Topic 1:

What is (are)Neural Networks?
What is Machine Learning?

Review and introduction

(we'll get back to different topics later)

Introduction

- Machine learning
  Example---May,1997, IBM supercomputer called Deep Blue beat Kasparov, the world chess champion.
- What is machine learning?
  Machine learning involves adaptive mechanisms that enable computers to learn form experience, learn by example and learn by analogy.
- Artificial neural network (ANN) and Artificial Intelligence (AI) in general is the most popular approach to machine learning.

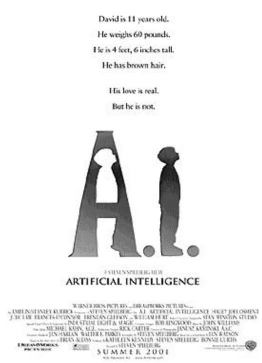
What is AI?

- Director:

Steven Spielberg

Stars: Jude Law, Haley Joel Osment, Frances O'Connor

Plot: In the wake of an environmental disaster, a new kind of self-aware computer is created



What is AI again?:

- Although the term of AI has been widely used for quite a long time with steadily increasing amount of research and applications, there is no anonymously accepted definition. AI can mean many things to different people and various techniques are considered as belonging to AI.
- The term coined in 1956 by J. McCarthy at MIT
- Two branches: engineering discipline dealing with the creation of intelligent machines and empirical science concerned with the computational modelling of human intelligence
- The goal of AI is developing methods, which allow producing thinking machines that can solve problems
- Which problems?
  - ill-defined and ill-structured
  - complicated taxonomy or classifying
  - Combinatorial optimisation

What is AI again?:

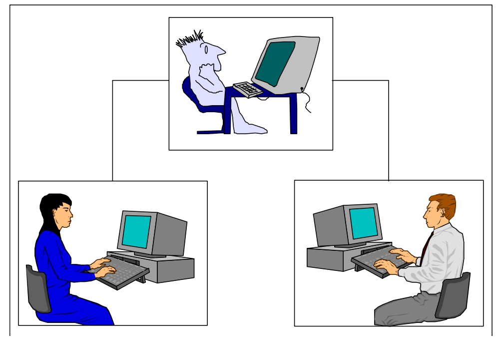
- The great variety of AI techniques have been developed and applied over the history for solving the problems mentioned above.
- Some of these methodologies are"conventional" or "old" methods (1950s):
  - search algorithms,
  - Probabilistic reasoning,
  - natural language processing,
  - belief networks, etc.
- Others are "new" (1960s) – soft computing and computational intelligence

One of the most significant papers on machine intelligence, ***"Computing Machinery and Intelligence"***, was written by the British mathematician ***Alan Turing*** over fifty years ago . However, it still stands up well under the test of time, and the Turing's approach remains universal.

He asked: ***Is there thought without experience? Is there mind without communication? Is there language without living? Is there intelligence without life?*** All these questions, as you can see, are just variations on the fundamental question of artificial intelligence, ***Can machines think?***

- Turing did not provide definitions of machines and thinking, he just avoided semantic arguments by inventing a game, the ***Turing Imitation Game***.
- The imitation game originally included two phases. In the first phase, the interrogator, a man and a woman are each placed in separate rooms. The interrogator's objective is to work out who is the man and who is the woman by questioning them. The man should attempt to deceive the interrogator that *he* is the woman, while the woman has to convince the interrogator that *she* is the woman.

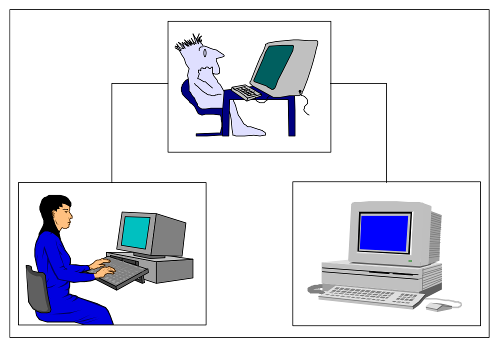
Turing Imitation Game: Phase 1



Turing Imitation Game: Phase 2

- In the second phase of the game, the man is replaced by a computer programmed to deceive the interrogator as the man did. It would even be programmed to make mistakes and provide fuzzy answers in the way a human would. If the computer can fool the interrogator as often as the man did, we may say this computer has passed the intelligent behaviour test.

Turing Imitation Game: Phase 2



The Turing test has two remarkable qualities that make it really universal.

