

Первоисточник: [\[http://fuzzy.kstu.ru/fulltext.htm\]](http://fuzzy.kstu.ru/fulltext.htm)

## **Использование Neuro-Fuzzy технологии для генерации и оптимизации эвристических правил - программный продукт Лингвогенератор**

Гиляров В.Н, Крюков А. Ю, Бондарчук П. В.

Санкт-Петербургский Технологический Институт  
(Технический Университет)  
кафедра Систем Автоматизированного Проектирования и Управления  
Московский проспект 26  
198013 Санкт-Петербург, Россия

### **Введение**

Нечеткое лингвистическое описание предполагает наличие базы данных (БД), которая соединяет между собой лингвистические переменные (ЛП) с соответствующими функциями принадлежности (ФП). Детальное описание Теории Нечетких Множеств и ее применений можно найти в литературе, например [Zimmermann, 1991], [Zadeh, 1965].

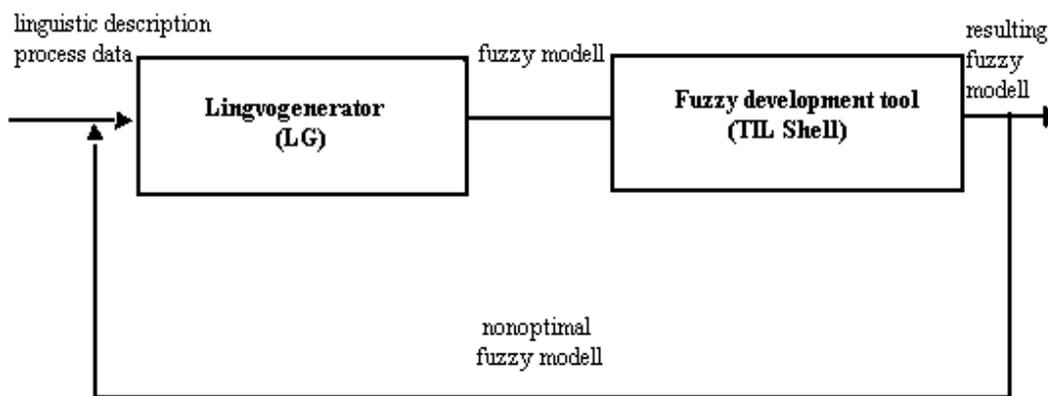
После множества успешных работ по созданию нечетких систем, в которых были достаточно качественно и быстро реализованы знания человека-эксперта (особенно в области нечеткого управления) [Mamdani,1974], [Sugeno,1985], на первый план вышла проблема создания подобных систем в условиях неполной, противоречивой, зашумленной информации. В математическом смысле это задача оптимизации нелинейных структур. В рамках решения этой проблемы мы видим ставший в последние годы популярным подход, который заключается в создании комбинированных Нейро-Фаззи систем.

Нечеткие системы работают со слабо структурированной качественной информацией. Нейронные сети напротив не используют какую-либо информацию об объекте кроме измеренных данных. Максимальная эффективность модели по-видимому, может быть достигнута, если мы попытаемся использовать *всю* доступную информацию об объекте. Эта идея и отражена в стремлении объединить нечеткую логику и нейронные сети. Технология, разработанная на этой основе объединяет соответствующим образом способность НС к самообучению и способность нечетких систем обрабатывать качественную информацию.

Важной особенностью представленных алгоритмов (ЛГ), которая отличает его от многих известных Neuro-Fuzzy систем, является его гибкость в отношении исходных данных для моделирования, другими словами, способность использовать различные источники информации.

Синтез нечеткой лингвистической модели (особенно при моделировании сложных систем) представляет собой циклическую процедуру. После применения ЛГ может быть использована какая-либо другая Fuzzy-оболочка (Рис.1).

В качестве выхода ЛГ мы получаем предварительную лингвистическую модель объекта. Затем результаты обучения могут быть визуализированы с помощью выше названной оболочки. Мы можем также внести вручную некоторые исправления и далее сгенерировать С-код. При этом может потребоваться повторное применение ЛГ. На этот раз полученная ранее лингвистическая модель может быть использована в рамках ЛГ для предварительной структуризации. Этот цикл выполняется до тех пор, пока не будет получена удовлетворительная модель объекта.



