Computational Intelligence
Lecture 19:Neuro-Fuzzy Systems

Farzaneh Abdollahi

Department of Electrical Engineering

Amirkabir University of Technology

Fall 2011

Adaptive Neuro-Fuzzy Systems (ANFIS)

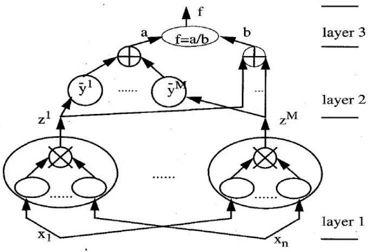
- ANFIS is using neural networks for defining fuzzy membership fcns.
- In this approach
  - The structure of the fuzzy system is specified.
  - Then some parameters in the structure are determined according to the input-output pairs.
  - Gradient descent technique is employed to find the parameters
- **Structure of the fuzzy system:**
  - Inference engine: Product; Fuzzifier: singleton
  - Defuzzifier: center average; Membership function: Gaussian

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l (\prod_{i=1}^{n} a_i^l exp[-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2])}{\sum_{l=1}^{M} (\prod_{i=1}^{n} a_i^l exp[-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2])} \tag{1}$$

  - M is fixed, $a_i^l = 1$
  - $\bar{y}^l, \bar{x}_i^l, \sigma_i^l$ are the params for choose

Determining The Parameters

- Represent the fuzzy system (1) as a feedforward network.
- It will be a three layer network.
- To map $x \in U$ to the output $f(x) \in V \subset R$:
  1. Pass x through a product Gaussian operator: $z^l = \prod_{i=1}^{n} exp[-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2]$
  2. Pass z^l through a summation operator: $b = \sum_{l=1}^{M} z^l$
  3. Pass z^l through a weighted summation operator: $a = \sum_{l=1}^{M} \bar{y}^l z^l$
  4. The output of the fuzzy system is: $f(x) = a/b$



Designing the Parameters by Gradient Descent

- The params are defined s.t minimize the error: $e^p = \frac{1}{2}[f(x_0) - y_0^p]^2$
- $\bar{y}^l(q+1) = \bar{y}^l(q) - \alpha\frac{\partial e}{\partial \bar{y}^l} = \bar{y}^l(q) - \alpha\frac{f-y}{b}z^l, \quad l = 1, .., M; \; q = 0, 1, ...$
- $\bar{x}_i^l(q+1) = \bar{x}_i^l(q) - \alpha\frac{\partial e}{\partial \bar{x}_i^l} = \bar{x}_i^l(q) - \alpha\frac{f-y}{b}(\bar{y}^l(q) - f)z^l\frac{2(x_{i0}^p - \bar{x}_i(q))}{\sigma_i^{l2}(q)}$,
  $l = 1, .., M; \; q = 0, 1, ...; , i = 1, ..., n$
- $\sigma_i^l(q+1) = \sigma_i^l(q) - \alpha\frac{\partial e}{\partial \sigma_i^l} = \sigma_i^l(q) - \alpha\frac{f-y}{b}(\bar{y}^l(q) - f)z^l\frac{2(x_{i0}^p - \bar{x}_i(q))^2}{\sigma_i^{l3}(q)}$,
  $l = 1, .., M; \; q = 0, 1, ...; , i = 1, ..., n$

Design of Fuzzy Systems Using Gradient Descent Training

1. Structure and initial parameters
    - Choose the fuzzy system (1)
    - Determine M.
        - The larger M results more parameters computation, but better approximation accuracy.
    - Specify the initial parameters $\bar{y}^l(0), \bar{x}_i^l(0), \sigma_i^l(0)$.
        - They can be determined according to the linguistic rules from experts,
        - or based on the idea that membership functions uniformly cover the input and output spaces.
2. Present input and calculate the output
    - For a given input-output pair (x_0^p, y_0^p), $p = 1, 2, ...,$ and at the q'th stage of training, $q = 0, 1, 2, ...,$ present x_0^p to the input layer of the fuzzy system
    - Compute the outputs $z^l = \prod_{i=1}^n exp[-(\frac{x_i - \bar{x}_i^l}{\sigma_i^l})^2]$