- By maintaining communication between the human and the machine via terminals, the test gives us an objective standard view on intelligence.
- The test itself is quite independent from the details of the experiment. It can be conducted as a two-phase game, or even as a single-phase game when the interrogator needs to choose between the human and the machine from the beginning of the test.

- Turing believed that by the end of the 20th century it would be possible to program a digital computer to play the imitation game. Although modern computers still cannot pass the Turing test, it provides a basis for the verification and validation of knowledge-based systems.
- **A program thought intelligent in some narrow area of expertise is evaluated by comparing its performance with the performance of a human expert.**
- To build an intelligent computer system, we have to capture, organise and use human expert knowledge in some narrow area of expertise.

The history of artificial intelligence

The birth of artificial intelligence (1943 – 1956)

The first work recognised in the field of AI was presented by **Warren McCulloch** and **Walter Pitts** in 1943. They proposed a model of an artificial neural network and demonstrated that simple network structures could learn.

McCulloch, the second "founding father" of AI after Alan Turing, had created the corner stone of neural computing and artificial neural networks (ANN).

- The third founder of AI was **John von Neumann**, the brilliant Hungarian-born mathematician. In 1930, he joined the Princeton University, lecturing in mathematical physics. He was an adviser for the Electronic Numerical Integrator and Calculator project at the University of Pennsylvania and helped to design the **Electronic Discrete Variable Calculator**. He was influenced by McCulloch and Pitts's neural network model. When **Marvin Minsky** and **Dean Edmonds**, two graduate students in the Princeton mathematics department, built the first neural network computer in 1951, von Neumann encouraged and supported them.

- Another of the first generation researchers was **Claude Shannon**. He graduated from MIT and joined Bell Telephone Laboratories in 1941. Shannon shared Alan Turing's ideas on the possibility of machine intelligence. In 1950, he published a paper on chess-playing machines, which pointed out that a typical chess game involved about 10^{120} possible moves (Shannon, 1950). Even if the new von Neumann-type computer could examine one move per microsecond, it would take 3×10^{106} years to make its first move. Thus Shannon demonstrated the need to use heuristics in the search for the solution.

- In 1956, **John McCarthy**, **Marvin Minsky** and **Claude Shannon** organised a summer workshop at Dartmouth College. They brought together researchers interested in the study of machine intelligence, artificial neural nets and automata theory. Although there were just ten researchers, this workshop gave birth to a new science called ***artificial intelligence***.

The rise of artificial intelligence, or the era of great expectations (1956 – late 1960s)

- The early works on neural computing and artificial neural networks started by McCulloch and Pitts was continued. Learning methods were improved and **Frank Rosenblatt** proved the ***perceptron convergence theorem***, demonstrating that his learning algorithm could adjust the connection strengths of a perceptron.

- One of the most ambitious projects of the era of great expectations was the **General Problem Solver (GPS)**. **Allen Newell** and **Herbert Simon** from the Carnegie Mellon University developed a general-purpose program to simulate human-solving methods.
- Newell and Simon postulated that a problem to be solved could be defined in terms of ***states***. They used the mean-end analysis to determine a difference between the current and desirable or *goal state* of the problem, and to choose and apply *operators* to reach the goal state. The set of operators determined the solution plan.

- However, GPS failed to solve complex problems. The program was based on formal logic and could generate an infinite number of possible operators. The amount of computer time and memory that GPS required to solve real-world problems led to the project being abandoned.
- In the sixties, AI researchers attempted to simulate the thinking process by inventing *general methods* for solving *broad classes of problems*. They used the general-purpose search mechanism to find a solution to the problem. Such approaches, now referred to as ***weak methods***, applied weak information about the problem domain.

- By 1970, the euphoria about AI was gone, and most government funding for AI projects was cancelled. AI was still a relatively new field, academic in nature, with few practical applications apart from playing games. So, to the outsider, the achieved results would be seen as toys, as no AI system at that time could manage real-world problems.

Unfulfilled promises, or the impact of reality (late 1960s – early 1970s)

The main difficulties for AI in the late 1960s were:

- Because AI researchers were developing general methods for broad classes of problems, early programs contained little or even no knowledge about a problem domain. To solve problems, programs applied a search strategy by trying out different combinations of small steps, until the right one was found. This approach was quite feasible for simple **toy problems**, so it seemed reasonable that, if the programs could be "scaled up" to solve large problems, they would finally succeed.

