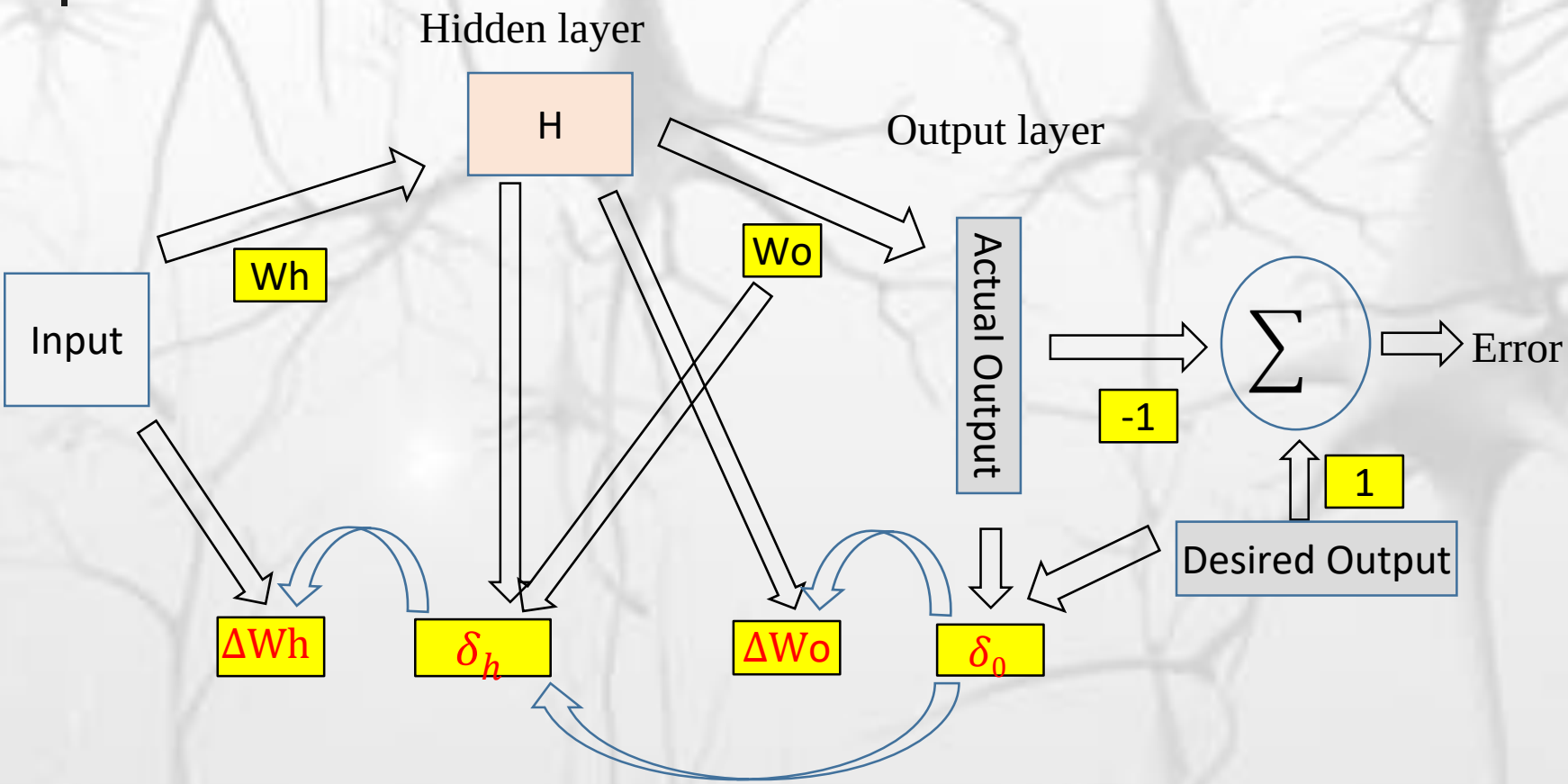
For logistic sigmoid activation function $f_o(n) = 1/(1+\exp(-x_o(n)))$ $\dfrac{\partial f_o(n)}{\partial x_o(n)} = f_o(n)(1 - f_o(n))$

Now the local gradient of output unit (δ_o) becomes $\delta_o = (d(n) - f_o(n))f_o(n)(1 - f_o(n))$

For the same sigmoid activation function the local gradient of hidden unit (δ_h) becomes $\delta_h = f_h(n)(1 - f_h(n))\sum_o \delta_o w_o$

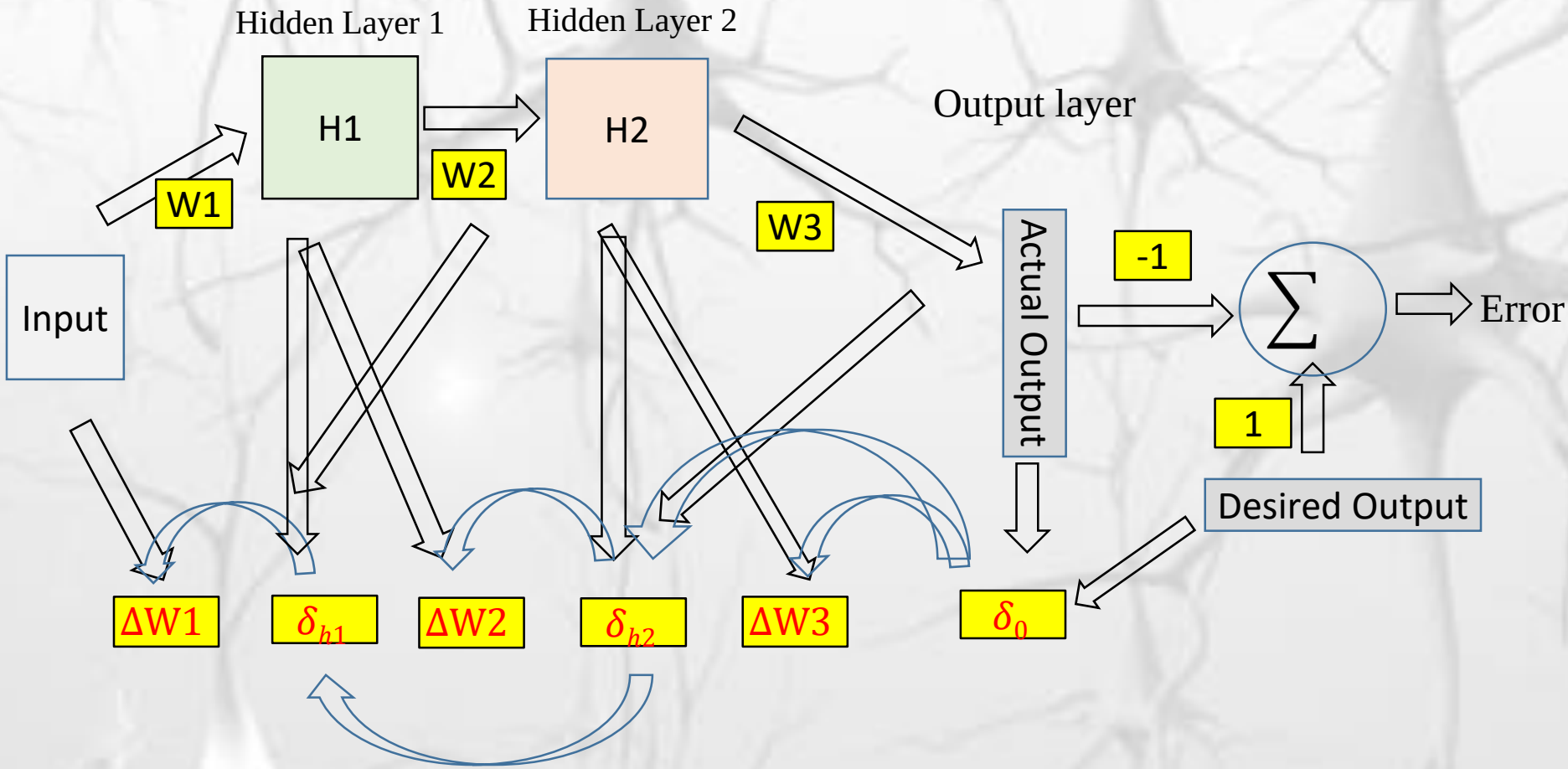
54

Operational flowchart of a NN with Single Hidden Layer



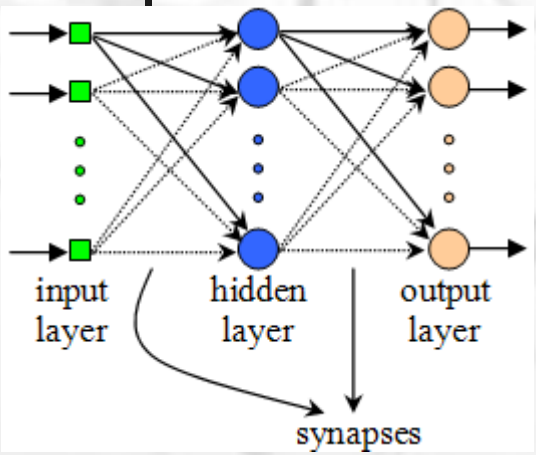
55

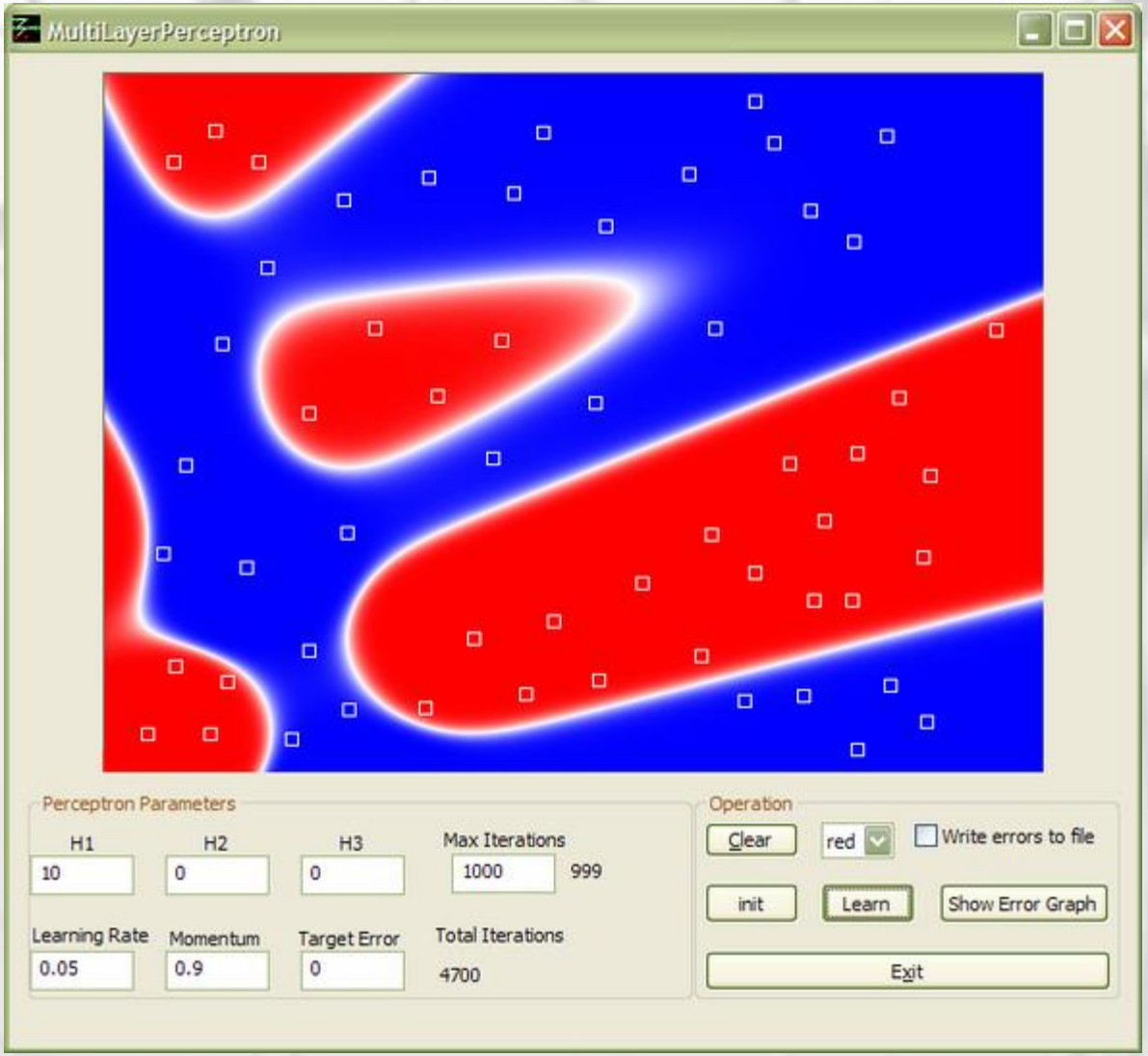
Operational flowchart of a NN with Two Hidden Layer



56

A MLP Simulator for Simple Application





57

Hossein Khosravi : The Developer of the Simulator

https://www.codeproject.com/Articles/9447/Neural-Network-Classifier

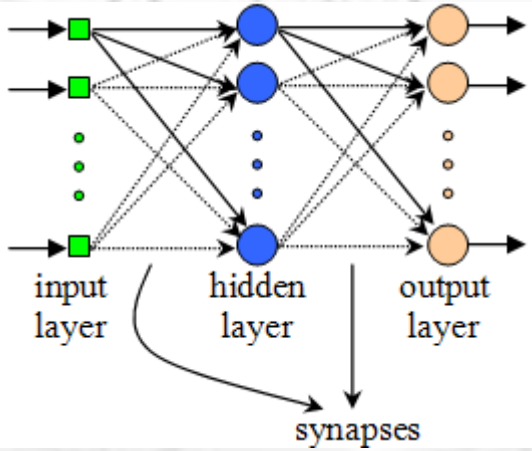
Member Profile: Hossein Khosravi

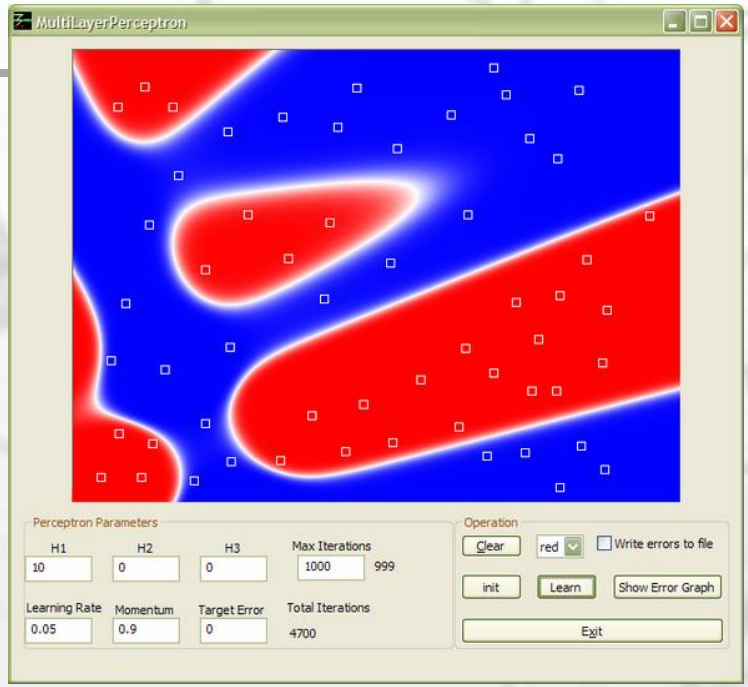
About Member

Friendly Url	http://www.codeproject.com/Members/Hossein-Khosravi
Status	**Silver**. Member No. 429414
	View Member's Blog
Messages Posted	26 - Poster
Articles Submitted	2 - Contributor
Biography	I am an electronic engineer intrested in Pattern Recognition (specially OCR), Neural Networks and Image Processing. I was born in Khezri, a small town in Khorasan, the largest province of Iran, I have taken My BS. from Sharif University of thechnology and by now I'm studying in Tarbiat Modarres University as a Ph.D. candidate.
Location	Iran, Islamic Republic Of
Job Title	Web Developer
Company	
Member since	Sunday, June 08, 2003 (6 years, 4 months)
Homepage	

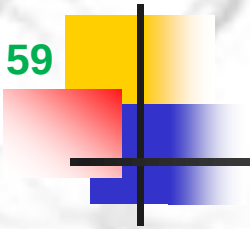
Blog:

58





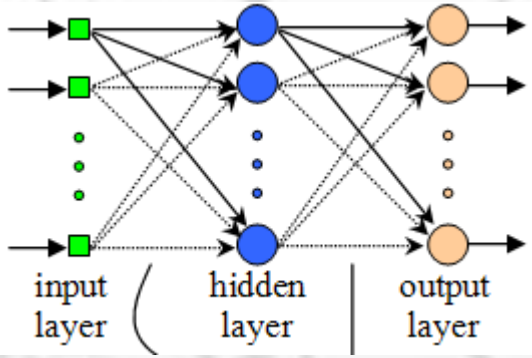
Lets run the simulator



Practical Aspects of BP

60

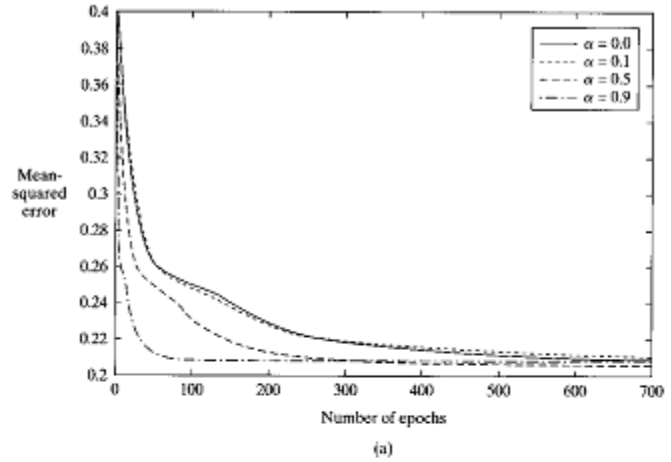
Practical Considerations

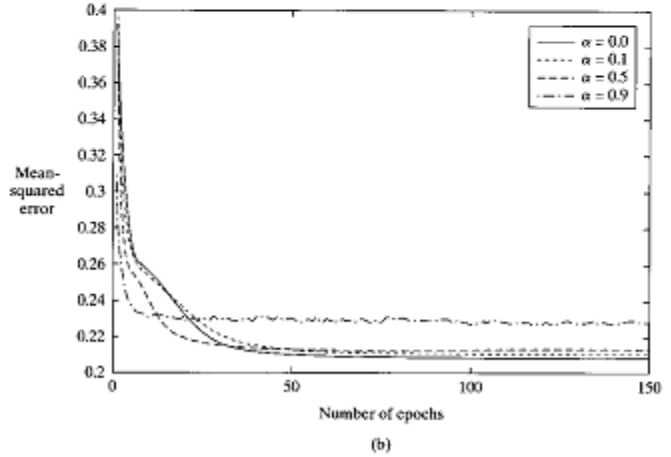


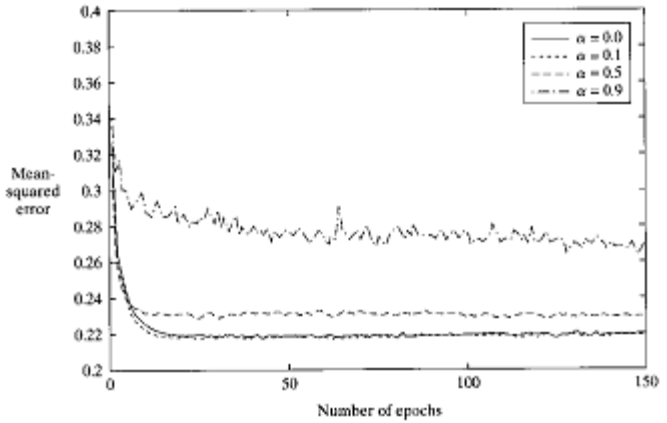
- **Training Data:** Sufficient / proper training data is require for proper input-output mapping.
- **Network Size:** Complex problem require more hidden neurons and may perform better with multiple hidden layers.
- **Weight Initialization:** NN works on numeric data. Before training all the synaptic weights are initialized with small random values, e.g., -0.5 to +0.5.
- **Learning Rate (η) and Momentum Constant (α):** A small η results slower convergence and its larger value gives oscillation. Momentum also speed up training but larger α with large η arise oscillation.
- **Stopping Training:** Training should stop at better generalization position.

