

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА

Лекции по курсу «Интеллектуальные системы»

Для студентов направления 090401

Очное обучение 2023-2024 учебный год

Лектор: Солдатова Ольга Петровна, к.т.н., доцент

Нечёткие нейронные сети

Лекция 5

**Нечёткие множества и нечёткий вывод.
Основные понятия.**

Треугольные нормы.

Системы нечёткого вывода.

Фуззификация и дефуззификация.

Нечёткий многослойный персептрон.

Алгоритм обучения C-means.

Алгоритм разностного группирования.

Нечёткая нейронная сеть Ванга-Менделя.

Адаптивный алгоритм обучения.

Нечёткая нейронная сеть TSK.

Гибридный алгоритм обучения.

Алгоритм наискорейшего спуска.

Литература по курсу

1. Куприянов А.В. «Искусственный интеллект и машинное обучение» <https://do.ssau.ru/moodle/course/view.php?id=1459>
2. Столбов В.Ю. Интеллектуальные информационные системы управления предприятием / В.Ю. Столбов, А.В. Вожаков, С.А. Федосеев. – Москва : Издательство Литрес, 2021. – 470 с. – Режим доступа: по подписке. – URL: <https://www.litres.ru/book/artem-viktorovich-vo/intellektualnye-informacionnye-sistemy-upravleniya-pr-66403444/>.
3. Осовский С. Нейронные сети для обработки информации / Пер. с пол. И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.: ил.
4. Горбаченко, В. И. Интеллектуальные системы: нечеткие системы и сети : учебное пособие для вузов / В. И. Горбаченко, Б. С. Ахметов, О. Ю. Кузнецова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2018. — 105 с. — (Университеты России). — ISBN 978-5-534-08359-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/424887> – Режим доступа: <https://urait.ru/bcode/424887>
5. 4. Гафаров Ф.М. Искусственные нейронные сети и приложения: учеб. пособие / Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Изд-во Казан. ун-та, 2018. – 121 с. – Текст : электронный. – Режим доступа: https://repository.kpfu.ru/?p_id=187099
6. Хайкин С. Нейронные сети: Полный курс: Пер. с англ. - 2-е изд. – М.: Вильямс, 2006. – 1104 с.: ил.
7. Борисов В.В., Круглов В.В., Федулов А.С. Нечёткие модели и сети. – М.: Горячая линия–Телеком, 2007. -284 с.: ил.
8. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечёткие системы: Пер. с польск. И.Д.Рудинского, - М.: Горячая линия – Телеком, 2007. – 452 с. ил.
9. Николенко С., Кадуринов А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2018. — 480 с.: ил. — (Серия «Библиотека программиста»).
10. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.: цв. ил.

Основные понятия и определения теории нечётких множеств

Понятие **нечётких множеств** (*fuzzysets*) как обобщение чётких множеств было введено **Лотфи Заде** в **1965** году. Причиной создания теории нечётких множеств стала необходимость описания таких явлений и понятий, которые имеют многозначный и неточный характер.

Перед формулированием определения нечёткого множества необходимо задать область рассуждений или пространство рассуждений, которое является чётким множеством. В случае неоднозначного понятия «много денег» большой будет признаваться одна сумма, если мы ограничимся диапазоном [0, 1000 руб.] и совсем другая в диапазоне [0, 1000000 руб.].

Традиционный способ представления элементов множества состоит в применении характеристической функции $\mu_A(x)$, которая равна 1, если этот элемент принадлежит множеству A , или равна 0 в противном случае. В нечётких системах элемент может частично принадлежать к любому множеству.

Нечётким множеством A в некотором непустом пространстве X , что обозначается $A \subseteq X$, называется множество пар $A = \{(x, \mu_A(x)); x \in X\}$, где $\mu_A : X \rightarrow [0,1]$ - **функция принадлежности нечёткого множества** A [3].

Основные понятия и определения теории нечётких множеств

Эта функция для каждого элемента $x \in X$ приписывает степень его принадлежности к нечёткому множеству A , при этом можно выделить три случая:

1. $\mu_A(x) = 1$ означает полную принадлежность элемента x множеству A , то есть $x \in A$;
2. $\mu_A(x) = 0$ означает отсутствие принадлежности элемента x множеству A , то есть $x \notin A$;
3. $0 < \mu_A(x) < 1$ означает частичную принадлежность элемента x множеству A .

Степень принадлежности элемента к множеству A , представляющая собой обобщение характеристической функции, называется функцией принадлежности $\mu_A(x)$, причём $\mu_A(x) \in [0,1]$. Конкретное значение функции принадлежности называется **степенью или коэффициентом принадлежности**.

Эта степень может быть определена явно функциональной зависимостью $\mu_A(x)$, либо дискретно – путём задания конечной последовательности значений $x \in \{x_n\}$ в виде:

$$A(x) = \left\{ \frac{\mu(x_1)}{x_1}, \frac{\mu(x_2)}{x_2}, \dots, \frac{\mu(x_n)}{x_n} \right\} \quad (132)$$

Основные понятия и определения теории нечётких множеств

Существуют стандартные функции принадлежности, которые можно определить следующим образом:

1. **Функция принадлежности класса s** определяется как:

$$s(x; a, b, c) = \begin{cases} 0 & \text{для } x \leq a, \\ 2\left(\frac{x-a}{c-a}\right)^2 & \text{для } a \leq x \leq b, \\ 1 - 2\left(\frac{x-c}{c-a}\right)^2 & \text{для } b \leq x \leq c, \\ 1 & \text{для } x \geq c, \end{cases} \quad (133)$$

где $b = (a + c)/2$. Функция, принадлежащая к данному классу, имеет сигмоидальный вид, причём её форма зависит от параметров a, b и c . В точке $x = b = (a + c)/2$ она имеет значение, равное 0.5.

2. **Функция принадлежности класса π** определяется через функцию принадлежности класса s :

$$\pi(x; b, c) = \begin{cases} s(x; c-b, c-b/2, c) & \text{для } x \leq c, \\ 1 - s(x; c, c+b/2, c+b) & \text{для } x \geq c. \end{cases} \quad (134)$$

Функция, принадлежащая к данному классу, имеет колоколообразный вид, и принимает нулевые значения в точках $x \geq c+b$ и $x \leq c-b$. В точках $x = c \pm b/2$ её значение равно 0.5.

Основные понятия и определения теории нечётких множеств

3. **Функция принадлежности класса \mathcal{Y}** задаётся выражением:

$$\gamma(x; a, b) = \begin{cases} 0 & \text{для } x \leq a, \\ \frac{x-a}{b-a} & \text{для } a \leq x \leq b, \\ 1 & \text{для } x \geq b. \end{cases} \quad (135)$$

Функции данного класса похожи на функции класса s .

4. **Функция принадлежности класса t** имеет вид:

$$t(x; a, b, c) = \begin{cases} 0 & \text{для } x \leq a, \\ \frac{x-a}{b-a} & \text{для } a \leq x \leq b, \\ \frac{c-x}{c-b} & \text{для } b \leq x \leq c, \\ 0 & \text{для } x \geq c. \end{cases} \quad (136)$$

Функции данного класса имеют треугольный вид.

5. **Функция принадлежности класса L** определяется выражением:

$$L(x; a, b) = \begin{cases} 0 & \text{для } x \leq a, \\ \frac{b-x}{b-a} & \text{для } a \leq x \leq b, \\ 1 & \text{для } x \geq b. \end{cases} \quad (137)$$

Основные понятия и определения теории нечётких множеств

Некоторые определения свойств нечётких множеств:

1. Два множества $A(x)$ и $B(x)$ равны между собой, когда $\mu_A(x) = \mu_B(x)$ для каждого элемента обоих множеств.
2. Нечёткое множество является *нормальным*, если хотя бы один элемент этого множества имеет коэффициент принадлежности, равный 1.
3. Нечёткое множество называется пустым и обозначается $A = \emptyset$ тогда и только тогда, когда $\mu_A(x) = 0$ для каждого $x \in X$.
4. Нечёткое множество A содержится в нечётком множестве B , $A \subset B$ тогда и только тогда, когда $\mu_A(x) \leq \mu_B(x)$ для каждого $x \in X$.

Операции на нечётких множествах

На нечётких множествах можно определить ряд математических операций, являющихся обобщением аналогичных операций, выполняемых на чётких множествах:

1. *Логическая сумма множеств* $A \cup B$

$$\mu_{A \cup B}(x) = \mu_A(x) \cup \mu_B(x) = \text{Max}[\mu_A(x), \mu_B(x)] \quad , (138)$$

где знак \cup обозначает оператор Max .

2. *Логическое произведение множеств* $A \cap B$

$$\mu_{A \cap B}(x) = \mu_A(x) \cap \mu_B(x) = \text{Min}[\mu_A(x), \mu_B(x)] \quad , (139)$$

где знак \cap обозначает оператор Min .

3. *Отрицание (дополнение) множества* A называется множество \bar{A} , с функцией принадлежности

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad . (140)$$

В отличие от обычных множеств, где отрицание элементов, принадлежащих к множеству, даёт пустое множество, отрицание нечёткого множества определяет непустое множество, состоящее из элементов, функции принадлежности которых, также определены на интервале $[0,1]$.

Операции на нечётких множествах

4. Равенство множеств A и B .

