Presentation of Anasua Sarkar

For Soft Computing (CSE/T/425E)

At Computer Science and Engineering Department, Jadavpur University, India

13/01/2020

Topics of Interests

- Hard COMPUTING AND Soft Computing
- Reasoning under Uncertainty
- Fuzzy sets, Rough Sets, Shadow sets
- Fuzzy Rough sets and Rough Fuzzy sets
- Type-2 Fuzzy sets
- Applications with Fuzzy Sets and Autoencoders

What is Hard Computing?

- Hard computing, i.e., conventional computing, requires a precisely stated analytical model and often a lot of computation time.
- Many analytical models are valid for ideal cases.
- Real world problems exist in a non-ideal environment.
- Premises and guiding principles of Hard Computing are
  - Precision, Certainty, and rigor.
- Many contemporary problems do not lend themselves to precise solutions such as
  - Recognition problems (handwriting, speech, objects, images)
  - Mobile robot coordination, forecasting,
  - Combinatorial problems etc.

• Introduction to Soft Computing

The idea of Soft Computing was initiated by Lotfi A. Zadeh.

Definition: " Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve wactability, robustness, and low solution cost. Its principal constituents are fuzzy logic, neurocomputing, and probabilistic reasoning. Soft computing is likely to play an increasingly important role in many application areas, including software engineering. The role model for Soft computing is the human mind."[Lotfi A. Zadeh, 1994]

Zadeh defines SC into one multidisciplinary system as the fusion (Union or Combination) of the fields of Fuzzy Logic, Neuro-Computing, Genetic Computing and Probabilistic Computing.

Fusion of methodologies designed to model and enable solutions to real world problems, which are not modeled or too difficult to model mathematically.

They are composed of two features: "adaptively" & "knowledge".

Soft computing can also be seen as a foundation for the growing field of computational intelligence (CI). The difference between traditional artificial intelligence (AI) and computational intelligence is that AI is based on hard computing whereas CI is based on soft computing.

https://sites.google.com/site/7csesoftcomputing/

- The Soft Computing – development history

SC	=	EC	+	NN	+	FL
Soft Computing		**Evolutionary Computing**		**Neural Network**		**Fuzzy Logic**
Zadeh 1981		**Rechenberg 1960**		**McCulloch 1943**		**Zadeh 1965**

EC	=	GP	+	ES	+	EP	+	GA
Evolutionary Computing		***Genetic Programming***		***Evolution Strategies***		***Evolutionary Programming***		***Genetic Algorithms***
Rechenberg 1960		**Koza 1992**		**Rechenberg 1965**		**Fogel 1962**		**Holland 1970**

Overview of the History of SC

	Conventional AI	Neural networks	Fuzzy systems	Other methodologies
1940s	1947 Cybernetics	1943 McCulloch-Pitts neuron model		
1950s	1956 Artificial Intelligence	1957 Perceptron		
1960s	1960 Lisp language	1960s Adaline Madaline	1965 Fuzzy sets	
1970s	mid- 1970s Knowledge Engineering (expert systems)	1974 Birth of Back-propagation algorithm 1975 Cognitron Neocognitron	1974 Fuzzy controller	1970s Genetic algorithm
1980s		1980 Self-organizing map 1982 Hopfield Net 1983 Boltzmann machine 1986 Backpropagation algorithm boom	1985 Fuzzy modeling (TSK model)	mid- 1980s Artificial life Immune modeling
1990s			1990s Neuro-fuzzy modeling 1991 ANFIS 1994 CANFIS	1990 Genetic programming

Soft Computing

- The real world problems are pervasively imprecise and uncertain
- Precision and certainty carry a cost
- The guiding principle of soft computing is:
  - Exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost.
  - Learning from experimental data
    - **Approximation** : here the model features are similar to the real ones, but not the same.
    - **Uncertainty :** here we are not sure that the features of the model are the same as that of the entity (belief).
    - **Imprecision :** here the model features (quantities) are not the same as that of the real ones, but close to them.

S.NO	SOFT COMPUTING	HARD COMPUTING
1.	Soft Computing is liberal of inexactness, uncertainty, partial truth and approximation.	Hard computing needs a exactly state analytic model.
2.	Soft computing produces approximate results.	Hard computing produces precise results.
3.	Soft computing will emerge its own programs.	Hard computing requires programs to be written.
4.	Soft computing incorporates randomness .	Hard computing is settled.
5.	Soft computing will use multivalued logic.	Hard computing uses two-valued logic.
6.	Soft computing works on ambiguous and noisy data.	Hard computing works on exact data.
7.	Soft computing is stochastic in nature.	Hard computing is deterministic in nature.
8.	Soft computing has the features of approximation and dispositionality.	Hard computing has the features of exactitude(precision) and categoricity.
9.	Soft Computing relies on formal logic and probabilistic reasoning.	Hard computing relies on binary logic and crisp system.

Why soft computing approach?

