

Синергетические базы знаний

Г.Б. Евгеньев

Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия

Аннотация

Современные базы знаний должны в значительной мере соответствовать человеческому мышлению и реальности мира. Синергетические базы знаний создают возможность совместного использования как «жёстких» вычислений, которые требуют точности и единственности решения, так и «мягких» вычислений, допускающих заданную погрешность и неопределённость для конкретной задачи. Предложена методология создания синергетических систем для представления знаний с использованием технологий искусственного интеллекта. Методология основана на методах создания баз знаний и может использоваться для разработки систем проектирования и управления в отраслях промышленности. Предложена модель представления лингвистических переменных. Рассмотрен метод создания нечётких баз знаний и этапы механизма логического вывода. Описан нечёткий вывод на примере механизма Мамдани. Предложена функциональная схема создания систем нечёткого логического вывода на основе структурированного чёткого модуля знаний. Рассмотрен метод создания баз знаний для реализации нейросетевых моделей. Приведён пример базы знаний для обучения нейронных сетей.

Ключевые слова: мягкие вычисления, базы знаний, модуль знаний, интеллектуальные системы, системы проектирования.

Цитирование: Евгеньев, Г.Б. Синергетические базы знаний / Г.Б. Евгеньев // Онтология проектирования. – 2021. – Т.11, №1(39). – С.76-88. – DOI: 10.18287/2223-9537-2021-11-1-76-88.

Введение

Актуальным направлением развития интеллектуальных систем (ИС) является совместное использование методов и средств различных способов представления знаний. В настоящее время широко развиты работы по созданию гибридных систем [1-2]. Гибридизация представляет собой такое объединение методов и технологий, когда различные блоки системы реализуют какой-либо метод решения задач и взаимодействуют между собой. Гибридизация охватывает методы, объединяемые термином «мягкие вычисления» (*softcomputing*), которые включают: нечёткие системы [3, 4], нейронные сети [4-8], генетические алгоритмы [4, 9], и др. [10]. Представляется целесообразным расширить круг объединяемых методов, в который включить методы чётких вычислений «классического» искусственного интеллекта (ИИ) [11].

Синергетические системы ИИ обладают такими основными характеристиками, как множеством неоднородных компонентов и гибких взаимосвязей между компонентами, активностью, автономностью, семиотической природой взаимосвязей, открытостью, высоким эволюционным потенциалом. Архитектура и технологии синергетического ИИ отличаются неоднородностью, гибридность, объединением различных информационных технологий [12].

1 Словарь базы знаний

Для создания синергетической системы ИИ необходимо рассмотреть возможность гибридизации передовых методов теории программирования и классического ИИ [13]. Ядром для построения синергетической системы ИИ являются чёткие знания классического ИИ. Нечёткие знания строятся на базе лингвистических переменных. Для наполнения математи-

ческих знаний семантикой над ними нужно надстроить систему чётких знаний. При этом формальные математические переменные, связанные с переменными словаря базы знаний (БЗ), получают смысловое содержание.

Фундаментом для построения чётких знаний является словарь. Словарь имеет имя и методы, обеспечивающие сортировку и поиск слов, а также импорт слов из текстовых документов. Словарь состоит из слов, каждое из которых обладает именем-идентификатором, общепринятым наименованием и типом (целый, действительный или символьный). Имеются методы добавления и удаления слов, а также определения входимости слов в модули знаний (МЗ). Со словами могут быть связаны ассоциативные списки допустимых значений. Ассоциативные списки, также как и слова, имеют имя-идентификатор, наименование списка и тип значений. С ассоциативными списками связаны методы добавления, удаления, сортировки и поиска списка. Списки состоят из элементов, каждый из которых должен иметь значение и может быть добавлен или удалён.

На рисунке 1 приведён пример фрагмента словаря и ассоциативных списков для построения нечёткой БЗ. На основе словаря строятся МЗ [14]. Каждый МЗ имеет наименование, имя-идентификатор, имя предусловия и версию. С модулем связаны методы добавления, выбора модуля-аналога, трансляции и тестирования модуля, определения входимости МЗ в БЗ и другие модули, а также удаления модуля. МЗ имеет свой словарь, представляющий собой подмножество терминов из словаря БЗ и включающий входные и выходные переменные. Предусловие определяет область определения модуля и содержит набор взаимосвязанных логических выражений.