Нечёткие множества $A(x)$ и $B(x)$ равны между собой, когда для всех элементов x обоих множеств выполняется условие $\mu_A(x) = \mu_B(x)$.

5. Ограниченная разность двух нечётких множеств $A|-B$

$$\mu_{A|-B}(x) = \max\{0, \mu_A(x) - \mu_B(x)\} \quad (141)$$

6. Декартово произведение нечётких множеств $A \times B$

$$\mu_{A \times B}(x) = \mu_A(x) \cap \mu_B(x) = \min[\mu_A(x), \mu_B(x)] \quad (142)$$

или

$$\mu_{A \times B}(x) = \mu_A(x) * \mu_B(x) \quad (143)$$

для каждого $x \in X$ и $y \in Y$.

Операции на нечётких множествах

Определённые выше операции обладают свойствами ассоциативности, коммутативности и дистрибутивности, которые определяются следующим образом:

ассоциативность: $(A * B) * C = A * (B * C)$;

коммутативность: (за исключением ограниченной разности)

$$A * B = B * A ;$$

дистрибутивность: $A * (B \circ C) = (A * B) \circ (A * C)$,

где операторы $*$ и \circ обозначают любую операцию на нечётких множествах. Из свойств нечётких множеств следует, что в отличие от произведения обычных множеств логическое произведение множества и его отрицания не обязательно образуют пустое множество, что можно записать в виде:

$$A \cap \bar{A} \neq \emptyset \quad .(144)$$

Точно также логическая сумма нечёткого множества и его отрицание не образуют полное множества , что можно записать в виде:

$$A \cup \bar{A} \neq U \quad (145)$$

Треугольные нормы

Операции произведения и суммы нечётких множеств могут быть определены не только как **минимум** и **максимум** соответственно. Часто встречаются следующие определения:

$$1. \quad \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x) \quad A * B \quad (146)$$

2. Алгебраическое произведение двух множеств

$$\mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x) \quad (147)$$

3. Ограниченная сумма двух нечётких множеств

$$\mu_{A \cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\} \quad A | + | B \quad (148)$$

4. Ограниченное произведение двух нечётких множеств $A | \cdot | B$

$$\mu_{A \cap B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\} \quad (149)$$

$$5. \quad \mu_{A \cup B}(x) = \begin{cases} \mu_A(x), & \text{если } \mu_B(x) = 0 \\ \mu_B(x), & \text{если } \mu_A(x) = 0 \\ 1, & \text{если } \mu_A(x), \mu_B(x) > 0 \end{cases} \quad (150)$$

$$6. \quad \mu_{A \cap B}(x) = \begin{cases} \mu_A(x), & \text{если } \mu_B(x) = 1 \\ \mu_B(x), & \text{если } \mu_A(x) = 1 \\ 0, & \text{если } \mu_A(x), \mu_B(x) < 1 \end{cases} \quad (151)$$

Данные соотношения можно определить как примеры действия более общих понятий треугольных норм, S -нормы и T -нормы соответственно:

$$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x)) \quad (152)$$

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)) \quad (153)$$

Нечёткий вывод

В нечётких продукционных моделях используются в основном два способа вывода заключений: прямой и обратный. Базовое правило прямого вывода типа «если-то» (продукционное правило), называется нечёткой импликацией, принимающей форму:

$$\text{если } x \text{ это } A, \text{ то } y \text{ это } B \quad (154)$$

где X – область определения условия правила, Y – область определения заключения, $x \in X$, $y \in Y$, A и B – это нечёткие множества, идентифицированные через соответствующие функции принадлежности для переменных x и y . Часть продукционного правила « $x \text{ это } A$ » называется *условием (предпосылкой)*, а другая часть « $y \text{ это } B$ » – *следствием (заключением)*. Это обобщённое (нечёткое) правило *modusponens*.

Нечёткое рассуждение – это процедура, которая позволяет определить заключение, вытекающее из множества правил «если – то». Такое множество при N переменных x_i может принять вид:

$$\text{если } x_1 \text{ это } A_1 \text{ и } x_2 \text{ это } A_2 \text{ и } \dots \text{ и } x_N \text{ это } A_N, \text{ то } y \text{ это } B \quad (155)$$

Переменные x_1, x_2, \dots, x_N образуют N -мерный входной вектор x , составляющий аргумент условия, в котором A_1, A_2, \dots, A_N и B обозначают величины соответствующего коэффициента принадлежности $\mu_A(x)$ и $\mu_B(y)$

Нечёткий вывод

Необходимо обратить внимание на то, что здесь присутствуют индивидуальные функции принадлежности для каждой переменной x_i , и отдельно для y . Случайное значение функции принадлежности $\mu_A(x)$, где x — это вектор $x = x_1, x_2, \dots, x_N$, уровень активации правила, должно в дальнейшем интерпретироваться с использованием введенных ранее нечётких операций.

Операция нечёткой импликации является основной в нечётких продукционных моделях. В настоящее время существует большое число (несколько десятков) интерпретаций нечёткой импликации, наиболее эффективными из них являются следующие:

1. Правило типа минимум (правило Мамдани):

$$\mu_{A \rightarrow B}(x, y) = \min[\mu_A(x), \mu_B(y)] \quad (156)$$

2. Правило типа произведение (правило Ларсена):

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) * \mu_B(y) \quad (157)$$

3. Правило Лукашевича (Лукасевича):

$$\mu_{A \rightarrow B}(x, y) = \min[1, 1 - (\mu_A(x) + \mu_B(y))] \text{ или} \quad (158)$$

$$\mu_{A \rightarrow B}(x, y) = \max[0, \mu_A(x) + \mu_B(y) - 1]$$

Нечёткий вывод

4.Правило типа максмин (правило Заде):

$$\mu_{A \rightarrow B}(x, y) = \max\{\min[\mu_A(x), \mu_B(y)], 1 - \mu_A(x)\} \quad (159)$$

5.Бинарное правило Клине-Дэнса:

$$\mu_{A \rightarrow B}(x, y) = \max[1 - \mu_A(x), \mu_B(y)], \text{ при } \mu_A(x) \geq \mu_B(y) \quad (160)$$

6.Правило ограниченной суммы:

$$\mu_{A \rightarrow B}(x, y) = \min\{1, [\mu_A(x) + \mu_B(y)]\} \quad (161)$$

7.Правило Гёделя :

$$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1, \text{ если } \mu_A(x) \leq \mu_B(y), \\ \mu_B(y), \text{ если } \mu_A(x) > \mu_B(y) \end{cases} \quad (162)$$

8. Правило Гогуэна:

$$\mu_{A \rightarrow B}(x, y) = \min[1, \mu_B(y) / \mu_A(x)], \text{ при } \mu_A(x) > 0 \quad (163)$$

9. Правило стандартной чёткой импликации:

$$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1, \text{ если } \mu_A(x) \leq \mu_B(y), \\ 0, \text{ если } \mu_A(x) > \mu_B(y) \end{cases} \quad (164)$$

Набор этих правил не исчерпывает все известные определения нечёткой импликации.

Приписывание единственного значения функции принадлежности, описывающей многомерное условие, называется **агрегированием предпосылки**. Приписывание единственного значения функции принадлежности всей импликации называется **процедурой агрегирования на уровне импликации**.

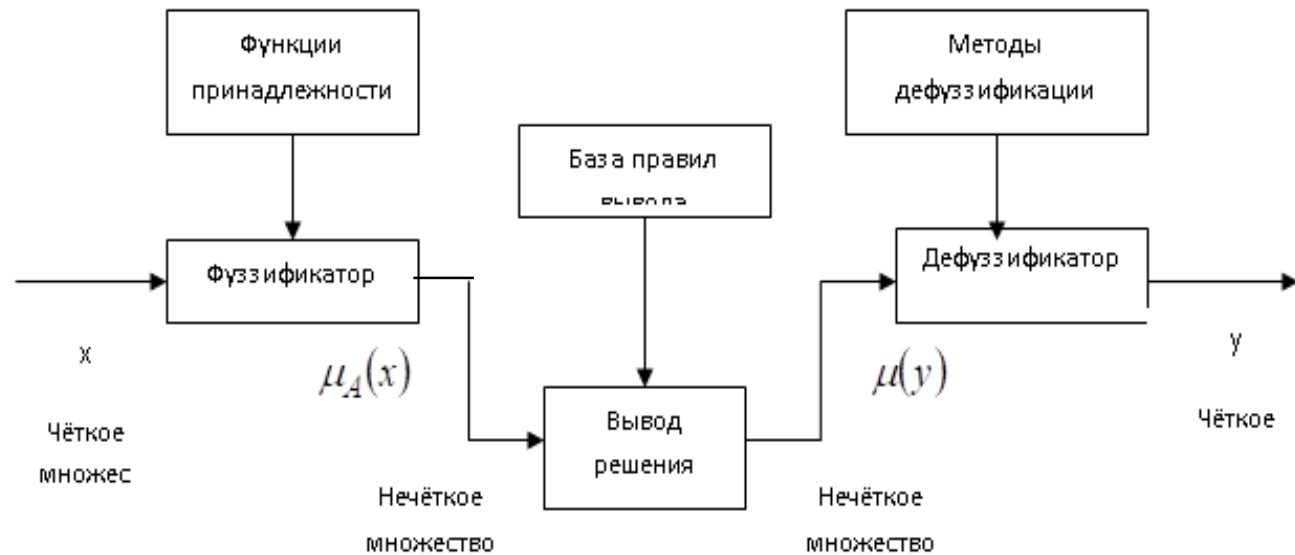
Система нечёткого вывода Мамдани-Заде

Элементы теории нечётких множеств, правила импликации и нечётких рассуждений образуют систему нечёткого вывода. В ней можно выделить базу правил, содержащую множество используемых в системе нечётких правил и описания функций принадлежности, механизм вывода и агрегирования, который формируется применяемыми правилами нечёткой импликации. Кроме того, если в качестве входных и выходных сигналов системы выступают чёткие величины, то в состав системы должны входить *фузификатор* и *дефузификатор*. Структура такой системы представлена на рисунке 18.

