Chapter3
ECOS for Supervised Learning

Prof. Nik Kasabov

nkasabov@aut.ac.nz

http://www.kedri.info

Overview

- Principles and Architectures
- Evolving Fuzzy Neural Networks (EFuNN)
- Knowledge manipulation in EFuNNs
- On-line evaluation, feature modification and parameter adaptation in EFuNNs

Principles & architectures of systems for on-line supervised learning

- Can use global or local goal function to optimise the structure of the learning system
- On-line supervised learning systems learn from a stream of pairs of data (x, y) where the desired output vector y is either known when the input vector x arrives, or will become known at a later stage.
- Output vector will be used to incrementally adapt the system’s structure and function
- Models – MLP, RBF, RAN, RFWR, ZISC, EFuNN

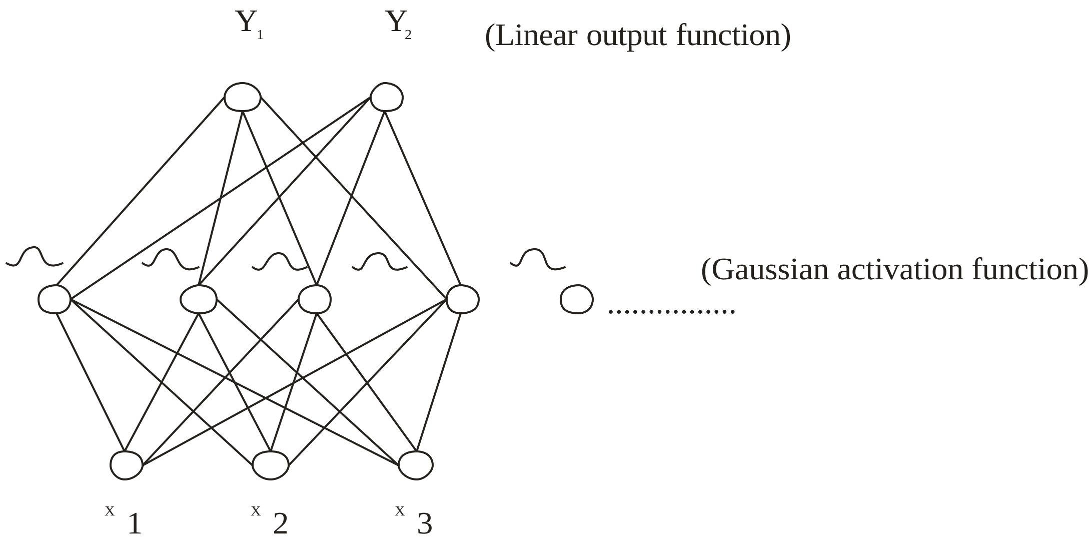
MLP for on-line training

- MLPs trained with a BP algorithm use a global optimisation function in both on-line (pattern) mode and batch mode training (typical mode)
- On-line pattern learning mode:
  - » A training example is presented to system and propagated
  - » Error calculated
  - » Connections modified in a backward manner
  - » Catastrophic forgetting phenomenon

Radial Based Functions (RBFs)

- The basis of many connectionist models for on-line and knowledge-based learning
- RBF Network Layers
  - » Input – clustering of training data
  - » Radial basis activation functions of the hidden neurons
  - » Output – nodes perform a summation function with a linear thresholding activation function;
  - » Uses a gradient descent (eg, Back Propagation) function during training to adjust the 2nd layer of connections – supervised learning phase

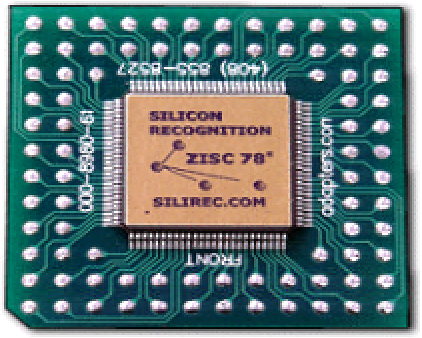
RBF Structure



The General Structure of an RBF network.(Fig. 3.2)