**Рисунок 1.** Стратегия для синтеза и оптимизации нечетких лингвистических моделей.

### Гибридные Нейро-Фаззи модели

Дадим следующее определение:

**Определение 1.** **Базовая Нейро-Фаззи Структура (БНФС)** это 3-слойная вперед направленная нейронная сеть ( $FUZ$ ,  $INF$ ,  $DEF$ ), которая состоит из 3 подсетей и описывается следующим образом:

- $U = \bigcup_{i \in I} U_i$  - непустое множество подсетей,  $I = \{FUZ, INF, DEF\}$  - множество индексов для  $U$ .  $U_{FUZ}$  - подсеть фаззификации;  $U_{INF}$  - подсеть, реализующая машину вывода;  $U_{DEF}$  - подсеть дефаззификации.  $\forall i, U_i = \bigcup_{j \in N_i} U_j$  - множество элементов (слоев),  $N_i$  - количество слоев в  $i$ -той подсети. Каждый слой состоит из вычислительных элементов (нейронов).  $J = \sum_{i \in I} N_i$  - общее количество слоев в  $U$ .  $\forall i, j \in J; U_i \cap U_j = \emptyset$ , при  $i \neq j$ .
- Структура сети (связи) определена:  $M: U \times U \rightarrow \Phi(\mathfrak{R})$  такое, что существуют только связи  $M(u, v)$ , с  $u \in U_i; v \in U_{i+1}, i \in J$  ( $\Phi(\mathfrak{R})$  - множество всех нечетких подмножеств на  $\mathfrak{R}$ ).
- $U_{FUZ}$  - 2-слойная вперед направленная НС со следующими характеристиками:  $U_{FUZ} = \bigcup_{i=1,2} U_i$  - непустое множество элементов (слоев).  $U_1 \neq \emptyset, U_2 \neq \emptyset$ .  $U_1$  - входной слой,  $U_2$  - слой функций принадлежности. Для  $U_1: F_C, F_I, F_N: \mathfrak{R} \rightarrow \mathfrak{R}$ . Для  $U_2: F_C, F_I: \mathfrak{R} \rightarrow \mathfrak{R}, F_N: (\mathfrak{R} \times \Phi(\mathfrak{R})) \rightarrow [0,1]$ .
- $U_{INF}$  - 2-слойная вперед направленная НС со следующими характеристиками:  $U_{INF} = \bigcup_{i=3,4} U_i$  - непустое множество элементов (слоев).  $U_3$  - агрегирующий слой,  $U_4$  - инференционный слой. For  $U_3: F_C: [0,1] \rightarrow [0,1], F_I: ([0,1] \times [0,1]) \rightarrow [0,1]$  - подходящая  $t$ -норм функция;  $F_N: [0,1] \rightarrow [0,1]$ . Для  $U_4: F_C, F_N: [0,1] \rightarrow [0,1], F_I: ([0,1] \times [0,1]) \rightarrow [0,1]$  - подходящая  $t$ -конорм функция.

е)  $U_{DEF}$  - 1-слойная вперед направленная НС со следующими характеристиками:  
 $U_{DEF} = U_5 \neq \emptyset$ .  $U_5$ - выходной слой.  $F_C:([0,1] \times \Phi(\mathfrak{X})) \rightarrow \Phi(\mathfrak{X})$ ,  $F_I:\Phi(\mathfrak{X}) \rightarrow \mathfrak{X}$  -  
 подходящая функция дефаззификации,  $F_N:\mathfrak{X} \rightarrow \mathfrak{X}$ .

Соответствующая определению 1 структура показана на рисунке 2.

### НС со специальной структурой

На базе БНФС мы разработали НС со специальной структурой (НССС). Подсеть  $U_{DEF}$  в БНФС была модифицирована и содержит 2 слоя. Это было сделано для реализации MAXDOT- и MAXMIN-центроид методов дефаззификации и чтобы дать возможность настройки соответствующих параметров.