Фузификатор преобразует точное множество входных данных в нечёткое множество, определяемое с помощью значений функций принадлежности, **дефузификатор** решает обратную задачу – формирует однозначное решение значения выходной переменной на основании многих нечётких выводов, вырабатываемых исполнительным модулем нечёткой системы.

Выходной сигнал модуля вывода может иметь вид нечётких множеств, определяющих изменения выходной переменной. **Дефузификатор** преобразует этот диапазон в одно конкретное значение, принимаемое в качестве выходного сигнала всей системы. Конкретная форма функции дефузификации зависит от применяемой *T*-нормы, определения нечёткой импликации и от способа определения декартова произведения.

Структура системы нечёткого вывода (рисунок 18)



Система нечёткого вывода Мамдани-Заде

Так как допускается применение множества нечётких правил, в модуле вывода предусматривается блок агрегирования, чаще всего реализуемый в виде логического суммы. Описанная система вывода называется системой **Мамдани - Заде**. Как правило, в модели Мамдани - Заде присутствуют следующие операторы:

- оператор логического или алгебраического произведения для агрегации всех компонентов вектора x условия;
- оператор логического или алгебраического произведения для определения значения функции принадлежности для всей импликации $A \rightarrow B$;
- оператор логической суммы как агрегатор равнозначных результатов импликации многих правил;
- оператор $y^{(t)}$ дефузификации, трансформирующий нечёткий результат $\mu(y)$ в чёткое значение переменной y .

Фuzziфикатор

Фuzziфикатор преобразует N -мерный входной вектор $x = [x_1, x_2, \dots, x_N]^T$ в нечёткое множество A , характеризуемое функцией принадлежности $\mu_A(x)$ с чёткими переменными. Нечёткие системы могут иметь функции принадлежности произвольной структуры, в частности, стандартные формы, приведённые в (133-137), но наиболее распространёнными являются функции **гауссова** типа, а также треугольные и трапецеидальные функции. Обобщённая гауссова функция с центром c радиусом (шириной) σ и параметром b формы кривой определяется формулой:

$$\mu_A(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^{2b} \right] \quad .(165)$$

Значение $b = 1$ соответствует стандартной функции Гаусса, также можно подобрать значения параметра b , при которых формула (165) будет определять **треугольную** и **трапецеидальную** функции. На практике часто используется симметричная треугольная функция:

Фуззификатор

$$\mu_A(x) = \begin{cases} 1 - \frac{|x - c|}{d} & \text{для } x \in [c - d, c + d] \\ 0 & \text{для остальных} \end{cases}, \quad (166)$$

где c - точка середины основания треугольника, а d - половина основания треугольника.

Обобщением треугольной функции является трапецеидальная функция. Если обозначить y и z соответственно координаты точек начала и конца нижнего основания, c - длину верхнего основания, t - точку середины нижнего основания, а s - угол наклона между боковыми сторонами и нижним основанием, то трапецеидальная функция описывается следующей зависимостью:

$$\mu_A(x) = \begin{cases} 0 & \text{для } x > z \text{ или } x < y \\ 1 & \text{для } c - \frac{t}{2} \leq x \leq c + \frac{t}{2} \\ s(z - x) & \text{для } c + \frac{t}{2} \leq x \leq z \\ s(z - y) & \text{для } y \leq x \leq c - \frac{t}{2} \end{cases} \quad (167)$$

Изменение значения параметра t может преобразовать трапецеидальную функцию в треугольную.

Дефузификатор

Дефузификатор трансформирует нечёткое множество в полностью детерминированное точечное решение y . Нечёткое множество представляет зависимость $\mu(y) = \mu_{A \rightarrow B}(y)$ как функцию y от выходной переменной. Преобразование этого множества в единственное точечное решение возможно следующими наиболее распространёнными способами:

1. Дефузификация относительно центра области (если на выходе блока выработки решения формируется k нечётких множеств):

$$y_c = \frac{\sum_i \mu(y_i) y_i}{\sum_i \mu(y_i)} ; \quad (168)$$

2. Дефузификация относительно среднего центра (если на выходе блока выработки решения формируется k нечётких множеств):

$$y_c = \frac{\sum_i \mu(y_{ci}) y_{ci}}{\sum_i \mu(y_{ci})} , \quad (169)$$

где y_{ci} обозначает центр i -го нечёткого множества, в котором функция $\mu(y_{ci})$ - принимает максимальное значение, то есть $\mu(y_{ci}) = \max_{y_{ci}} \mu(y_{ci})$;

На практике чаще всего применяется дефузификация относительно среднего центра.

Модель вывода Такаги-Сугено-Канга

Наибольшую популярность среди нечётких систем адаптивного типа приобрела модель вывода **Такаги-Сугено-Канга (TSK)**. В этой модели функция заключения определяется функциональной зависимостью. Благодаря этому, дефузификатор на выходе системы не требуется, а модель вывода значительно упрощается.

Общая форма модели TSK:

если x_1 это A_1 И x_2 это A_2 И...И x_N это A_N , то $y = f(x_1, x_2, \dots, x_N)$ (170)

В векторной записи её можно записать:

если X это A , то $y = f(x)$, (171)

где $f(x) = f(x_1, x_2, \dots, x_N)$ - чёткая функция. Условие модели TSK аналогично модели Мамдани-Заде, принципиальное отличие касается заключения, которое представляется в форме функциональной зависимости, чаще всего – в виде полиномиальной функции нескольких переменных.

Классическое представление этой функции – это полином первого порядка:

$$y = f(x) = p_0 + \sum_{j=1}^N p_j x_j, (172)$$

в котором коэффициенты p_0, p_1, \dots, p_N – это веса, подбираемые в процессе обучения. Если в модели TSK используется M правил вывода, то выход системы определяется как среднее нормализованное взвешенное значение.

Модель вывода Такаги-Сугено-Канга

Если приписать каждому правилу вес w_i (интерпретируются как $\mu_A^{(i)}(x)$ в форме алгебраического произведения), то выходной сигнал можно представить в виде:

$$y = \frac{\sum_{i=1}^M w_i y_i}{\sum_{i=1}^M w_i}, \quad (173)$$

или

$$y = \sum_{i=1}^M \frac{w_i}{\sum_{i=1}^M w_i} y_i = \sum_{i=1}^M w'_i y_i. \quad (174)$$

Необходимо отметить, что в выражении (173) веса w_i отвечают условию нормализации:

$$\sum_{i=1}^M \frac{w_i}{\sum_{i=1}^M w_i} = 1.$$

Если для каждого i -го правила реализуется функция вида (174), то можно получить описание выходной функции модели TSK в виде:

$$y = \sum_{i=1}^M \frac{w_i}{\sum_{i=1}^M w_i} \left(p_{io} + \sum_{j=1}^N p_{ij} x_j \right), \quad (175)$$

которая линейна относительно всех входных переменных системы x_j для $j=1,2,\dots,N$. Веса w_i являются нелинейными параметрами функции y , которые уточняются в процессе обучения.

Модель вывода Цукамото

В модели **Цукамото** в качестве функций заключения используются монотонные (возрастающие или убывающие) функции. Заключения правил f^{-1} формируются путём обратного преобразования этих функций по полученным значениям предпосылок данных правил:

$$\text{если } x_1 \text{ это } A_1 \text{ И } x_2 \text{ это } A_2 \text{ И...И } x_N \text{ это } A_N, \text{ то } y = f^{-1}(w). \quad (176)$$

где w - уровень срабатывания предпосылки правила.

Условие модели Цукамото аналогично модели Мамдани-Заде, принципиальное отличие касается заключения.

Если в модели Цукамото используется M правил вывода, то выход системы определяется как среднее взвешенное значение. Если приписать каждому правилу вес w_i (интерпретируются как $\mu_A^{(i)}(x)$ в форме алгебраического произведения), то выходной сигнал можно представить в виде:

$$y = \frac{\sum_{i=1}^M w_i f_i^{-1}(w_i)}{\sum_{j=1}^M w_j}, \quad (177)$$

где $w_i = f_i(y_i)$ - значение аргумента функции f_i , при котором $y_i = f_i^{-1}(w_i)$

Модель вывода Ларсена

Правила вывода в модели **Ларсена** формируются аналогично модели Мамдани-Заде. В модели Ларсена присутствуют следующие операторы:

- ✓ оператор алгебраического или логического произведения для агрегации всех компонентов вектора x условия;
- ✓ оператор алгебраического произведения для определения значения функции принадлежности для всей импликации $A \rightarrow B$;
- ✓ оператор логической суммы как агрегатор равнозначных результатов импликации многих правил;
- ✓ оператор $\mu(y)$ дефузификации, трансформирующий нечёткий результат в чёткое значение переменной Y .