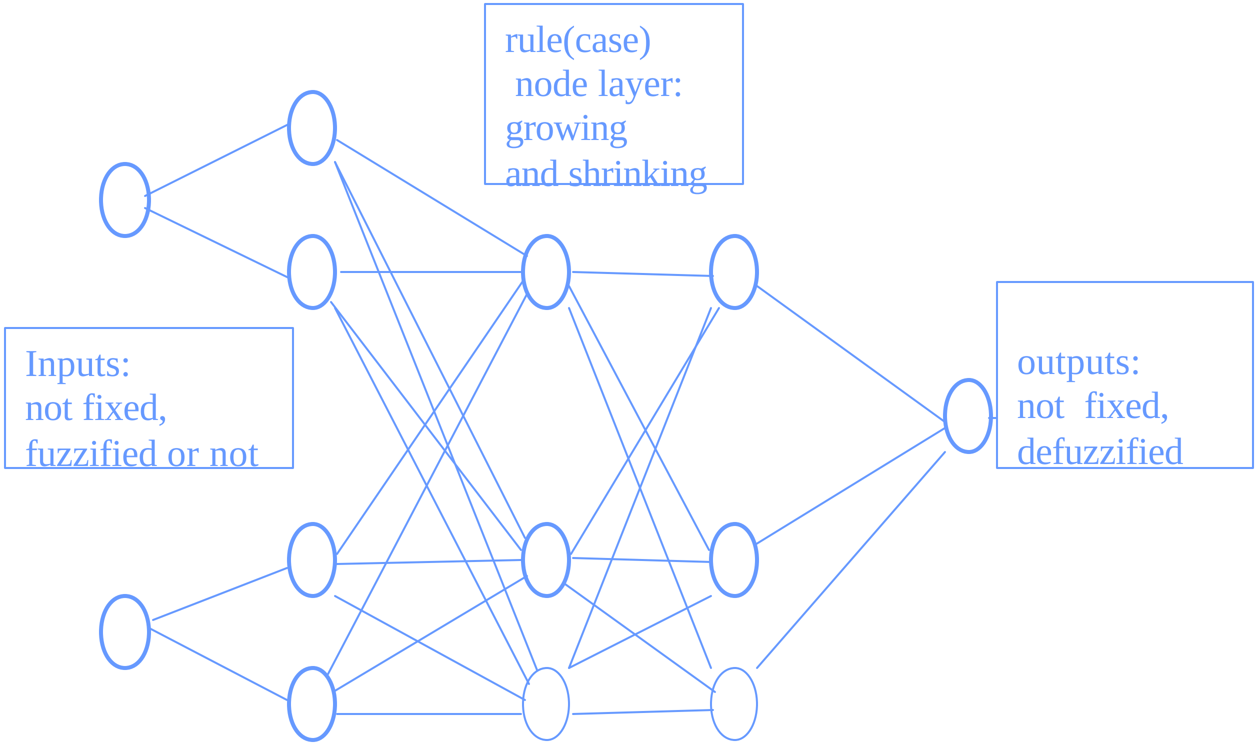
Zero Instruction Set Computer (ZISC)

- Supervised learning system in a chip that realises a growing RBF network.
- Each hidden node has a receptive field (filed of interest) that has a maximum value initially.
- A node is linked to yes/no output class type (depending on the example)



EFuNN

- Fuzzy neural networks – structures that can be interpreted in fuzzy rules
- EFuNN nodes are created an connected as data examples are presented



EFuNN Layer Architecture

- Inputs – input variables
- Fuzzy Input – fuzzy quantisation of each input variable space
- Rule node – nodes that evolve during supervised and/or unsupervised learning. Linear activation or Gaussian function is used.
- Fuzzy output – fuzzy quantisation of output variable space. Weighted sum function is used to calculate the membership degrees to which the input vectors belong to the Membership Functions of the output variables.
- Output – linear activation function used to calculate the defuzzified values for the output variables