    $b = \sum_{l=1}^M z^l, \;\; a = \sum_{l=1}^M \bar{y}^l z^l \rightsquigarrow f(x) = a/b$

3. Update the Parameters
    - Apply the updating rules given in slide 5.

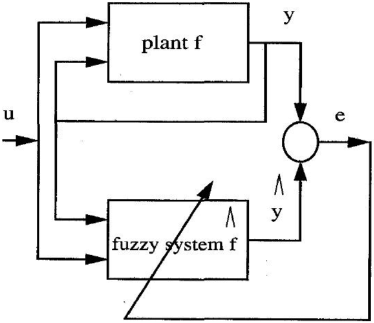
4. Repeat by going to Step 2 with $q = q + 1$, until the error $|f - y_0^p|$ is less than a prespecified number E_{max}, or until the q equals a prespecified q_{max}.

5. $p = p + 1$ and use the next input-output pair (x_0^{p+l}, y_0^{p+l}), go to step 2.

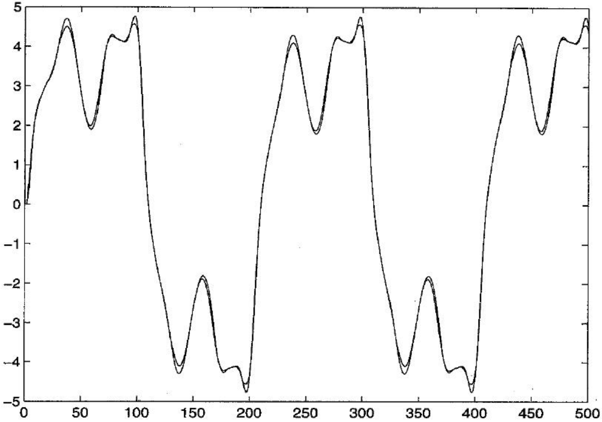
6. If the performance is not desirable and feasible, set $p = 1$ and repeat steps 2-5 again
    - For on-line control and dynamic system identification, this step is not feasible because the input-output pairs are provided one-by-one in a real-time fashion.
    - For pattern recognition problems where the input-output pairs are provided off-line, this step is usually desirable.

Application to Nonlinear Dynamic System Identification

- Considering the fuzzy systems as powerful universal approximators, it is reasonable to use them as identifier.
- Consider the nonlinear dynamics $y(k+1) = f(y(k), ..., y(k-n+1), u(k), ..., u(k-m+1))$
  - f is unknown
  - u is input; y is output
- The identified model will be $\hat{y}(k+1) = \hat{f}(y(k), ..., y(k-n+1), u(k), ..., u(k-m+1))$
- Objective: adjust the parameters in $\hat{f}(x)$ s.t. the output of the identification model $\hat{y}(k+1)$ converges to the output of the true system $y(k+1)$ as $k \to \infty$



- The I/O pairs: $(x_0^{k+l}; y_0^{k+l})$,
  - $x_0^{k+1} = (y(k), ..., y(k-n+1); u(k), ..., u(k-m+1))$
  - $y_0^{k+1} = y(k+1)$, and $k = 0, 1, 2, ...$
- Note that in the previous formula p is k and the n is $n+m$.
- **Example:** Consider the following system to be identified: $y(k+1) = 0.3y(k) + 0.6y(k-1) + g(u(k))$
  - $g(u(k))$ is unknown, for simulation define is as $g(u(k)) = 0.6sin(\pi u) + 0.3sin(3\pi u) + 0.1sin(5\pi u)$
  - The identification model will be $\hat{y}(k+1) = 0.3y(k) + 0.6y(k-1) + \hat{g}(u(k))$
  - Choose $M = 10, \alpha = 0.5, k_{max} = 200$
  - $u = sin(2\pi k/200)$
  - Online train the system


5
4
3
2
1
0
−1
−2
−3
−4
−5
0
50
100
150
200
250
300
350
400
450
500

J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*.
Plenum Press, 1981.