61

Effect of Learning Rate (η) and Momentum Constant (α) values







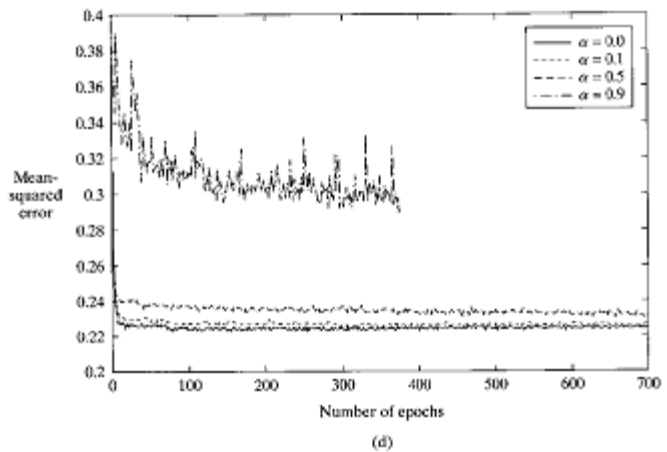


FIGURE 4.15 Ensemble-averaged learning curves for varying momentum α, and the following values of learning-rate parameters: (a) $\eta = 0.01$, (b) $\eta = 0.1$ (c) $\eta = 0.5$, and (d) $\eta = 0.9$.

62

Learning and Generalization

Learning and generalization are the most important topics in NN research. Learning is the ability to approximate the training data while generalization is the ability to predict well beyond the training data.

- Generalization is more desirable because the common use of a NN is to make good prediction on new or unknown objects.
- It measures on the testing set that is reserved from available data and not use in the training.
- Testing error rate (TER), i.e., rate of wrong classification on testing set, is widely acceptable quantifying measure, which value minimum is good.

$$TER = \frac{\text{Total testing set misclassified patterns}}{\text{Total testing set patterns}}$$

63

Learning and Overfitting

Available Data

Training Set
(Use for learning)

Testing Set
(Reserve to measure generalization)

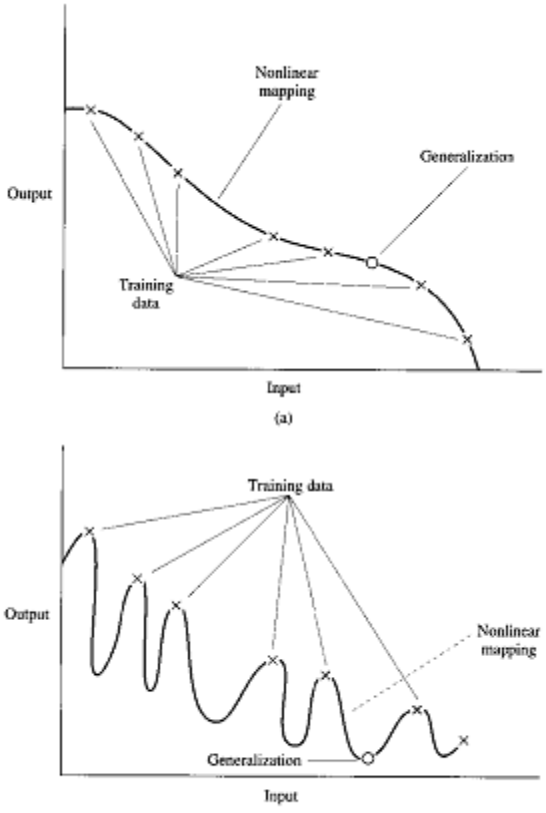
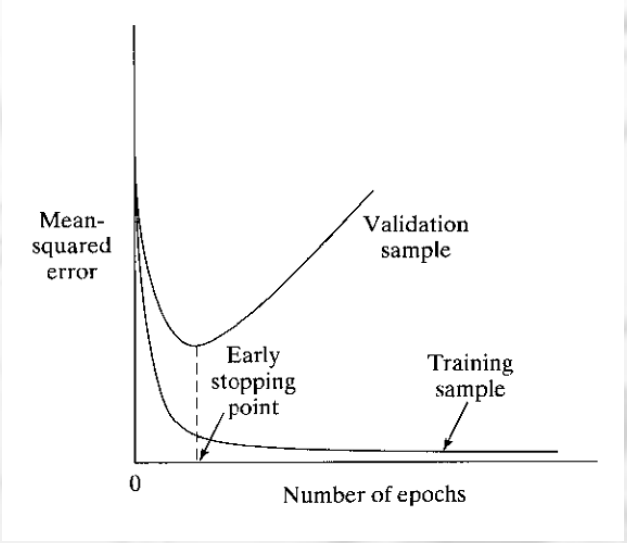
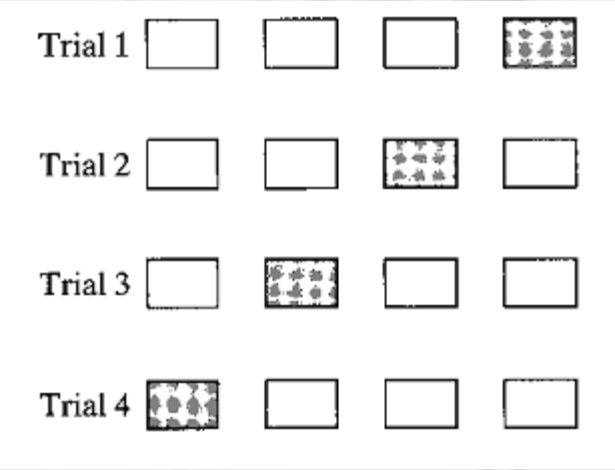


FIGURE 4.19 (a) Properly fitted data (good generalization) (b) Overfitted data (poor generalization).





Where we find benchmark data for testing NNs or machine learning?

64

Benchmark Problems for Evaluation

A benchmark is a point of reference by which something can be measured.

- For NN or machine learning, the most popular benchmark dataset collection is the University of California, Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/).

- UCI contains raw data that require preprocessing to use in NN. Some preprocessed data is also available at Proben1 (ftp://ftp.ira.uka.de/pub/neuron/).

- Various persons or groups also maintain different benchmark datasets for specific purpose: Delve (www.cs.toronto.edu/~delve/data/datasets.html), Orange (www.ailab.si/orange/datasets.asp), etc.

UCI Machine Learning Repository - Windows Internet Explorer

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

UCI Machine Learning Repository

About Citation Policy Donate a Data Set Contact

Search

Repository Web

View ALL Data Sets

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 174 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. Our old web site is still available, for those who prefer the old format. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If you wish to donate a data set, please consult our donation policy. For any other questions, feel free to contact the Repository librarians. We have also set up a mirror site for the Repository.

Supported By: In Collaboration With: Rexa.info

Latest News:

07-23-2008: Repository mirror has been set up.
03-24-2008: New data sets have been added!
06-25-2007: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope
04-13-2007: Research papers that cite the repository have been associated to specific data sets.
04-09-2007: Three new data sets have been added: Poker Hand, Callt2 Building People Counts, Dodgers Loop Sensor.
09-08-2006: The Beta site has been launched.
09-01-2006: SPECTF.test has been modified by the donor.

Featured Data Set: Statlog (Shuttle)

Task: Classification
Data Type: Multivariate
 # Attributes: 9
 # Instances: 58000

The shuttle dataset contains 9 attributes all of which are numerical. Approximately 80% of the data belongs to class 1

Newest Data Sets:

06-26-2008: Parkinsons
04-21-2008: Ozone Level Detection
04-03-2008: Abscisic Acid Signaling Network
03-20-2008: Hill-Valley
03-12-2008: Bag of Words
03-08-2008: Reuters Transcribed Subset
02-29-2008: Gisette
02-29-2008: Dorothea
02-29-2008: Madelon

Most Popular Data Sets (hits since 2007):

39351: Iris
31585: Adult
26458: Wine
23553: Breast Cancer Wisconsin (Diagnostic)
18449: Abalone
18321: Poker Hand
13471: Yeast
12996: Internet Advertisements
11793: SPECT Heart

UCI Machine Learning Repository - Windows Internet Explorer

Thesis | idea of neural ne... | UCI Machine Lea... | FTP directory: /p... | Ensemble of Div... | H-Bibliography.d... | Desktop | 4:39 PM

66

Benchmark Problems

Problems Related to Human Life

Problem	Task
Breast Cancer Wisconsin	Predicts whether a tumor is benign (not dangerous to health) or malignant (dangerous) based on a sample tissue taken from a patient's breast.
BUPA Liver Disorder	Identify lever disorders based on blood tests along with other related information such as alcohol consumption.
Diabetes	Investigate whether the patient shows or not the signs of diabetes.
Heart Disease Cleveland	Predicting whether at least one of four major heart vessels is reduced in diameter by more than 50%.
Hepatitis	Anticipate status (i.e., live or die) of hepatitis patient.
Lymphography	Predict the situation of lymph nodes and lymphatic vessels.
Lungcancer	Identify types of pathological lung cancers.
Postoperative	Determine place to send patients for postoperative recovery.

67

Benchmark Problems

Problems Related to Finance

Problem	Task
Australian Credit Card	Classify people as good or bad credit risks depend on applicants' particulars.
Car	Evaluate cars based on price and facilities.
Labor Negotiations	Identify a worker as good or bad i.e., contract with him beneficial or not.
German Credit Card	Like Australian Card, this problem also concerns to predict the approval or non-approval of a credit card to a customer.

Problems Related to Plants

Problem	Task
Iris Plants	Classify iris plant types.
Mushroom	Identify whether a mushroom is edible or not based on a description of the mushroom's shape, color, odor, and habitat.
Soybean	Recognize 19 different diseases of soybeans.

68

Benchmark Problems and NN Architecture

After Preprocessing

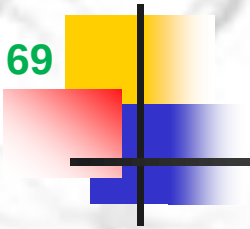
Depends on Problem

Abbr.	Problem	Total Examp	Input Features		NN Architecture		
			Cont.	Discr.	Inputs	Class	Hidd. Node
ACC	Australian Credit Card	690	6	9	51	2	10
BLN	Balance	625	-	4	20	3	10
BCW	Breast Cancer Wisconsin	699	9	-	9	2	5
CAR	Car	1728	-	6	21	4	10
DBT	Diabetes	768	8	-	8	2	5
GCC	German Credit Card	1000	7	13	63	2	10
HDC	Heart Disease Cleveland	303	6	7	35	2	5
HPT	Hepatitis (HPT)	155	6	13	19	2	5
HTR	Hypothyroid	7200	6	15	21	3	5
HSV	House Vote	435	-	16	16	2	5
INS	Ionosphere	351	34	-	34	2	10
KRP	King+Rook vs King+Pawn	3196	-	36	74	2	10
LMP	Lymphography	148	-	18	18	4	10
PST	Postoperative	90	1	7	19	3	5
SBN	Soybean	683	-	35	82	19	25
SNR	Sonar	208	60	-	60	2	10
SPL	Splice Junction	3175	-	60	60	3	10
WIN	Wine	178	13	-	13	3	5
WVF	Waveform	5000	21	-	21	3	10
ZOO	Zoo	101	15	1	16	7	10

Input Features of Diabetes

1. Number of times pregnant
2. Plasma glucose concentration
3. Diastolic blood pressure
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index
7. Diabetes pedigree function
8. Age

❖Problems show variations in number of examples, input features and classes.



An Application of Feed Forward Neural Network Optical Character Recognition(OCR)

70

Optical Character Recognition(OCR)

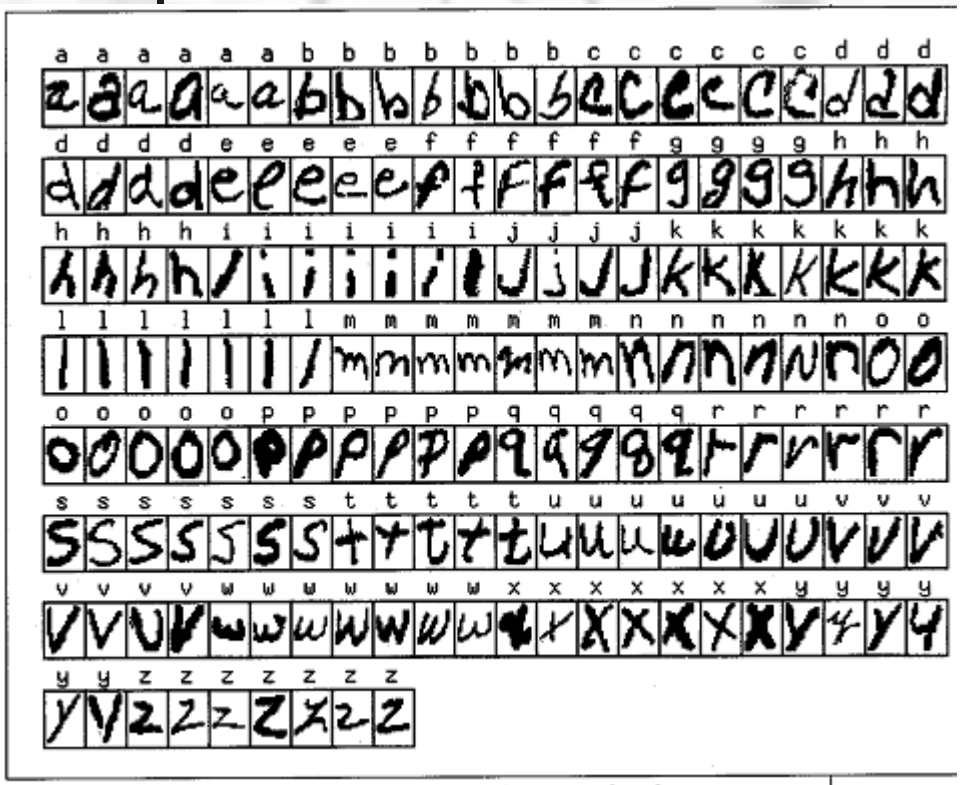


Figure 10. A sample set of characters in the NIST database.

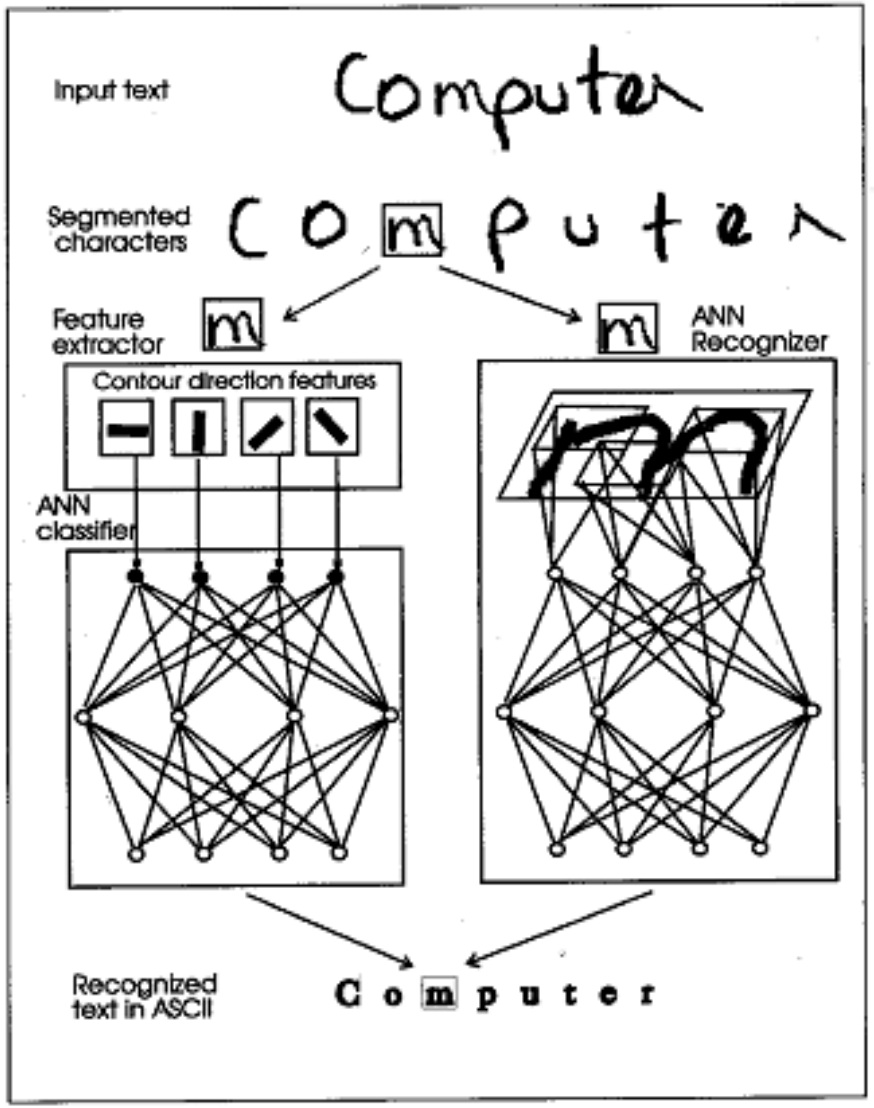
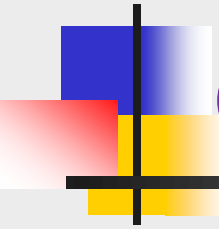


Figure 11. Two schemes for using ANNs in an OCR system.

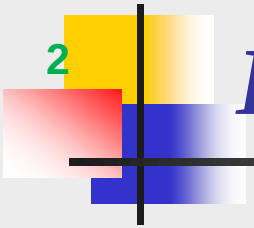
MCSE 666:Pattern and Speech Recognition



Classification with Decision Trees

Dr. Md. Aminul Haque Akhand
Dept. of CSE, SUB

Reference/Source Slides: Maria Simi Michael Crawford



2 Inductive Inference with Decision Trees

- *Decision Trees (DT)* is one of the most widely used and practical methods of *inductive inference*
- Features:
  - Method for approximating *discrete-valued* functions (including boolean)
  - Learned functions are represented as *decision trees* (or *if-then-else* rules)
  - Expressive hypotheses space, including disjunction
  - Robust to noisy data

3

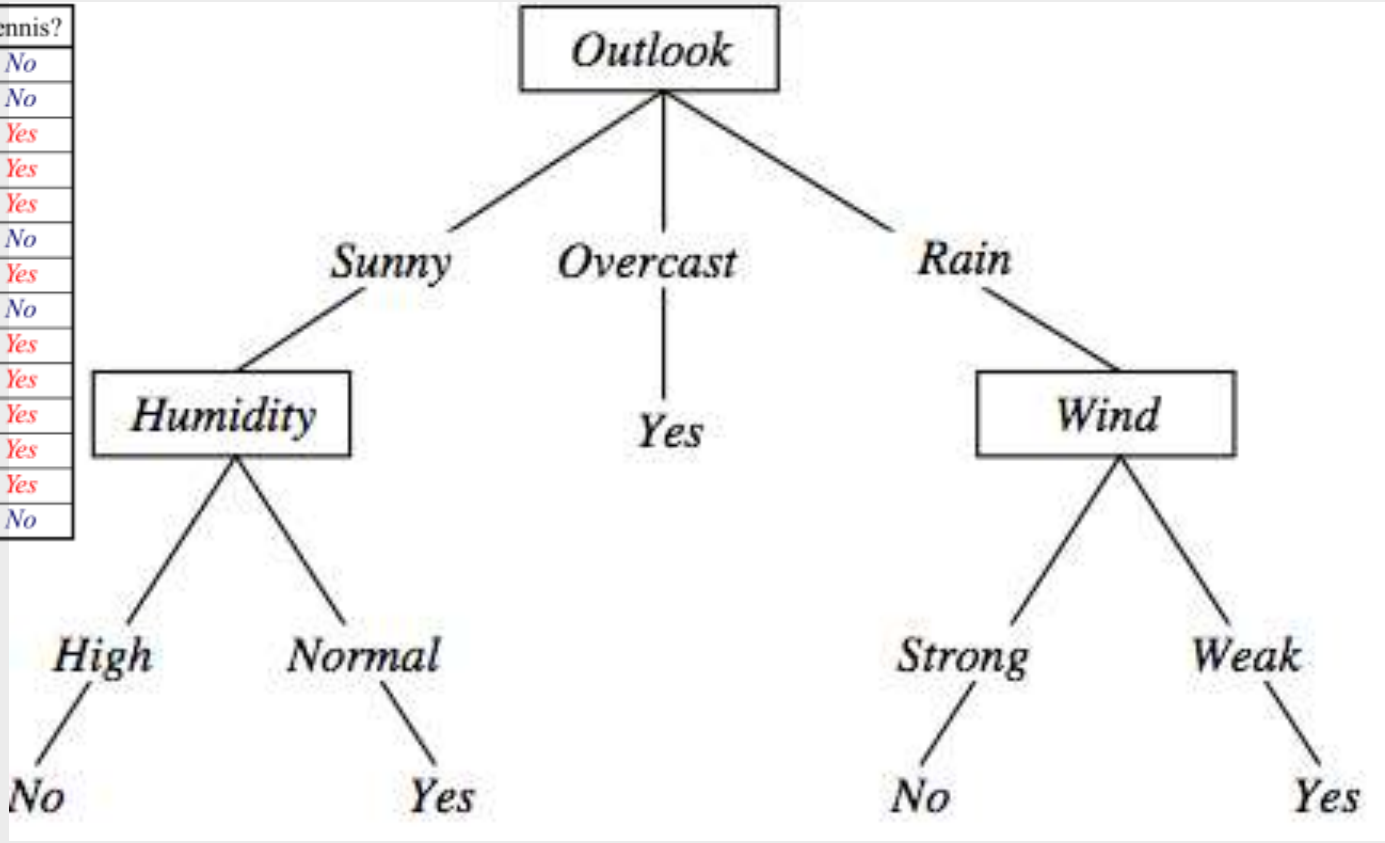
Play Tennis based on Weather Condition

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	*No*
D2	Sunny	Hot	High	Strong	*No*
D3	Overcast	Hot	High	Weak	*Yes*
D4	Rain	Mild	High	Weak	*Yes*
D5	Rain	Cool	Normal	Weak	*Yes*
D6	Rain	Cool	Normal	Strong	*No*
D7	Overcast	Cool	Normal	Strong	*Yes*
D8	Sunny	Mild	High	Weak	*No*
D9	Sunny	Cool	Normal	Weak	*Yes*
D10	Rain	Mild	Normal	Weak	*Yes*
D11	Sunny	Mild	Normal	Strong	*Yes*
D12	Overcast	Mild	High	Strong	*Yes*
D13	Overcast	Hot	Normal	Weak	*Yes*
D14	Rain	Mild	High	Strong	*No*