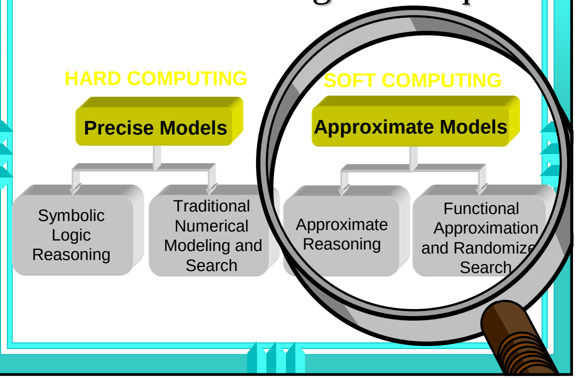
- Many of the problems that AI attempted to solve were **too broad and too difficult**. A typical task for early AI was machine translation. For example, the National Research Council, USA, funded the translation of Russian scientific papers after the launch of the first artificial satellite (Sputnik) in 1957. Initially, the project team tried simply replacing Russian words with English, using an electronic dictionary. However, it was soon found that translation requires a general understanding of the subject to choose the correct words. This task was too difficult. In 1966, all translation projects funded by the US government were cancelled.

- In 1971, the British government also suspended support for AI research. Sir **James Lighthill** had been commissioned by the Science Research Council of Great Britain to review the current state of AI. He did not find any major or even significant results from AI research, and therefore saw no need to have a separate science called "artificial intelligence".

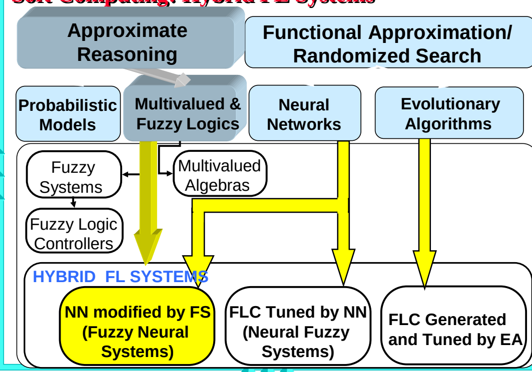
Soft Computing

- Soft Computing (SC): the symbiotic use of many emerging problem-solving disciplines.
  - According to Prof. Zadeh:
    "...in contrast to traditional hard computing, soft computing exploits the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution-cost, and better rapport with reality"
  - Soft Computing Main Components:
    - -Approximate Reasoning:
      - » Probabilistic Reasoning, Fuzzy Logic
    - -Search & Optimization:
      - » Neural Networks, Evolutionary Algorithms

Problem Solving Techniques



Soft Computing: Hybrid FL Systems



Fuzzy Logic Genealogy

- Origins: MVL for treatment of imprecision and vagueness
  - 1930s: Post, Kleene, and Lukasiewicz attempted to represent *undetermined, unknown*, and other possible intermediate truth-values.
  - 1937: Max Black suggested the use of a *consistency profile* to represent vague (ambiguous) concepts
  - 1965: Zadeh proposed a complete theory of fuzzy sets (and its isomorphic fuzzy logic), to represent and manipulate ill-defined concepts

This is fuzzy math!

Fuzzy Logic : Linguistic Variables

- Fuzzy logic give us a language (with syntax and local semantics), in which we can translate our qualitative domain knowledge.
- Linguistic variables to model dynamic systems
- These variables take linguistic values that are characterized by:
  - a label - a sentence generated from the syntax
  - a meaning - a membership function determined by a local semantic procedure