EFuNN

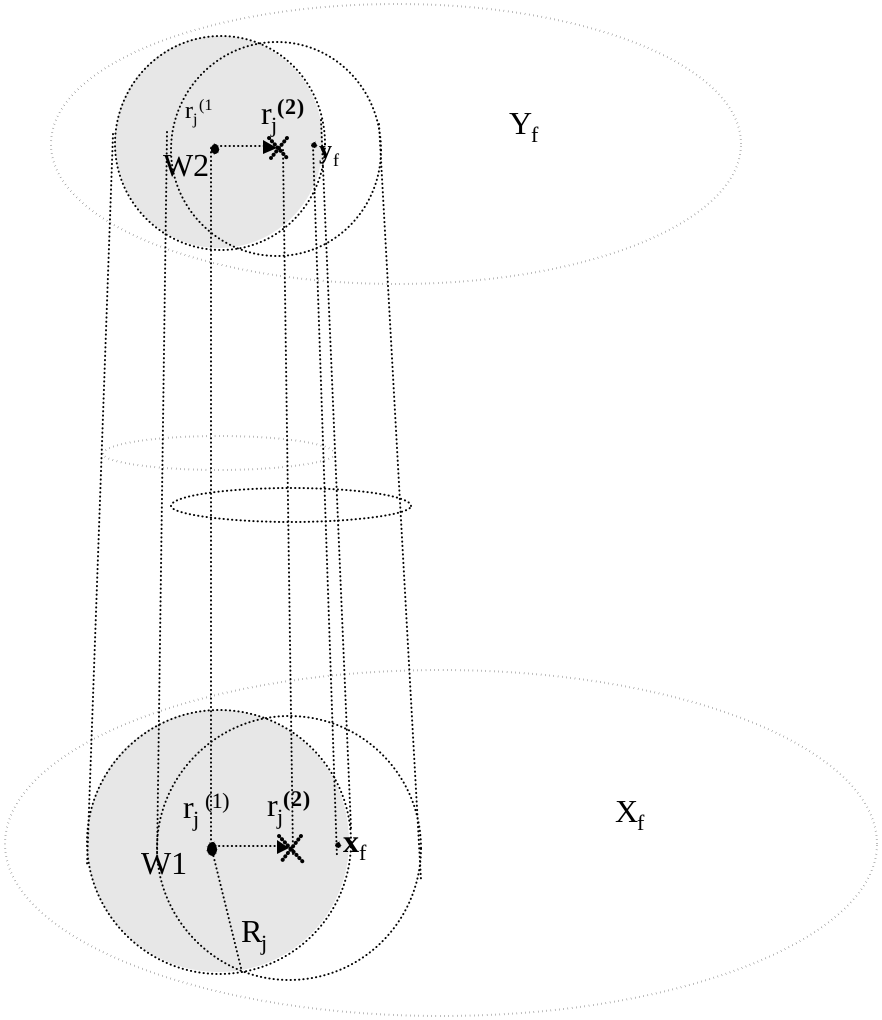
- EFuNN learning is based on either one of these assumptions
  - » No rule nodes exist prior to learning and all of them are created during the evolving process

  OR

  - » There is an initial set of rule nodes that are not connected to the input and output nodes and become connected through the learning (evolving) process

Adaptive Learning in EFuNN

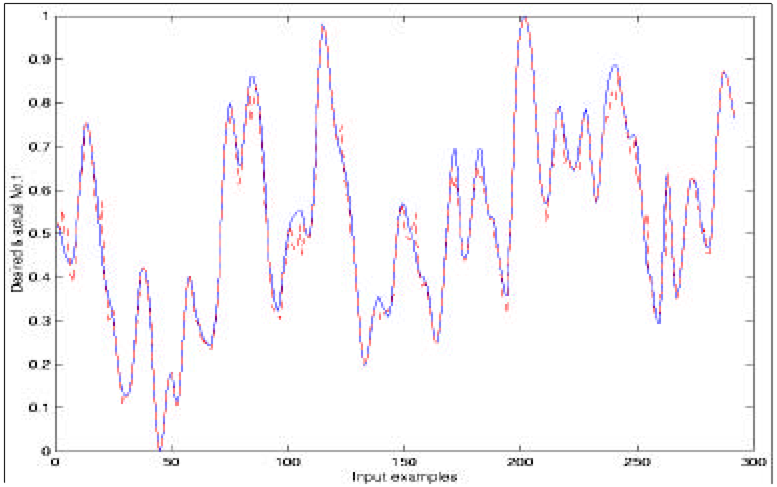
A rule node represents an association of two hyperspheres from the fuzzy input space and the fuzzy output space (fig. 3.11)



Knowledge Manipulation in EFuNN

- Important for an ECOS not only to learn in lifelong learning mode, but also to “explain” at any time the essence/knowledge that the system has acquired

- Rule Insertion and Extraction
  - » Fuzzy or exact rules can be inserted and extracted at any phase of the learning process