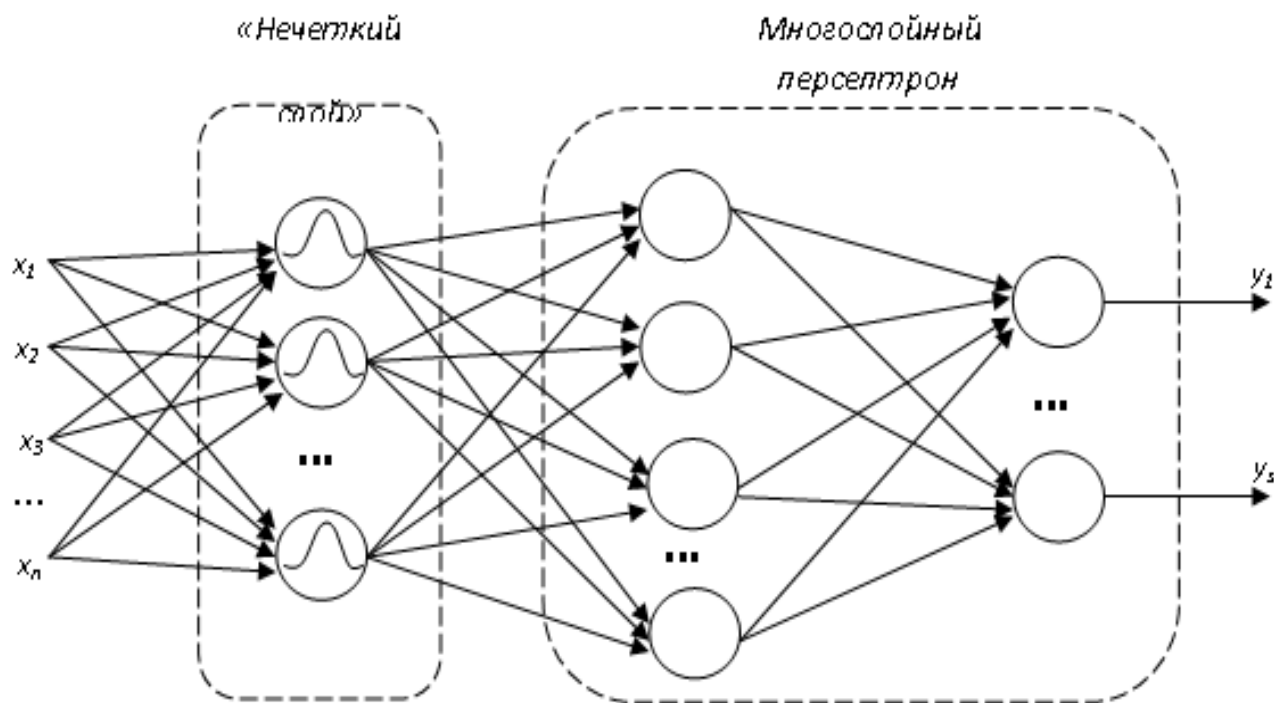
Гибридный нечёткий многослойный персептрон

В случае линейной неразделимости классов, то есть когда классы пересекаются, а обучающие примеры приводят к неустойчивому поведению алгоритма обучения персептрона, имеет смысл использовать гибридную сеть, включающую в себя слой с нечёткой самоорганизацией и многослойный персептрон.

Основная идея построения нечеткого персептрона и алгоритма его обучения заключается в уменьшении влияния векторов, лежащих в зоне перекрытия классов, на изменение весовых коэффициентов.

На рисунке 19 приведена структура **нечеткого многослойного персептрона**. Он состоит из двух частей: нейронов «нечеткого слоя» и собственно **многослойного персептрона**. Функции активации нейронов «нечеткого слоя» такой сети являются радиальными базисными функциями (в виде **функции Гаусса**), моделирующими функции принадлежности. Эти нейроны предназначены для определения степеней принадлежности компонентов входных векторов (которые могут быть и нечеткими). На выходах нейронов этого слоя формируются коэффициенты в требуемой для дальнейшего распознавания форме. Выходы нейронов «нечеткого слоя» употребляются в качестве входов традиционного многослойного персептрона.

Гибридный нечёткий многослойный персептрон (рисунок 19)



Гибридный нечёткий многослойный персептрон

Если на вход сети подается $x = [x_1, x_2, \dots, x_n]^T$, то на выходе «нечеткого слоя» формируется вектор $\mu(x) = [\mu_1(x), \mu_2(x), \dots, \mu_n(x)]^T$, состоящий из степеней принадлежности x к конкретным центрам (радиально базисным функциям). Конкретные компоненты $\mu_i(x)$ рассчитываются таким образом, чтобы удовлетворять условию нормализации $\sum_{i=1}^m \mu_i(x^{(t)}) = 1$ для каждого вектора $x^t, t = 1, \dots, p$, где p – число векторов в обучающей выборке.

Выходы нечёткого многослойного персептрона трактуются как степени принадлежности предъявленного объекта соответствующему классу.

С учётом ярко выраженной двухкомпонентной структуры гибридной сети для её обучения применяется алгоритм, состоящий из двух этапов. На первом из них проводится обучение самоорганизующегося слоя, состоящее в подборе центров. Для этого может быть использован алгоритм *C-means* или другие подобные алгоритмы (например, алгоритм Густавсона-Кесселя).

Гибридный нечёткий многослойный персептрон

По завершении первого этапа, начинается второй этап обучения, на котором уточняются только веса персептронной компоненты. Это обычное обучение многослойного персептрона, для которого входом является множество коэффициентов принадлежности вектора x к центрам самоорганизующегося слоя. В зависимости от типа решаемой задачи выходом сети может быть код класса, к которому принадлежит входной вектор x , либо ожидаемое значение d выходного вектора.

Для обучения весов скрытых слоёв и выходного слоя нечёткого многослойного персептрона могут быть использованы градиентные алгоритмы обучения или эвристические алгоритмы обучения, используемые для обучения чёткого многослойного персептрона.

Для уточнения всех параметров нечёткого многослойного персептрона, включая параметры функции Гаусса нейронов слоя фuzziфикации можно использовать градиентный алгоритм наискорейшего спуска или его модификацию с моментами по аналогии с обучением радиально-базисных сетей. При этом алгоритм самоорганизации может быть использован для первичной инициализации центров функции Гаусса.

Алгоритм нечёткой самоорганизации C-means

Алгоритм самоорганизации приписывает вектор x к соответствующей группе данных, представляемых центром c_i , с использованием обучения конкурентного типа подобно сетям Кохонена. При обучении этого типа процесс самоорганизации становится возможным при предъявлении вектора x .

Допустим, что в сети существует M нечётких нейронов с центрами в точках $c_i (i = 1, 2, \dots, M)$. Начальные значения этих центров могут быть выбраны случайным образом из областей допустимых значений соответствующих компонентов векторов $x (t = 1, 2, \dots, p)$, использованных для обучения. Пусть функция фуззификации задана в форме обобщенной функции Гаусса, выраженной формулой (179).

$$\mu(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^{2b} \right] \quad (179)$$

Подаваемый на вход сети вектор будет принадлежать к различным группам, представляемым центрами в степени u_{it} , а суммарная степень принадлежности ко всем группам, очевидно, равна единице. Поэтому

$$\sum_{i=1}^M u_{it} = 1 \quad (180)$$

для $t = 1, 2, \dots, p$. Функцию погрешности, соответствующую такому представлению, можно определить как сумму частных погрешностей принадлежности к центрам c_i с учетом степени принадлежности u_{it} .

Алгоритм нечёткой самоорганизации C-means

Следовательно

$$E = \sum_{i=1}^M \sum_{t=1}^P (u_{it})^m \|c_i - x_t\|^2, \quad (181)$$

где m — это весовой коэффициент, который принимает значения из интервала $[1, \infty)$, на практике часто принимают $m=2$. Цель обучения с самоорганизацией состоит в таком подборе центров c_i , чтобы для заданного множества обучающих векторов $x^{(t)}$ обеспечить достижение минимума функции (181) при одновременном соблюдении условий ограничения (180). Таким образом возникает задача минимизации нелинейной функции (181) с p ограничениями типа (180). Решение этой задачи можно свести к минимизации функции Лагранжа, определенной в виде.

$$LE = \sum_{i=1}^M \sum_{t=1}^P (u_{it})^m \|c_i - x_t\|^2 + \sum_{t=1}^P \lambda_t \left(\sum_{i=1}^M u_{it} - 1 \right), \quad (182)$$

где $\lambda_t (t=1, 2, \dots, p)$ — это множители Лагранжа.

Доказано, что решение задачи (182) можно представить в виде:

$$c_i = \frac{\sum_{t=1}^P (u_{it})^m x_t}{\sum_{t=1}^P (u_{it})^m}, \quad (183)$$

Алгоритм нечёткой самоорганизации C-means

$$u_{it} = \frac{1}{\sum_{k=1}^M \left(\frac{d_{it}^2}{d_{kt}^2} \right)^{\frac{1}{m-1}}}, \quad (184)$$

где d_{it} - это эвклидово расстояние между центром C_i и вектором x_t , $d_{it} = \|c_i - x_t\|$. Поскольку точные значения центров C_i в начале процесса не известны, алгоритм обучения должен быть итерационным:

1. Выполнить случайную инициализацию коэффициентов u_{it} , выбирая их значения из интервала $[0,1]$ таким образом, чтобы соблюдать условие (180).
2. Определить M центров в соответствии с (183).
3. Рассчитать значение функции погрешности в соответствии с (181). Если её значение ниже установленного порога, либо если уменьшение этой погрешности относительно предыдущей итерации пренебрежимо мало, то закончить вычисления. Иначе, перейти к п.4.
4. Рассчитать новые значения по формуле (184) и перейти к п.2.