Soft computing deals with imprecision, uncertainty, partial truth and approximation to achieve tractability, robustness and low solution cost. It extends its application to various disciplines of Engg. and science. Typically human can:

1. Take decisions
2. Inference from previous situations experienced
3. Expertise in an area
4. Adapt to changing environment
5. Learn to do better
6. Social behaviour of collective intelligence.
7. Intelligent control strategies have emerged from the above mentioned characteristics of human/ animals. The first two characteristics have given rise to Fuzzy logic;2 nd , 3 rd and 4 th have led to Neural Networks; 4 th , 5 th and 6 th have been used in evolutionary algorithms.

http://www.vssut.ac.in/lecture_notes/lecture1423723637.pdf

Characteristics of Neuro-Fuzzy & Soft Computing:

1. Human Expertise
2. Biologically inspired computing models
3. New Optimization Techniques
4. Numerical Computation
5. New Application domains
6. Model-free learning
7. Intensive computation
8. Fault tolerance
9. Goal driven characteristics
10. Real world applications

- Intelligent Control Strategies (Components of Soft Computing): The popular soft computing components in designing intelligent control theory are:
- 1. Fuzzy Logic
- 2. Neural Networks
- 3. Evolutionary Algorithms.

http://www.vssut.ac.in/lecture_notes/lecture1423723637.pdf

Neural Networks

- Imitation of the natural intelligence of the brain
- Parallel processing with incomplete information
- Nerve cells function about 10^6 times slower than electronic circuit gates, but human brains process visual and auditory information much faster than modern computers
- The brain is modeled as a continuous-time non linear dynamic system in connectionist architectures
- Distributed representation in the form of weights between a massive set of interconnected neurons

Fuzzy Set Theory

- Human brains interpret imprecise and incomplete sensory information provided by perceptive organs

  - Fuzzy set theory provides a systematic calculus to deal with such information linguistically

  - It performs numerical computation by using linguistic labels stimulated by membership functions

  - It lacks the adaptability to deal with changing external environments ==> incorporate NN learning concepts in fuzzy inference systems: NF modeling

Evolutionary Computation

- Natural intelligence is the product of millions of years of biological evolution

  - Simulation of complex biological evolutionary processes

  - GA is one computing technique that uses an evolution based on natural selection

  - GA and SA population-based systematic random search (RA) techniques

NF and SC characteristics

- With NF modeling as a backbone, SC can be characterized as:

  - Human expertise (fuzzy if-then rules)

  - Biologically inspired computing models (NN)

  - New optimization techniques (GA, SA, RA)

  - Numerical computation (no symbolic AI, only numerical)

https://notendur.hi.is/benedikt/Courses/ch01_2005.pdf

Goal of Soft Computing

- It is a new multidisciplinary field, to construct a new generation of Artificial Intelligence, known as **Computational Intelligence**.
- The main **goal** is: to develop intelligent machines to provide solutions to real world problems, which are not modeled or too difficult to model mathematically.
- Its aim is to exploit (develop) the tolerance for Approximation, Uncertainty, Imprecision, and Partial Truth in order to achieve close resemblance with human like decision making.
- Soft computing methods have been applied to many real-world problems. Applications can be found in signal processing, pattern recognition, quality assurance and industrial inspection, business forecasting, speech processing, credit rating, adaptive process control, robotics control, natural-language understanding, etc.

Fuzzy Logic & NN

- Fuzzy logic is mainly associated to imprecision, approximate reasoning and computing with words, neurocomputing to learning and curve fitting (also to classification), and probabilistic reasoning to uncertainty and belief propagation (belief networks). These methods have in common that they
- 1. are nonlinear,
- 2. have ability to deal with non-linearities,
- 3. follow more human-like reasoning paths than classical methods,
- 4. utilize self-learning,
- 5. utilize yet-to-be-proven theorems,
- 6. are robust in the presence of noise or errors.
- The main dissimilarity between fuzzy logic system (FLS) and neural network is that FLS uses heuristic knowledge to form rules and tunes these rules using sample data, whereas NN forms "rules" based entirely on data.

- Kosko lists the following similarities between fuzzy logic systems and neural networks [Kosko, 1992]:
- estimate functions from sample data
- do not require mathematical model
- are dynamic systems
- can be expressed as a graph which is made up of nodes and edges
- convert numerical inputs to numerical outputs
- process inexact information inexactly
- have the same state space
- produce bounded signals
- a set of n neurons defines n -dimensional fuzzy sets
- learn some unknown probability function p (x)
- can act as associative memories
- can model any system provided the number of nodes is sufficient.

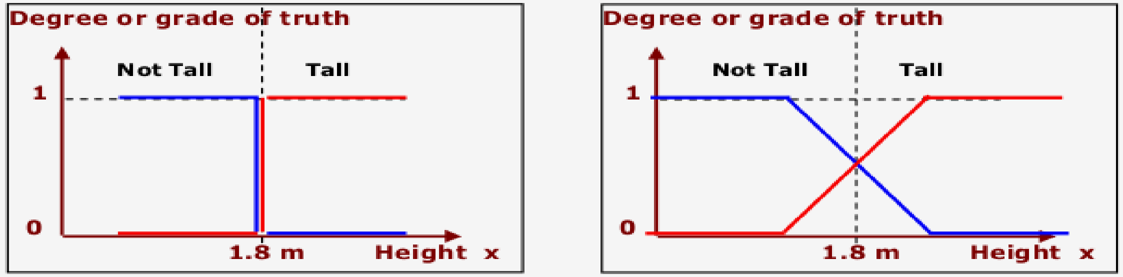
Fuzzy Computing

- In the real world there exists much fuzzy knowledge, that is, knowledge which is vague, imprecise, uncertain, ambiguous, inexact, or probabilistic in nature.
- Human can use such information reasoning frequently involve fuzzy because information, the human thinking and possibly originating from inherently inexact human concepts and matching of similar rather then identical experiences.
- The logic,computing systems, based upon classical set theory and two-valued can not answer to some questions, as human does, because they do not have completely true answers.
- We want, the computing systems should not only give human like answers but also describe their reality levels. These levels need to be calculated using imprecision and the uncertainty of facts and rules that were applied.