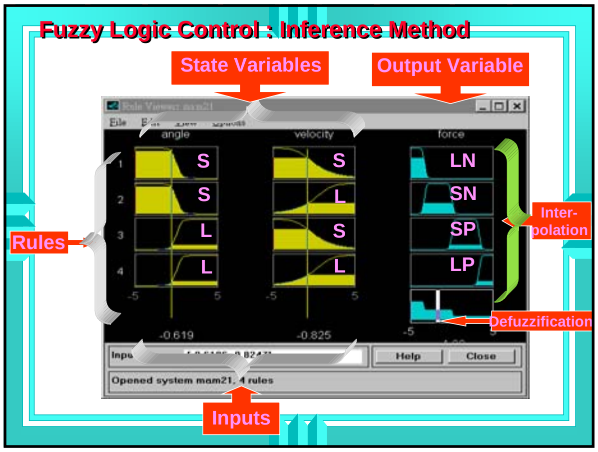
Fuzzy Logic : Reasoning Methods

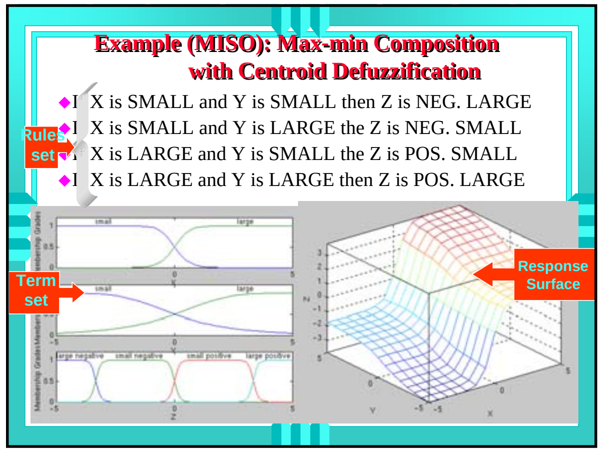
- The meaning of a linguistic variable may be interpreted as a elastic constraint on its value.
- These constraints are propagated by fuzzy inference operations, based on the generalized

•A FL Controller (FLC) applies this reasoning system to a Knowledge Base (KB) containing the problem domain heuristics.

• The inference is the result of interpolating among the outputs of all relevant rules.

• The outcome is a membership distribution on the output space, which is defuzzified to produce a crisp output.





How to make a machine learn, or the rebirth of neural networks (mid-1980s – onwards)

- In the mid-eighties, researchers, engineers and experts found that building an expert system required much more than just buying a reasoning system or expert system shell and putting enough rules in it. Disillusions about the applicability of expert system technology even led to people predicting an *AI "winter"* with severely squeezed funding for AI projects. AI researchers decided to have a new look at neural networks.

- By the late sixties, most of the basic ideas and concepts necessary for neural computing had already been formulated. However, only in the mid-eighties did the solution emerge. The major reason for the delay was technological: there were no PCs or powerful workstations to model and experiment with artificial neural networks.
- In the eighties, because of the need for brain-like information processing, as well as the advances in computer technology and progress in neuroscience, the field of neural networks experienced a dramatic resurgence. Major contributions to both theory and design were made on several fronts.

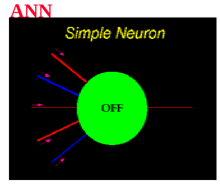
- Grossberg established a new principle of self-organisation (***adaptive resonance theory***), which provided the basis for a new class of neural networks (Grossberg, 1980).
- Hopfield introduced neural networks with feedback – ***Hopfield networks***, which attracted much attention in the eighties (Hopfield, 1982).
- Kohonen published a paper on ***self-organising maps*** (Kohonen, 1982).
- Barto, Sutton and Anderson published their work on ***reinforcement learning*** and its application in control (Barto et al., 1983).

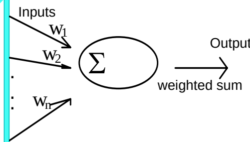
- But the real breakthrough came in 1986 when the ***back-propagation learning algorithm***, first introduced by Bryson and Ho in 1969 (Bryson & Ho, 1969), was reinvented by Rumelhart and McClelland in *Parallel Distributed Processing* (1986).
- Artificial neural networks have come a long way from the early models of McCulloch and Pitts to an interdisciplinary subject with roots in neuroscience, psychology, mathematics and engineering, and will continue to develop in both theory and practical applications.

AI: Is there any future?

- Microsoft, Amid Dwindling Interest, Talks Up Computing as a Career March 1, 2004 By STEVE LOHR
  Mr. Gates scoffed at the notion, advanced by some, that the computer industry was a mature business of waning opportunity. In one question-and-answer session, a student asked if there could ever be another technology company as successful as Microsoft.
  "If you invent a breakthrough in artificial intelligence, so machines can learn," Mr. Gates responded, "that is worth 10 Microsofts."
  Http://www.nytimes.com/2004/03/01/technology/01bill.html?ex=1079243652&ei=1&en=c1b404b245663b75

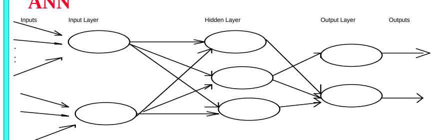
ANN





- The artificial neuron is a mathematical construct that emulates the more salient function of biological neurons, namely this signal integration and threshold firing behavior. Just as in the biological case, such neurons are bound together by various connection weights that determine how the outputs from one neuron are to be algebraically weighted before arriving at receiving neurons. The intelligence within these collective structures of artificial neurons (i.e., ANNs) is stored within these sundry algebraic connection weights.