4

DT Representation (Play Tennis)

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	*No*
D2	Sunny	Hot	High	Strong	*No*
D3	Overcast	Hot	High	Weak	*Yes*
D4	Rain	Mild	High	Weak	*Yes*
D5	Rain	Cool	Normal	Weak	*Yes*
D6	Rain	Cool	Normal	Strong	*No*
D7	Overcast	Cool	Normal	Strong	*Yes*
D8	Sunny	Mild	High	Weak	*No*
D9	Sunny	Cool	Normal	Weak	*Yes*
D10	Rain	Mild	Normal	Weak	*Yes*
D11	Sunny	Mild	Normal	Strong	*Yes*
D12	Overcast	Mild	High	Strong	*Yes*
D13	Overcast	Hot	Normal	Weak	*Yes*
D14	Rain	Mild	High	Strong	*No*



⟨*Outlook=Sunny, Temp=Hot, Humidity=High, Wind=Strong*⟩ No

5

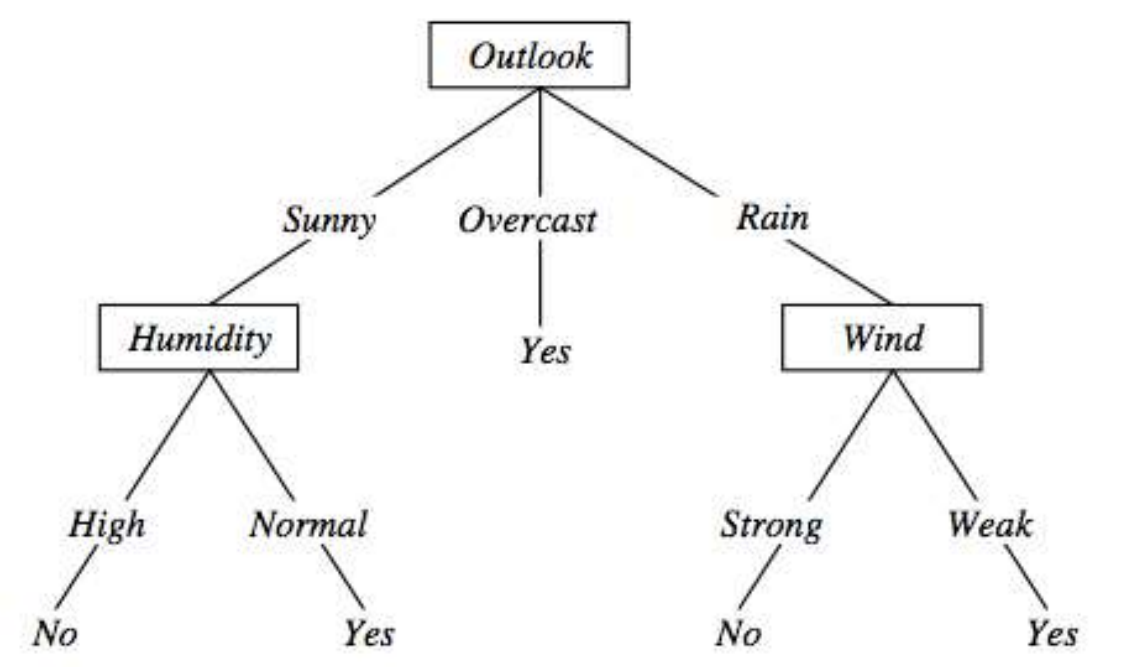
DT Expressivity

DTs represent a disjunction of conjunctions on constraints on the value of attributes:

(*Outlook* = *Sunny* $\wedge$ *Humidity* = *Normal*) $\vee$

(*Outlook* = *Overcast*) $\vee$

(*Outlook* = *Rain* $\wedge$ *Wind* = *Weak*)



6

When to use Decision Trees

- **Problem characteristics:**
  - Instances can be described by attribute value pairs
  - Target function is discrete valued
  - Disjunctive hypothesis may be required
  - Possibly noisy training data samples
    - Robust to errors in training data
    - Missing attribute values
- **Different classification problems:**
  - Medical diagnosis
  - Credit risk analysis
  - ----------

7

Top-Down Induction of DTs

- ID3 (Quinlan, 1986) is a basic algorithm for learning DT's
- Given a training set of examples, the algorithms for building DT performs search in the space of decision trees
- The construction of the tree is top-down. The algorithm is greedy.
- The fundamental question is "which attribute should be tested next? Which question gives us more information?"
- Select the *best* attribute
- A descendent node is then created for each possible value of this attribute and examples are partitioned according to this value
- The process is repeated for each successor node until all the examples are classified correctly or there are no attributes left

8

Which attribute is the best classifier?

- A statistical property called *information gain*, measures how well a given attribute separates the training examples
- Information gain uses the notion of *entropy*, commonly used in information theory
- *Information gain = expected reduction of entropy*

9

Entropy in Binary Classification

Entropy measures the impurity of a collection of examples. It depends from the distribution of the random variable *p*.

- *S* is a collection of training examples
- p_+ the proportion of positive examples in *S*
- p_- the proportion of negative examples in *S*

$$Entropy\ (S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_- \quad [0 \log_2 0 = 0]$$

$$Entropy\ ([14+, 0-]) = -14/14 \log_2 (14/14) - 0 \log_2 (0) = 0$$

$$Entropy\ ([9+, 5-]) = -9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = 0{,}94$$

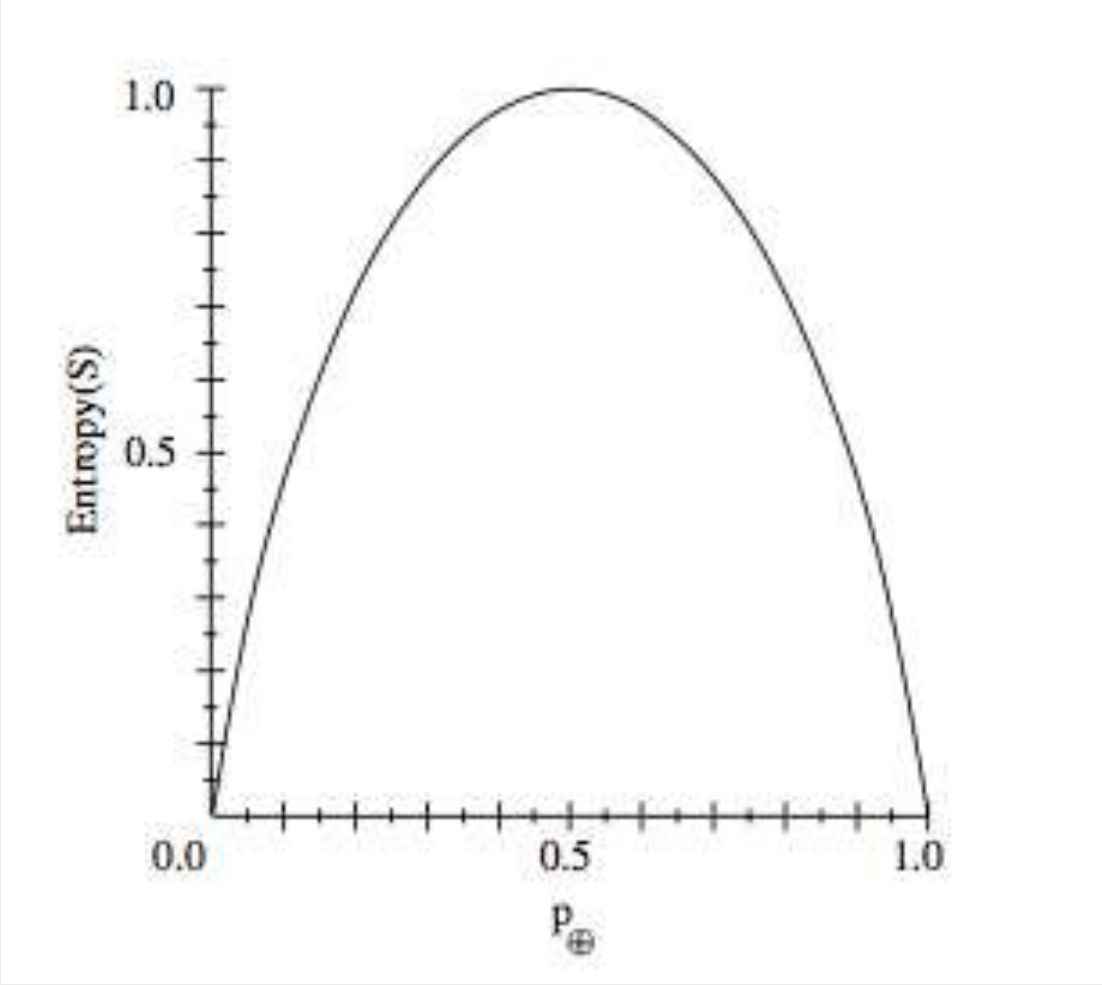
$$Entropy\ ([7+, 7-]) = -7/14 \log_2 (7/14) - 7/14 \log_2 (7/14) =$$

$$= 1/2 + 1/2 = 1 \qquad [\log_2 1/2 = -1]$$

Note: the log of a number < 1 is negative, $0 \leq p \leq 1$, $0 \leq entropy \leq 1$

10

Entropy



11

Entropy in General

Entropy measures the amount of information in a random variable

- For binary classification [two-valued random variable]

$$H(X) = -p_+ \log_2 p_+ - p_- \log_2 p_- \qquad X = \{+, -\}$$

- For classification in c classes

$$H(X) = -\sum_{i=1}^{c} p_i \log_2 p_i = \sum_{i=1}^{c} p_i \log_2 1/p_i \qquad X = \{i, \ldots, c\}$$

Example: rolling a die with 8, equally probable, sides

$$H(X) = -\sum_{i=1}^{8} 1/8 \log_2 1/8 = -\log_2 1/8 = \log_2 8 = 3$$

12

Entropy and Information Theory

- Entropy specifies the number the average length (in bits) of the message needed to transmit the outcome of a random variable. This depends on the probability distribution.
- Optimal length code assigns $\lceil -log_2\, p \rceil$ bits to messages with probability p. Most probable messages get shorter codes.
- Example: 8-sided [unbalanced] die

1	2	3	4	5	6	7	8
4/16	4/16	2/16	2/16	1/16	1/16	1/16	1/16
2 bits	*2 bits*	*3 bits*	*3 bits*	*4bits*	*4bits*	*4bits*	*4bits*

$E = (1/4\ log_2\ 4) \times 2 + (1/8\ log_2\ 8) \times 2 + (1/16\ log_2\ 16) \times 4 = 1+3/4+1 = 2{,}75$

13

Information Gain as Entropy Reduction

- *Information gain* is the *expected* reduction in entropy caused by partitioning the examples on an attribute.
- The higher the information gain the more effective the attribute in classifying training data.
- Expected reduction in entropy knowing A

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} Entropy(Sv)$$

$Values(A)$ possible values for A

Sv subset of S for which A has value v

14

Example: Expected Information Gain

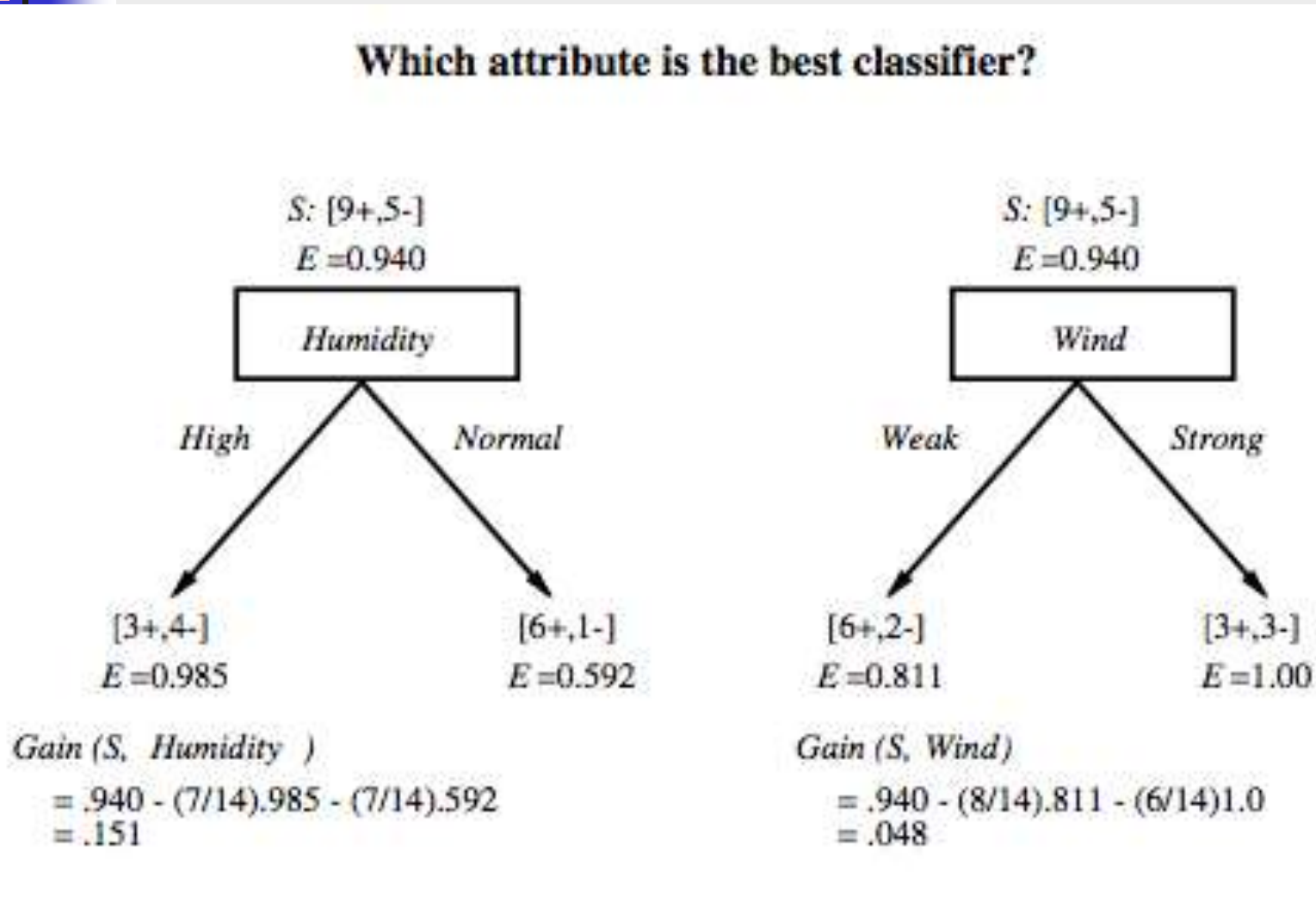
Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	*No*
D2	Sunny	Hot	High	Strong	*No*
D3	Overcast	Hot	High	Weak	*Yes*
D4	Rain	Mild	High	Weak	*Yes*
D5	Rain	Cool	Normal	Weak	*Yes*
D6	Rain	Cool	Normal	Strong	*No*
D7	Overcast	Cool	Normal	Strong	*Yes*
D8	Sunny	Mild	High	Weak	*No*
D9	Sunny	Cool	Normal	Weak	*Yes*
D10	Rain	Mild	Normal	Weak	*Yes*
D11	Sunny	Mild	Normal	Strong	*Yes*
D12	Overcast	Mild	High	Strong	*Yes*
D13	Overcast	Hot	Normal	Weak	*Yes*
D14	Rain	Mild	High	Strong	*No*

- Let
  - $Values(Wind) = \{Weak, Strong\}$
  - $S = [9+, 5-]$
  - $S_{Weak} = [6+, 2-]$
  - $S_{Strong} = [3+, 3-]$
- Information gain due to knowing *Wind*:

$$Gain(S, Wind) = Entropy(S) - 8/14\ Entropy(S_{Weak}) - 6/14\ Entropy(S_{Strong})$$
$$= 0{,}94 - 8/14 \times 0{,}811 - 6/14 \times 1{,}00$$
$$= 0{,}048$$

15

Which attribute is the best classifier?



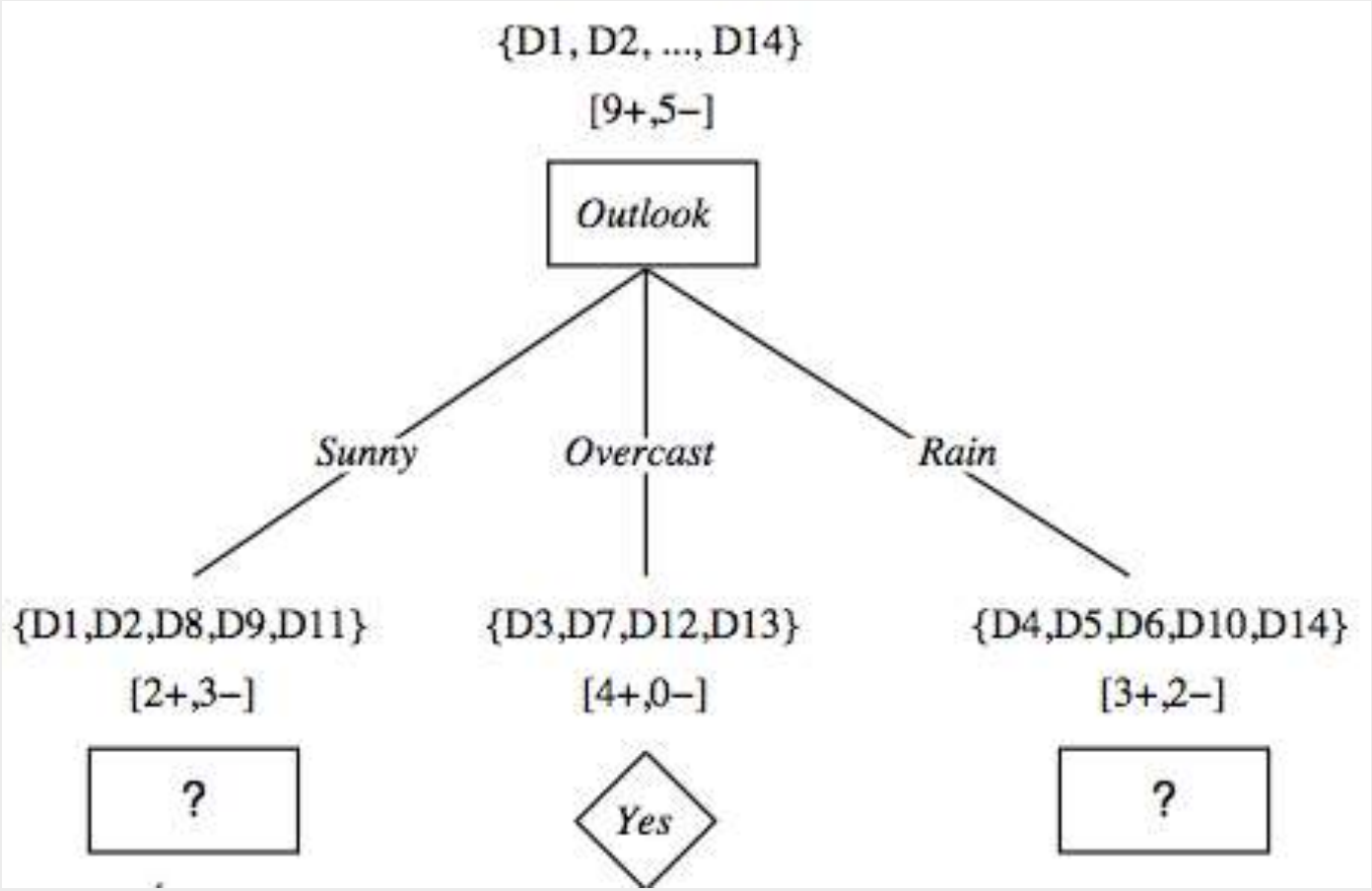
16

First Step: Which Attribute to Test at the Root?