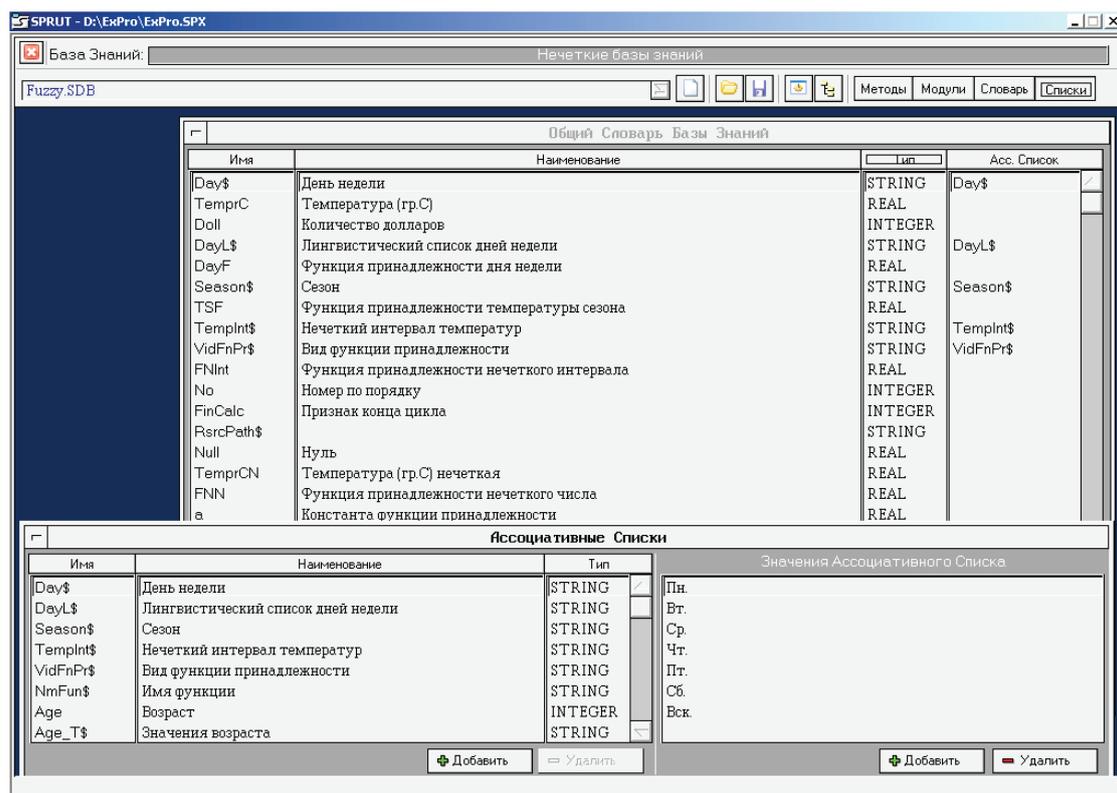


Рисунок 1 - Пример фрагмента словаря для построения нечёткой базы знаний

На рисунке 2 приведён пример модуля расчёта по формуле. Модуль имеет наименование «Расчёт функции принадлежности нечёткого числа по Гауссу» и имя «RsFNNG». Механизм модуля реализует формула $FNN = 2.718^{-(a_*(quant-quantn)^2)}$.

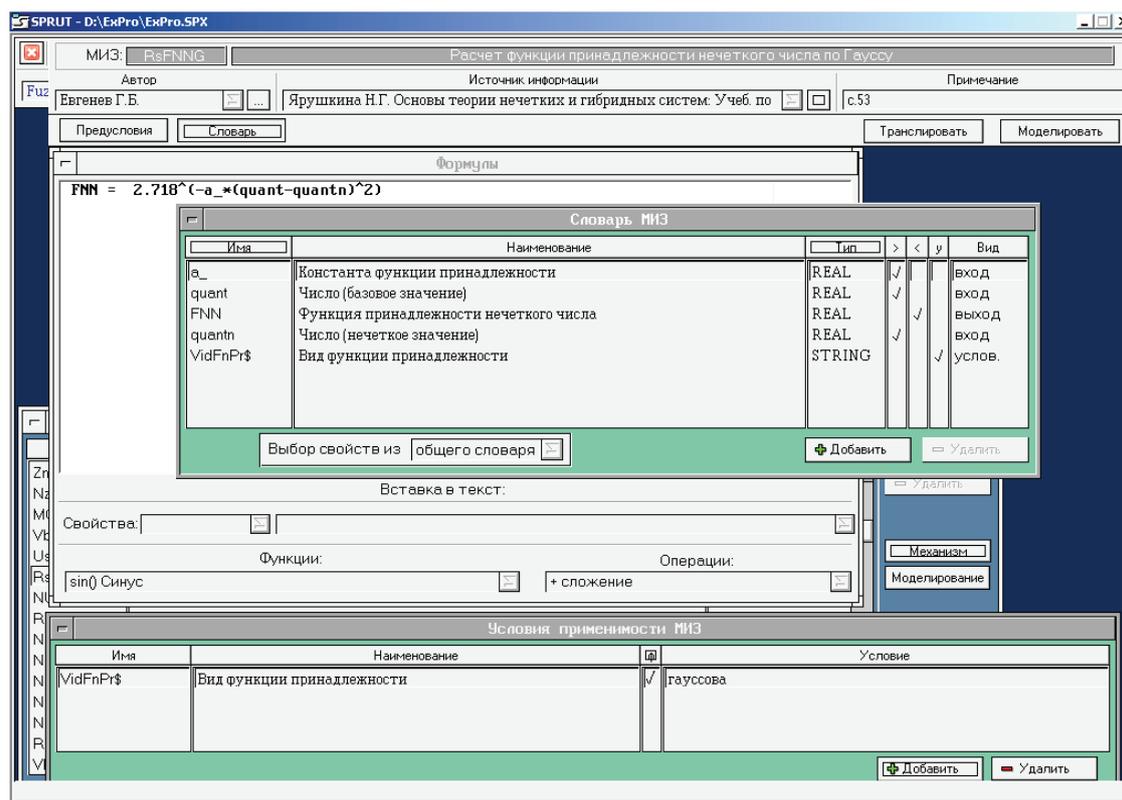


Рисунок 2 - Пример модуля знаний с механизмом вычисления по формуле

Словарь модуля включает входные переменные: a – константа функции принадлежности, $quant$ – число (базовое значение), $quantn$ – число (нечёткое значение), а также выходную переменную FNN – функция принадлежности нечёткого числа и управляющую переменную $VidFnPr\$$ – вид функции принадлежности. Модуль применяется, если вид функции принадлежности имеет значение «функция Гаусса» [14].