ANN

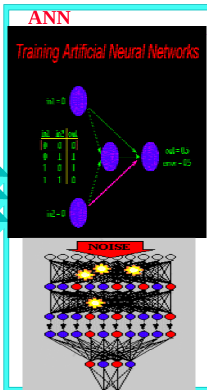


- **All of the information stored within an artificial neural network (i.e., its virtual computer programs) takes the form of connection strengths between neurons. These are values by which the signals from one artificial neuron to another are multiplied before being summed up within the receiving neuron. Important to note is that these weights are not 'hand wired' into these networks by computer nerds. Instead, special computer programs mathematically 'spank' the net until it consistently yields the correct outputs for any given set of inputs.**

The new era of knowledge engineering, or computing with words (late 1980s – onwards)

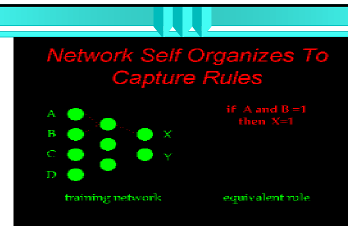
- Neural network technology offers more natural interaction with the real world than do systems based on symbolic reasoning. Neural networks can learn, adapt to changes in a problem's environment, establish patterns in situations where rules are not known, and deal with fuzzy or incomplete information. However, they lack explanation facilities and usually act as a black box. The process of training neural networks with current technologies is slow, and frequent retraining can cause serious difficulties.

ANN



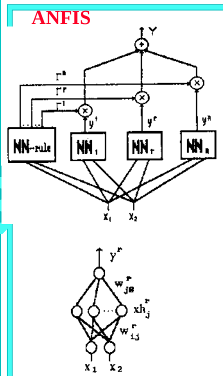
- **ANN Training -we successively apply all known inputs to the net (here the Exclusive Or data) propagating signals in the forward direction, observe network output, and then backwardly propagate corrections to the respective connections in the net. We continue this process until the net yields the correct output for all known test cases. At this point we say that we have a neural network model of some conceptual space. Applications of inputs unencountered during the network's training phase should yield reasonable estimates for network outputs (i.e., the model's predictions). The most important aspect of this process is that the network discovers on its own what the underlying rules actually are.**

ANN



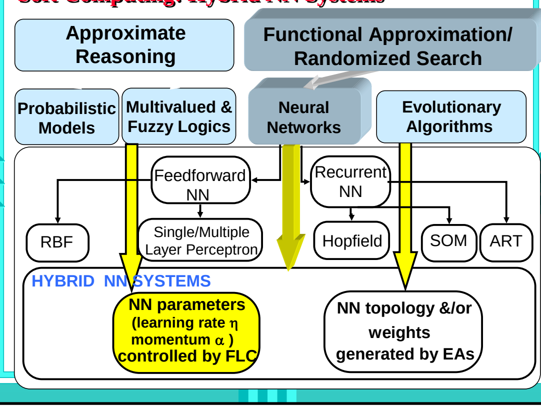
- **How ANNs Capture Rules -** Artificial Neural Networks have taken a rap for being 'black boxes'. That is, they give the right results, but don't explain why they do so. In reality, they internally develop connection traces that embody the rules behind the conceptual space they are training on. Here we see a network learning three implicit rules hidden within a database of numbers.

ANFIS



- **The ways to combine FL and ANN:**
  1) fuzzy systems where ANN learn the shape of the surface of membership functions, the rules and output membership values,
  2) fuzzy systems that are expressed in the form of ANN and are designed using a learning capability of the ANN,
  3) fuzzy systems with ANN which are used to tune the parameters of the fuzzy controller as a design tool but not as a component of the final fuzzy system.

Soft Computing: Hybrid NN Systems



L. Reznik

Different NN Types

- **Radial Basis Network**
- Introduced as a solution to multi-variate interpolation by Powell in 1985
- Usually formulated with a hidden layer of RB neurons followed by a linear output layer
- The hidden layer contains radially symmetric basis functions, hence the name of NN
  - In the case of a single input system, the basis function input is the absolute difference between the input and weight value, however when considering multiple inputs, the euclidian distance is used.
- Training of these networks is accomplished using that back-propagation algorithm.
  - A network with equidistant spaced RB weights within the input space is proposed to reduce the memory required to implement a NN. Effectively this reduces the number of weights needed to recall the network by a factor of three for a two input NN. This type of NN will be referred to as a radial basis with a fixed first layer (RBF).
- **Multi-Layer Perceptron**
- Introduced by Frank Rosenblatt initially in early 1960s
- Many modifications and variations
- Trained with a Levenberg-Marquardt optimisation technique which is more sophisticated than the steepest descent technique