Многократное повторение итерационной процедуры ведёт к достижению минимума функции E , который необязательно будет глобальным.

Алгоритм разностного группирования

Алгоритм разностного группирования данных является модификацией алгоритма пикового группирования, предложенного З.Егером и Д.Филёвым. В качестве меры плотности размещения векторов x_t используются так называемые пиковые функции. В алгоритме разностного группирования в качестве потенциальных центров пиковых функций используются обучающие векторы x_t . Пиковая функция задаётся в виде:

$$D(x_i) = \sum_{t=1}^p \exp \left(- \frac{\|x_i - x_t\|^{2b}}{\left(\frac{r_a}{2}\right)^2} \right) \quad (185)$$

Значение коэффициента r_a определяет сферу соседства векторов. При большой плотности векторов x_t вокруг точки x_i , значение функции $D(x_i)$ велико, следовательно, точка x_i является «удачным» кандидатом в центры. После расчёта значений пиковой функции для всех точек x_t , отбирается та, для которой значение функции $D(x_i)$ оказалось максимальным. Именно эта точка становится первым отобранным центром c_1 . Выбор следующего центра возможен после исключения предыдущего и всех точек, лежащих в его окрестности. Для этого пиковая функция переопределяется в виде:

$$D_{new}(x_i) = D(x_i) - D(c_1) \exp \left(- \frac{\|x_i - c_1\|^{2b}}{\left(\frac{r_b}{2}\right)^2} \right) \quad (186)$$

Алгоритм разностного группирования

Обычно при выборе новой константы r_b соблюдается условие $r_b \geq r_a$. Пиковая функция $D_{new}(x_i)$ принимает нулевое значение при $x_i = c_1$ и близка к нулю в ближайшей окрестности этой точки.

После модификации пиковой функции отыскивается следующая точка x_i , для которой величина $D_{new}(x_i)$ оказывается максимальной. Эта точка становится следующим центром c_2 . Процесс поиска очередного центра продолжается после исключения всех предыдущих центров и их окрестностей. Процесс инициализации центров завершается в момент фиксации всех центров, предусмотренных начальными условиями.

Независимо от способа реализации алгоритма обучения сеть с нечёткой самоорганизацией выполняет нечёткое группирование данных путём приписывания их к различным центрам на основании коэффициентов принадлежности, значения которых изменяются от нуля до единицы. Это означает, что каждый вектор x представляется множеством центров, причём влияние каждого из них на значение вектора различно и зависит от величины коэффициента принадлежности. Если считать, что вектор x_t представляется M центрами $c_i (i = 1, 2, \dots, M)$, а принадлежность вектора к каждому центру задана коэффициентом u_{it} (формула 180), то реконструкция исходного вектора происходит согласно выражению:

$$x_t = \sum_{i=1}^M u_{it} c_i \quad . \quad (187)$$

Нечеткая нейронная сеть Ванга-Менделя

Если в модели Мамдани-Заде в качестве агрегатора использовать оператор алгебраического произведения, то дефуззификация относительно среднего центра, приводит к модели Менделя-Ванга. Следует отметить, что $\mu(y)$ состоит из суммы нечетких функций для импликаций всех M правил, образующих систему нечеткого вывода. В модели Мамдани-Заде каждое из этих M правил определяется уровнем активации условия, $\mu(y_i) = \prod_{j=1}^N \mu_{A_j}(x_j)$, тогда как y_i - это значение $\mu(y)$, при котором величина y становится максимальной (либо принимает среднее из максимальных значений). Пусть величина v_i обозначает центр нечеткого множества заключения i -го правила вывода. Тогда дефуззификация относительно среднего центра в модели Менделя-Ванга, в соответствии с которой:

$$y = \frac{\sum_{i=1}^M v_i \left[\prod_{j=1}^N \mu_{A_j}(x_j) \right]}{\sum_{i=1}^M \left[\prod_{j=1}^N \mu_{A_j}(x_j) \right]} \quad (188)$$

Допустим, что существует нечеткая система, описываемая зависимостью (188), на вход которой подается последовательность векторов $x = [x_1, x_2, \dots, x_N]$. При использовании фуззификатора в виде обобщенной гауссовой функции выходной сигнал y этой системы определяется по формуле:

$$\mu(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^{2b} \right]$$

Нечеткая нейронная сеть Ванга-Менделя

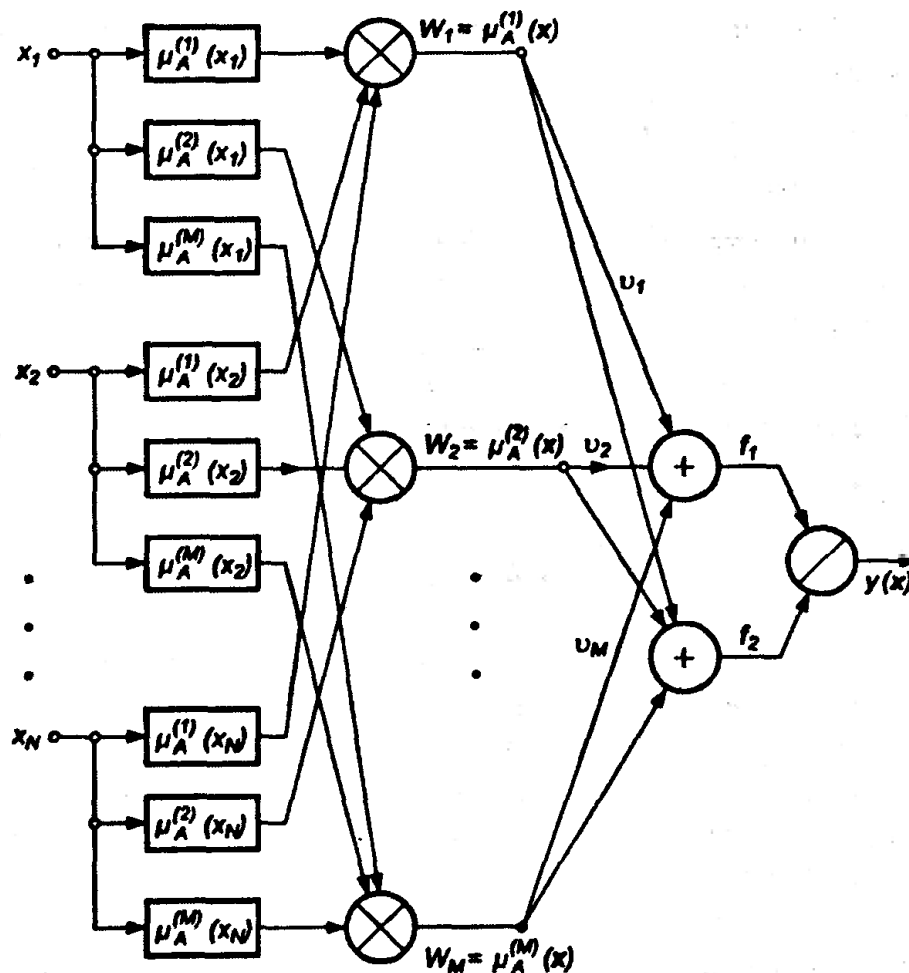
$$y = f(x) = \frac{\sum_{i=1}^M v_i \left[\prod_{j=1}^N \exp \left[- \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}} \right] \right]}{\sum_{i=1}^M \left[\prod_{j=1}^N \exp \left[- \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}} \right] \right]} \quad (189)$$

в которой $(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)})$ обозначают параметры центра, ширины и формы (условия) j -го компонента вектора x для i -го нечеткого правила вывода.

Выражение (189) определяет непрерывную функцию, которая может использоваться для аппроксимации произвольно заданной непрерывной функции $g(x)$ от многих переменных x_j , образующих вектор x . При соответствующем подборе параметров условия $(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)})$ и заключения w_i , функция (189) может аппроксимировать заданную функцию $g(x)$ с произвольной точностью ε . Способность нечёткой системы, характеризующейся рядом нелинейных функций от одной переменной, к аппроксимации нелинейной функции от многих переменных, свидетельствует о возможностях практического применения нечётких систем.

Если использовать в качестве основы дальнейших рассуждений выражение (189), можно получить структуру нечеткой сети (рисунок 20), определенную Л.Вангом и Дж.Менделем.

Нечеткая нейронная сеть Ванга-Менделя (рисунок 20)



Нечеткая нейронная сеть Ванга-Менделя

Это четырехслойная структура, в которой первый слой выполняет фуззификацию входных переменных, второй – агрегирование значений отдельных переменных x_j в условии i -го правила вывода, третий (линейный) – агрегирование M правил вывода (первый нейрон) и генерацию нормализующего сигнала (второй нейрон), тогда как состоящий из одного нейрона выходной слой осуществляет нормализацию, формируя выходной сигнал $y(x)$.