2 Лингвистические переменные

Системы нечётких продукций строятся на основе понятия «лингвистическая переменная» [15]. Лингвистической переменной называется пятерка $\{x, T(x), X, G, M\}$, где x - имя переменной; $T(x)$ - множество имён лингвистических значений переменной x , каждое из которых является нечётким множеством на множестве X ; G - синтаксическое правило для образования имён значений x ; M - семантическое правило для ассоциирования каждой величины значения с её понятием.

Например, лингвистическую переменную «возраст человека», можно представить так:

x : «возраст»;

X : множество целых чисел из интервала $[1, 120]$;

$T(X)$: значения «молодой», «зрелый», «старый»;

G : «очень», «не очень» – модификатор возраста (например, «очень молодой», «не очень старый» и пр.);

M : правило, определяющее вид функции принадлежности для каждого значения из множества T .

В структуре чётких знаний лингвистическая переменная представляется в виде структурированного МЗ (см., например, таблицу 1).

Таблица 1 – Пример МЗ лингвистической переменной «возраст человека»

имя	наименование	тип	вид
x_	Переменная x	REAL	вход
Age_G\$	Модификаторы возраста	STRING	выход
Age_T\$	Значения возраста	STRING	выход
FnPr	Функция принадлежности	REAL	выход
a_	Константа функции принадлежности	REAL	локал.
dax	Разность a - x	REAL	локал.
dxb	Разность x - b	REAL	локал.
b_	Константа 2 функции принадлежности	REAL	локал.

В описываемом модуле использована шкала возрастов. Функция принадлежности равна 1: для значения «молодой» – до 30 лет, для значения «зрелый» – от 30 до 60 лет, для значения «старый» – от 60 лет. Модификатор возраста действует при отклонениях от этих значений в пределах 5 лет (например, до 35 лет – не очень молодой).

Например, для значения возраста «молодой» используется функция принадлежности класса L , которая определяется как:

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

Структуры МЗ для расчёта этой функции имеют следующий вид.

МЗ: MRsFPPrL - Метод расчёта функции принадлежности класса L

Предусловия запуска

имя	наименование	тип	условие
Age_T\$	Значения возраста	STRING	молодой

Входные свойства

имя	наименование	тип	значение
x_	Переменная x	REAL	
a_	Константа функции принадлежности	REAL	
b_	Константа 2 функции принадлежности	REAL	

Механизм - Внешний Метод

База Знаний:	текущая База Знаний - Fuzzy.SDB
Метод:	RsFPPrL/Расчет функции принадлежности класса L

Согласование свойств внешнего метода и свойств МЗ

a_	Константа функции принадлежности	= a_	Константа функции принадлежности
b_	Константа 2 функции принадлежности	= b_	Константа 2 функции принадлежности
x_	Переменная x	= x_	Переменная x
FnPr	Функция принадлежности	= FnPr	Функция принадлежности

Выходные свойства

имя	наименование	тип	значение
FnPr	Функция принадлежности	REAL	

Используемые здесь константы в соответствии с принятой шкалой назначаются с помощью модуля.

МЗ: NzPrAge - Назначение параметров возраста

Предусловия запуска

имя	наименование	тип	условие
-----	--------------	-----	---------

Механизм - Формула

a_ = 30, b_ = 60

Выходные свойства

имя	наименование	тип	значение
b_	Константа 2 функции принадлежности	REAL	
a_	Константа функции принадлежности	REAL	

M3: NzAgG - Назначение модификатора возраста**Предусловия запуска**

имя	наименование	тип	условие

Входные свойства

имя	наименование	тип	значение
FnPr	Функция принадлежности	REAL	

Механизм - Таблица

Конфигурация свойств в таблице

FnPr	Age_G\$
------	---------

Таблица

1	очень
[0,1)	не очень

Выходные свойства

имя	наименование	тип	значение
Age_G\$	Модификаторы возраста	STRING	

3 Нечёткие базы знаний

Основой для проведения операции нечёткого логического вывода является база правил, содержащая нечёткие высказывания в форме «ЕСЛИ-ТО», и функции принадлежности для соответствующих лингвистических переменных (термов). При этом должны соблюдаться следующие условия.

- 1) Существует хотя бы одно правило для каждого лингвистического термина выходной переменной.
- 2) Для любого термина входной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве предпосылки (левая часть правила).

В противном случае имеет место неполная база нечётких правил.

Пусть в базе правил имеется m правил вида:

R_1 : ЕСЛИ x_1 это A_{11} ... И ... x_n это A_{1n} , ТО y это B_1

...

R_i : ЕСЛИ x_1 это A_{i1} ... И ... x_n это A_{in} , ТО y это B_i

...

R_m : ЕСЛИ x_1 это A_{m1} ... И ... x_n это A_{mn} , ТО y это B_m ,

где $x_k, k=1..n$ – входные переменные; y – выходная переменная; A_{ik} – заданные нечёткие множества с функциями принадлежности.

Результатом нечёткого вывода является чёткое значение переменной y^* на основе заданных чётких значений x_k .

В общем случае механизм логического вывода включает четыре этапа: введение нечёткости (фазификация), нечёткий вывод, композиция и приведение к чёткости, или дефазификация. Алгоритмы нечёткого вывода различаются главным образом видом используемых правил, логических операций и разновидностью метода дефазификации.