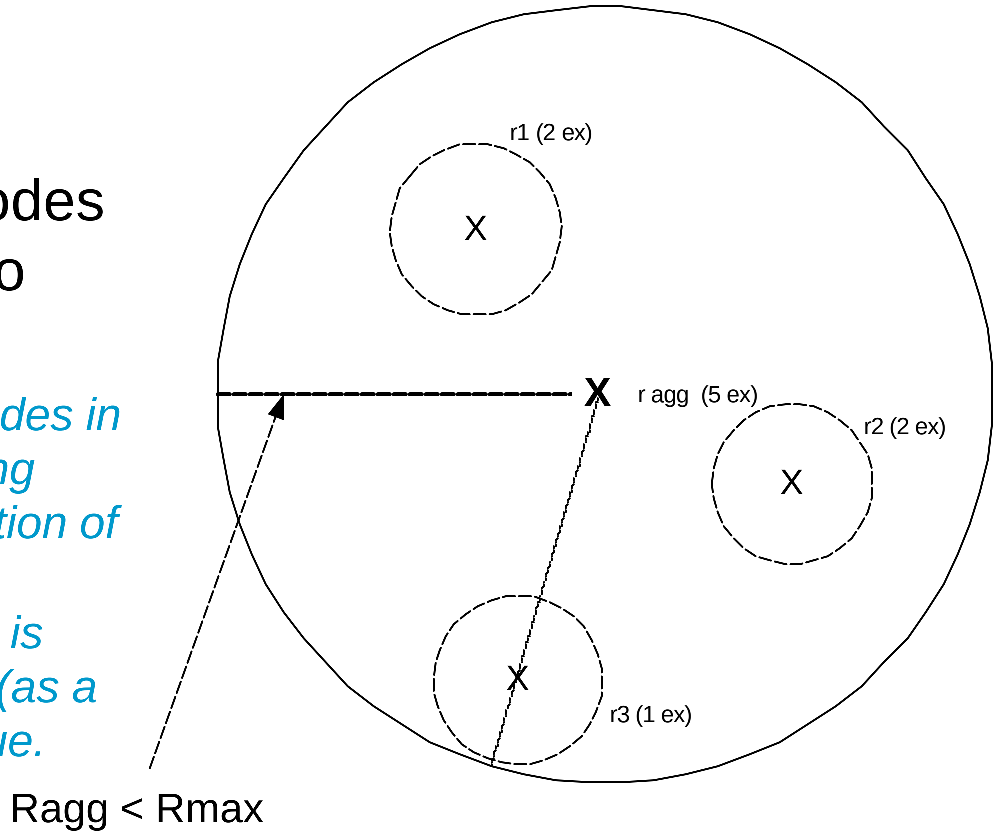
Rule 1 :IF input [1] is (Small 0.46) and (Medium 0.540) and input [2] is (Large 0.809) THEN output is (Large 0.685); receptive field 0.106; accommodated examples = 2

Rule 2: IF input [1] is (Medium 0.527) and (Large 0.473) and input [2] is (Small 0.461) and (Medium 0.539) THEN output is (Small 0.496) and (Medium 0.504); receptive field = 0.124; examples = 5

Rule Node Aggregation in EFuNN

- Rule Aggregation
  - » Several rule nodes are merged into one

Aggregation of Rule nodes in an EFuNN - the resulting node from the aggregation of the three rules has a receptive radius, which is less than a predefined (as a system parameter) value. (fig.3.16c)



On-line Evaluation, Feature Modification and Parameter Adaptation

- EFuNNs are considered Universal Classifiers and Universal Function Approximators

- Once set, the EFuNN parameter values can either be kept fixed or can be adapted (optimised) during the system's operation.

- GAs and Evolutionary Programming techniques can be applied to optimise the EFuNN's structural and functional parameters

Summary

- EFuNNs incorporate important AI features
  - » adaptive learning
  - » non-monotonic reasoning
  - » knowledge manipulation
  - » knowledge acquisition and explanation
- EFuNNs can solve difficult engineering tasks through self organisation and adaptation during the learning process
- EFuNNs can be applied to many problems from the information science, life sciences and engineering domains.

Further Reading

- ART architectures and the stability-plasticity dilemma (Grossberg, 1976 and 1998).
- ARTMAP (Carpenter, Grossberg and Reynolds, 1991).
- FuzzyARTMAP (Carpenter et al, 1992).
- On-line Q-learning (Rummery and Niranjan, 1994).
- On-learning in ZISC (Zero Instruction Set Computer) (ZISC Manual, 2001)
- Life-long learning cell structures (Hamker,2001; Bruske, 1998; Bruske et al, 1996; Hamker and Gross, 1997).
- Hybrid neuro-fuzzy systems for adaptive and continuous learning (Berenji, 1992; Lim and Harrison, 1997).
- On-line learning in RBF networks (Karayinnis and Mi, 1997; Berthold and Diamond, 1995; Obradovich, 1996; Platt, 1991; Fritzke, 1992 and1994; Freeman and Saad, 1997).
- Quantizable RBF networks (Poggio and Girosi, 1990).
- Prediction of chaotic time-series with a resource-allocating RBF network (Rosipal, Koska and Farkas, 1997).