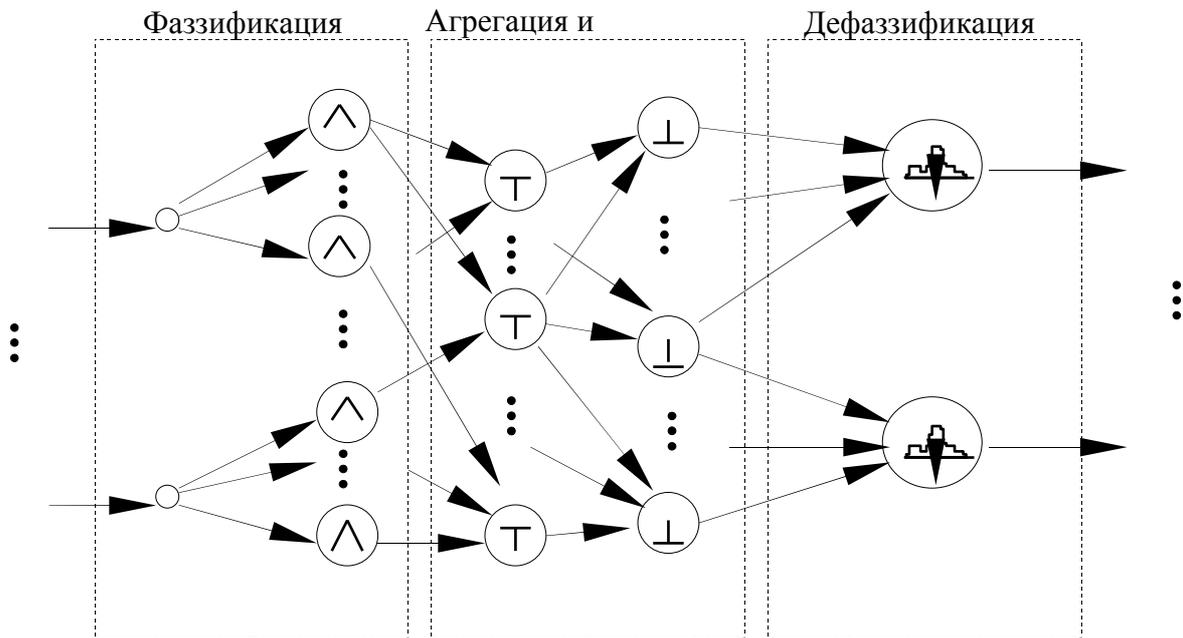


Рисунок 2. Базовая Нейро-Фаззи структура

**Определение 2.** Нейронная сеть со специальной структурой (НССС) это базовая Нейро-Фаззи структура со следующими характеристиками:

а) для слоя  $U_1$  (Input - входной слой):  $S^i_{Input} = x$

$$b) \text{ для слоя } U_2 (H_1 - \text{первый скрытый слой}): S_{H_1}^i = \begin{cases} 0, & x < a_1 \\ sfunc(a_1, b_1, x), & a_1 \leq x \leq b_1 \\ 1, & b_1 < x < b_2 \\ sfunc(b_2, a_2, x), & b_2 \leq x \leq a_2 \\ 0, & x > a_2 \end{cases}$$

$$\text{, где } sfunc(a, b, x) = \begin{cases} \frac{(x-a)^2}{2(c-a)^2}, & x \leq c \\ 1 - \frac{(b-x)^2}{2(c-a)^2}, & x > c \end{cases}, \text{ при } c = \frac{a+b}{2}$$

$$c) \text{ для слоя } U_3 (H_2 - \text{второй скрытый слой}): S_{H_2}^i = \prod_j^L \rho_j^{(1-\gamma)} \left( 1 - \prod_j^L (1 - \rho_j) \right)^\gamma$$

$$\text{, при } 0 \leq \gamma \leq 1, \rho_j = x_j + (1 + x_j)(1 - m_{ji}^{1-2}), m_{ji}^{1-2} = 1 \vee 0, \sum_j m_{ji}^{1-2} \leq 1$$

d) для слоя  $U_4 (H_3 - \text{третий скрытый слой}):$

$$S_{H_3}^i = \prod_j^L \rho_j^{(1-\gamma)} \left( 1 - \prod_j^L (1 - \rho_j) \right)^\gamma, \text{ при } \gamma \approx 1, \rho_j = m_{ji}^{2-3} x_j, m_{ji}^{2-3} - \text{веса}$$

правил;  $0 \leq m_{ji}^{2-3} \leq 1$

e)  $U_{DEF} = \bigcup_{i=5,6} U_i$  - непустое множество элементов (слоев).  $U_5 \neq 0, U_6 \neq 0$

f) для слоя  $U_5 (H_4 - \text{четвертый скрытый слой}): S_{H_4}^i = \sum_j^L m_{ji}^{3-4} x_j$ , при

$i = \overline{1,2}$ ,  $m_{j1}^{3-4}$  - моменты ФП,  $m_{j2}^{3-4}$  - площади ФП

g) для слоя  $U_6 (Output - \text{выходной слой}): S_{Output} = \frac{x_1}{x_2}$ , (MaxDot Centroid

метод дефаззификации)

Все функции, представленные выше дифференцируемы, что позволяет применять градиентные методы оптимизации. Для обучения НССС мы применяем варианты хорошо известных методов обучения: *back propagation* и *генетический алгоритм*. Результаты подробно описаны в [Giljarow, 1995].