Нечёткий вывод приведён на примере механизма Мамдани (Mamdani) [1,2]. Это наиболее распространённый способ логического вывода в нечётких системах. В нём используется ми-

максимальная композиция нечётких множеств. Данный механизм включает в себя: процедуру фазификации; нечёткий вывод; композицию, использующую максимальную композицию нечётких множеств; дефазификацию, или приведение к чёткости. Используются несколько методов дефазификации, например, метод среднего центра, или центроидный метод.

Пример применения нечёткой БЗ в задаче определения времени стирки на машине.

На входе БЗ имеются чёткие переменные: масса белья в килограммах (*Mass*) со множеством чисел из интервала от 0 до 5 и температура воды в градусах (*Tv*) со множеством значений от 0 до 90. На выходе определяется чёткая переменная - время стирки в минутах (*Dlt*) со множеством значений от 0 до 150. С массой белья связана лингвистическая переменная «количество белья» со значениями «много» и «мало». С температурой связана лингвистическая переменная «температура воды» со значениями «высокая» и «низкая». Со временем стирки связана лингвистическая переменная «длительность», имеющая значения «высокая» и «малая».

БЗ состоит из двух правил:

R₁: ЕСЛИ <количество_белья> = <много> И
<температура_воды> = <высокая>, ТО
<длительность> = <высокая>

R₂: ЕСЛИ <количество_белья> = <мало> И
<температура_воды> = <низкая>, ТО
<длительность> = <малая>.

Метод фазификации переменных осуществляется с помощью соответствующих МЗ.

МЗ Выполнение нечёткого вывода для стиральной машины

ранг	имя	наименование
0	<u>NzImFn</u>	Назначение имени функции
1	<u>NzZnPr22</u>	Назначение значения низкая переменной температура воды
1	<u>NzZnPr21</u>	Назначение значения мало переменной количество белья
1	<u>NzZnPr11</u>	Назначение значения много переменной количество белья
1	<u>NzZnPr12</u>	Назначение значения высокая переменной температура воды
2	<u>MFuzif21</u>	Метод фазификации 1 переменной 2 правила
2	<u>MFuzif22</u>	Метод фазификации 2 переменной 2 правила
2	<u>MFuzif12</u>	Метод фазификации 2 переменной 1 правила
2	<u>VpR1</u>	Выполнение правила R1
2	<u>VpR2</u>	Выполнение правила R2
2	<u>MFuzif11</u>	Метод фазификации 1 переменной 1 правила
3	<u>MRsTnR2</u>	Метод расчёта Т-нормы для правила R2
3	<u>MRsTnR1</u>	Метод расчёта Т-нормы для правила R1
4	<u>RsDlml</u>	Расчёт малой длительности
4	<u>RsDlvs</u>	Расчёт высокой длительности

Входные свойства

имя	наименование	тип	значение
<u>Mass</u>	Масса, кг	REAL	0.7
<u>Tv</u>	Температура, град	REAL	20.1

Выходные свойства

имя	наименование	тип	значение
<u>Dlt</u>	Длительность, мин	REAL	37.5

4 Нейросетевые модели знаний

Для построения ИС традиционно применяются два основных подхода. Первый основан на правилах. При этом подходе необходимо заранее знать весь набор закономерностей, описывающих предметную область. Он детально описан в [14]. При использовании другого подхода, основанного на примерах, надо иметь достаточное количество примеров для настройки адаптивной системы с заданной степенью достоверности. Нейронные сети (НС) представляют собой пример такого подхода. С точки зрения экспертного программирования НС может рассматриваться как механизм одного из правил [13]. При построении БЗ такое правило может использоваться совместно со всеми другими типами правил. Комбинированный подход обеспечивает получение правил на основе примеров, сохранённых в базе данных (БД). Это – извлечение знаний из данных.

Искусственные нейронные сети НС предназначены для решения широкого круга задач: принятия решений, распознавания образов, идентификации, прогнозирования, оптимизации, управления сложными объектами. Применительно к рассматриваемой тематике наибольший интерес представляет применение НС для принятия решений и управления. Эта задача близка к задаче классификации. Классификации подлежат ситуации, характеристики которых поступают на вход НС. На выходе сети при этом должен появиться признак решения, которое она приняла. При этом в качестве входных сигналов используются различные критерии описания состояния управляемой системы.

Формальным нейроном называется элемент, используемый в узлах НС. Ему может быть поставлена в соответствие структурная схема, представленная на рисунке 3.

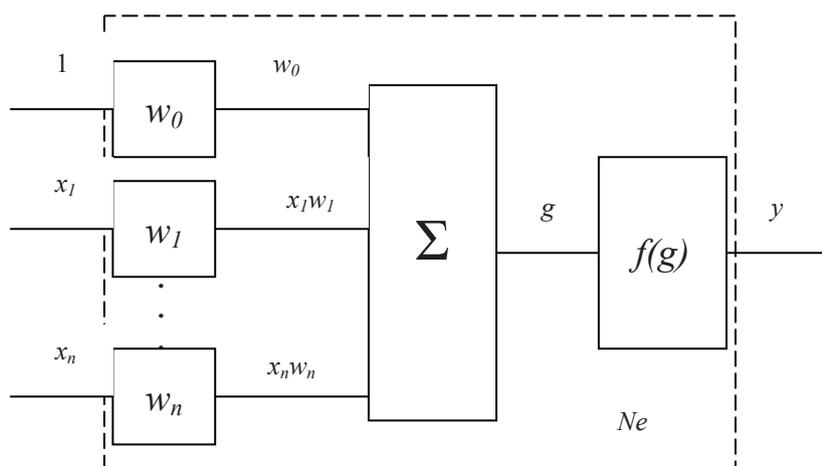


Рисунок 3 - Структурная схема формального нейрона

Схема формального нейрона включает n входных блоков умножения на коэффициенты w_i , один сумматор и выходной блок функционального преобразования. Функция, которую реализует выходной блок, получила название функции активации. Коэффициенты w_i получили название синаптических коэффициентов или коэффициентов межнейронной связи.