- Which attribute should be tested at the root?
  - *Gain*(*S*, *Outlook*) = 0.246
  - *Gain*(*S*, *Humidity*) = 0.151
  - *Gain*(*S*, *Wind*) = 0.084
  - *Gain*(*S*, *Temperature*) = 0.029
- *Outlook* provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Outlook*
  - partition the training samples according to the value of *Outlook*

17

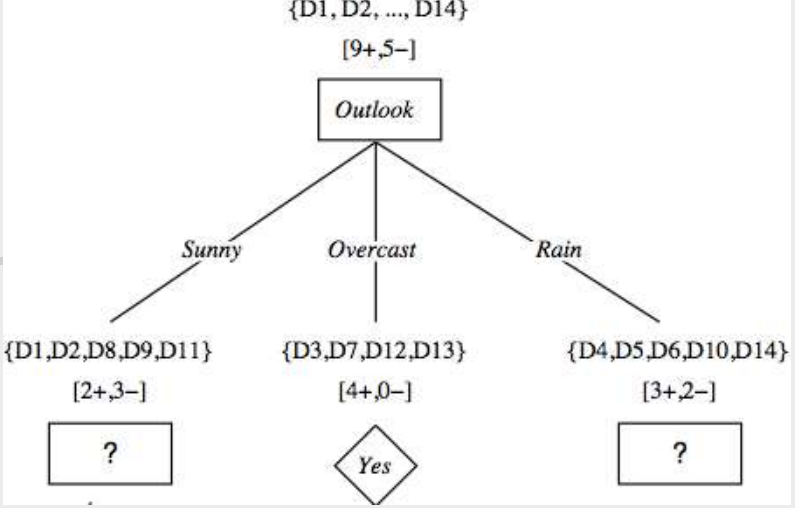
After First Step



18

Second Step

Outlook={Sunny, Overcast, Rain}



- Working on *Outlook=Sunny* node:

  $Gain(S_{Sunny}, Humidity) = 0.970 - 3/5 \times 0.0 - 2/5 \times 0.0 = 0.970$

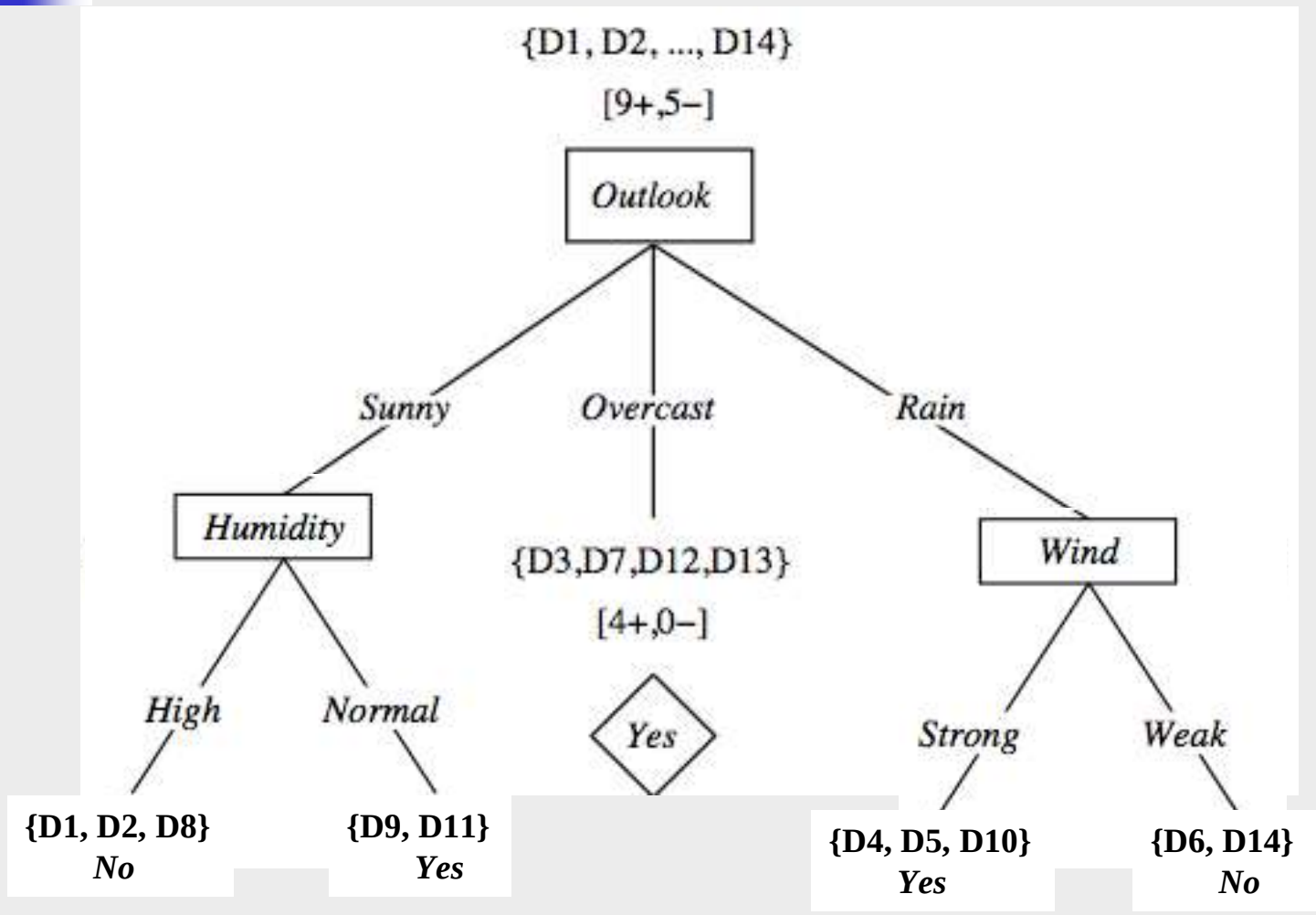
  $Gain(S_{Sunny}, Wind) = 0.970 - 2/5 \times 1.0 - 3.5 \times 0.918 = 0\ .019$

  $Gain(S_{Sunny}, Temp.) = 0.970 - 2/5 \times 0.0 - 2/5 \times 1.0 - 1/5 \times 0.0 = 0.570$

- *Humidity* provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Humidity*
  - partition the training samples according to the value of *Humidity*

19

Second and Third steps



20

ID3: Algorithm

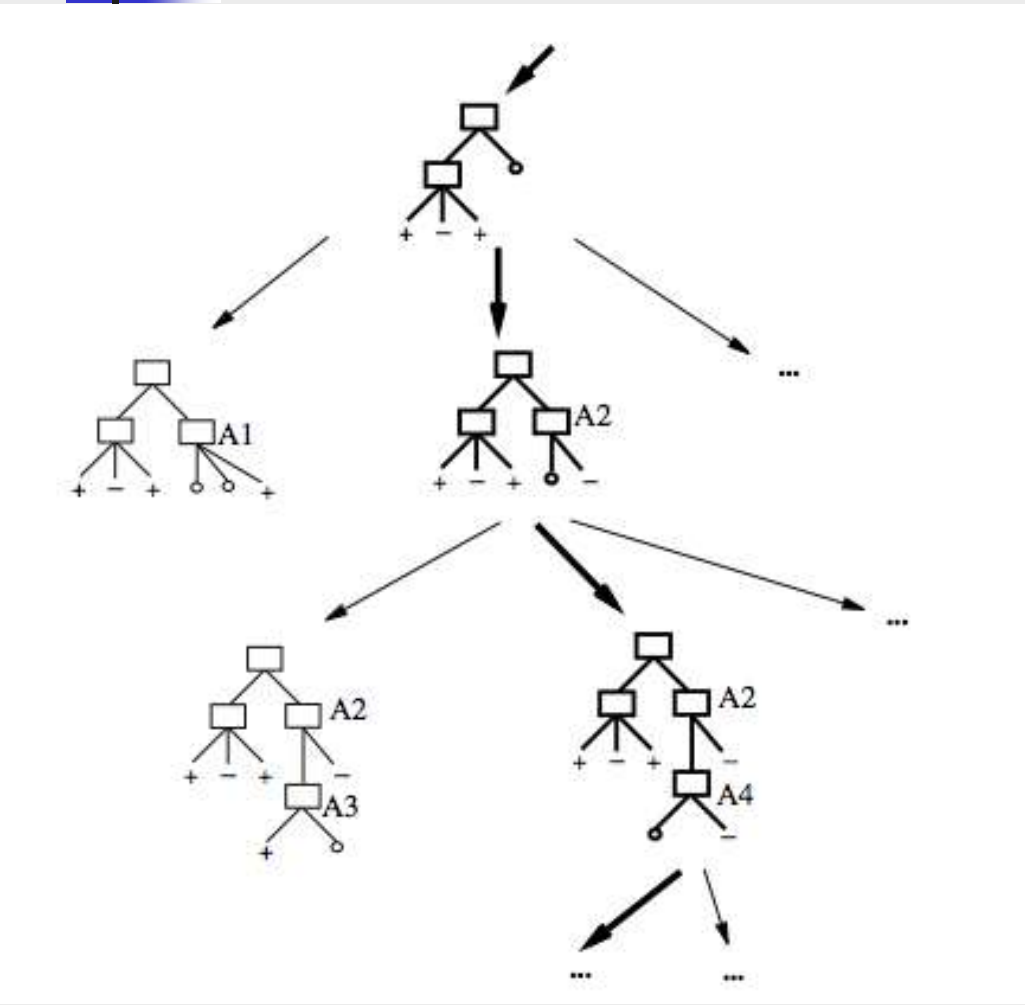
ID3(*X, T, Attrs*) *X*: training examples:
T: target attribute (e.g. *PlayTennis*),
Attrs: other attributes, initially all attributes

Create *Root* node
If all X's are +, *return Root* with class +
If all X's are –, *return Root* with class –
If Attrs is empty *return Root* with class most common value of *T* in *X*
else
$A \leftarrow$ best attribute; decision attribute for Root $\leftarrow A$
For each possible value v_i of *A:*
- add a new branch below *Root,* for test $A = v_i$
- $X_i \leftarrow$ subset of *X* with $A = v_i$
- *If* X_i is empty *then* add a new leaf with class the most common value of *T* in *X*
else add the subtree generated by ID3(X_i, *T, Attrs* – {*A*})

return Root

21

Search Space in DT Learning



- The search space is made by partial decision trees
- The algorithm is *hill-climbing*
- The evaluation function is *information gain*
- The hypotheses space is complete (represents all discrete-valued functions)
- The search maintains a single current hypothesis
- No backtracking; no guarantee of optimality
- It uses all the available examples (not incremental)
- May terminate earlier, accepting noisy classes

22

Inductive Bias in DT Learning

What is the inductive bias of DT learning?

1. *Shorter trees are preferred over longer trees*

   Not enough. This is the bias exhibited by a simple breadth first algorithm generating all DT's e selecting the shorter one

2. *Prefer trees that place high information gain attributes close to the root*

Note: DT's are not limited in representing all possible functions

23

Two Kinds of Biases

- Preference or search biases (due to the search strategy)
  - ID3 searches a *complete* hypotheses space; the search strategy is *incomplete*
- Restriction or language biases (due to the set of hypotheses expressible or considered)
  - *Candidate-Elimination* searches an *incomplete* hypotheses space; the search strategy is *complete*
- A combination of biases in learning a linear combination of weighted features in board games.

24

Prefer shorter hypotheses: Occam's rasor

- Why prefer shorter hypotheses?
- Arguments in favor:
  - There are fewer short hypotheses than long ones
  - If a short hypothesis fits data unlikely to be a coincidence
  - Elegance and aesthetics
- Arguments against:
  - Not every short hypothesis is a reasonable one.
- Occam's razor:*"The simplest explanation is usually the best one."*
  - a principle usually (though incorrectly) attributed14th-century English logician and Franciscan friar, William of Ockham.
  - lex parsimoniae ("law of parsimony", "law of economy", or "law of succinctness")
  - The term razor refers to the act of *shaving away* unnecessary assumptions to get to the simplest explanation.

25

Issues in DTs Learning

- Overfitting
  - Reduced error pruning
  - Rule post-pruning
- Extensions
  - Continuous valued attributes
  - Alternative measures for selecting attributes
  - Handling training examples with missing attribute values
  - Handling attributes with different costs
  - Improving computational efficiency
  - Most of these improvements in C4.5 (Quinlan, 1993)

26

Overfitting in DTs

- Building DTs that "adapt too much" to the training examples may lead to "overfitting".
- Consider error of hypothesis *h* over
  - training data: $error_D(h)$ — empirical error
  - entire distribution *X* of data: $error_X(h)$ — expected error
- Hypothesis *h* *overfits* training data if there is an alternative hypothesis $h' \in H$ such that

$$error_D(h) < error_D(h') \quad \text{and} \quad error_X(h') < error_X(h)$$

i.e. *h'* behaves better over unseen data

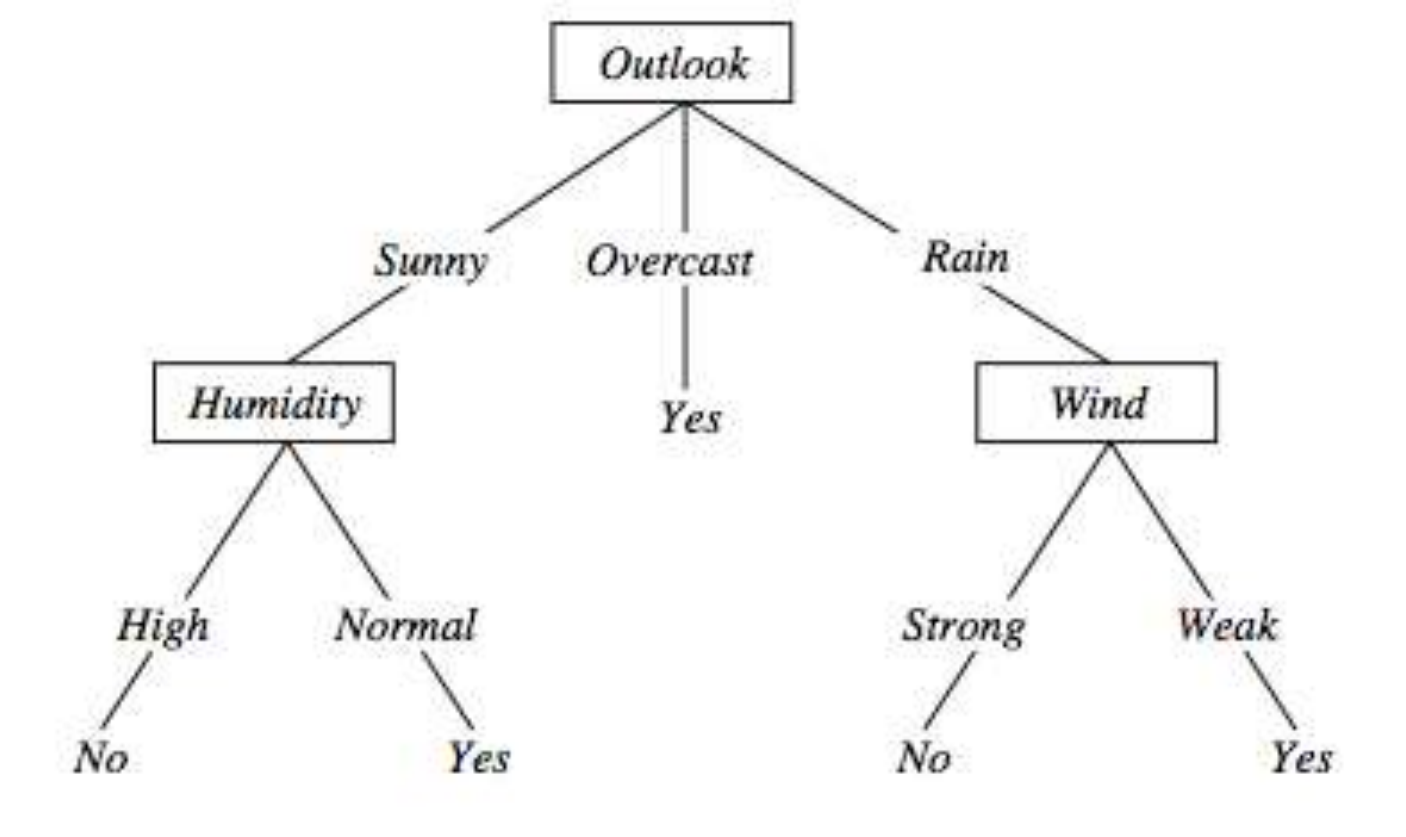
27

Example

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	*No*
D2	Sunny	Hot	High	Strong	*No*
D3	Overcast	Hot	High	Weak	*Yes*
D4	Rain	Mild	High	Weak	*Yes*
D5	Rain	Cool	Normal	Weak	*Yes*
D6	Rain	Cool	Normal	Strong	*No*
D7	Overcast	Cool	Normal	Strong	*Yes*
D8	Sunny	Mild	High	Weak	*No*
D9	Sunny	Cool	Normal	Weak	*Yes*
D10	Rain	Mild	Normal	Weak	*Yes*
D11	Sunny	Mild	Normal	Strong	*Yes*
D12	Overcast	Mild	High	Strong	*Yes*
D13	Overcast	Hot	Normal	Weak	*Yes*
D14	Rain	Mild	High	Strong	*No*
D15	Sunny	Hot	Normal	Strong	No

28

Overfitting in DTs

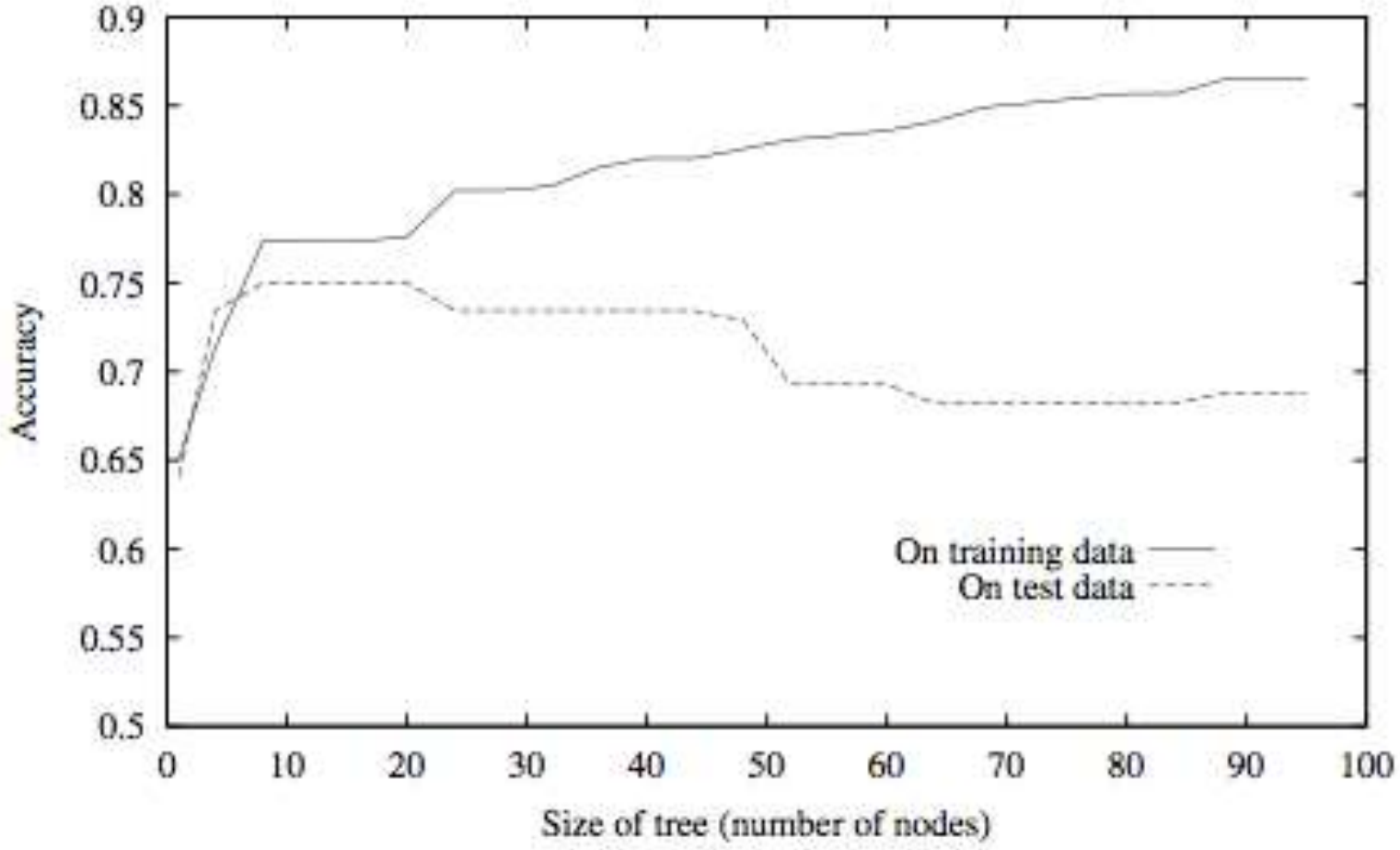


⟨*Outlook=Sunny, Temp=Hot, Humidity=Normal, Wind=Strong, PlayTennis=No* ⟩

New noisy example causes splitting of second leaf node.