L. Reznik

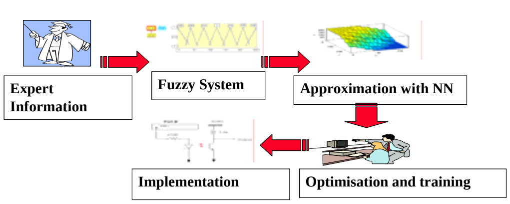
Techniques and Comparison of Results

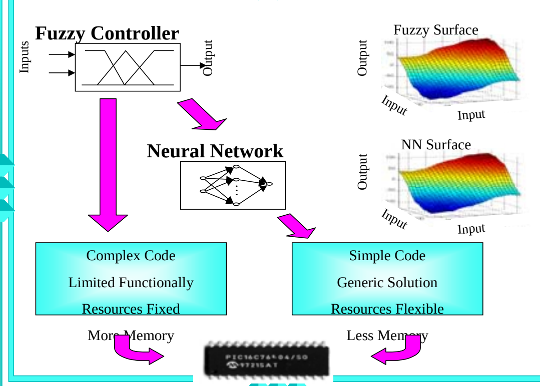
- Commonalities:
- Both are non-linear feedforward NNs and can replace each other.
- Performance
- Evaluated on four practical FCs:
- FC to control motors that mechanically feed banknotes in automatic tellers.
- FC for an anti-lock braking system in a car.
- FC used by a robotic arm for force feedback movement
- FC applied for automatic cruise control.

L. Reznik

Implementation of FS with NN

- **Advantages:**
- Knowledge acquisition ability of FS
- Learning ability of NN
- Optimisation and adjustment against any criteria (including multi-criteria)
- Simpler and cheaper implementation



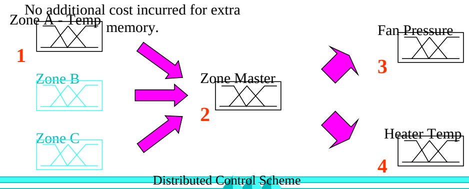


Multi-Zone Thermostat Controller

Project Benefited from 4 different Fuzzy Controllers.

Implemented with the same Neural Network Engine, using different neural weights for each controller.

No additional cost incurred for extra memory.



Evolutionary Algorithms (EA)

EA are part of the Derivative-Free Optimization and Search Methods:

- **Evolutionary Algorithms**
- **Simulated annealing (SA)**
- **Random search**
- **Downhill simplex search**
- **Tabu search**

EA consists of:

- **Evolution Strategies (ES)**
- **Evolutionary Programming (EP)**
- **Genetic Algorithms (GA)**
  - **Genetic Programming (GP)**

Evolutionary Algorithms Characteristics

◆Most Evolutionary Algorithms can be described by

$$\mathbf{x}[t+1] = s(v(\mathbf{x}[t]))$$

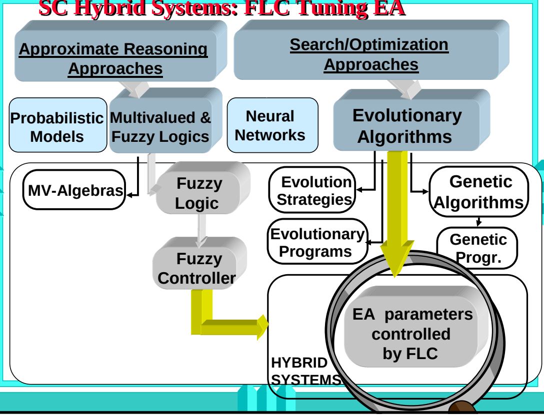
- $x[t]$: the population at time t under representation x
- v : is the variation operator(s)
- s : is the selection operator

Evolutionary Algorithms Characteristics

◆**EA exhibit an *adaptive behavior* that allows them to handle non-linear, high dimensional problems without requiring differentiability or explicit knowledge of the problem structure.**

◆**EA are very robust to time-varying behavior, even though they may exhibit low speed of convergence.**

SC Hybrid Systems: FLC Tuning EA



Evolutionary Algorithms: ES

◆Evolutionary Strategies (ES)

- Originally proposed for the optimization of continuous functions
- (m , l)-ES and (m + l)-ES
  - A population of m parents generate l offspring
  - Best m offspring are selected in the next generation
  - (m , l)-ES: parents are **excluded** from selection
  - (m + l)-ES: parents are **included** in selection
- Started as (1+1)-ES ***(Reschenberg)*** and evolved to (m + l)-ES ***(Schwefel)***
- Started with Mutation only (with individual mutation operator) and later added a recombination operator
- Focus on behavior of individuals