Формальный нейрон может быть представлен с помощью двух МЗ. Первый реализует входные блоки и сумматор, а второй – функцию активации. В качестве простого примера модуля, осуществляющего расчёт взвешенной суммы входов нейрона, приведено внешнее представление МЗ RSNeur11. Модуль не имеет предусловия запуска. Такой модуль запускается на исполнение, когда будут определены все входные свойства. В качестве входных свойств выступают входные переменные и синаптические коэффициенты. Выходом модуля является взвешенная сумма входов нейрона, которая вычисляется по формуле.

МЗ: RSNeur11 - Расчёт взвешенной суммы входов нейрона 1 слоя 1**Предусловия запуска**

имя	наименование	тип	условие
-----	--------------	-----	---------

Входные свойства

имя	наименование	тип	значение
w14	Вес связи 14	REAL	
x_2	Входная переменная 2	REAL	
x_0	Входная переменная 0	REAL	
x_3	Входная переменная 3	REAL	
x_8	Входная переменная 8	REAL	
x_9	Входная переменная 9	REAL	
w11	Вес связи 11	REAL	
w13	Вес связи 13	REAL	
w12	Вес связи 12	REAL	
x_1	Входная переменная 1	REAL	
w16	Вес связи 16	REAL	
w17	Вес связи 17	REAL	
w15	Вес связи 15	REAL	
w19	Вес связи 19	REAL	
w18	Вес связи 18	REAL	
x_6	Входная переменная 6	REAL	
x_4	Входная переменная 4	REAL	
x_5	Входная переменная 5	REAL	
x_7	Входная переменная 7	REAL	

Механизм - Формула

$$s11 = x_{0+x_1} * w11 + x_2 * w12 + x_3 * w13 + x_4 * w14 + x_5 * w15 + x_6 * w16 + x_7 * w17 + x_8 * w18 + x_9 * w19$$

Выходные свойства

имя	наименование	тип	значение
s11	Взвешенная сумма входов нейрона 1 слоя 1	REAL	

Вычисление значения функции активации осуществляется модулем RFNeur11. Модуль имеет предусловие, которое определяет вид выбранной функции. В данном случае используется логистическая функция. Входные свойства модуля включают параметр функции активации и взвешенную сумму входов нейрона. Параметр функции активации выбирается из БД со свойствами сети нейронов, а взвешенная сумма входов вычисляется предыдущим МЗ.

МЗ: RFNeur11 - Расчёт функции активации нейрона 1 слоя 1**Предусловия запуска**

имя	наименование	тип	условие
VidAktFn\$	Вид активационной функции	STRING	логистическая

Входные свойства

имя	наименование	тип	значение
a_	Параметр функции активации	REAL	
s11	Взвешенная сумма входов нейрона 1 слоя 1	REAL	

Механизм - Формула

$$y11 = 1 / (1 + 2.71828^{(-a_*s11)})$$

Выходные свойства

имя	наименование	тип	значение
y11	Активационная функция нейрона 1 слоя 1	REAL	

Приведённые выше два МЗ реализуют формальную модель нейрона. С помощью описанной технологии возможно генерировать модели с другими количествами входных переменных и синаптических коэффициентов, а также с другими активационными функциями и с другими моделями нейронов. Хотя один нейрон и способен выполнять простейшие процедуры распознавания, эффективность нейронных вычислений проявляется в нейронных сетях. Простейшая сеть состоит из группы нейронов, образующих слой (см. рисунок 4).

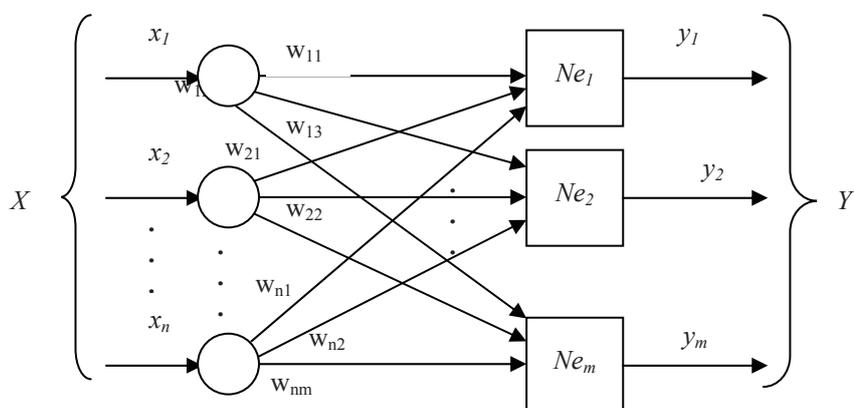


Рисунок 4 - Простейшая однослойная нейронная сеть

Вершины-круги, представленные на рисунке 4, предназначены для распределения входных сигналов. Каждый элемент из множества входов X соединён с каждым искусственным нейроном Ne_i отдельной связью, которой приписан вес (синаптический коэффициент). Удобно считать веса элементами матрицы W . Матрица имеет n строк и m столбцов, где n – число входов, а m – число нейронов.