Только первый и третий слои являются параметрическими. В первом слое это параметры функции фуззификации $(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)})$, а в третьем слое – веса v_1, v_2, \dots, v_M , интерпретируемые как центры v_i функции принадлежности следствия i -ого нечеткого правила вывода.

Представленная на рисунке 20 сетевая структура реализует функцию аппроксимации (190), которую с учетом введенных обозначений можно записать в виде

$$y(x) = \frac{1}{\sum_{i=1}^M \left[\prod_{j=1}^N \mu_A^{(i)}(x_j) \right]} \sum_{i=1}^M v_i \left[\prod_{j=1}^N \mu_A^{(i)}(x_j) \right] \quad (190)$$

Адаптивный алгоритм обучения нечёткой сети Ванга- Менделя

Важным вопросом обучения сети является подбор начальных значений для параметров нейронной сети. Существует несколько подходов к решению этого вопроса. Так, начальное приближение для весов сети выбирается случайным образом. Вопрос об инициализации центров c_j требует более тонкого подхода, так как положение функции в пространстве входных данных тесно связано с классифицирующей способностью сети. Естественным в таком случае ориентироваться на характер обучающих данных, и в качестве начального значения центра функции выбрать среднее арифметическое всех значений входных переменных, т.е. разместить функцию равноудаленно от всех точек пространства входных данных. Дальнейший подбор центров будет произведен при работе адаптивного метода подбора параметров так же, как и общее количество центров.

Адаптивный алгоритм был сформулирован только для гауссовой функции с использованием обобщенной модели Ванга-Менделя ($b=1$). В результате его реализации определяются: количество центров и их расположение в части, соответствующей условиям (множество векторов x_i) и заключениям (множество скалярных ожидаемых значений d_i). Этот алгоритм можно описать следующим образом.

Адаптивный алгоритм обучения нечёткой сети Ванга- Менделя

1. При старте с первой пары данных (x_1, d_1) создается первый кластер с центром $c_1=x_1$. Принимается, что $v_1=d_1$ и что мощность множества $L_1=1$. Пусть r обозначает предельное эвклидово расстояние между вектором x и центром, при котором данные будут трактоваться как принадлежащие к созданному кластеру (то есть максимальный радиус функции). Для сохранения общности решения принимается, что в момент начала обучения существует M кластеров с центрами c_1, c_2, \dots, c_M и соответствующие им значения v_i и L_i ($i=1,2,\dots,M$).
2. После считывания t -ой обучающей пары (x_t, d_t) рассчитываются расстояния между вектором x_t и всеми существующими центрами $\|x_t-c_i\|$ для $i=1, 2, \dots, M$ и выбирается центр, ближайший к x_t . Допустим, что ближайший центр — это c_{it} . В таком случае в зависимости от значения $\|x_t-c_{it}\|$ может возникнуть одна из двух ситуаций:
 - ✓ если $\|x_t-c_{it}\|>r$, то создается новый кластер $c_{M+1}=x_t$, причем $v_{M+1}(t)=d_t$, $L_{M+1}(t)=1$. Параметры созданных до этого кластеров не изменяются, т.е. $v_i(t)=v_i(t-i)$, $L_i(t)=L_i(t-i)$ для $i=1, 2, \dots, M$. Количество кластеров M увеличивается на 1 ($M \leftarrow M+1$);

Адаптивный алгоритм обучения нечёткой сети Ванга- Менделя

- ✓ если $\|x_t - c_{it}\| \leq r$, то данные включаются в it -й кластер, параметры которого следует уточнить в соответствии с формулами:

$$v_{it}(t) = v_{it}(t-1) + d_t \quad , (191)$$

$$L_{it}(t) = L_{it}(t-1) + 1 \quad , (192)$$

$$c_{it}(t) = \frac{c_{it}(t-1)L_{it}(t-1) + x_t}{L_{it}(t)} \quad . (193)$$

тогда как остальные кластеры не изменяются, т.е. при $i \neq it$,
 $v_i(t) = v_i(t-1)$, $L_i(t) = L_i(t-1)$ и $c_i(t) = c_i(t-1)$, для $i = 1, 2, \dots, M$.

В другой версии алгоритма фиксируется положение центров c_{it} после инициализации, и их координаты уже не изменяются. Во многих случаях такой прием улучшает результаты адаптации.

3. После уточнения параметров нечеткой системы функция, аппроксимирующая входные данные системы, определяется в виде:

$$f(x) = \frac{\sum_{i=1}^M v_i(t) \exp\left(-\frac{\|x - c_i(t)\|^2}{\sigma^2}\right)}{\sum_{i=1}^M L_i(t) \exp\left(-\frac{\|x - c_i(t)\|^2}{\sigma^2}\right)} \quad , (194)$$

Адаптивный алгоритм обучения нечёткой сети Ванга- Менделя

При повторении перечисленных этапов алгоритма до $t=p$ с уточнением каждый раз значения M , пространство данных разделяется на M кластеров, при этом мощность каждого из них определяется как $L_i=L_i(t)$, центр - как $c_i=c_i(t)$, а значение приписанной ему накопленной функции d - как $v_i = v_i(t)$.

Этот алгоритм называется самоорганизующимся, поскольку разделение пространства данных на кластеры происходит самостоятельно и без участия человека, в соответствии с заданным значением порога r . При малом значении r количество кластеров возрастает, в результате чего аппроксимация данных становится более точной, однако это достигается за счет более сложной функции и увеличения объема необходимых вычислений при одновременном ухудшении обобщающих свойств сети. Если значение r слишком велико, то вычислительная сложность уменьшается, однако возрастает погрешность аппроксимации.

Адаптивный алгоритм обучения нечёткой сети Ванга- Менделя

При подборе оптимальной величины порога r должен соблюдаться компромисс между точностью отображения и вычислительной сложностью. Как правило, оптимальное значение r подбирается методом проб и ошибок с использованием вычислительных экспериментов, по существу r это максимальное значение параметра σ функции Гаусса. Таким образом, следует подобрать оптимальные значения σ и затем взять максимальное из них. Оптимальные значения σ можно подобрать в соответствии с формулой (65), то есть так же, как для радиально-базисных сетей.

Нейронная сеть Ванга-Менделя может быть представлена как частный случай нечёткой продукционной сети *TSK*, в случае, когда полином в заключении правил вывода содержит только свободный член. В этой связи, алгоритмы обучения и методы модификации структуры для сети *TSK*, могут быть использованы и для сети Ванга-Менделя.

Нечёткая сеть TSK

Структура нечёткой сети TSK основана на системе нечёткого вывода Такаги-Сугэно- Канга. При этом в качестве функции фуззификации для каждой переменной используется дробно-рациональная функция Гаусса в виде:

$$\mu_A(x_j) = \frac{1}{1 + \left(\frac{x_j - c_j}{\sigma_j} \right)^{2b_j}} \quad (195)$$

Для агрегации условия i -го правила в системе вывода TSK используется операция алгебраического произведения:

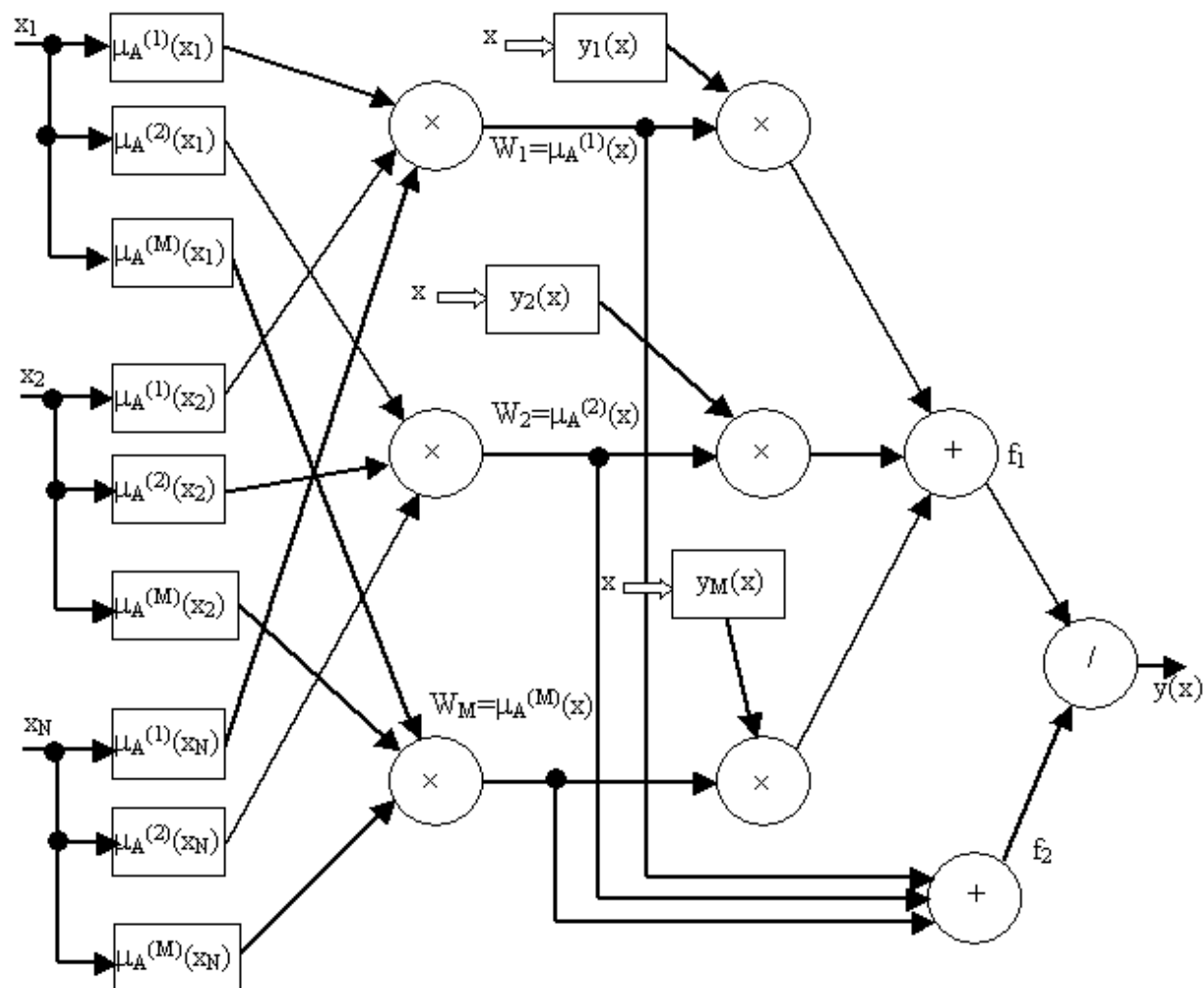
$$\mu_A^{(i)}(x) = \prod_{j=1}^N \left[\frac{1}{1 + \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}}} \right] \quad (196)$$