Evolutionary Algorithms: EP

- Evolutionary Programming (EP)
  - Originally proposed for sequence prediction and optimal gaming strategies
  - Currently focused on continuous parameter optimization and training of NNs
  - Could be considered a special case of $(\mu + \mu)$-ES without recombination operator
  - Focus on behavior of species (hence no crossover)
  - Proposed by ***Larry Fogel (1963)***

Evolutionary Algorithms: GA

- Genetic Algorithms (GA)
  - Perform a randomized search in solution space using a genotypic rather than a phenotypic
  - Each solution is encoded as a chromosome in a population (a binary, integer, or real-valued string)
    - Each string's element represents a particular feature of the solution
  - The string is evaluated by a fitness function to determine the solution's quality
    - Better-fit solutions survive and produce offspring
    - Less-fit solutions are culled from the population
  - **Strings are evolved using mutation & recombination operators.**
  - **New individuals created by these operators form next generation of solutions**
  - **Started by *Holland (1962; 1975)***

Evolutionary Algorithms: GP

- Genetic Programming (GP)
  - A special case of Genetic Algorithms
    - –Chromosomes have a **hierarchical** rather than a **linear** structure
    - –Their sizes are not predefined
    - –Individuals are tree-structured programs
    - –Modified operators are applied to sub-trees or single nodes
  - **Proposed by *Koza (1992)***

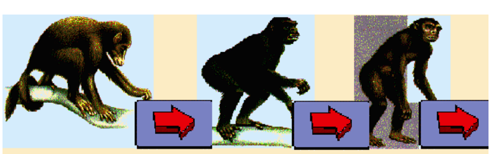
GA Structure (cont.)

- GA Structural Design Selections:
  - **Parent Selection Method:**
    - – Proportional Roulette, Tournament, Rank, Uniform, ... }
  - **Crossover Operator:**
    - –{Once-cut, Two-cuts, Uniform, BLX, Parent Weighted, ...}
  - **Mutation Operator:**
    - –Mutation Rate: {Exponentially Decreasing, Uniform, ..}
    - –Value: {Exponentially Decreasing

L. Reznik

VICTORIA UNIVERSITY

FUZZY CONTROLLER DESIGN METHODOLOGY EVOLUTION



1970s	1980s	1990s
Expert Systems approach	*Control Engineering approach*	*Combined (syntetic) approach*
Fuzzy controller parameters are chosen by an expert(s) on the base of their opinions	Fuzzy controller parameters are assigned on the base of control theory analysis	Fuzzy controller parameters are partly assigned by an expert and then tuned by learning

---

L. Reznik

VICTORIA UNIVERSITY

DESIGN APPROACHES CLASSIFICATION

- 1) expert systems approach
  - **originates from the methodology of expert systems**
  - **justified by a consideration of a FC as an expert system applied to control problem solving.**
  - **fuzzy sets are applied to represent the knowledge or behaviour of a control practitioner (an application expert, an operator) who may be acting only on the subjective or intuitive knowledge.**
  - **too subjective and prone to errors**
- 2) control engineering approach
  - **to evaluate a quality of a FC the criteria commonly used in control engineering practice are applied**
  - **the feedback structure of the FC is commonly applied with the error signal chosen as one of the inputs**
  - **fuzzy PID-like (as well as PD-like, PI-like) controllers are extremely popular**
  - **the membership functions and scaling factors are selected on the base of their influence on the FC control surface, and rules are formulated considering the control trajectory**
  - **proposes to design a FC by investigating how the FC stability and performance indicators depend upon different FC parameters**
- 3) intermediate approaches,
  - **suppose setting some of the parameters (e.g. membership functions) by the experts and fixing the others (e.g. rules) with the methods inherited from the control system design.**
- 4) combined approaches and synthetic approaches
  - **include the initial choice of the FC structure and parameters made by the expert and further their adjustment performed with the control engineering methods**

L. Reznik

AI vs. CE approaches

- AI approach
  - allows to capture in a FC design the vagueness of a human knowledge and express the design framework with natural languages.
  - leads to that feature of FC which becomes more and more important, especially in design applications: the design process of a FC becomes more understandable, looks less sophisticated and superficial to a human designer and becomes more attractive and threfore cheaper than a conventional one
- Control engineering
  - allows to apply in a FC design traditional criteria and develop design methodologies to satisfy conventional design specifications including such parameters as e.g. overshoot, integral and/or steady-state eerror.
  - enhancing FC engineering methods with an ability to learn and a development of an adaptive FC design would significantly improve the quality of a FC, making it much more robust and expanding an area of possible apllications