Для построения продукционной модели НС генерируется метод, состоящий из набора МЗ. Этот метод в дальнейшем может использоваться в качестве модуля в сложных БЗ.

На рисунке 5 представлена функциональная модель НС в соответствии со стандартом *IDEF0*. Функция модели заключается в вычислении компонент y_j вектора Y . Для этого в качестве входных переменных используются компоненты x_i вектора X . Механизмом выполнения этой функции является МЗ и элементы матрицы весов W , которые хранятся в БД. НС характеризуется параметрами: количество входов и выходов, количество слоёв и нейронов в каждом слое, параметры активационной функции.

Ниже приведён пример внешнего представления автоматически сгенерированного метода расчёта простой однослойной НС, включающей три нейрона, десять входных и три выходных переменных.

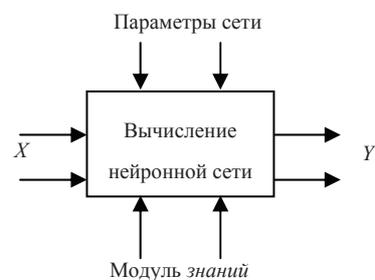


Рисунок 5 - Функциональная модель нейронной сети

МЗ RsNeurNet - Расчёт нейронной сети

ранг	имя	наименование
0	<u>VzPrNet</u>	Вызов параметров сети
0	<u>NUstN</u>	Начальная установка номера входной переменной
0	<u>NUstPer</u>	Начальные установки переменных
1	<u>MVvW</u>	Метод ввода весовых коэффициентов 1-го слоя
2	<u>RSNeur21</u>	Расчёт взвешенной суммы входов нейрона 2 слоя 1

2	RSNeur11	Расчёт взвешенной суммы входов нейрона 1 слоя 1
2	RSNeur31	Расчёт взвешенной суммы входов нейрона 3 слоя 1
3	RFNeur21	Расчёт логистической активационной функции нейрона 2 слоя 1
3	RFNeur11	Расчёт логистической активационной функции нейрона 1 слоя 1
3	RFNeur31	Расчёт логистической активационной функции нейрона 3 слоя 1
4	FrOut11	Формирование выхода однослойной сети с одним нейроном
4	FrOut12	Формирование выхода однослойной сети с двумя нейронами
4	FrOut13	Формирование выхода однослойной сети с тремя нейронами

Входные свойства

имя	наименование	тип	значение
x_10	Входная переменная 10	REAL	
x_8	Входная переменная 08	REAL	
x_7	Входная переменная 07	REAL	
x_4	Входная переменная 04	REAL	
x_6	Входная переменная 06	REAL	
x_5	Входная переменная 05	REAL	
x_9	Входная переменная 09	REAL	
x_2	Входная переменная 02	REAL	
x_3	Входная переменная 03	REAL	
x_1	Входная переменная 01	REAL	

Выходные свойства

имя	наименование	тип	значение
y_1	Выходная переменная 1	REAL	0.
y_2	Выходная переменная 2	REAL	0
y_3	Выходная переменная 3	REAL	0

М3 метода автоматически ранжируются в порядке определения необходимых для их работы переменных. В данном случае в ранге 0 размещаются модуль вызова из БД параметров сети и служебные модули начальных установок переменных. Ранг 1 включает метод ввода весовых коэффициентов, вызываемых из БД.

Ранг 2 состоит из модулей расчёта взвешенных сумм входов всех нейронов, а ранг 3 – из модулей расчёта активационных функций для этих нейронов. В заключительном ранге 4 располагаются модули формирования выходных переменных.

Важнейшим компонентом построения БЗ на основе НС является обучение. В качестве примера построения алгоритма обучения рассмотрен метод обучения, названный δ -правило [5]. Пусть имеется обучающая выборка, состоящая из пар векторов (X^k, Y^k) , $k = 1, 2, \dots, p$. НС называется обученной на данной обучающей выборке, если при подаче на входы сети каждого вектора X^k на выходах всякий раз получается соответствующий вектор Y^k . Обучение заключается в итерационной подстройке матрицы весов W , последовательно уменьшающей ошибку в выходных векторах. Алгоритм включает следующие шаги.

Шаг 0. Проинициализировать элементы весовой матрицы W случайными значениями.

Шаг 1. Подать на входы один из входных векторов X^k и вычислить его выход Y^k .

Шаг 2. Если выход правильный ($Y = Y^k$), перейти на шаг 4. Иначе, вычислить вектор ошибки – разницу между заданным и полученным значениями выхода: $\delta = Y^k - Y$.

Шаг 3. Матрица весов модифицируется по следующей формуле: $w^{t+1}_{ij} = w^t_{ij} + v \cdot \delta \cdot x_i$. Здесь t и $t+1$ – номера соответственно текущей и следующей итераций; v – коэффициент скорости обучения ($0 < v \leq 1$); x_i – i -я компонента входного вектора X^k ; j – номер нейрона в слое.

Шаг 4. Шаги 1–3 повторяются для всех обучающих векторов. Обучение завершается, когда сеть перестаёт ошибаться.