При M правилах вывода агрегирование выходного результата сети производится по формуле (218), которую можно представить в виде:

$$y(x) = \frac{1}{\sum_{i=1}^M w_i} \sum_{i=1}^M w_i y_i(x) \quad (197)$$

где $y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij} x_j$, агрегация импликации. Присутствующие в этом выражении веса w_i интерпретируются как компоненты $\mu_A^{(i)}(x)$, определённые формулой (196). При этом формуле (197) можно сопоставить многослойную структуру сети, изображённую на рисунке 21.

Нечёткая сеть TSK (рисунок 21)



Нечёткая сеть TSK

В такой сети выделяется пять слоёв:

- ✓ Первый слой выполняет отдельную фуззификацию каждой переменной $x_j (j=1,2,...,N)$, определяя для каждого i -го правила вывода значение коэффициента принадлежности $\mu_A^{(i)}(x_j)$ в соответствии с применяемой функцией фуззификации. Это параметрический слой с параметрами $(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)})$, подлежащими адаптации в процессе обучения.
- ✓ Второй слой выполняет агрегирование отдельных переменных x_j , определяя результирующее значение коэффициента принадлежности $W_i = \mu_A^{(i)}(x)$ для вектора x в соответствии с формулой (196). Этот слой непараметрический.
- ✓ Третий слой представляет собой генератор функции TSK, рассчитывающий значения $y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij} x_j$. В этом слое также происходит умножение сигналов $y_i(x)$ на значения W_i сформированные в предыдущем слое. Это параметрический слой, в котором адаптации подлежат линейные веса p_{ij} для $i=1,2,...,M$ и $j=1,2,...,N$, определяющие функцию следствия модели TSK.
- ✓ Четвёртый слой составляют два нейрона- сумматора, один из которых рассчитывает взвешенную сумму сигналов $y_i(x)$, а второй определяет сумму весов $\sum_{i=1}^M W_i$. Это непараметрический слой.

Нечёткая сеть TSK

- ✓ Пятый слой состоит из одного выходного нейрона- это нормализующий слой, в котором веса подвергаются нормализации в соответствии с формулой (198). Выходной сигнал $y(x)$ определяется выражением, соответствующим зависимости (197),

$$y(x) = f(x) = \frac{f_1}{f_2} \quad (198)$$

Это также непараметрический слой.

При уточнении функциональной зависимости (10.10) для сети TSK получаем:

$$y(x) = \frac{1}{\sum_{i=1}^M \left[\prod_{j=1}^N \mu_A^{(i)}(x_j) \right]} \sum_{i=1}^M \left[\prod_{j=1}^N \mu_A^{(i)}(x_j) \right] \left[p_{i0} + \sum_{j=1}^N p_{ij} x_j \right] \quad (199)$$

Если принять, что в конкретный момент времени параметры условия зафиксированы, то функция $y(x)$ является линейной относительно переменных p_{ij} x_j ($j=1,2,\dots,N$). При наличии N входных переменных каждое правило формирует $N+1$ переменныхлинейной зависимости TSK. При M правилах вывода это даёт $M(N+1)$ линейных параметров сети. В свою очередь каждая функция принадлежности использует три параметра, подлежащих адаптации. Так как каждая переменная характеризуется собственной функцией принадлежности, то мы получим $3MN$ нелинейных параметров. В сумме это даёт $M(4N+1)$ линейных и нелинейных параметров, значения которых должны подбираться в процессе обучения сети.

Гибридный алгоритм обучения нечеткой сети TSK

Гибридный алгоритм обучения может применяться как для сетей Ванга-Менделя, так и для сетей TSK. Сеть Ванга - Менделя может при этом трактоваться как сеть TSK, у которой все параметры p_{ij} , кроме p_{i0} , равны нулю.

В гибридном алгоритме подлежащие адаптации параметры разделяются на две группы: линейных параметров p_{ij} третьего слоя и параметров нелинейной функции принадлежности первого слоя. Уточнение параметров проводится в два этапа:

На первом этапе при фиксации определенных значений параметров функции принадлежности путем решения системы линейных уравнений рассчитываются линейные параметры (в первом цикле – это значения, полученные в результате инициализации). При известных значениях функции принадлежности зависимость (199) можно представить в линейной форме :

$$y(x) = \sum_{i=1}^M w'_i \left(p_{i0} + \sum_{j=1}^N p_{ij} x_j \right) \quad , (200)$$

где

$$w'_i = \frac{\prod_{j=1}^N \mu_A^{(i)}(x_j)}{\sum_{i=1}^M \left[\prod_{j=1}^N \mu_A^{(i)}(x_j) \right]} = const \quad (201)$$

для $i=1, 2, \dots, M$.

Гибридный алгоритм обучения нечеткой сети TSK

При p обучающих выборках $(x(t), d(t))$ ($t= 1, 2, \dots, p$) и замене выходного сигнала сети ожидаемым значением $d(t)$ получим систему из p линейных уравнений вида

$$\begin{bmatrix} w'_{11} & w'_{11} x_1^{(1)} & \dots & w'_{11} x_N^{(1)} & \dots & w'_{1M} & w'_{1M} x_1^{(1)} & \dots & w'_{1M} x_N^{(1)} \\ w'_{21} & w'_{21} x_1^{(2)} & \dots & w'_{21} x_N^{(2)} & \dots & w'_{2M} & w'_{2M} x_1^{(2)} & \dots & w'_{2M} x_N^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ w'_{p1} & w'_{p1} x_1^{(p)} & \dots & w'_{p1} x_N^{(p)} & \dots & w'_{pM} & w'_{pM} x_1^{(p)} & \dots & w'_{pM} x_N^{(p)} \end{bmatrix} \begin{bmatrix} p_{10} \\ \dots \\ p_{1N} \\ \dots \\ p_{M0} \\ \dots \\ p_{MN} \end{bmatrix} = \begin{bmatrix} d^{(1)} \\ d^{(2)} \\ \dots \\ d^{(p)} \end{bmatrix}, \quad (202)$$

где w'_{ii} обозначает уровень активации (вес) условия i -го правила при предъявлении t -го входного вектора x . Это выражение можно записать в сокращенной матричной форме

$$Ap = d, \quad (203)$$

Размерность матрицы A равна $p \times (N+1)M$, при этом количество строк значительно больше количества столбцов $(N+1)M$.

При помощи псевдоинверсии матрицы A решение можно получить за один шаг:

$$p = A^+ d, \quad (204)$$

где A^+ псевдоинверсия матрицы A . Псевдоинверсия матрицы A заключается в проведении декомпозиции SVD с последующим сокращением её размерности.