This example explains the grade of truth value.

- **tall students** qualify and **not tall students** do not qualify
- if students 1.8 m tall are to be qualified, then
  should we exclude a student who is $^{1}/_{10}$" less? or
  should we exclude a student who is 1" shorter?

- Non-Crisp Representation to represent the notion of a tall person.



Crisp logic | Non-crisp logic

Fig. 1 Set Representation – Degree or grade of truth

A student of height 1.79m would belong to both tall and not tall sets with a particular degree of membership.

As the height increases the membership grade within the tall set would increase whilst the membership grade within the not-tall set would decrease.

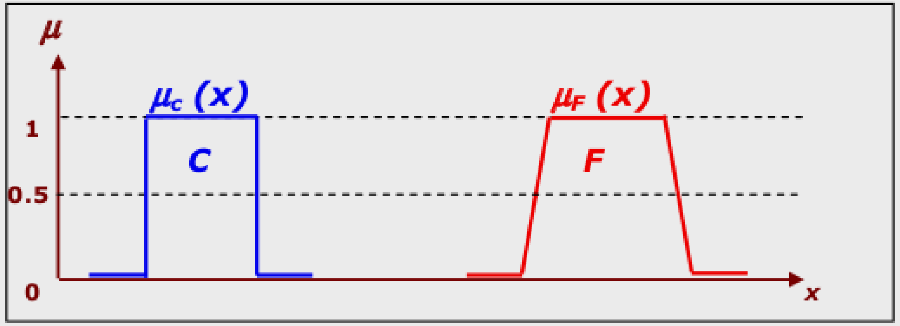


Fig. 2 Membership function of a Crisp set C and Fuzzy set F

- In the case of Crisp Sets the members of a set are :

  either out of the set, with membership of degree " **0** ",
  or in the set, with membership of degree " **1** ",

Therefore, **Crisp Sets ⊆ Fuzzy Sets**

In other words, Crisp Sets are Special cases of Fuzzy Sets.

Fuzzy Sets

- Vagueness in concept formation and representation comes from our inability to describe a precisely defined concept in situations with incomplete information.
- The vagueness is described by differences in representations of the same vague concept in various possible worlds. A fuzzy set is a weighted combination.
- A fuzzy set is defined by a membership function from a universe U to the unit interval [0, 1].
- Typically explicit fuzzy sets are defined in some universe of discourse:
  - Each elopement of the universe is associated with a membership degree.

$$\mu_{\mathcal{A}} : U \longrightarrow [0, 1],$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}}$$

$$d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|^2$$

  - i = assigned cluster among c clusters, k = pattern to allocate, v=centroid

Rough Sets

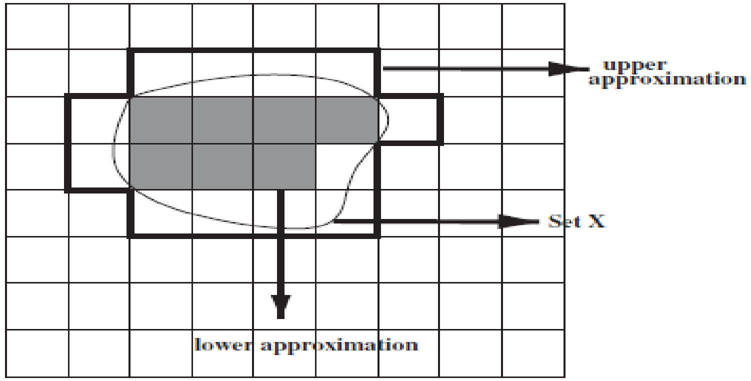


Fig. 1. Lower and upper approximations in a rough set.

- Rough sets = Start with a finite collection of information granules using which any given granule can be expressed in terms of so-called lower and upper bounds.
- The lower approximation of a set is union of all granules which are entirely included in the set.

$$\underline{apr}(A) = \{x \in U \mid [x]_{\Re} \subseteq A\},$$

- The upper approximation - is union of all granules which have non-empty intersection with the set.

$$\overline{apr}(A) = \{x \in U \mid [x]_{\Re} \cap A \neq \emptyset\},$$

- The boundary region of set is the difference between the upper and the lower approximation. The Set X is Rough, if the boundary region of the set is nonempty.
- Using terminologies of fuzzy sets, lower and upper approximations are the core and support of fuzzy set μA.

$$A \subseteq U \qquad \mu_A(x) = \frac{|A \cap [x]_{\Re}|}{|[x]_{\Re}|},$$

$$core(\mu_A) = \{x \mid \mu_A(x) = 1\} = \underline{apr}(A),$$
$$support(\mu_A) = \{x \mid \mu_A(x) > 0\} = \overline{apr}(A).$$