### RBF-Fuzzy

Выбрав произведение в качестве t-norm оператора и суммирование в качестве t-conorm оператора в рамках БНФС, а кроме того используя в качестве  $f_N$  функцию Гаусса и в качестве  $S_{DEF}$  MAXDOT-Centroid метод дефаззификации, приходим к структуре, которую мы назовем *Radial Basic Function - Fuzzy сеть* (RBF-Fuzzy). RBF сети более подробно описаны в [Hafner (Hrsg), 1994].

**Определение 3.** RBF-Fuzzy сеть это БНФС со следующими характеристиками:

a) для слоя  $U_1: S_{H_1}^i = x$

$$b) \text{ для слоя } U_2: S^i_{H_2} = \begin{cases} e^{-\frac{(x_j - \mu_{jl})^2}{2\delta_{jl}^2}} & x_j < \mu_{j1}, l = 1 \\ e^{-\frac{(x_j - \mu_{jl})^2}{2\delta_{jl}^2}} & x_j > \mu_{j2}, l = 2 \\ 1, \mu_{j1} \leq x_j \leq \mu_{j2} \end{cases}$$

$$c) \text{ для слоя } U_3: S^i_{H_3} = \prod_{j=1}^L x_j$$

$$d) U_4 = \emptyset$$

$$e) \text{ для слоя } U_5: f_C^i = m_{ji} x_j, f_l = \frac{\sum_{j=1}^L d_j x_j}{\sum_{j=1}^L x_j}$$

Находясь в рамках структуры, описанной в определении 3, мы можем вычислить выход системы следующим образом:

$$y_i = \frac{\sum_{k=1}^n m_{ik} d_{ik} f_k}{\sum_{k=1}^n m_{ik} f_k}, \quad i = 1, \dots, q, \quad \text{при } f_k = \prod_{j=1}^p e^{-\frac{(x_j - \mu_{jkl})^2}{2\delta_{jkl}^2}} = e^{-\sum_{j=1}^p \frac{(x_j - \mu_{jkl})^2}{2\delta_{jkl}^2}}$$

Благодаря замкнутому описанию RBF-Fuzzy сеть может быть оптимизирована без использования back propagation алгоритма. Соответствующие формулы процедуры модификации системных параметров показаны в таблице.

Применение RBF-Fuzzy моделей дало хорошие результаты в классе проблем с отсутствием информации о моделируемом объекте. На основе только экспериментальных данных удается получить нечеткое описание с ошибкой моделирования менее 5%.

Промежуточные величины	Коррекция весов
$\delta_i = y_i - \tilde{y}_i$	$\Delta d_{ik} = -\alpha \delta_i m_{ik} g_{ik}$ $\Delta m_{ik} = -\alpha \psi_{ik}$
$i = 1, \dots, q$	$k = 1, \dots, n$ $i = 1, \dots, q$
$g_{ik} = \frac{f_k}{\sum_{k=1}^n m_{ik} f_k}$ $\psi_{ik} = \delta_i g_{ik} (d_{ik} - y_i)$	$\Delta \mu_{jkl} = -\alpha \frac{(x_j - \mu_{jkl})}{\delta_{jkl}^2} \sum_{i=1}^q m_{ik} \psi_{ik}$ $\Delta \delta_{jkl} = -\alpha \frac{(x_j - \mu_{jkl})^2}{\delta_{jkl}^3} \sum_{i=1}^q m_{ik} \psi_{ik}$
$i = 1, \dots, q$ $k = 1, \dots, n$	$l = \begin{cases} 1, x_j \leq \mu_{jk1} \\ 2, x_j \geq \mu_{jk2} \end{cases}$ $k = 1, \dots, n$ $j = 1, \dots, p$

**Таблица.** Формулы оптимизации для RBF-Fuzzy сети ( $p$  входов,  $q$  выходов и  $n$  нейронов в слое  $U_2$ )

## **Программный продукт Лингвогенератор.**

В рамках нашей работы был разработан программный продукт Лингвогенератор (ЛГ), который представляет собой программную реализацию алгоритмов описанных выше. ЛГ ориентирован на поддержку процесса сетевого обучения при генерации или оптимизации нечетких систем на основе экспериментальных данных.