M3 ObNeurNet - Обучение нейронной сети

ранг	имя	наименование
0	<u>NUstN</u>	Начальная установка номера входной переменной
0	<u>NUstNc</u>	Начальная установка номера цикла обучения
0	<u>VzPrNet</u>	Вызов параметров сети
1	<u>PerAdrOb</u>	Переадресация цикла обучения
1	<u>VvNObVb</u>	Ввод номера обучающей выборки
2	<u>MVvXob</u>	Метод ввода входных переменных обучающей выборки
2	<u>MInitW</u>	Метод инициации весов матрицы W
2	<u>MVvYob</u>	Метод ввода выходных переменных обучающей выборки
2	<u>NzFnClOb</u>	Назначение признака конца цикла обучения
3	<u>MModW</u>	Метод модификации матрицы весов

В ранг 0 входят служебные M3, которые не включаются в тело цикла обучения сети. Они осуществляют начальные значения переменных, необходимых для дальнейшего использования и вызов параметров сети, включающих количества входных и выходных переменных, слоёв нейронов, а также параметр функции активации.

В ранг 1 входят M3, осуществляющие переадресацию цикла обучения и ввод номера обучающей выборки.

В ранг 2 входят следующие M3. Структурированный модуль инициации весов матрицы W , выполняющий функции шага 0. Этот модуль выполняется однократно в начале цикла обучения. Два модуля ввода входных и выходных переменных обучающей выборки, выполняющие функции шага 1. Модуль назначения признака конца цикла обучения, выполняющий функции шага 2.

В ранг 3 входит модуль модификации матрицы весов, выполняющий функции шага 3.

Заключение

Базовыми принципами мягких вычислений являются: терпимость к неточности, неопределённости и частичной истинности, робастности, низкой стоимости решения и лучшего согласия с реальностью. Интеграция жёстких вычислений с мягкими может осуществляться посредством создания синергетических БЗ.

В работе представлен подход к созданию синергетических БЗ в формате структурированных M3.

Список источников

- [1] **Батыршин, И.З.** Нечёткие гибридные системы. Теория и практика / И.З. Батыршин, А.О. Недосекин, А.А. Стецко, В.Б. Тарасов, А.В. Язенин, Н.Г. Ярушкина // Под ред. Н.Г. Ярушкиной. – М.: ФИЗМАТЛИТ. 2007. 208 с.
- [2] **Ярушкина, Н.Г.** Основы теории нечётких и гибридных систем / Н.Г. Ярушкина. – М. Финансы и статистика, 2004. 320 с.
- [3] **Штовба, С.Д.** Проектирование нечётких систем средствами MATLAB / С.Д. Штовба. М.: Горячая линия-Телеком. 2007. 288 с.
- [4] **Рутковская, Д.** Нейронные сети, генетические алгоритмы и нечёткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. — М.: Горячая линия-Телеком, 2008. 383 с. ISBN 5-93517-103-1.
- [5] **Рутковский, Л.** Искусственные нейронные сети. Теория и практика / Лешек Рутковский. — М.: Горячая линия-Телеком, 2010. 520 с. ISBN 978-5-9912-0105-6.
- [6] **Усков, А.А.** Интеллектуальные технологии управления. Искусственные нейронные сети и нечёткая логика / А.А. Усков, А.В. Кузьмин. — М.: Горячая Линия-Телеком, 2004. 143 с.

- [7] **Голубев, Ю.Ф.** Нейросетевые методы в мехатронике / Ю.Ф. Голубев. — М.: Изд-во Моск. ун-та, 2007. 157 с. ISBN 978-5-211-05434-9.
- [8] **Хайкин, С.** Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation / С. Хайкин. 2-е изд. — М.: Вильямс, 2006. 1104 с. ISBN 0-13-273350-1.
- [9] **Гладков, Л.А.** Генетические алгоритмы: / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. — 2-е изд. — М.: Физматлит, 2006. 320 с. ISBN 5-9221-0510-8.
- [10] **Ульянов, С.В.** Интеллектуальное робастное управление: технологии мягких вычислений / С.В. Ульянов, Л.В. Литвинцева, В.Н. Добрынин, А.А. Мишин. - М.: ВНИИгеосистем, 2011. 406 с.
- [11] **Hutter, M.** Universal Artificial Intelligence / Marcus Hutter. Berlin: Springer. 2005. ISBN 978-3-540-22139-5.
- [12] **Тарасов, В.Б.** От многоагентных систем к интеллектуальным организациям: философия, психология, информатика / В.Б. Тарасов. — М.: Эдиториал УРСС, 2002. 352 с.
- [13] **Евгениев, Г.Б.** Экспертопедия как средство создания онтологического Интернета знаний / Г.Б. Евгениев // Онтология проектирования. — 2019. — Т.9, №3(33). — С.7-23. DOI: 10.18287/2223-9537-2019-9-1-7-23.
- [14] **Евгениев, Г.Б.** Основы автоматизации технологических процессов и производств. Т. 2: Методы проектирования и управления / Г.Б. Евгениев // Москва, Изд-во МГТУ им. Н.Э. Баумана, 2015. 479 с.
- [15] **Заде, Л.** Понятие лингвистической переменной и его применение к принятию приближённых решений / Л. Заде. — М.: Мир, 1976. 166 с.

Сведения об авторе

Евгениев Георгий Борисович, 1938 г. рождения. Окончил Московский авиационный технологический институт в 1960 г., к.т.н. (1964), д.т.н. (1975). Лауреат премии Совета Министров СССР (1991). Профессор кафедры Компьютерные системы автоматизации производства МГТУ им. Н.Э. Баумана. Член Российской ассоциации искусственного интеллекта. В списке научных трудов более 190 работ в области САПР и ИИ. AuthorID (РИНЦ): 17685. g.evgeniev@mail.ru.