29

Overfitting in DTs



30

Avoid Overfitting in DTs

- Two strategies:
  1. Stop growing the tree earlier, before perfect classification
  2. Allow the tree to *overfit* the data, and then *post-prune* the tree
- Training and validation set
  - split the training in two parts (training and validation) and use validation to assess the utility of *post-pruning*
    - *Reduced error pruning*
    - *Rule pruning*
- Other approaches
  - Use a statistical test to estimate effect of expanding or pruning *Minimum description length principle*: uses a measure of complexity of encoding the DT and the examples, and halt growing the tree when this encoding size is minimal

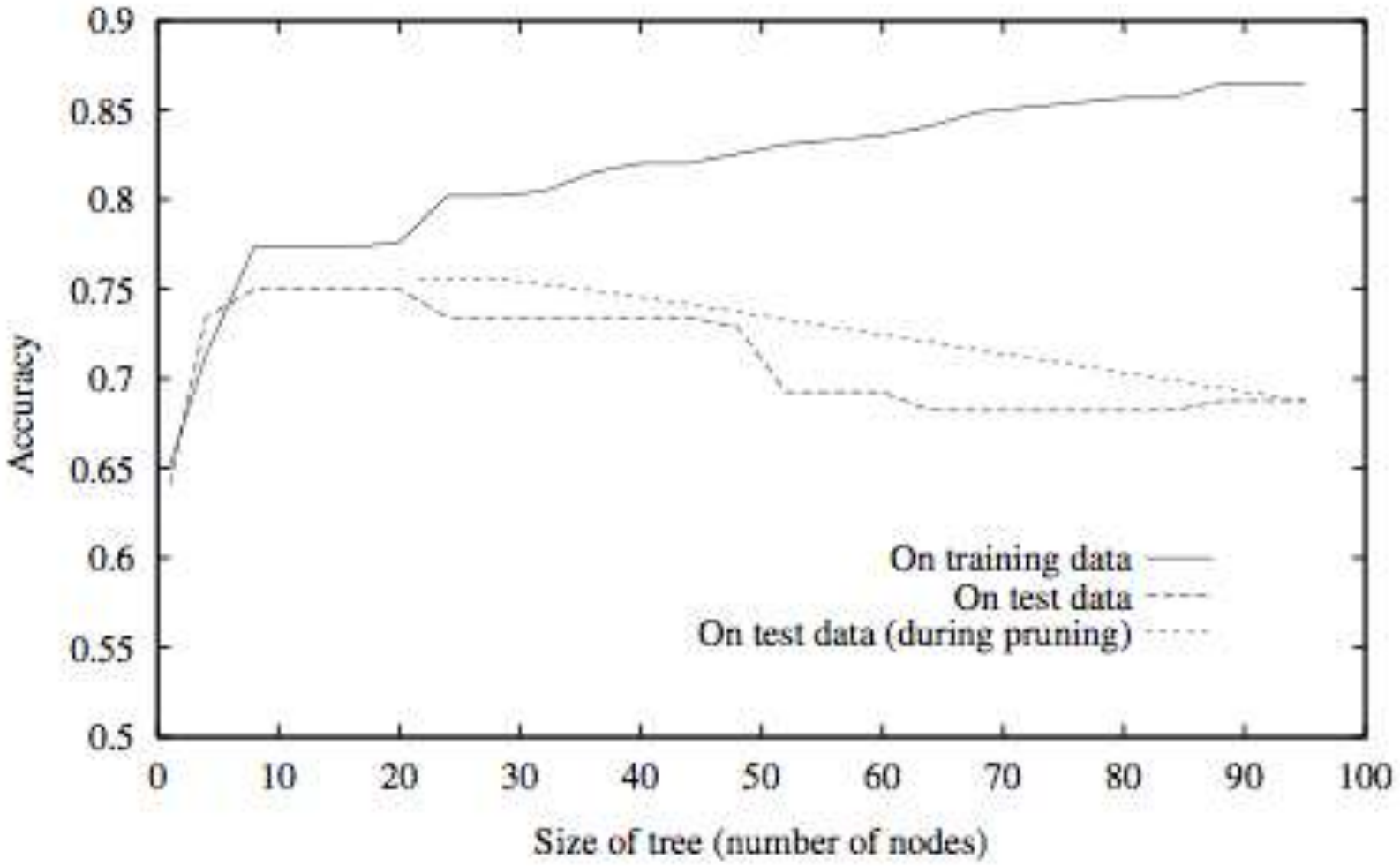
31

Reduced-error Pruning (Quinlan 1987)

- Each node is a candidate for pruning
- *Pruning* consists in removing a subtree rooted in a node: the node becomes a leaf and is assigned the most common classification
- Nodes are removed only if the resulting tree performs no worse on the validation set.
- Nodes are pruned iteratively: at each iteration the node whose removal most increases accuracy on the validation set is pruned.
- Pruning stops when no pruning increases accuracy

32

Effect of Reduced Error Pruning



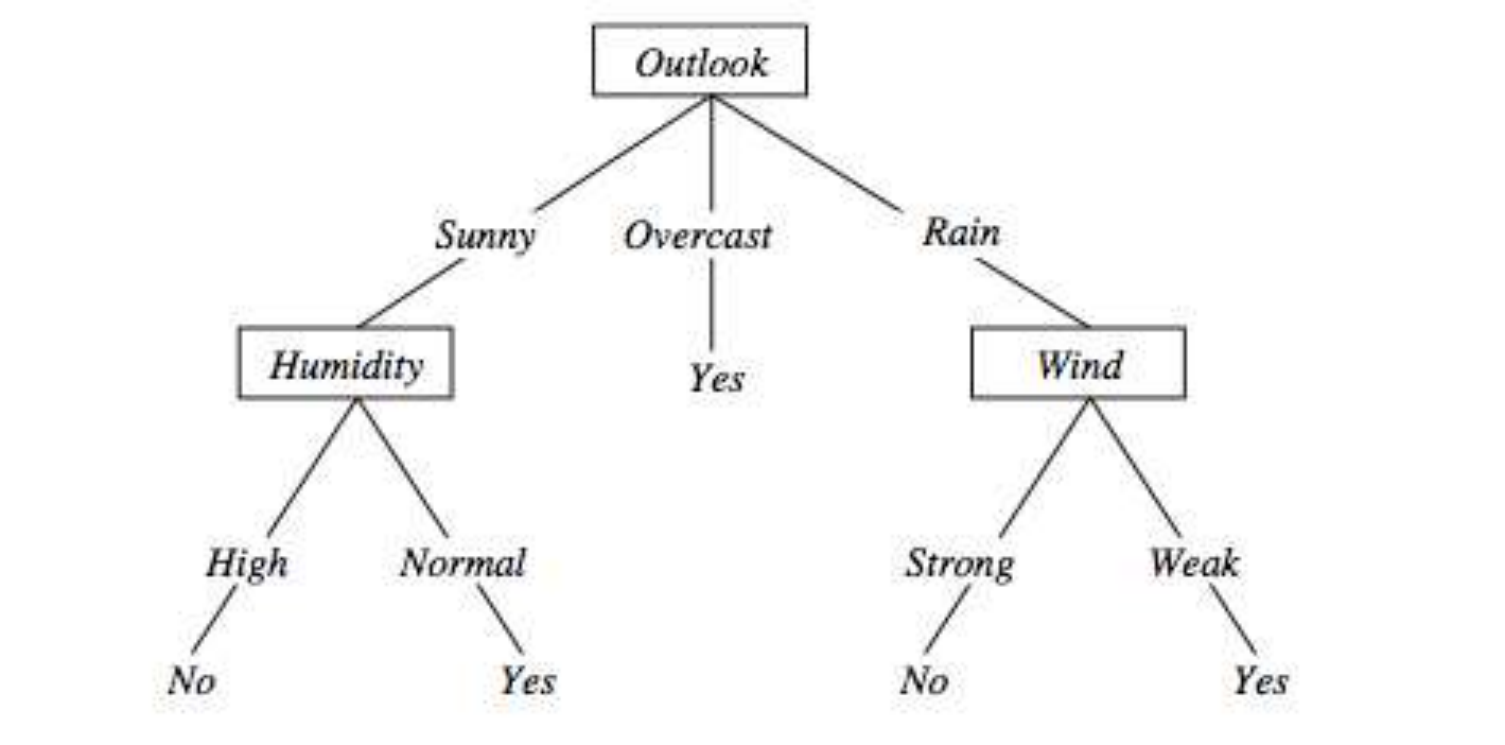
33

Rule Post-Pruning

1. Create the decision tree from the training set
2. Convert the tree into an equivalent set of rules
   - Each path corresponds to a rule
   - Each node along a path corresponds to a pre-condition
   - Each leaf classification to the post-condition
3. Prune (generalize) each rule by removing those preconditions whose removal improves accuracy …
   - … over validation set
   - … over training with a pessimistic, statistically inspired, measure
4. Sort the rules in estimated order of accuracy, and consider them in sequence when classifying new instances

34

Converting to Rules



$$(Outlook=Sunny) \wedge (Humidity=High) \Rightarrow (PlayTennis=No)$$

35

Why Converting to Rules?

- Each distinct path produces a different rule: a condition removal may be based on a local (contextual) criterion. Node pruning is global and affects all the rules
- In rule form, tests are not ordered and there is no book-keeping involved when conditions (nodes) are removed
- Converting to rules improves readability for humans

36

Dealing with Continuous-valued Attributes

- So far discrete values for attributes and for outcome.
- Given a continuous-valued attribute A, dynamically create a new attribute A_c

$$A_c = \text{True } if\ A < c, \text{ False } otherwise$$

- How to determine threshold value c ?
- Example. *Temperature* in the *PlayTennis* example
  - Sort the examples according to *Temperature*

Temperature	40	48	54	60	72	80	85	90
PlayTennis	No	No		Yes	Yes	Yes		No

  - Determine candidate thresholds by averaging consecutive values where there is a change in classification: (48+60)/2=54 and (80+90)/2=85
  - Evaluate candidate thresholds (attributes) according to information gain. The best is $Temperature_{>54}$.The new attribute competes with the other ones

37

Handling Incomplete Training Data

- How to cope with the problem that the value of some attribute may be missing?
  - *Example*: Blood-Test-Result in a medical diagnosis problem
- The strategy: use other examples to guess attribute
  - Assign the value that is most common among the training examples at the node
  - Assign a probability to each value, based on frequencies, and assign values to missing attribute, according to this probability distribution
- Missing values in new instances to be classified are treated accordingly, and the most probable classification is chosen (C4.5)

38

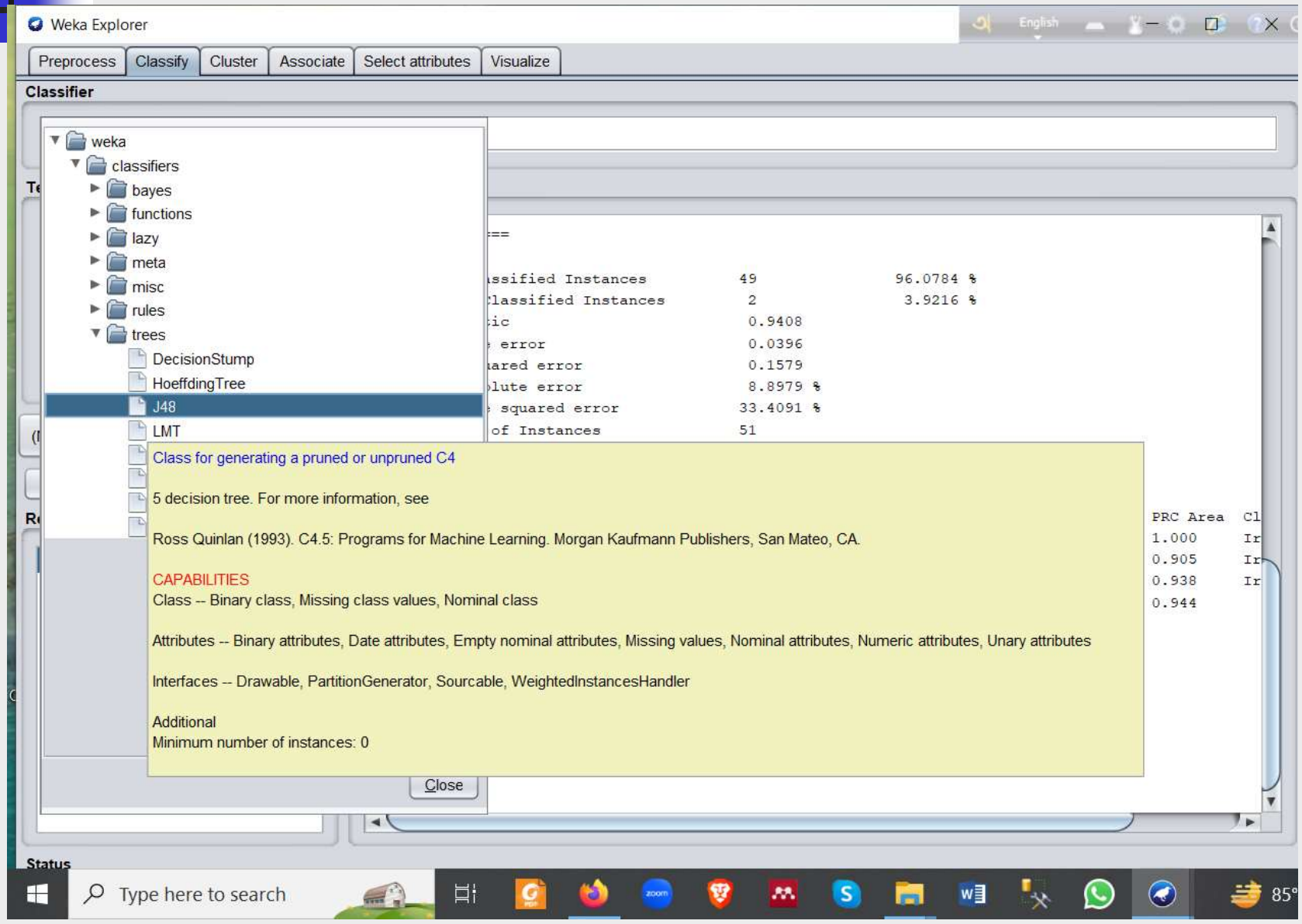
Reference and Software

Important Reference: Machine Learning, Tom Mitchell, Mc Graw-Hill International Editions, 1997 (Cap 3).

- In R:
  - Packages tree and rpart
- C4.5:
  - http://www.cse.unwe.edu.au/~quinlan
- Weka
  - http://www.cs.waikato.ac.nz/ml/weka

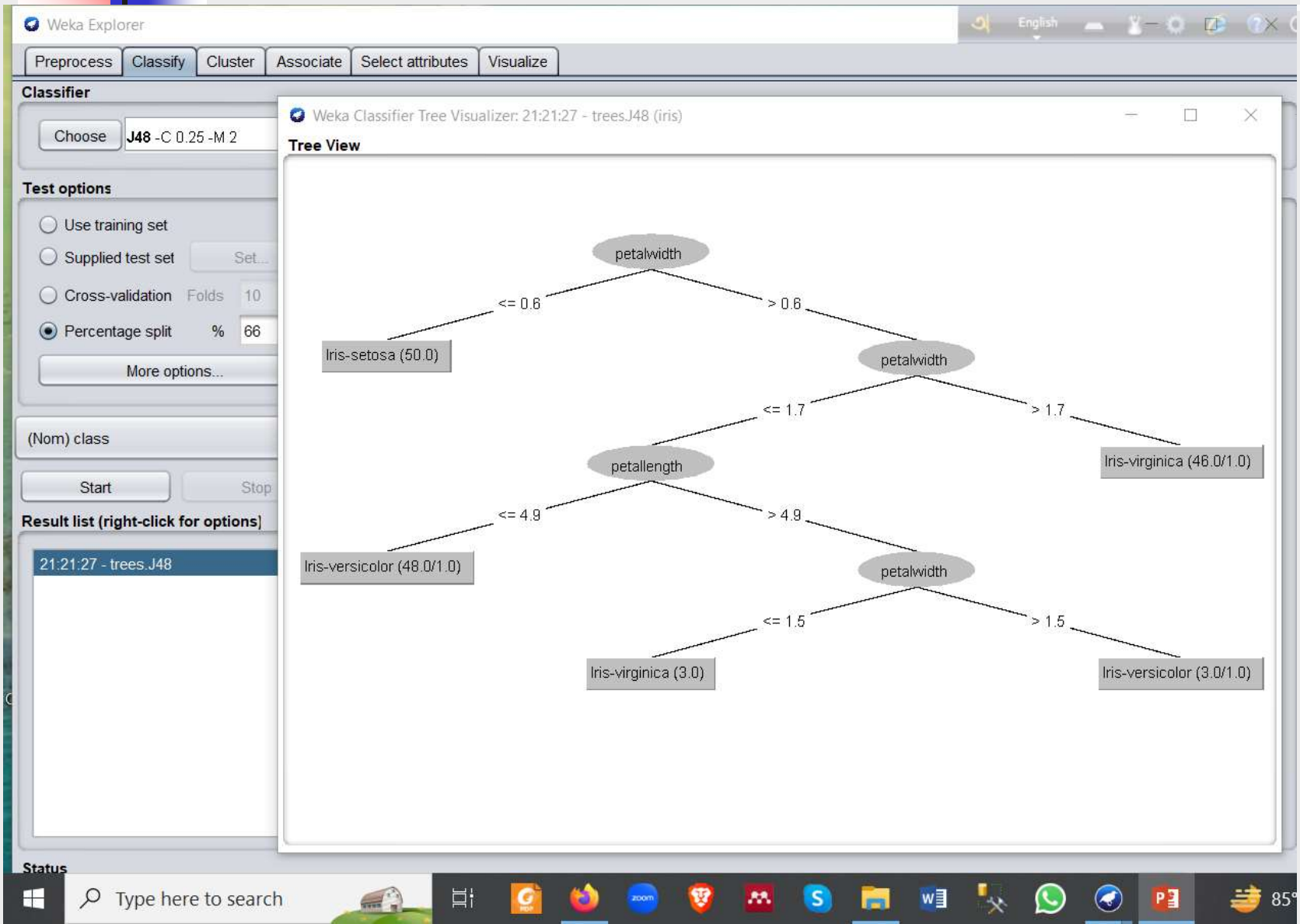
39

DT for Iris Dataset in Weka J48 (C4.5)



40

DT for Iris Dataset in Weka J48 (C4.5)



41

Open Discussion

An alternative measure: *gain ratio*

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}$$

- S_i are the sets obtained by partitioning on value i of A
- *SplitInformation* measures the entropy of S with respect to the values of A. The more uniformly dispersed the data the higher it is.