Fuzzy rough sets and rough fuzzy sets

- By using an equivalence relation on U, one can introduce lower and upper approximations in fuzzy set theory to obtain an extended notion called **rough fuzzy sets.**
- A fuzzy similarity relation can be used to replace an equivalence relation, the result is a deviation of rough set theory called **fuzzy rough sets.**

RFCM

1) Assign initial means $\mathbf{v}_i$ for the c clusters.
2) Compute u_{ik} by (3) for c clusters and N data objects.
3) Assign each data object (pattern) $\mathbf{x}_k$ to the lower approximation $\underline{B}U_i$ or upper approximation $\overline{B}U_i$, $\overline{B}U_j$ of cluster pairs U_i and U_j by computing the difference in its membership $u_{ik} - u_{jk}$ to cluster centroid pairs $\mathbf{v}_i$ and $\mathbf{v}_j$.
4) Let u_{ik} be maximum and u_{jk} be the next to maximum.
   If $u_{ik} - u_{jk}$ is less than some threshold,
   then $\mathbf{x}_k \in \overline{B}U_i$ and $\mathbf{x}_k \in \overline{B}U_j$ and $\mathbf{x}_k$ cannot be a member of any lower approximation,
   else $\mathbf{x}_k \in \underline{B}U_i$ such that membership u_{ik} is maximum over the c clusters.
5) Compute new mean for each cluster U_i, incorporating (2) and (3) into (4), as in (9), shown at the bottom of the page.
6) **Repeat** Steps 2)–5) **until** convergence, i.e., there are no more new assignments.

MITRA et al.: ROUGH–FUZZY COLLABORATIVE CLUSTERING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 36, NO. 4,pp. 3795-805,2006

Type-2 fuzzy sets

- Membership degree treated as a single number in [0,1]
- Type-2 fuzzy set: admit membership modeled as fuzzy sets defined in [0,1]
- For type-II fuzzy set $\hat{B}$, a membership function for type-II is defined as: $\mu_{\hat{B}}(y,u)$,

  where $y \in Y$ and $u \in J_y \subseteq [0,1]$ where, $0 \le \mu_{\hat{B}}(y,u) \le 1$

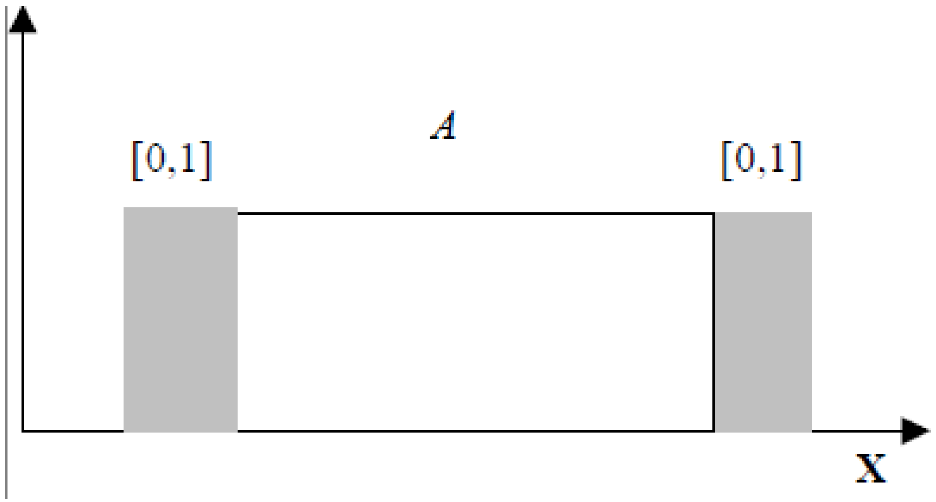
$$\hat{B} = \{(y, \mu_U(y), \mu_L(y)) \mid y \in Y,$$

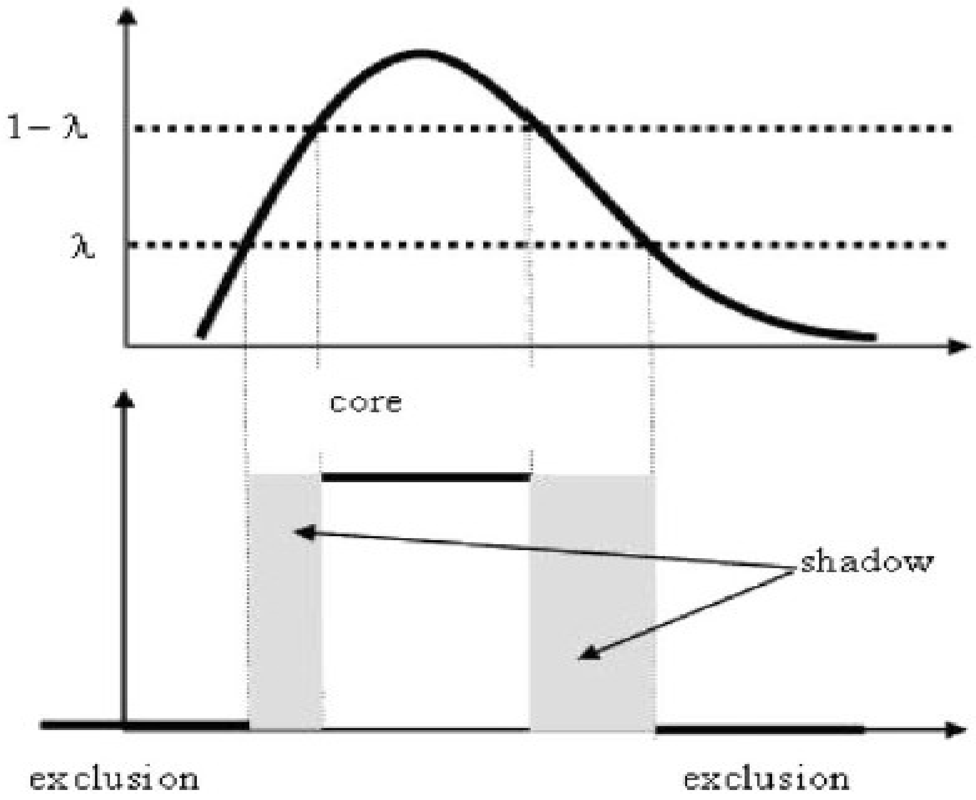
$$\mu_L(y) \le \mu(y) \le \mu_U(y), \mu \in [0,1]\}$$