### **Рисунок 3. Интерфейс ЛГ**

Пользовательский интерфейс - графическая поверхность, в которой интегрированы следующие функции (рис. 3):

- первичная обработка данных (сглаживание, фильтрация, нормализация, генерация ФП как S-функций, прямоугольников и синглетонов);
- определение структуры и стартовых параметров нечеткой системы и/или соответствующей нейронной сети с возможностью фиксировать параметры модели;
- обучение НС с использованием ряда алгоритмов (back propagation, генетический и др);
- запуск сети в вычислительном режиме

Для ввода грубого лингвистического описания также, как и для вывода результирующей нечеткой модели могут быть использованы .FPL-файлы. Они содержат описание нечеткой системы на языке FPL и могут обрабатываться как файлы проекта в широко известной системе TIL Shell (рис. 1).

## **Применения**

ЛГ был тестирован на некоторых примерах из области химической технологии: управление электропечью в производстве электрокорунда и проектирование теплообменников.

В первом примере управляющие правила, полученные от персонала в процессе плавки, были объединены в фаззи модель и ЛГ использовался для оптимизации ФП четырех входных переменных контроллера.

В примере проектирования теплообменной аппаратуры ЛГ использовался для генерации неизвестных эвристических правил. Для случая двух горячих и одного холодного потоков на математической модели были вычислены оптимальные решения с разделением и без деления холодного потока для различных начальных и конечных температур трех потоков. На основе этих данных были найдены эвристические правила.

Подробное описание применения ЛГ можно найти в [Giljarow, 1995].

## **Литература**

1. Giljarow, W. Systematische Untersuchungen zur Gewinnung und Präzisierung von Zugehörigkeitsfunktionen heuristischer Regeln zur Prozeßsynthese verfahrenstechnischer Systeme. Unveröffentlicher Bericht über Forschungsergebnisse. /TU Berlin. 1995
2. Hafner, S. [Hrsg.] Neuronale Netze in der Automatisierungstechnik. /R. Oldenbourg Verlag München Wien. 1994

3. Henning, H. Tryba, V. Mühlenfeld, E. Automatische Generierung von Fuzzy-Systemen mit genetischen Algorithmen. /Fuzzy Logic. Theorie und Praxis, B.Reusch [Hrsg.], Springer-Verlag 1994, ss.167-174
4. Ishibuchi, H. Kwon, K. Tanaka, H. Learning of fuzzy neural network from fuzzy inputs and fuzzy targets. /Fifth IFSA World Congress (1993) 147-150
5. Kosko, B. Neural Networks and Fuzzy Systems: a dynamical systems approach to machine intelligence. /Prentice Hall, Englewood Cliffs. 1992
6. Krjukov, A. Unveröffentlicher Bericht über Forschungsergebnisse im Rahmen des DAAD-Stipendiums. /TU Berlin. 1996
7. Kruse, R. Nauck, D. Learning methods for fuzzy systems. /3<sup>rd</sup> German GI-Workshop „Neuro-Fuzzy-Systeme“. Darmstadt, 15.-17. November 1995
8. Mamdani, E.H. Application of fuzzy algorithms for the control of a simple dynamic plant. /Proc. IEEE 121, 1585-1488. 1974
9. Nauck, D. Klawonn, F. Kruse, R. Neuronale Netze und Fuzzy-Systeme /Vieweg, Braunschweig /Wiesbaden, 1994
10. Shann, J.J. Fu, H.C. A fuzzy neural network for knowledge learning. /Fifth IFSA World Congress (1993) 151-154
11. Sugeno, M. ed., Industrial applications of fuzzy control. North-Holland. 1985
12. Wagenknecht, M. Otto, O. Hartmann, K. Theoretische und praktische Aspekte der unscharfen Modellierung. Unveröffentlicher Vortrag /TU Berlin. 1995
13. Zadeh, L.A. Fuzzy Sets. Information and Control, (1965) 8:338-353
14. Zimmermann, H.-J. Fuzzy sets theory and its application. / Kluwer Academic Publishers Boston /Dordrecht /London. 1991