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

- *GainRatio* penalizes attributes that split examples in many small classes such as *Date*. Let $|S|=n$, *Date* splits examples in n classes
  - $SplitInformation(S, Date) = -[(1/n \log_2 1/n) + \ldots + (1/n \log_2 1/n)] = -\log_2 1/n = \log_2 n$
- Compare with A, which splits data in two even classes:
  - $SplitInformation(S, A) = -[(1/2 \log_2 1/2) + (1/2 \log_2 1/2)] = -[-1/2 - 1/2] = 1$

Adjusting *gain-ratio*

- Problem: $SplitInformation(S, A)$ can be zero or very small when $|S_i| \approx |S|$ for some value i
- To mitigate this effect, the following heuristics has been used:
  1. compute $Gain$ for each attribute
  2. apply $GainRatio$ only to attributes with $Gain$ above average
- Other measures have been proposed:
  - *Distance-based* metric [Lopez-De Mantaras, 1991] on the partitions of data
  - Each partition (induced by an attribute) is evaluated according to the distance to the partition that perfectly classifies the data.
  - The partition closest to the *ideal* partition is chosen

Handling attributes with different costs

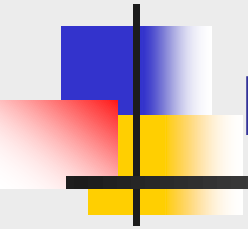
- Instance attributes may have an associated cost: we would prefer decision trees that use low-cost attributes
- ID3 can be modified to take into account costs:

1. Tan and Schlimmer (1990)

$$\frac{Gain^2(S, A)}{Cost(S, A)}$$

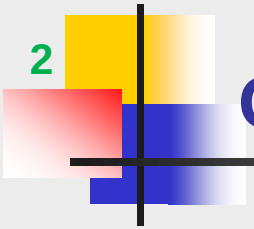
2. Nunez (1988)

$$\frac{2^{Gain(S, A)} - 1}{(Cost(A) + 1)^w} \quad w \in [0,1]$$



Ensembles of Diverse Neural Networks

Dr. Md. Aminul Haque Akhand



Outline of the Presentation

1. Introduction
2. Background of Ensemble Construction
3. Data Sampling for Ensemble Construction
4. Candidate Ensemble Methods
5. Experimental Studies
6. Motivation to better NNE Construction

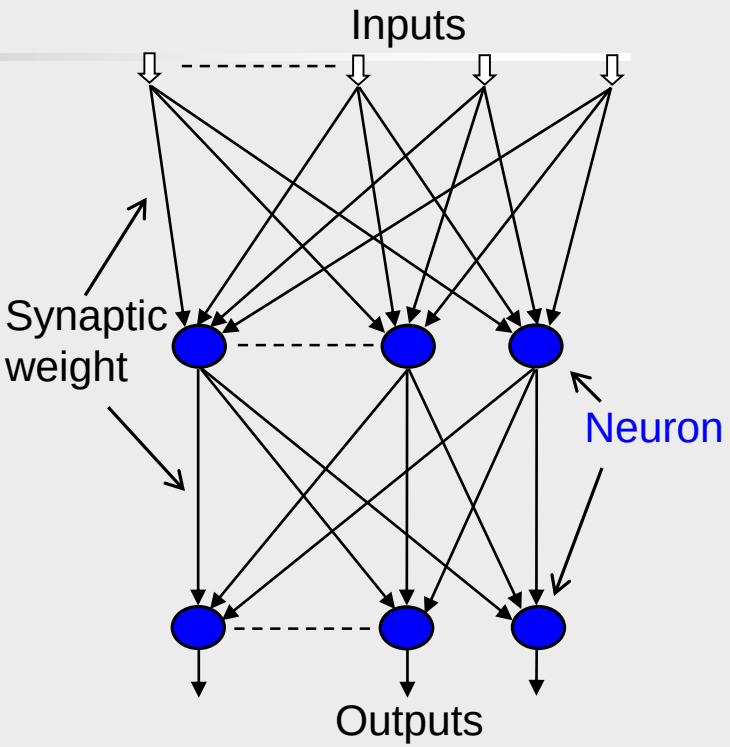
3

Artificial Neural Network

According to Haykin, an artificial neural network (NN) is a collection of simple processing units and it has ability to store experimental knowledge.

NN resembles the human brain in two respects:

1. *Knowledge is acquired by a NN from its environment through a learning process.*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the knowledge.*



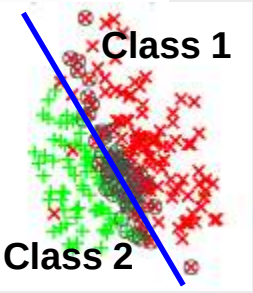
Artificial Neural Network

❖ Functionality of a NN depends on the synaptic weight values and the aim of learning is to get proper weight set.

NNs have been successfully applied for various task; for classification it performs well.

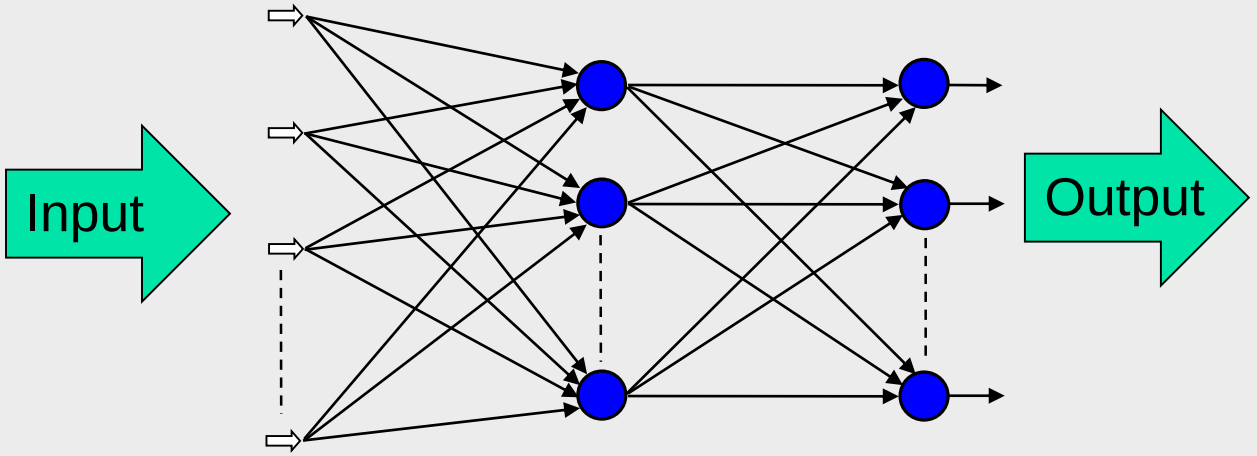
4

NNs for Classification Tasks



- Classification is one of the most frequently encountered decision making tasks of human activity.
  - It occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object.

Back-Propagation (BP) Learning Algorithm for Classification



Error = Desired Output - Actual Output

Forward Pass: Error is calculated based on desired and actual output.

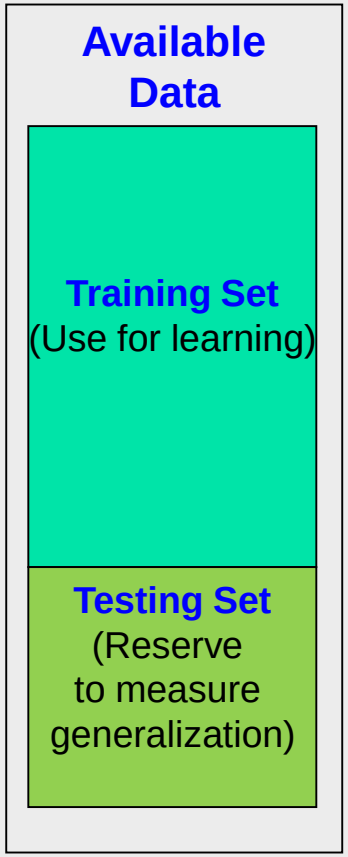
Backward Pass: Synaptic weights are adjusted based on calculated error.

- At a time small fraction of a weight is corrected w. r. t demand correction for smooth learning; a parameter learning rate(η) defines the relative size to change.

5

Performance Measures

Learning and generalization are the most important topics in NN research. Learning is the ability to approximate the training data while generalization is the ability to predict well beyond the training data.



- Generalization is more desirable because the common use of a NN is to make good prediction on new or unknown objects.
- It measures on the testing set that is reserved from available data and not use in the training.
- Testing error rate (TER), i.e., rate of wrong classification on testing set, is widely acceptable quantifying measure, which value minimum is good.

$$\text{TER} = \frac{\text{Total testing set misclassified patterns}}{\text{Total testing set patterns}}$$

Benchmark problems are used to measure TER.

6

Benchmark Problems for Evaluation

A benchmark is a point of reference by which something can be measured.

- For NN or machine learning, the most popular benchmark dataset collection is the University of California, Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/).
- UCI contains raw data that require preprocessing to use in NN. Some preprocessed data is also available at Proben1 (ftp://ftp.ira.uka.de/pub/neuron/).
- Various persons or groups also maintain different benchmark datasets for specific purpose: Delve (www.cs.toronto.edu/~delve/data/datasets.html), Orange (www.ailab.si/orange/datasets.asp), etc.
- In this study 32 benchmark problems are considered from UCI. Well defined benchmark methodology is followed for preprocessing.

UCI Machine Learning Repository - Windows Internet Explorer

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

UCI Machine Learning Repository

UCI Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 174 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. Our old web si format. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If donation policy. For any other questions, feel free to contact the Repository librarians. We have also set up a mirror site for the Repository.

Supported By: In Collaboration With: Rexa.info

Latest News:

07-23-2008: Repository mirror has been set up.

03-24-2008: New data sets have been added!

06-25-2007: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope

04-13-2007: Research papers that cite the repository have been associated to specific data sets.

04-09-2007: Three new data sets have been added: Poker Hand, Callt2 Building People Counts, Dodgers Loop Sensor.

09-08-2006: The Beta site has been launched.

09-01-2006: SPECTF.test has been modified by the donor.

Newest Data Sets:

06-26-2008: Parkinsons

04-21-2008: Ozone Level Detection

04-03-2008: Abscisic Acid Signaling Network

03-20-2008: Hill-Valley

Most Popular Data

39351:

31585:

26458:

23553:

8

Benchmark Problems

Problems Related to Human Life

Problem	Task
Breast Cancer Wisconsin	Predicts whether a tumor is benign (not dangerous to health) or malignant (dangerous) based on a sample tissue taken from a patient's breast.
BUPA Liver Disorder	Identify lever disorders based on blood tests along with other related information such as alcohol consumption.
Diabetes	Investigate whether the patient shows or not the signs of diabetes.
Heart Disease Cleveland	Predicting whether at least one of four major heart vessels is reduced in diameter by more than 50%.
Hepatitis	Anticipate status (i.e., live or die) of hepatitis patient.
Lymphography	Predict the situation of lymph nodes and lymphatic vessels.
Lungcancer	Identify types of pathological lung cancers.
Postoperative	Determine place to send patients for postoperative recovery.

9

Benchmark Problems

Problems Related to Finance

Problem	Task
Australian Credit Card	Classify people as good or bad credit risks depend on applicants' particulars.
Car	Evaluate cars based on price and facilities.
Labor Negotiations	Identify a worker as good or bad i.e., contract with him beneficial or not.
German Credit Card	Like Australian Card, this problem also concerns to predict the approval or non-approval of a credit card to a customer.

Problems Related to Plants

Problem	Task
Iris Plants	Classify iris plant types.
Mushroom	Identify whether a mushroom is edible or not based on a description of the mushroom's shape, color, odor, and habitat.
Soybean	Recognize 19 different diseases of soybeans.

10

Summary of Benchmark Problems

Abbr.	Problem	Total Examp	Input Features		NN Architecture		
			Cont.	Discr.	Inputs	Class	Hidd. Node
ACC	Australian Credit Card	690	6	9	51	2	10
BLN	Balance	625	-	4	20	3	10
BCW	Breast Cancer Wisconsin	699	9	-	9	2	5
CAR	Car	1728	-	6	21	4	10
DBT	Diabetes	768	8	-	8	2	5
GCC	German Credit Card	1000	7	13	63	2	10
HDC	Heart Disease Cleveland	303	6	7	35	2	5
HPT	Hepatitis (HPT)	155	6	13	19	2	5
HTR	Hypothyroid	7200	6	15	21	3	5
HSV	House Vote	435	-	16	16	2	5
INS	Ionosphere	351	34	-	34	2	10
KRP	King+Rook vs King+Pawn	3196	-	36	74	2	10
LMP	Lymphography	148	-	18	18	4	10
PST	Postoperative	90	1	7	19	3	5
SBN	Soybean	683	-	35	82	19	25
SNR	Sonar	208	60	-	60	2	10
SPL	Splice Junction	3175	-	60	60	3	10
WIN	Wine	178	13	-	13	3	5
WVF	Waveform	5000	21	-	21	3	10
ZOO	Zoo	101	15	1	16	7	10

Input Features of Diabetes

1. Number of times pregnant
2. Plasma glucose concentration
3. Diastolic blood pressure
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index
7. Diabetes pedigree function
8. Age

❖Problems show variations in number of examples, input features and classes.

11

Why Ensemble of Neural Networks?

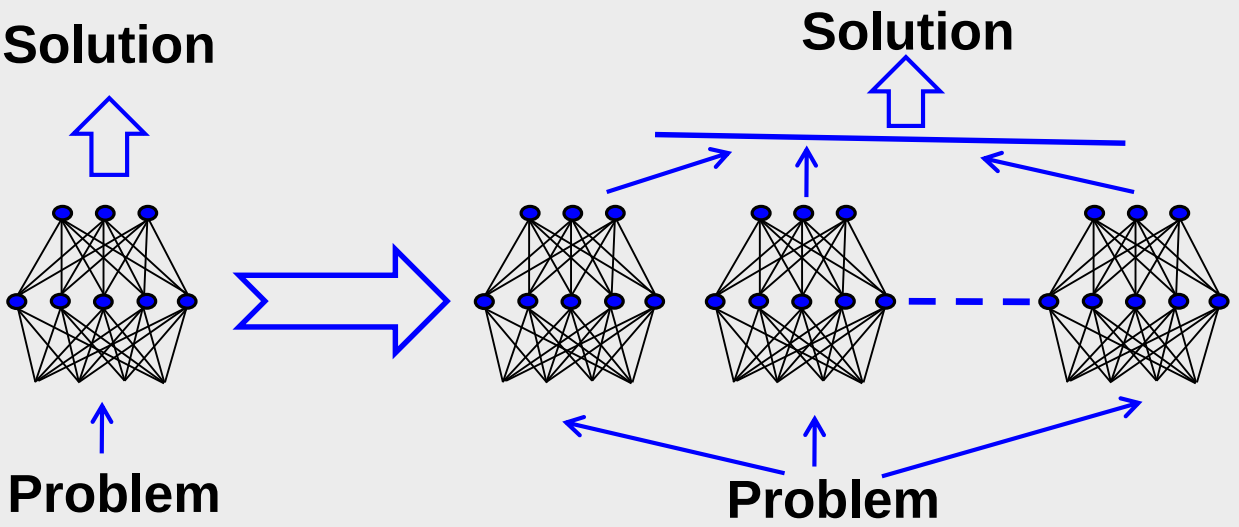
- The idea of building an ensemble with several NNs is taken from sociology.
- A committee of people for an important task or building board of doctors for a major operation is a common matter.

- Each member of the committee should be as competent as possible, but they should be complementary to one another. If one or a few members make an error, the probability is high that the remaining members can correct his error.
- Several NNs together might perform better than single NN when they maintain proper diversity to compensate failure of one by others.

The goal of ensemble is to achieve better generalization (i.e., lower TER) through producing diverse NNs.

12

Neural Network Ensemble (NNE)



In an NNE, component NNs solve the problem individually and combine their outputs for NNE's output. For better performance diversity among NNs is important.

- Diversity means disagreement among the NNs. Pair wise plain disagreement (*PD*) measure is the most popular among various measuring techniques.
- For a NNs pair, diversity *(div)* is equal to the proportion of the patterns on which NNs reply different class predictions. The total NNE diversity (*div_ens)* is the average for all the pairs.

$$div_{i.j} = \frac{1}{N}\sum_{n=1}^{N} Diff\,(C_i(x_n), C_j(x_n)),$$

$$div\_ens = \frac{\left[div_{i.j} \text{ for all NNs pairs }\right]}{\text{Total NNs pair}} = \frac{\sum_{i=1}^{M-1}\sum_{j=i+1}^{M} div_{i.j}}{\sum_{m=1}^{M-1} m}$$

13

Data Sampling for NNE Construction

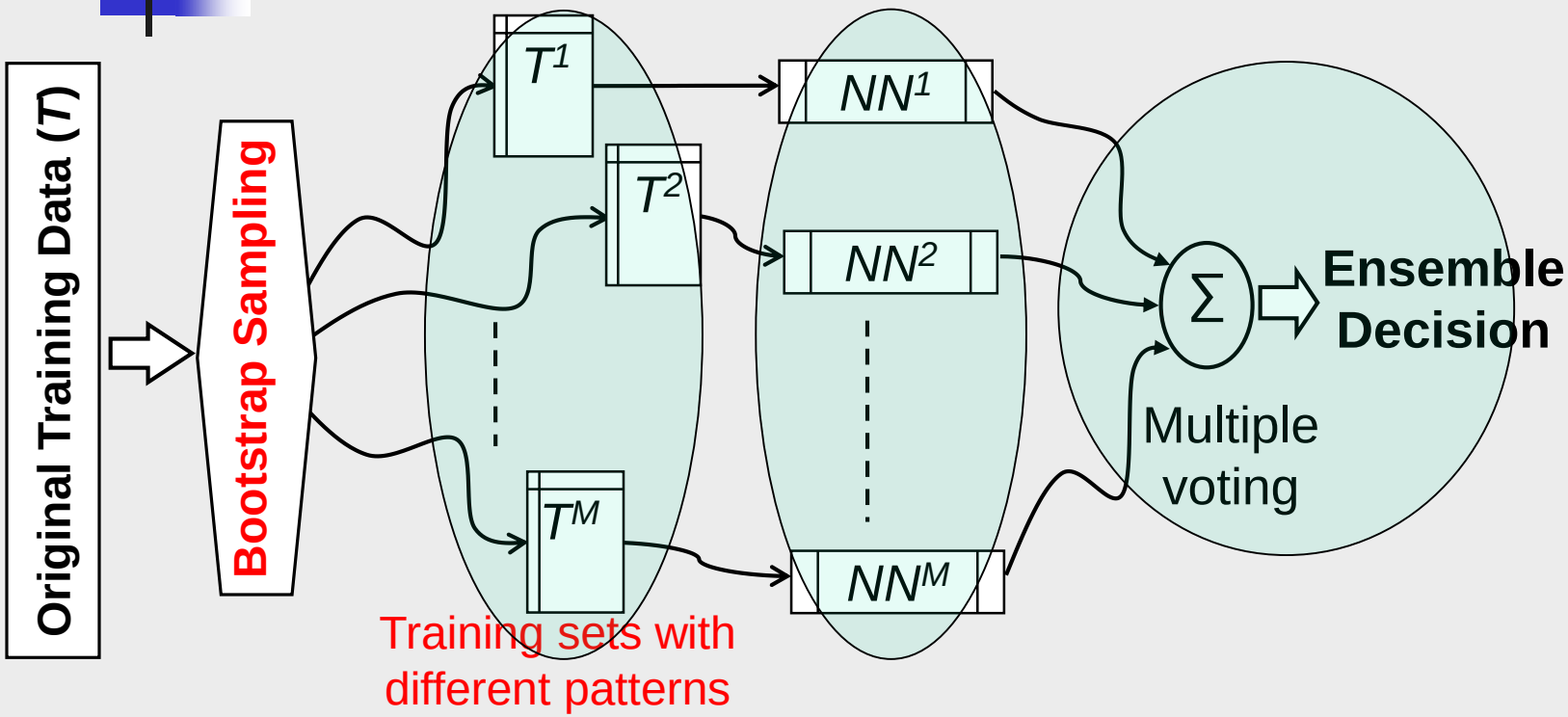
Proper diversity among component NNs is an important parameter for ensemble construction so that failure of one may compensate by others.

- There are various ways one can produce diverse NNs, such as varying initial random weights, algorithm employed and training data.
- Since functionality of a NN depends on its training data, data sampling is considered as the most effective for diversity than other approaches.
- Data sampling (i.e. variation in training data) involves: bootstrapping of original training data, generation artificial data, sampling of input features, etc.