- μ_U and μ_L are the upper and lower membership degrees of the skeleton (main) membership function μ

Shadowed sets

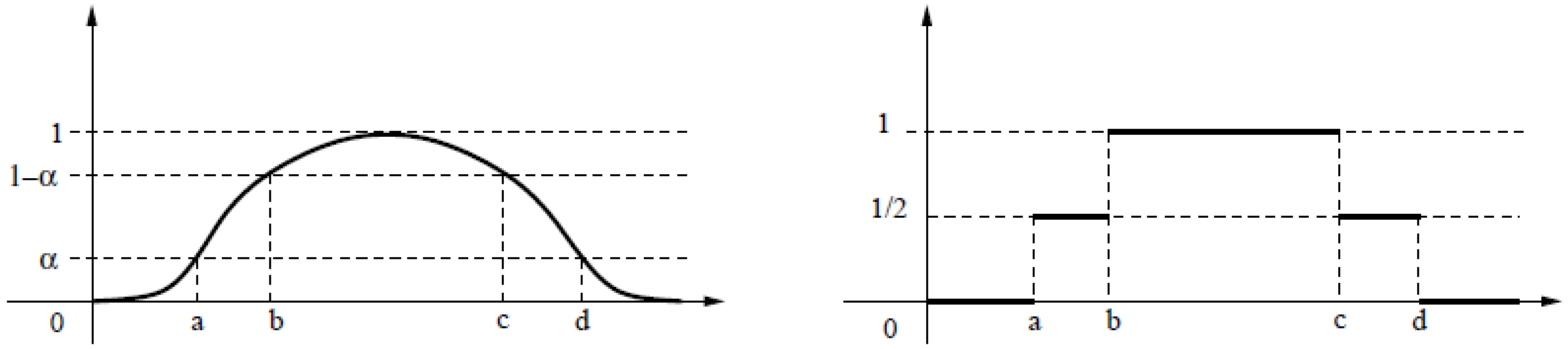
- Information granule A in which :
- Full membership
- Full exclusion, and
- Shadow – range of [0,1]
- Lambda - Threshold





“Shadow” of the induced shadowed set

- A fuzzy set and its corresponding shadowed set



- Once fixed a value α, we can define the α–approximation function of a fuzzy set f, denoted by $s_\alpha(f)$, as the following shadowed set:

$$s_\alpha(f)(x) := \begin{cases} 0 & f(x) \leq \alpha \\ 1 & f(x) \geq 1 - \alpha \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

Neural Networks

- Imitation of the natural intelligence of the brain
- Parallel processing with incomplete information
- Nerve cells function about 10^6 times slower than electronic circuit gates, but human brains process visual and auditory information much faster than modern computers
- The brain is modeled as a continuous-time non linear dynamic system in connectionist architectures
- Distributed representation in the form of weights between a massive set of interconnected neurons

Neural Networks (NN)

- NN are simplified models of the biological neuron system.
- **Neural network:** *information processing paradigm(model) inspired by biological nervous systems, such as our brain*
- **Structure:** large number of highly interconnected processing elements (*neurons*) working together. Inspired by brain.
- Like people, they learn ***<u>from experience</u>*** (by example), therefore **train with known example of problem** to acquire knowledge.
- NN adopt various learning mechanisms (**Supervised and Unsupervised are very popular**)