L. Reznik

Fuzzy Controller Structure and Parameter Choice

Choice ofthe structure	Apply the hierarchical structure whenever there is any doubt in the stability of a fuzzy control system or in applications requiring high reliability
Choice of the inputs	The same as for a conventional control systemThe error and change_of_error (derivative) signals are often applied as the inputs for a fuzzy controller (fuzzy PID-like controller)Additional: choose the inputs regarding to which some control rules, expressing the dependence of the output on these inputs, can be easily formulated
Choice of the scaling factors	Initially choose the scaling factors to satisfy to the operational ranges (the universe of discourse) for the inputs and outputs, if they are known.Change the scaling factors to satisfy to the performance parameters given in the specifications on the base of recommendations provided
Choice of the number of the classes (membership functions)	There are several issues to consider when determining the number of membership functions and their overlap characteristics. The number of membership functions is quite often odd - generally, anywhere from 3 to 9.As a rule of thumb, the greater control required (i.e. the more sensitive the output should be to the input changes) the greater the membership function density in that input region.
Choice of the membership functions	1) the expert approach - choose the membership functions determined by the expert(s) 2) the control engineering approach - 1) initially choose the width of the membership functions to provide the whole overlap about 12-14%, 2) in order to improve the steady-state error and the response time decrease the membership functions whole overlap, 3) in order to improve dynamic characteristics (oscillation, settling time, overshoot) increase the whole overlap, 4) the use of a fuzzy controller with wider membership functions and a large overlap can be recommended in the presence of large disturbances.

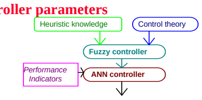
L. Reznik

Fuzzy Controller Structure and Parameter Choice

Choice of the rules	Main methods: 1) expert experience and knowledge, 2) operator's control actions learning, 3) fuzzy model of the process or object under control usage, 4) learning technique application.The whole rules set should be: - complete, - consistent, - continuous.
Choice of the defuzzification method	Choose the method according to the criteria The most widely used are: The Centre_of_Area and Middle_of_Maxima
Choice of the fuzzy reasoning method	Choose Mamdani method if:- the rules are expected to be formulated by a human expert Choose Sugeno method if:- computational efficiency and convenience in analysis are very important.
Choice of the t-norm and s-norm calculation method	The most widely used are: for t-norm Min or product operators, for s-norm Max or algebraic sum

L. Reznik

Tuning (adjustment) of fuzzy controller parameters



1. Conventional methods:
   - 1.1. least-square method variations,
   - 1.2. gradient descent method variations
2. Intelligent methods with applications of fuzzy logic, neural networks, and genetic algorithms
   - 2.1. tuning with fuzzy meta-rules,
   - 2.2 adjustment with neural networks,
   - 2.3 optimisation with genetic/evolutionary algorithms.

- In an intelligent design fuzzy logic is utilised to incorporate the available knowledge into the controller design, and ANN and/or GA technology are applied to adaptively develop an optimal control strategy. The control system structure in this case can be presented as in Fig. 8 [11]. One should note that there exist another trend in combining FL and ANN technologies and creating new synergisms such as adaptive network based fuzzy inference systems (ANFIS) [7]. In this approach the controller design originates from the ANN framework.

Combined structure for a FC

Where to next ?

Synergy in SC: Reasons & Approaches

- Hybrid Soft Computing
  - **Leverages** *tolerance for imprecision*, **uncertainty, and incompleteness - intrinsic to the problems to be solved**
  - **Generates** ***tractable, low-cost,*** *robust* **solutions to such problems** *by*
- Tight Hybridization
  - Data-driven Tuning of Knowledge-derived Models
    - » Translate domain knowledge into initial structure and parameters
    - » Use Global or local data search to tune parameters
  - Knowledge-driven Search Control
    - » Use Global or local data search to derive models (Structure + Parameters)
    - » Translate domain knowledge into an algorithm's controller to improve/manage solution convergence and quality

Synergy in SC: Reasons & Approaches

- Loose Hybridization (Model Fusion)
  - **Does not combine** *features* **of methodologies - only their results**
  - **Their** *outputs* **are compared, contrasted, and aggregated, to increase reliability**
- Hybrid Search Methods
  - Intertwining ***local*** search within ***global*** search
  - Embedding knowledge in operators for global search
- Future:
  - Circle of SC's related technologies will probably widen beyond its current constituents.
  - Push for low-cost solutions and intelligent tools will result in deployment of hybrid SC systems that ***integrate reasoning and search techniques***.