14

1. Bagging (Breiman, 1996)

L. Breiman, "Bagging Predictors" *Machine Learning,* vol. 24, pp. 23 –140, 1996.



T	T¹	T²
1	4	10
2	9	6
3	6	3
4	8	9
5	1	2
6	8	7
7	5	3
8	1	4
9	2	2
10	4	9

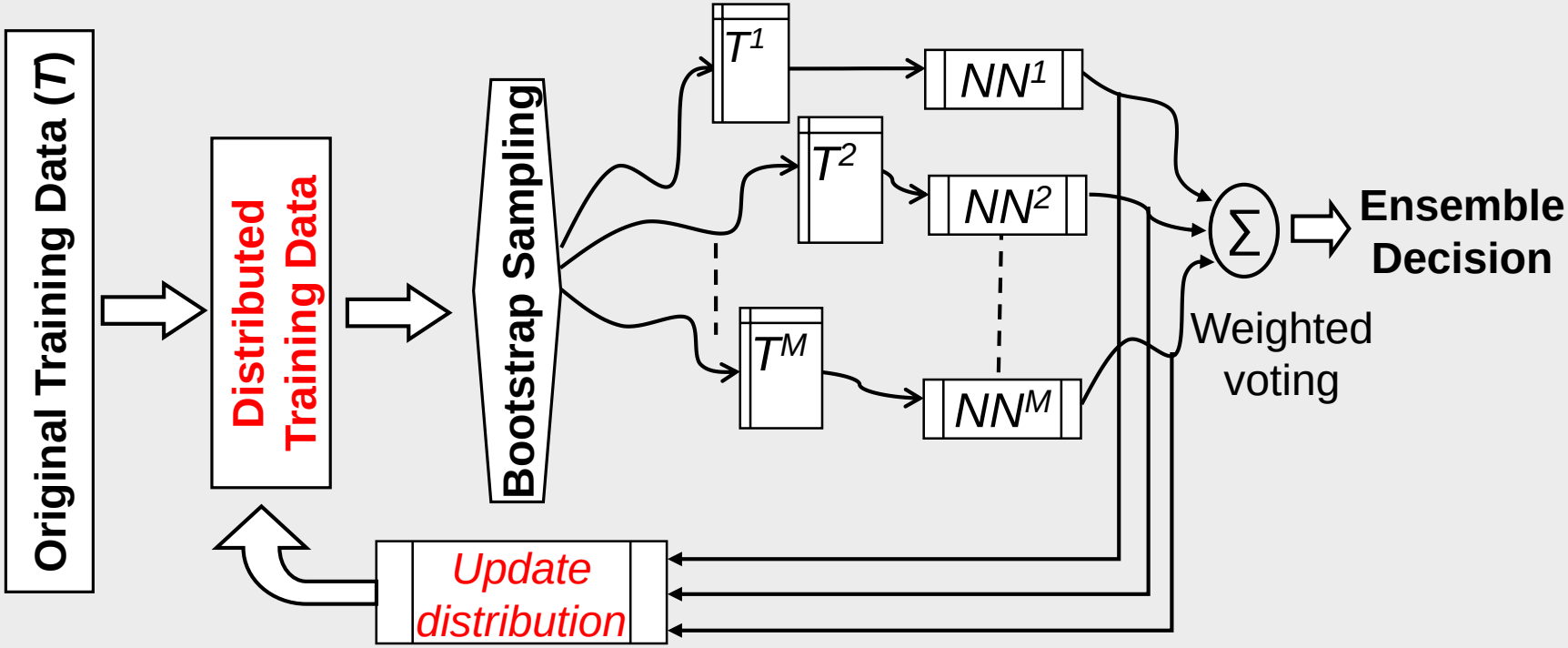
Bootstrap sampling: Randomly selects a pattern and replace it again in its original place for later use.

In a training set many patterns appear multiple times while others are left.

15

2. AdaBoost (Freund & Schapire, 1996)

Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm", *in Proc. of the 13th Int. Conf. on Machine Learning(Morgan Kaufmann, 1996)*, pp. 148–156



- NNs are trained one after another sequentially and after training a NN distribution of training data updates.
- Existence of previously miss classified patterns increases in coming training sets due to error base distribution.

16

Bagging and AdaBoost

1. Let M be the number of networks to be trained for an ensemble

Take original training set $T= \{(x(1), d(1)),\ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,.....,k\}$

2. *for i=1 to M* {

   a. Make a training set, T_i by sampling N patterns uniformly at random with replacement from T

   b. Train network NN_i by T_i

}

3. Ensemble decision is made in multiple voting way

Bagging

1. Let M be the number of networks to be trained for an ensemble.

Take original train ing set $T= \{(x(1), d(1)),\ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,.....,k\}$

Assign weight for each pattern of T, initially weights are the same, i.e., $w_1(n)=1/N$

2. *for i=1 to M* {

   a. Make a training set, T_i by sampling N patterns at random with replacement from T based on weight distribution $w_i(n)$

   b. Train network NN_i with T_i

   c. $\varepsilon_i = \sum_{(x(n),d(n))\in T: NN_i(x(n))\neq d(n)} w_i(n)$ (weighted error on training set)

   d. $\beta_i = (1-\varepsilon_i)/\varepsilon_i$

   e. *for each* $(x(n),d(n)) \in T$,

   if $NN^i(x_n) \neq d_n$ *then* $w_{i+1}(n) = w_i(n).\beta_i$, *otherwise* $w_{i+1}(n) = w_i(n)$

   f. Normalize the weights of T

}

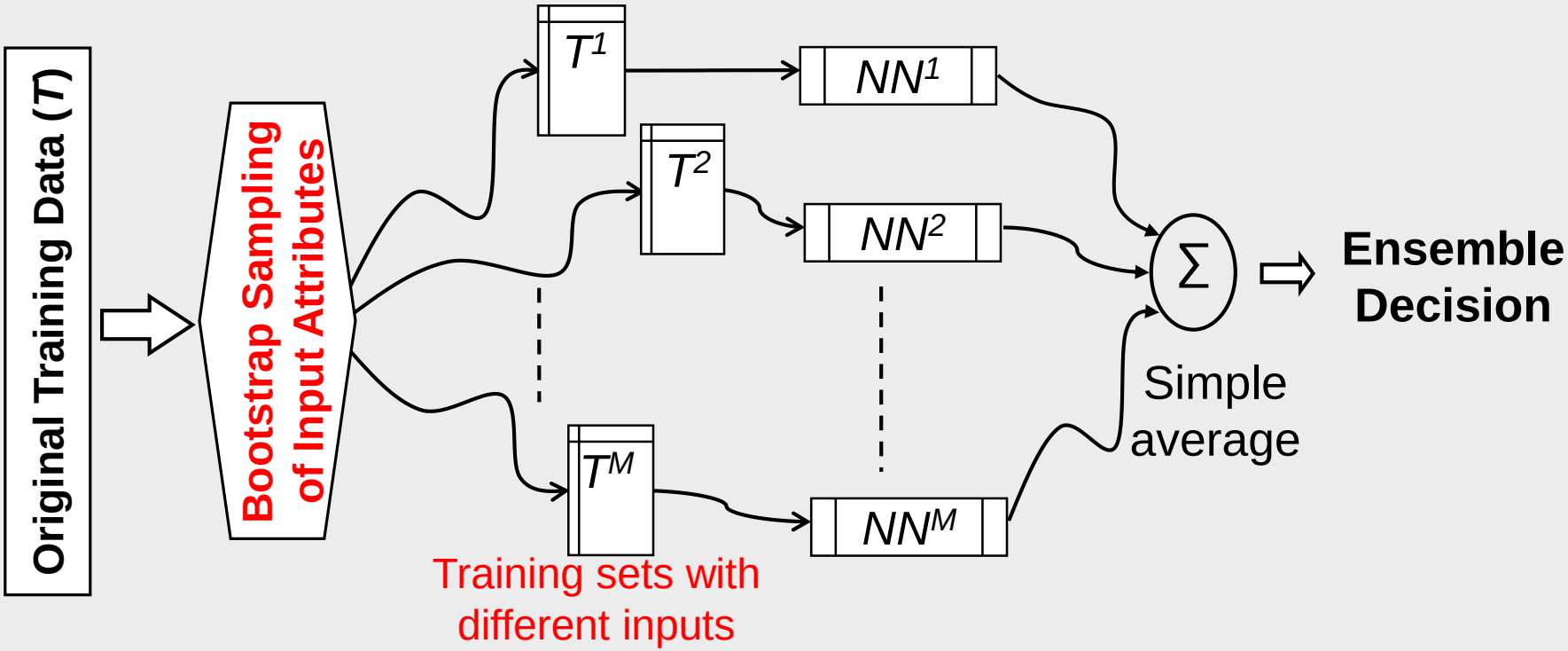
3. Ensemble decision is made in weighted voting way

AdaBoost

17

3. Random Subspace Method(RSM) (Ho, 1998)

T. K. Ho, "The random subspace method for constructing decision forests" *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.* 20, pp. 832–844, 1998



RSM require to maintain tagging with original inputs for every NN's inputs.

18 RSM

1. Let M be the number of networks to be trained for an ensemble

   Take original training set T= *{(x(1), d(1)),…, (x(N), d(N))}* with class label $d(n) \in K = \{1,2,.....,k\}$

   and feature set $F = \{1,2,....., f\}$

2. *for i=1 to M* {

   a. Make a feature subset, F_i by sampling $|F|$ features uniformly at random with replacement from F

   b. Train network NN_i by T with feature set F_i

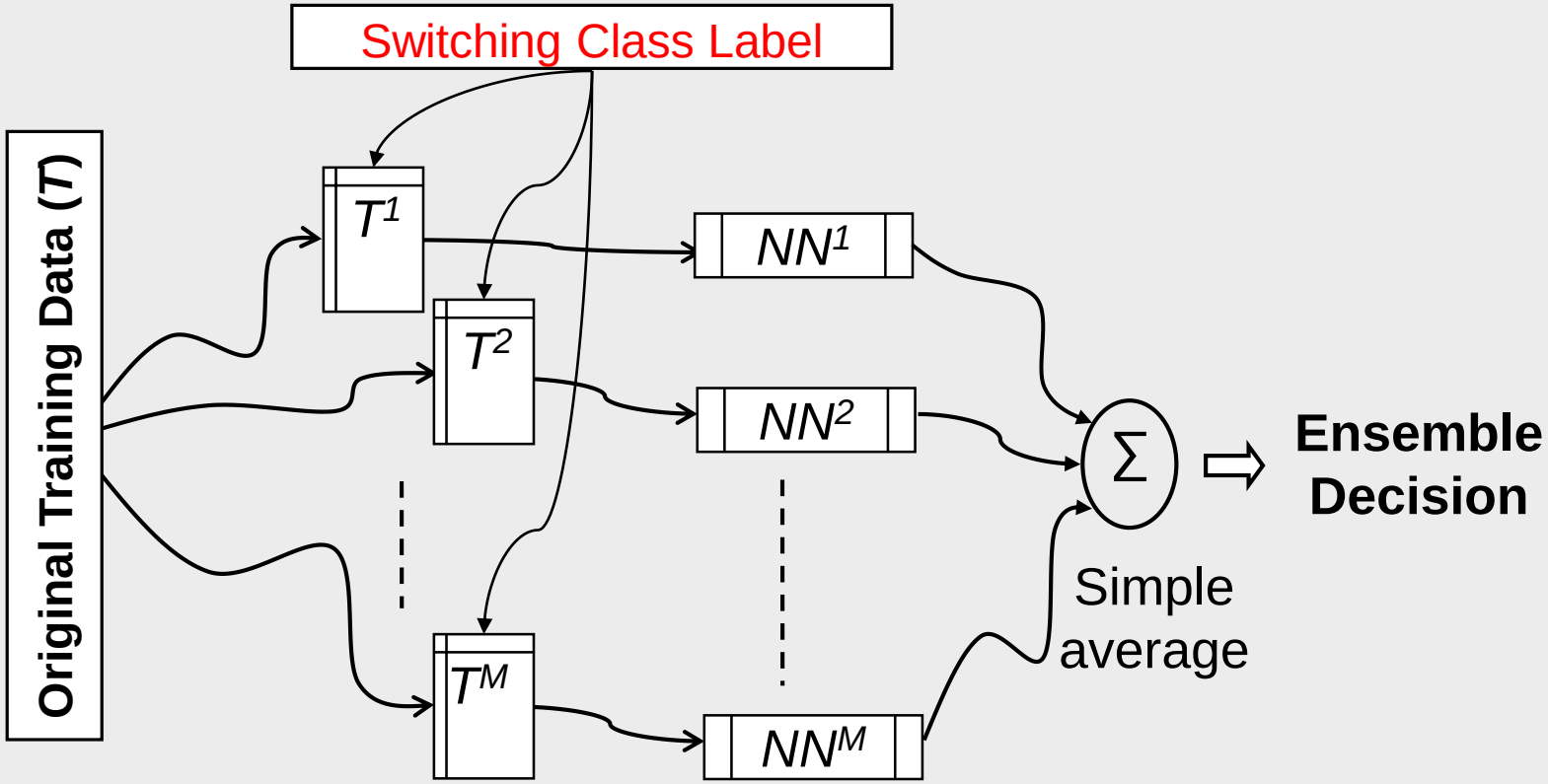
   }

3. Ensemble decision is made in simple average way

19

4. Class Label Switching (Martínez-Muñoz & Suárez, 2005)

G. Mart´ınez-Mu˜noz and A. Su´arez, “Switching class labels to generate classification ensembles,” *Pattern Recognition*, vol. 38, pp. 1483–1494, 2005.

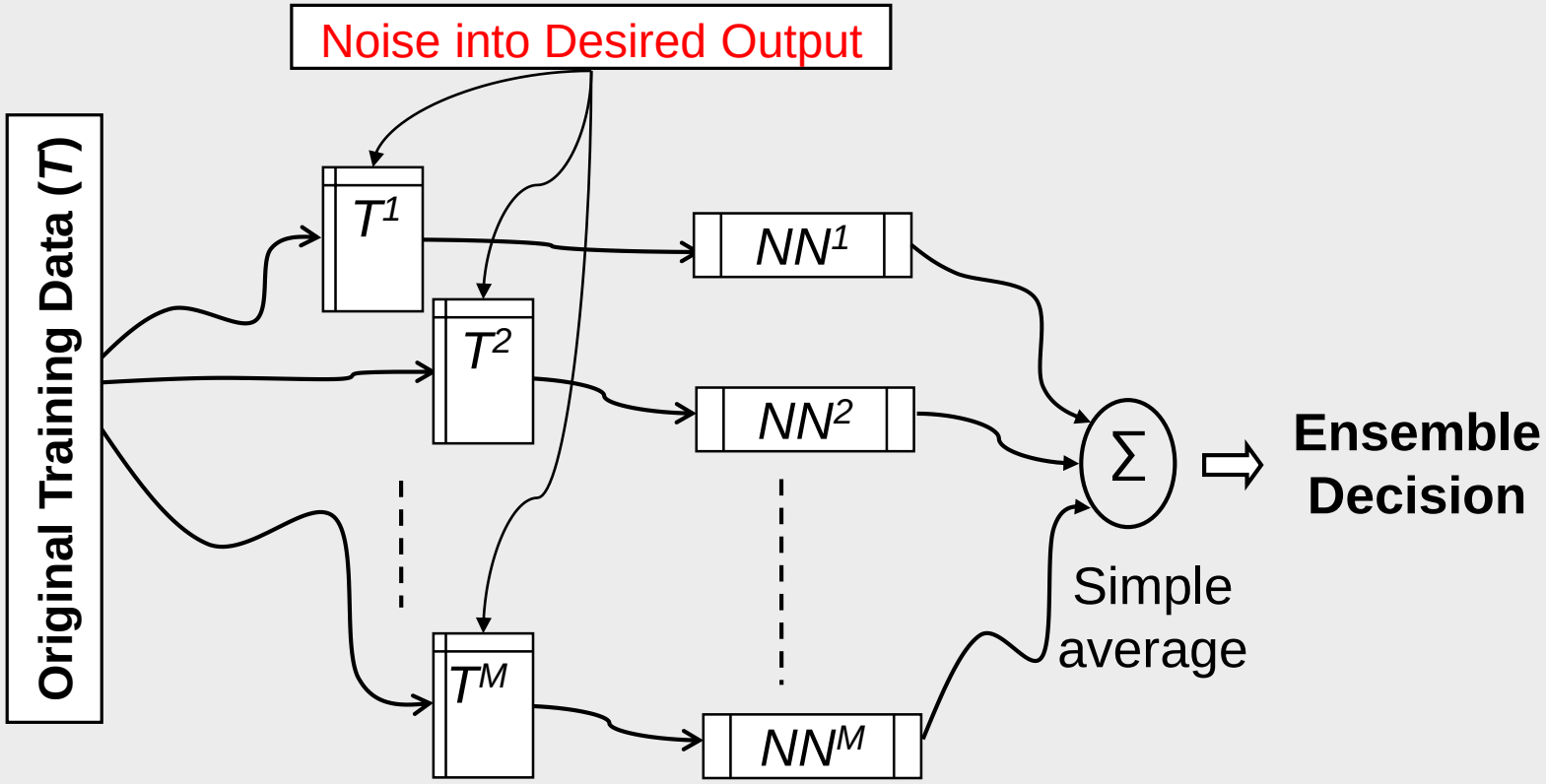


A parameter $S_{fraction}$ maintains number of examples to change class label.

20

5. Smearing (Breiman, 2000)

L. Breiman, "Randomizing outputs to increase prediction accuracy," Machine Learning, vol 40, pp. 229–242, 2000.



21

Class Label Switching and Smearing

1. Let M be the number of networks to be trained for an ensemble

   Take original training set $T= \{(x(1), d(1)),\ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,.....,k\}$

   Take $S_{Fraction}$ -factor that determines number of pattern to alter class label

   Number of patters to switch the class label, $S = S_{Fraction} * N$

2. *for i=1 to M* {

   a. Make a training set, $T_i = T$

   b. Randomly select S examples in T_i and assign different class label randomly

   c. Train network NN_i by T_i

   }

3. Ensemble decision is made in simple average way

Class Label Switching

1. Let M be the number of networks to be trained for an ensemble

   Take original training set $T= \{(x(1), d(1)),\ldots, (x(N), d(N))\}$ with class label $d(n) \in K = \{1,2,.....,k\}$

   Compute standard deviation measure (sd_k) is each class based on Eq. (2.1)

2. *for i=1 to M* {

   a. Make a training set, $T_i = T$

   b. Change the desired output of T_i based on Eq. (2.2)

   c. Train network NN_i by T_i

   }

3. Ensemble decision is made in simple average way

If p_k is the proportion of the patterns in class k, then standard deviation measure (sd)

$$sd_k = 2\,(p_k\,(1 - p_k))\text{^}0.5 \quad (2.1))$$

The new desired output for the k-th class for the n-th training pattern for a network is

$$d_k'(n) = d_k(n) + z_k(n) sd_k \quad k=1,\ldots,K\ n=1,\ldots,N\ (2.2)$$

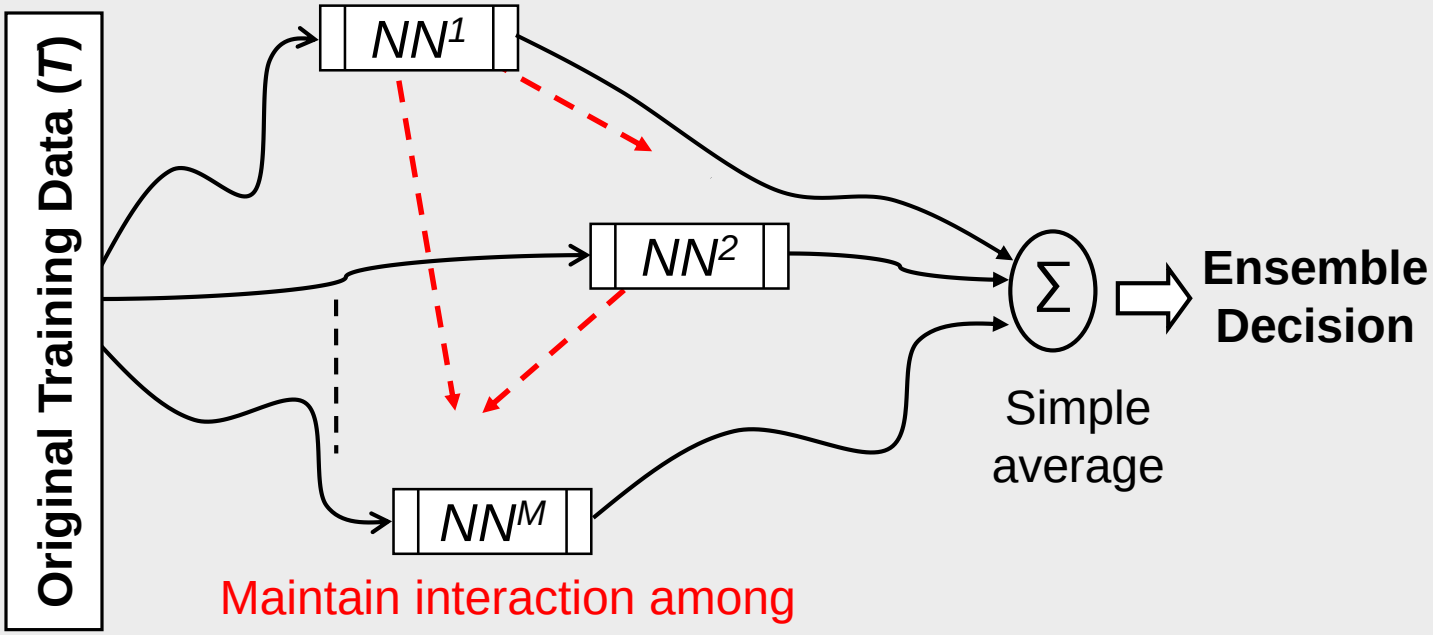
Where $z_k(n)$ is the random number from Gaussian distribution with a mean zero and a variance of one.