Гибридный алгоритм обучения нечеткой сети TSK

На втором этапе после фиксации значений линейных параметров рассчитываются фактические выходные сигналы $y(t)$ сети для $t=1, 2, \dots, p$, для чего используется линейная зависимость

$$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(p)} \end{bmatrix} = Ap, \quad (205)$$

и следом за ними – вектор ошибки $\varepsilon = y - d$. Сигналы ошибок направляются через подключенную сеть по направлению к входу сети (обратное распространение) вплоть до первого слоя, где могут быть рассчитаны компоненты градиента целевой функции относительно конкретных параметров $(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)})$. После формирования вектора градиента параметры уточняются с использованием одного из градиентных методов обучения, например, метода наискорейшего спуска.

$$c_j^{(i)}(t+1) = c_j^{(i)}(t) - \eta_c \frac{\partial E(t)}{\partial c_j^{(i)}} \quad (206)$$

Гибридный алгоритм обучения нечеткой сети TSK

$$\sigma_j^{(i)}(t+1) = \sigma_j^{(i)}(t) - \eta_\sigma \frac{\partial E(t)}{\partial \sigma_j^{(i)}} \quad (207)$$

$$b_j^{(i)}(t+1) = b_j^{(i)}(t) - \eta_b \frac{\partial E(t)}{\partial b_j^{(i)}} \quad (208)$$

После уточнения нелинейных параметров вновь запускается процесс адаптации линейных параметров функции TSK (первый этап) и нелинейных параметров (второй этап). Этот цикл повторяется вплоть до стабилизации всех параметров процесса. Формулы (206) – (208) требуют расчёта градиента целевой функции принадлежности и для одной пары обучающих данных (x, d) принимают значения:

$$\frac{\partial E}{\partial c_j^{(i)}} = (y(x) - d) \sum_{i=1}^M \left(\left[p_{i0} + \sum_{j=1}^N p_{ij} x_j \right] \frac{\partial w'_i}{\partial c_j^{(i)}} \right) \quad (209)$$

Гибридный алгоритм обучения нечеткой сети TSK

$$\frac{\partial E}{\partial \sigma_j^{(i)}} = (y(x) - d) \sum_{i=1}^M \left(\left[p_{i0} + \sum_{j=1}^N p_{ij} x_j \right] \frac{\partial w'_i}{\partial \sigma_j^{(i)}} \right) \quad (210)$$

$$\frac{\partial E}{\partial b_j^{(i)}} = (y(x) - d) \sum_{i=1}^M \left(\left[p_{i0} + \sum_{j=1}^N p_{ij} x_j \right] \frac{\partial w'_i}{\partial b_j^{(i)}} \right) \quad (211)$$

$$\frac{\partial w'_k}{\partial c_j^{(i)}} = \frac{\delta_{ki} m(x_j) - l(x_j)}{[m(x_j)]^2} \prod_{s=1, s \neq j}^N [\mu_A^{(i)}(x_s)] \frac{\left[\frac{2b_j^{(i)} \left(x_j - c_j^{(i)} \right)^{2b_j^{(i)}-1}}{\sigma_j^{(i)} \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)} \right]}{\left[1 + \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}} \right]^2} \quad (212)$$

$$\frac{\partial w'_k}{\partial \sigma_j^{(i)}} = \frac{\delta_{ki} m(x_j) - l(x_j)}{[m(x_j)]^2} \prod_{s=1, s \neq j}^N [\mu_A^{(i)}(x_s)] \frac{\left[\frac{2b_j^{(i)} \left(x_j - c_j^{(i)} \right)^{2b_j^{(i)}}}{\sigma_j^{(i)} \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)} \right]}{\left[1 + \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}} \right]^2} \quad (213)$$

Гибридный алгоритм обучения нечеткой сети TSK

$$\frac{\partial w'_k}{\partial b_j^{(i)}} = \frac{\delta_{ki} m(x_j) - l(x_j)}{[m(x_j)]^2} \prod_{s=1, s \neq j}^N [\mu_A^{(i)}(x_s)] \frac{\left[-2 \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}} \ln \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right) \right]}{\left[1 + \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}} \right]^2} \quad , (214)$$

для $k=1,2,...,M$, где обозначает дельту Кронекера, $l(x) = \prod_{s=1}^N \mu_A^{(k)}(x_s)$ $m(x) = \sum_{i=1}^M \left[\prod_{s=1}^N \mu_A^{(i)}(x_s) \right]$

(x это вектор, первая формула это формула 196, а вторая – формула 197 или f_2 на рисунке 21).

При практической реализации гибридного метода обучения нечетких сетей доминирующим фактором их адаптации считается первый этап, на котором веса P_{ij} подбираются с использованием псевдоинверсии за один шаг. Для уравнивания его влияния второй этап (подбор нелинейных параметров градиентным методом) многократно повторяется в каждом цикле. При практической реализации алгоритма на первом этапе первый раз разность $\varepsilon = y - d$ может быть очень маленькой, в связи с чем, второй этап гибридного алгоритм не работает. В этом случае имеет смысл один раз искусственно увеличить $\varepsilon = y - d$

Алгоритм наискорейшего спуска для сети TSK

Приведённый ниже алгоритм наискорейшего спуска и метод обратного распространения ошибки может быть использован и для обучения сети Ванга-Менделя.

Нечеткая нейронная сеть TSK имеет многослойную структуру с прямым распространением сигнала, значение выхода которой можно изменять, корректируя параметры элементов слоев, что позволяет для обучения этой сети использовать алгоритм обратного распространения ошибки. Для этого потребуется обучающая выборка в виде пар (x, d) , где $x = [x_1, \dots, x_N]^T$ – это входной вектор, а d – эталонный сигнал. Задача заключается в такой коррекции параметров сети, описанной выражением (206) чтобы мера погрешности, задаваемая выражением:

$$E = \frac{1}{2} (y(x) - d)^2 \quad (215)$$

была минимальной.

Если применяется простейший метод наискорейшего спуска, то соответствующие формулы адаптации принимают форму:

$$p_{ij}(t+1) = p_{ij}(t) - \eta_p \frac{\partial E(t)}{\partial p_{ij}}, \quad (216)$$

Алгоритм наискорейшего спуска для сети TSK

Параметры $(c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)})$ корректируются в соответствии с формулами (206)-(208).

Формулы (206-208 и 216) требуют расчета градиента целевой функции относительно параметров функции принадлежности. Окончательный вид этих формул зависит от используемого определения функции погрешности на выходе сети, так и от формы функции принадлежности. Например, при использовании функции Гаусса :

$$\mu_A(x_j) = \frac{1}{1 + \left(\frac{x_j - c_j}{\sigma_j} \right)^{2b_j}} \quad (217)$$

соответствующие формулы градиента целевой функции для одной пары обучающих данных (x, d) принимают вид:

$$\frac{\partial E}{\partial p_{ij}} = (y - d) \times \frac{\prod_{j=1}^N \mu_A^{(i)}(x_j)}{\sum_{i=1}^M \left[\prod_{j=1}^N \mu_A^{(i)}(x_j) \right]} \times x_j \quad (218)$$

Алгоритм наискорейшего спуска для сети TSK

$$\frac{\partial E}{\partial c_{ij}} = (y - d) \times \sum_{r=1}^M \left[(p_{r0} + \sum_{j=1}^N p_{rj} x_j) \times \frac{\delta_{ri} \times \sum_{i=1}^M \left(\prod_{j=1}^N \mu_A^{(i)}(x_j) - \prod_{j=1}^N \mu_A^{(r)}(x_j) \right)}{\left(\sum_{i=1}^M \left(\prod_{j=1}^N \mu_A^{(i)}(x_j) \right) \right)^2} \right] \times \quad (219)$$

$$\prod_{j=1, j \neq l}^N \mu_A^{(i)}(x_l) \times (\mu_A^{(i)}(x_j))^2 \times \frac{2b_{ij}}{\sigma_{ij}} \times \frac{(x_j - c_{ij})^{2b_{ij}-1}}{(\sigma_{ij})^{2b_{ij}}},$$

$$\frac{\partial E}{\partial \sigma_{ij}} = (y - d) \times \sum_{r=1}^M \left[(p_{r0} + \sum_{j=1}^N p_{rj} x_j) \times \frac{\delta_{ri} \times \sum_{i=1}^M \left(\prod_{j=1}^N \mu_A^{(i)}(x_j) - \prod_{j=1}^N \mu_A^{(r)}(x_j) \right)}{\left(\sum_{i=1}^M \left(\prod_{j=1}^N \mu_A^{(i)}(x_j) \right) \right)^2} \right] \times \quad (220)$$

$$\prod_{j=1, j \neq l}^N \mu_A^{(i)}(x_l) \times (\mu_A^{(i)}(x_j))^2 \times \frac{2b_{ij}}{\sigma_{ij}} \times \frac{(x_j - c_{ij})^{2b_{ij}}}{(\sigma_{ij})^{2b_{ij}+1}},$$

Алгоритм наискорейшего спуска для сети TSK

$$\frac{\partial E}{\partial b_{ij}} = (y - d) \times \sum_{r=1}^M \left[\left(p_{r0} + \sum_{j=1}^N p_{rj} x_j \right) \times \frac{\delta_{ri} \times \sum_{i=1}^M \left(\prod_{j=1}^N \mu_A^{(i)}(x_j) - \prod_{j=1}^N \mu_A^{(r)}(x_j) \right)}{\left(\sum_{i=1}^M \left(\prod_{j=1}^N \mu_A^{(i)}(x_j) \right) \right)^2} \right] \times \prod_{j=1, j \neq l}^N \mu_A^{(i)}(x_l) \times (-2) \times (\mu_A^{(i)}(x_j))^2 \times \left(\frac{x_j - c_{ij}}{\sigma_{ij}} \right)^{2b_{ij}} \times \ln \left(\frac{x_j - c_{ij}}{\sigma_{ij}} \right), \quad (221)$$

где y – фактический выход сети,

d – ожидаемый выход сети,

x_j – j -я компонента входного вектора,

$\mu_A^{(i)}$ – функция принадлежности i -му нечеткому множеству,

δ_{ri} – символ Кронекера.

Одна из модификаций получила название алгоритма обучения с моментом.

При этом подходе уточнение параметров сети производится по формуле:

$$par_{ij}(t+1) = par_{ij}(t) - \eta \cdot \frac{\partial E(t)}{\partial par_{ij}(t)} + \alpha (par_{ij}(t) - par_{ij}(t-1)) \quad (222)$$

где α - это коэффициент момента, принимающий значения в интервале $[0, 1]$.