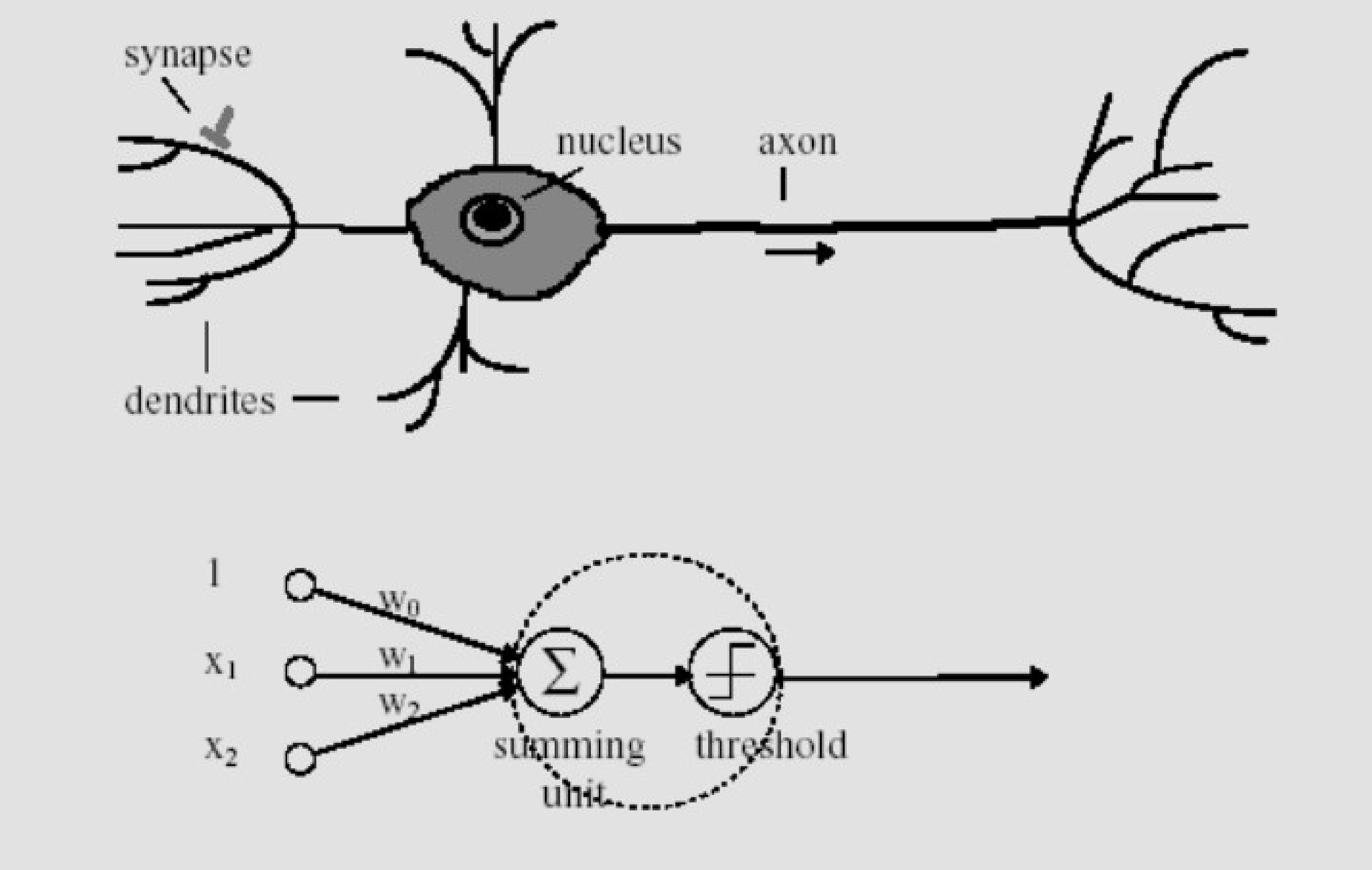


Figure 1.2 Simple illustration of biological and artificial neuron (perceptron).

https://sites.google.com/site/7csesoftcomputing/

Neural Networks (NN)

- **Characteristics**, such as:
  - Mapping capabilities or Pattern recognition.
  - Data classification.
  - Generalization.
  - High speed information processing.
  - Parallel Distributed Processing.
- In a biological system,
  - learning involves adjustments to the synaptic connections between neurons.
- **Architecture:**
  - Feed Forward (Single layer and Multi layer)
  - Recurrent.

https://sites.google.com/site/7csesoftcomputing/

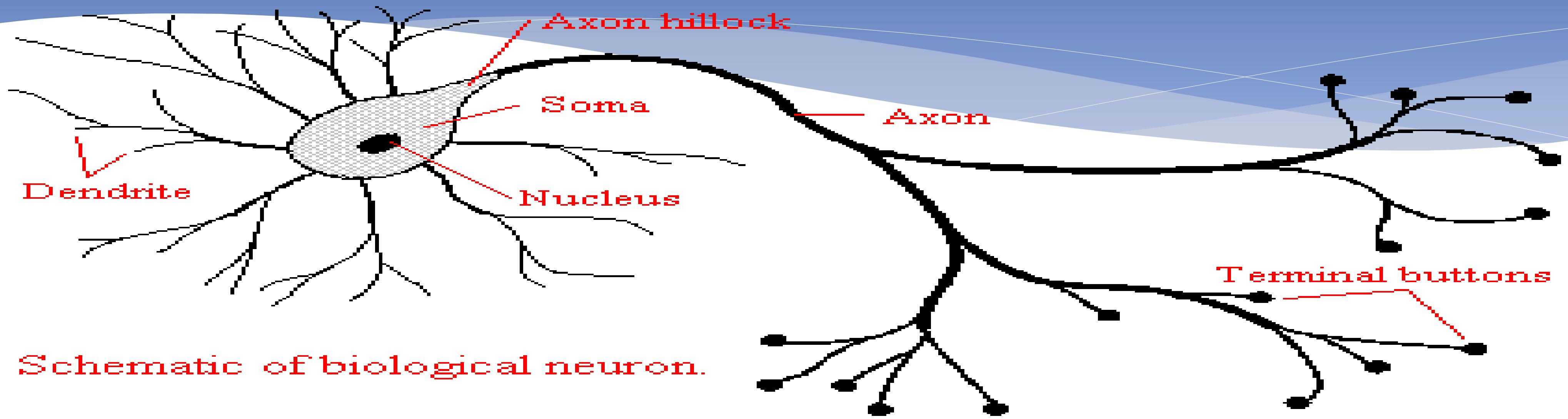
Neural Networks (NN)

Where can neural network systems help.