Smearing

6. Negative Correlation Learning(NCL) (Liu & Yao, 1999)

Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12 pp. 1399–1404, 1999.

Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Trans. Systems, Man, and Cybernetics — Part B*, vol. 29, pp. 716–725, 1999.



Maintain interaction among component NNs

$$\text{Error function, } e_i(n) = \frac{1}{2}\left(f_i(n) - d(n)\right)^2 + \lambda\left(\left(f_i(n) - \bar{f}(n)\right)\sum_{j\neq i}\left(f_j(n) - \bar{f}(n)\right)\right)$$

Normal BP portion — Correlation penalty term

- Interaction produces negatively correlated NNs and therefore NNs motivate different functional spaces.
- The coefficient of the penalty term (λ) maintain strength of interaction.

23

Negative Correlation Learning(NCL)

1. Let *M* be the number of networks to be trained for an ensemble

Take original training set *T= {(x(1), d(1)),…, (x(N), d(N))}* with class label $d(n) \in K = \{1,2,...,k\}$

Create *M* networks, NN_1 ----NN_M

2. *for n=1 to N* {

Prepare ensemble output for pattern *n*

for i=1 to M {

Train network NN_i for pattern *n* using error definition of Eq. (2.3)

}

}

3. Ensemble decision is made in simple average way

Negative Correlation Learning (NCL)

$$e_i(n) = \frac{1}{2}(d(n) - f_i(n))^2 + \lambda p_i(n) \quad \text{------- Eq. 2.3}$$

$$p_i(n) = (f_i(n) - \overline{f}(n)) \sum_{j \neq i} (f_j(n) - \overline{f}(n))$$

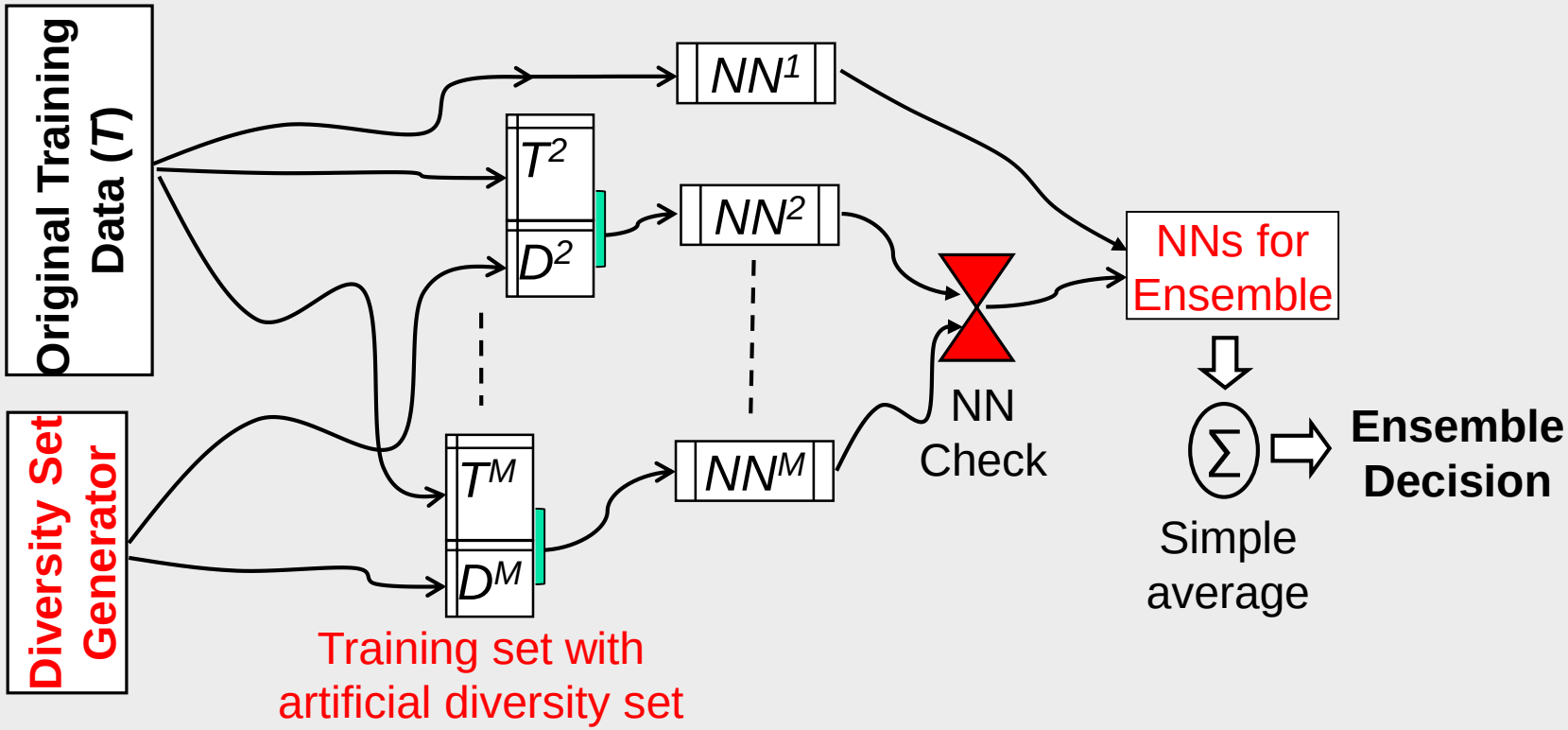
$$\overline{f}(n) = \frac{1}{M} \sum_{i=1}^{M} f_i(n)$$

24

7. DECORATE (Melville & Mooney, 2005)

Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples

P. Melville and R. J. Mooney, "Creating diversity in ensembles using artificial data," *Information Fusion, vol.* 6, pp. 99–111, 2005.



- Trains predefined large number of NNs to select NNs for final NNE
- A parameter R_{size} maintains size of diversity set.
- Training of large number of NNs and additional diversity set increase training time.

25

DECORATE

1. Let I_{max} be the number of networks to be trained for an ensemble and M is the desired number of networks

Take original training set T= *{(x(1), d(1)),…, (x(N), d(N))}* with class label $d(n) \in K = \{1,2,.....,k\}$

Take R_{Size} -factor that determines size of diversity set

i=1 (for network in ensemble) and *trails* =1 (for trial network)

Train network NN_i with T (first network is trained with original training data)

Initialize ensemble, *Ens* ={ NN_i}

Compute ensemble error, $\varepsilon = \left(\sum_{(x(n),d(n)) \in T: Ens(x(n)) \neq d(n)} 1 \right) / N$

2. *while trials* < I_{max} and $i < M$ {

a. Generate $R_{Size} \times |T|$ training examples, R

b. Label examples in R with probability of class labels inversely proportional to predictions of *Ens*

c. Prepare training set T_i, $T_{i=}$ $T \cup R$ and train network NN_i with T_i

d. *Ens* = *Ens* ∪ { NN_i}

e. Compute ε' based on Step 1

f. *if* $\varepsilon' \leq \varepsilon$ *then i = i + 1 and* $\varepsilon = \varepsilon'$, *otherwise Ens = Ens* - { NN_i}

g. *trials = trials +1*

}

3. Ensemble decision is made in simple average way

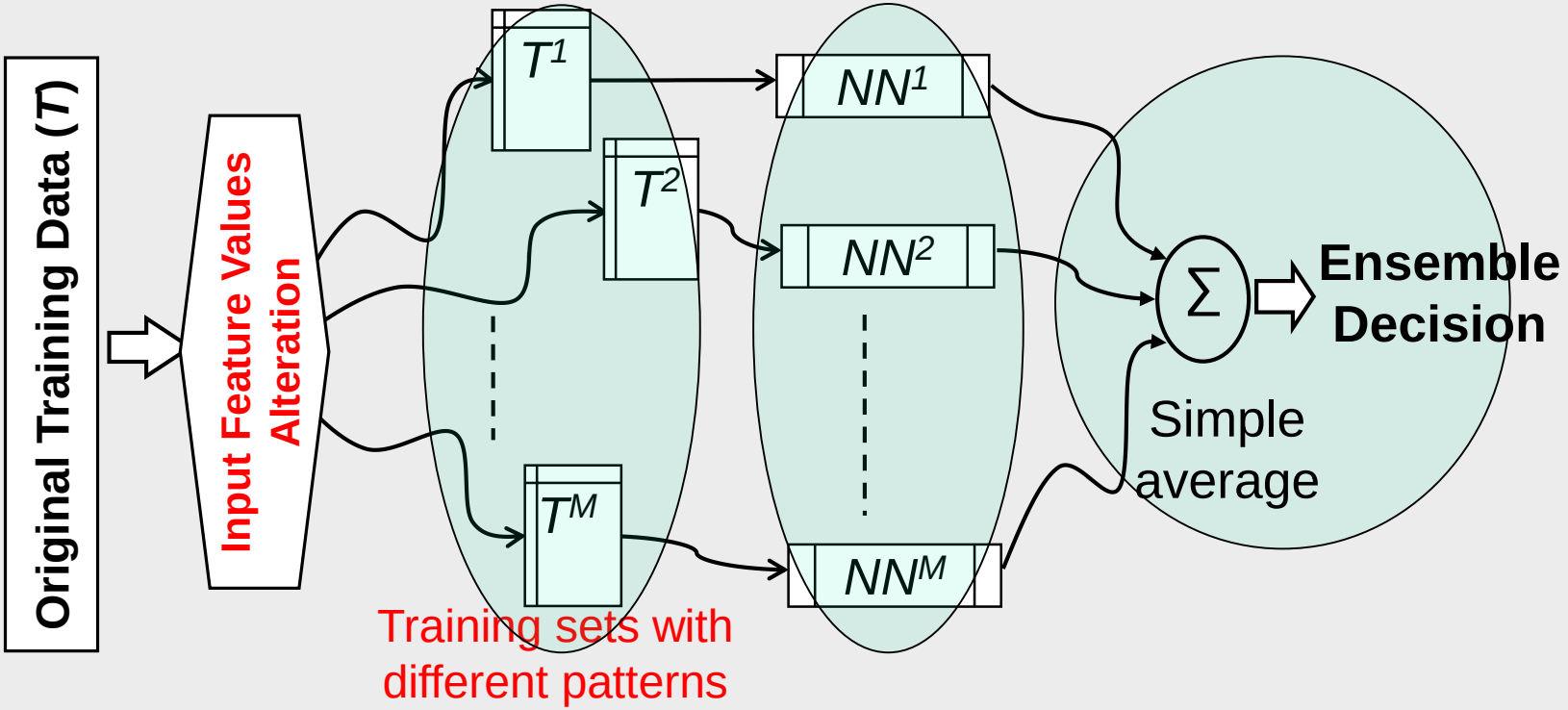
DECORATE

26

8. Ensemble through Input Values Alteration (EIVA)

(M.A.H. Akhand & K. Murase, 2012)

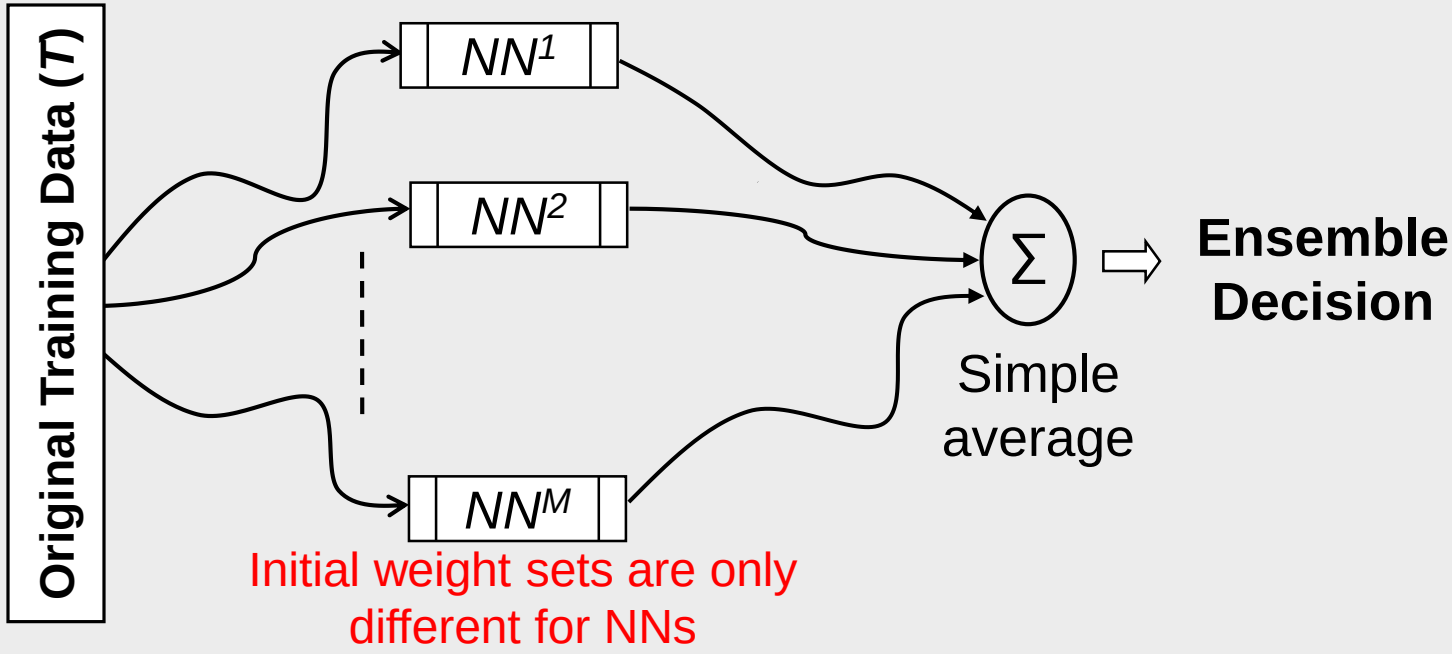
M. A. H. Akhand, and K. Murase , "Ensembles of Neural Networks based on the Alteration of Input Feature Values " *International Journal of Neural Systems*, vol. 22, no.1, pp. 77-87, 2012.



Feature Values Alteration: Randomly selects a pattern and alter some of its feature values with the value of another one.

27

9. Simple NNE(sNNE)



- To evaluate performance of data sampling based NNE methods, sNNE is considered as base line.
- Less diversity due to same data for all NNs; therefore, performance is worse.

28

Motivation to Comparative Study

- A number of ensemble methods already proposed using various data sampling techniques.
- Proposed methods are investigated on heterogeneous test bed.
- Some methods are proposed for decision trees and empirical result for NN is not available.

Comparative study of the proposed methods on a common ground is necessary to evaluate their effectiveness.

(Eight prominent NNE methods are considered for investigation.)

29

Experimental Studies

Common Settings:

- For an NNE 20 NNs are considered.
- Each NN is trained for equal 50 / 75 / 100 iteration.
- Learning rate of back propagation was set 0.15.
- 10-fold cross validations was followed for result presentation.

Built in Parameter Settings:

- For DECORATE R_{Size} value was 0.5, 0.75 or 1. Maximum trial NNs was 30.
- In class label switching $S_{fraction}$ was 0.1, 0.2 or 0.3.
- NCL was tested with λ value 0.25, 0.5 or 0.75.

30 TER Comparison over 50 Indp. Runs

Problem	sNNE 20NNs/NNE	Bagging 20NNs/NNE	AdaBoost 20NNs/NNE	DECORATE (NNs/NNE)	RSM 20NNs/NNE	Switching 20NNs/NNE	Smearing 20NNs/NNE	NCL 20NNs/NNE
ACC	0.1522	0.1417	0.1568	**0.14**(8.34)	0.1461	0.1423	0.1414	0.1443
BCW	0.0348	0.0322	0.0322	0.0299(6.40)	0.0296	**0.029**	0.0319	0.0313
CAR	0.1128	0.0995	**0.0799**	0.1203(2.00)	0.1647	0.118	0.1193	0.1036
DBT	0.2379	0.2321	**0.2305**	0.2342(1.14)	0.2318	0.2382	0.2366	0.2308
GCC	0.2462	0.2424	0.2476	0.2652(1.88)	0.2414	0.2416	0.2482	**0.2402**
HDC	0.1633	0.1573	0.1653	**0.152**(6.78)	**0.152**	0.1567	0.1567	0.1627
HPT	0.1547	0.1627	0.172	0.16(1.16)	0.1573	0.1587	0.1547	**0.152**
HTR	0.0531	0.0518	**0.0263**	0.0528(1.02)	0.0559	0.056	0.0558	0.0522
INS	0.1343	0.1297	0.1034	**0.0606**(12.9)	0.1429	0.132	0.1806	0.1366
IRP	**0.0267**	0.0293	0.028	**0.0267**(1.00)	0.0293	**0.0267**	**0.0267**	**0.0267**
LMP	0.1486	0.1529	0.1729	**0.1371**(4.74)	0.1486	0.14	0.1586	0.15
PRM	0.068	0.068	0.072	**0.066**(20.00)	0.068	0.07	0.078	0.068
SGM	0.07	0.0648	**0.0432**	0.0761(3.14)	0.0681	0.0724	0.0765	0.0677
SNR	0.195	0.194	0.181	**0.166**(7.58)	0.20	0.201	0.209	0.195
SPL	0.1419	0.1556	0.1529	0.1823(2.36)	**0.0873**	0.1527	0.1725	0.1409
STL	0.144	0.1379	**0.1358**	0.1525(2.98)	0.1435	0.1419	0.1556	0.145
WVF	0.1327	**0.1297**	0.132	0.1308(4.18)	0.1352	0.1339	0.1345	0.1312

31

Result Summary over 30 Problems

	sNNE	Bagging	AdaBoost	DECORATE	RSM	Switching	Smearing	NCL
Average TER	**0.1505**	**0.1393**	**0.1380**	**0.1441**	**0.1562**	**0.1512**	**0.1581**	**0.1462**
Best/Worst	**2/3**	**5/1**	**11/6**	**7/2**	**2/7**	**2/3**	**1/9**	**6/0**
NNE Method	**Pair Wise Win/Draw/Loss Summary**							
sNNE	-	23/1/6	19/0/11	17/1/12	12/2/16	11/1/18	8/2/20	20/4/6
Bagging		-	16/1/13	11/0/19	10/2/18	7/1/22	5/0/25	10/1/19
AdaBoost			-	13/0/17	11/0/19	12/0/18	10/0/20	15/0/15
DECORATE				-	11/1/18	10/1/19	5/1/24	16/1/13
RSM					-	17/0/13	12/0/18	22/1/7
Switching						-	6/2/22	19/1/10
Smearing							-	24/1/5

Conclusions from the Comparative Study:

- ❖ No one is superior to others for all the problems.
- ❖ DECORATE performs better for problems with limited examples, e.g., INS, LPM, PRM. NCL also good for small problems.
- ❖ For large sized problems bagging and AdaBoost are the best, e.g., CAR, HRT, WVF. AdaBoost might show very good result for very large problems.
- ❖ RSM performs well for sufficient input set, e.g., SPL.

32

Conclusions

- Basic introduction about NNs and NNE is presented.
- Comparative study of prominent existing methods is given and identified effectiveness of the methods.
- Given an Outline for better NNE construction.