Поступила в редакцию 17.09.2020, после рецензирования 01.02.2021. Принята к публикации 05.03.2021.

Synergetic knowledge bases

G.B. Evgeniev

Moscow State Technical University named after N.E. Bauman, Moscow, Russia

Abstract

Modern knowledge bases must largely correspond to human thinking and the reality of the world. Synergetic knowledge bases create the possibility of joint use of both "hard" computing, which require the accuracy and uniqueness of the solution, and "soft" computing, allowing a given error and uncertainty for a specific problem. A methodology for creating synergetic systems for the representation of knowledge using artificial intelligence technologies is proposed. The methodology is based on knowledge base methods and can be used to develop design and management systems in industries. A model for representing linguistic variables is proposed. The method of creating fuzzy knowledge bases and the stages of the inference mechanism are considered. The fuzzy inference is described using the example of the Mamdani mechanism. A functional diagram of the creation of fuzzy inference systems based on a structured clear knowledge module is proposed. A method for creating knowledge bases for the implementation of neural network models is considered. An example of a knowledge base for training neural networks is given.

Key words: *soft computing, knowledge bases, knowledge module, intelligent systems, design systems.*

Citation: *Evgeniev GB. Synergetic knowledge bases [In Russian]. Ontology of designing. 2021; 11(1): 76-88. DOI: 10.18287/2223-9537-2021-11-1-76-88.*

List of figures and tables

Figure 1 - An example of a dictionary fragment for building a fuzzy knowledge base

Figure 2 - An example of a knowledge module with a formula calculation engine

Figure 3 - Structural diagram of a formal neuron

Figure 4 - The simplest single layer neural network

Figure 5 - Functional neural network model

Table 1 - An example of a knowledge module of the linguistic variable "age of a person"

References

- [1] **Batyrshin IZ, Nedosekin AO, Stetsko AA, Tarasov VB, Yazenin AV, Yarushkina NG.** Fuzzy hybrid systems. Theory and Practice [In Russian]. Ed. N.G. Yarushkina. Moscow: FIZMATLIT. 2007. 208 p.
- [2] **Yarushkina NG.** Fundamentals of the theory of fuzzy and hybrid systems [In Russian]. Moscow. Finance and Statistics, 2004. 320 p.
- [3] **Shtovba SD.** Design of fuzzy systems by means of MATLAB [In Russian]. Moscow: Hotline - Telecom. 2007. 288 p.
- [4] **Rutkovskaya D, Pilinsky M, Rutkovsky L.** Neural networks, genetic algorithms and fuzzy systems [In Russian]. Moscow: Hotline-Telecom, 2008. 383 p. ISBN 5-93517-103-1.
- [5] **Rutkovsky Leszek.** Artificial neural networks. Theory and practice [In Russian]. Moscow: Hotline-Telecom, 2010. 520 p. ISBN 978-5-9912-0105-6.
- [6] **Uskov AA, Kuzmin AV.** Intellectual control technologies. Artificial neural networks and fuzzy logic [In Russian]. Moscow: Hotline-Telecom, 2004. 143 p.
- [7] **GolubevYuF.** Neural network methods in mechatronics [In Russian]. Moscow: Publishing house of Moscow University, 2007. 157 p. ISBN 978-5-211-05434-9.
- [8] **Haykin S.** Neural Networks: A Complete Course = Neural Networks: A Comprehensive Foundation [In Russian]. 2nd ed. Moscow: Williams, 2006. 1104 p. ISBN 0-13-273350-1.
- [9] **Gladkov LA, Kureichik VV, Kureichik VM.** Genetic algorithms [In Russian]. 2nd ed. Moscow: Fizmatlit, 2006. 320 p. ISBN 5-9221-0510-8.
- [10] **Ulyanov SV, Litvintseva LV, Dobrynin VN, Mishin AA.** Intelligent robust control: soft computing technologies [In Russian]. Moscow: VNIIGeosystem, 2011. 406 p.
- [11] **Hutter Marcus.** Universal Artificial Intelligence. Berlin: Springer. 2005, ISBN 978-3-540-22139-5.
- [12] **Tarasov VB.** From multi-agent systems to intelligent organizations: philosophy, psychology, informatics [In Russian]. Moscow: Editorial URSS, 2002. 352 p.
- [13] **Evgenev GB.** Expertopedia as a means of creating an ontological Internet of knowledge [In Russian]. *Ontology of designing*. 2019; 9(3): 7-23. DOI: 10.18287/2223-9537-2019-9-1-7-23.
- [14] **Evgenev GB,** ed. Basics of automation of technological processes and production. T.2: Methods of design and management [In Russian]. Moscow, Publishing house of MSTU named after N.E. Bauman, 2015. 479 p.
- [15] **Zade L.** The concept of a linguistic variable and its application to making approximate decisions [In Russian]. Moscow: Mir, 1976. 166 p.

About the author

Georgy Borisovich Evgenev (b. 1938) graduated from Moscow aviation technological Institute in 1960, PhD (1964), Doctor of engineering (1975). He is the winner of the Prize of the Council of Ministers of the USSR (1991). He is a Professor at BMSTU and a member of the Russian Association for Artificial Intelligence. He is a co-author of about 150 scientific articles and abstracts in the field of CAD and AI. Author ID (RSCI): 17685. g.evgenev@mail.ru.

Received September 17, 2020. Revised February 1, 2021. Accepted March 5, 2021.