- When we can't formulate an algorithmic solution.
- When we ***can*** get lots of examples of the behavior we require.
- ‘learning from experience’
- When we need to pick out the structure from existing data.
- Neural networks offer nonlinearity, input-output mapping, adaptivity and fault tolerance.
- Nonlinearity is a desired property if the generator of input signal is inherently nonlinear [Haykin,1994].
- The high connectivity of the network ensures that the influence of errors in a few terms will be minor, which ideally gives a high fault tolerance. (Note that an ordinary sequential computation may be ruined by a single bit error).

Neural Networks (NN)

- **Biological Neuron**.



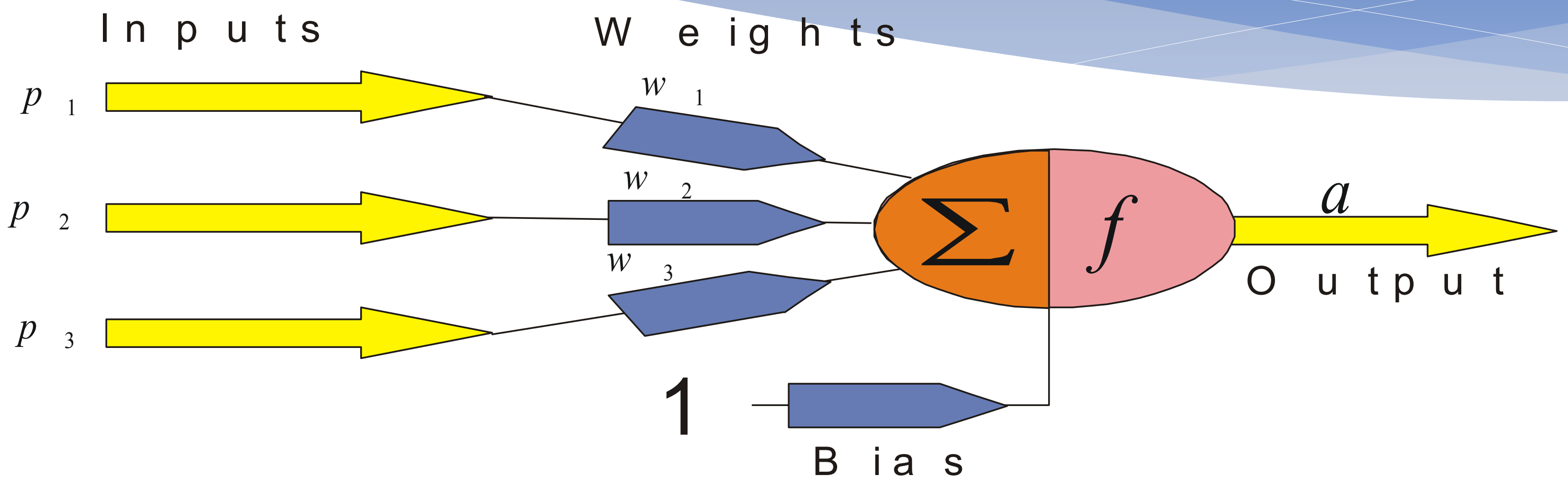
Schematic of biological neuron.

- Brain contain about 10^{10} basic unit called *neurons (Small cell)*.
- Connected to 10^{14} other neurons.
- It receives electro-chemical signals from its various source and transmit electrical impulses to other neurons.
- Average brain weight 1.5 kg, and neuron has $1.5 * 10^{-9}$ gms.

https://sites.google.com/site/7csesoftcomputing/

Key Elements of NN

- Neural computing requires a number of neurons, to be connected together into a neural network. Neuron(s) are arranged in layers.



$$a = f(p_1 w_1 + p_2 w_2 + p_3 w_3 + b) = f(\sum p_i w_i + b)$$

- Each neuron within the network is usually a simple processing unit which takes one or more inputs and produces an output. At each neuron, every input has an associated weight which modifies the strength of each input. The neuron simply adds together all the inputs and calculates an output to be passed on.

Probabilistic Reasoning

- As fuzzy set theory, the probability theory deals with the uncertainty, but usually the type of uncertainty is different.
- Stochastic uncertainty deals with the uncertainty toward the occurrence of certain event and this uncertainty is quantified by a degree of probability. Probability statements can be combined with other statements using stochastic methods. Most known is the Bayesian calculus of conditional probability.
- Probabilistic reasoning includes genetic algorithms, belief networks, chaotic systems and parts of learning theory [Zadeh, 1994].

Genetic Algorithm : INTRUDUCTION

- GA initiated and developed by John Holland (1970)
- GAs are computerized search and optimization algorithms based on mechanics of natural genetics and natural selection.
- It perform direct random searches through a given set of alternatives with the aim of finding the best alternative with respect to the given criteria of goodness.
- The criteria are required to be expressed in terms of objective function (Fitness function)

Genetic Algorithm (Continued)

- Fitness function can either maximized or minimized.
- The alternative (solution) be coded in some specific finite length which consists of symbol from some finite alphabet.
- These strings are called **chromosomes** and the symbols that form the chromosomes are known as **genes**. (color of eye like black,brown)
- For example in the case of binary alphabet (0, 1) the chromosomes are binary string.
- They create a child generation from parent generation according to a set of rules that mimic the genetic reproduction in biology. Randomness plays an important role, since
  - the parents are selected randomly, but the best parents have greater probability of being selected than the others
  - the number of ‘genes’ to be muted is selected randomly
  - all bits in new child string can be flipped with a small probability

Genetic Algorithm (Continued)

- In GA, starting with an initial population of chromosomes (Set of solution).
- Genetic inheritance operators are applied to generate **offspring (Child or Children)** that competes for survival to make up the next generation of population.
- The genetic inheritance operators are:
- ***Reproduction:*** applied on a population for selects good chromosome in a population to form a mating pool.
- ***Crosser over:*** Two strings are picked from the mating pool at random and some segment of the string are exchanged between the string.Cross over operators are:
  - **Single point cross over.**
  - **Two point cross over.**
  - **Matrix crossover.**

Genetic Algorithm (Continue)

- ***Mutation:***
- The operator changes a 1 to 0 and vise versa with a small probability P_m.
- The need for the operator is to keep the diversity of the population.

Fuzzy Applications to Deep Neural Networks

Fuzzy Neural Network with Generalized Hamming Network

- Generalized hamming network with induced fuzzy XOR
- A generalized hamming network (GHN) is any networks consisting of neurons, whose outputs

  $\mathbf{h} \in \mathcal{H}^L$ are related to neuron inputs $\mathbf{x} \in \mathcal{H}^L$ and weights $\mathbf{w} \in \mathcal{H}^L$ by element-wise $\boxed{\mathbf{h} = \mathbf{x} \oplus^L \mathbf{w}}$.
- For the generalized case where $U = \mathbb{R}$, the fuzzy membership μ can be defined by a sigmoid function such as logistic, tanh or any function : $U \to I$. In this work authors adopt the logistic function $\mu(a) = \frac{1}{1+\exp(0.5-a)}$ and the resulting fuzzy XOR connective is given by following membership function:

$$\mu_R(i, j) = \frac{1}{1 + \exp\left(0.5 - \mu^{-1}(i) \oplus \mu^{-1}(j)\right)},$$

- Where $\mu^{-1}(a) = -\ln(\frac{1}{a} - 1) + \frac{1}{2}$ is the inverse of $\mu(a)$ $\mu : U \to I : \mu(a) = i, \mu(b) = j$

generalized hamming distance (GHD), denoted by $\oplus$,

Revisit Fuzzy Neural Network: Demystifying Batch Normalization and ReLU with Generalized Hamming Network, Lixin Fan, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA., pp. 1-10.

Fuzzy Logic and Autoencoder Neural Network to improve data privacy

- Their proposed model is divided into two tasks.
  - The first task is based on hiding the sensitive information using fuzzy membership functions – Triangle, Gaussian and S-shaped.
  - The second task is t data to different o autoencoder.

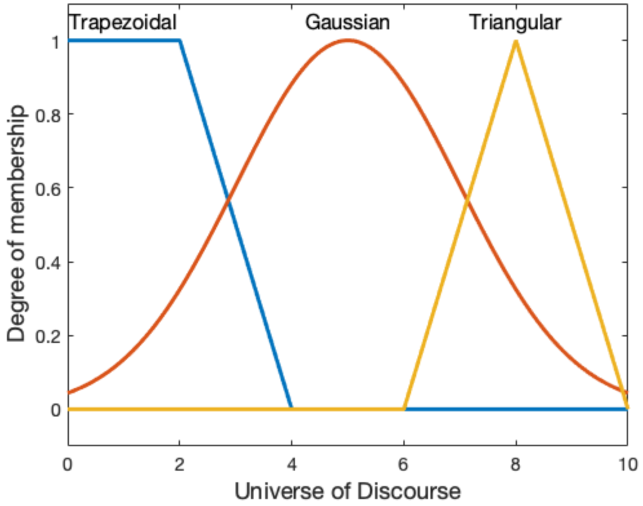


TABLE II
ACCURACY AND LOSS VALUE OF DROP COLUMN DATA SET AND WITH SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	0.1313	0.7409
Gaussian	MSE	0.1260	**0.8734**
Gaussian	Categorical_crossentropy	0.9451	0.0058
Gaussian	Logcosh	0.1263	0.8122
Gaussian	Hinge	0.7091	0.1135
S-shaped	Mean_Absolute	0.1326	0.8180
S-shaped	MSE	0.1261	0.8457
S-shaped	Categorical_crossentropy	1.0220	0.0291
S-shaped	Logcosh	0.1266	0.8239
S-shaped	Hinge	0.6954	0.1397
Triangular	Mean_Absolute	0.1313	0.8122
Triangular	MSE	0.1278	0.8457
Triangular	Categorical_crossentropy	0.0936	0.0015
Triangular	Logcosh	0.1267	0.8384
Triangular	Hinge	0.7059	0.1266