

СРАВНЕНИЕ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ НЕЙРО-НЕЧЕТКОЙ СЕТИ В MATLAB

Ханевский А.А.

*Ханевский Алексей Александрович – магистр,
Специальность: системный анализ, обработка и управления информации,
Белорусско–Российский университет, г. Могилев*

Аннотация: исследуя проблему моделирования объектов при помощи нейронных сетей, сталкиваешься с основной проблемой – проверкой результатов полученной модели. Предложена одна задача, которая будет решена разными путями, чтобы узнать есть ли связь результатов, полученных при помощи обучения нейро-нечеткой сети ANFIS и выбора различных функций принадлежности.

Ключевые слова: нейро–нечеткие сети, matlab, анализ данных, машинное обучение, нейроны, ANFIS.

Попытка использовать неточную информацию в математических моделях привела к разработке методов нечеткого моделирования. Напомним, что математические модели манипулируют переменными. В традиционных моделях значения рассматриваемой переменной берутся из некоторого набора чисел, называемого универсумом. Традиционные математические модели манипулируют непосредственно своими элементами. Однако в нечеткой модели переменные могут представлять собой нечеткие подмножества универсума. Следовательно, нечеткие модели требуют разбиения универсума на части, для которых специфично, что они не должны быть точно сформированы и могут пересекаться.

Нечеткое моделирование представляет собой группу специальных математических методов, которые позволяют включать в модель неточную или неопределенно сформулированную экспертную информацию, которая часто характеризуется естественным языком. Развитые модели (мы называем их нечеткими моделями) очень успешны, потому что они обеспечивают решение в различных ситуациях, когда традиционные математические модели терпят неудачу - либо из-за их неадекватности, либо из-за их неспособности использовать полную доступную информацию.

Обратите внимание, что идея включения неточной информации в наши модели противоречит тому, что всегда требуется: насколько возможно высокая точность. Однако есть веская причина для этого, а именно, мы сталкиваемся с расхождением между актуальностью и точностью. Так называемый принцип несовместимости, сформулированный Л.А. Заде, гласит следующее:

По мере усложнения системы наша способность делать абсолютные, точные и значимые утверждения о поведении системы уменьшается. В какой-то момент будет возможен обмен между точностью и релевантностью. Повышение точности может быть достигнуто только за счет уменьшения значимости; повышение значимости может быть достигнуто только за счет снижения точности.

С другой стороны, правила IF-THEN, используемые людьми почти всегда расплывчаты. Причина в том, что они содержат неопределенные выражения естественного языка, которые являются центральными для человеческого мышления.

ANFIS - это аббревиатура Adaptive-Network-Based Fuzzy Inference System - адаптивная сеть нечеткого вывода. Она была предложена Янгом (Jang) в начале девяностых [1]. ANFIS является одним из первых вариантов гибридных нейро-нечетких сетей - нейронной сети прямого распространения сигнала особого типа. Архитектура нейро-нечеткой сети изоморфна нечеткой базе знаний. В нейро-нечетких сетях используются дифференцируемые реализации треугольных норм (умножение и вероятностное ИЛИ), а также гладкие функции принадлежности. Это позволяет применять для настройки нейро-нечетких сетей быстрые алгоритмы обучения нейронных сетей, основанные на методе обратного распространения ошибки. Ниже описываются архитектура и правила функционирования каждого слоя ANFIS-сети.

ANFIS реализует систему нечеткого вывода Сугено в виде пятислойной нейронной сети прямого распространения сигнала. Назначение слоев следующее:

- первый слой - термы входных переменных;
- второй слой - антецеденты (посылки) нечетких правил;
- третий слой - нормализация степеней выполнения правил;
- четвертый слой - заключения правил;
- пятый слой - агрегирование результата, полученного по различным правилам.

Входы сети в отдельный слой не выделяются. На рисунке 1 изображена ANFIS-сеть с двумя входными переменными (x_1 и x_2) и четырьмя нечеткими правилами. Для лингвистической оценки входной переменной x_1 используется 3 терма, для переменной x_2 - 2 терма.

Для анализа базового алгоритма используем среду MATLAB с пакетом Fuzzy Logic Toolbox также гибридные сети реализованы в форме так называемой адаптивной системы нейро-нечеткого вывода или ANFIS.

Основным этапом создания такой сети является база правил IF-THEN, которая создается внутри нейронной сети, что делает её приближенной к человеку. Инструментом в нечетком моделировании являются нечеткие правила IF-THEN. Это специальные выражения, которые характеризуют отношения между частями двух или более универсумов. Например, рассмотрим электрический котел и два универсума: значения электрического тока (А) и температуры (°C). Тогда следующее типичное нечеткое правило IF-THEN:

R: если электрический ток очень сильный тогда температура высокая.

Для написания алгоритма нейро-нечеткой сети понадобится математическая модель нейрона человеческого мозга. С использованием данной модели можем описать, как взаимодействуют входные данные и что получаем на выходе.

Начало происходит с входных данных, это параметры в зависимости от самой задачи, потом данные параметры зависят от весов, которые зависят от конкретной задачи, после чего они проходят скрытые слои или hidden-layer, где суммируются как, а после проходят через функцию принадлежности и благодаря данной функции получаем нужный нам результат.

Функций принадлежности существует достаточно много для получения различных результатов, они выбираются в зависимости от задачи. Функции принадлежности бывают линейные и нелинейные.

Классификация данных функций: линейная или нелинейная.

К линейным относятся: кусочно-линейные, треугольные, трапециевидальные.

К нелинейным относятся: полиномиальные, сигмоидальные, гауссовские.

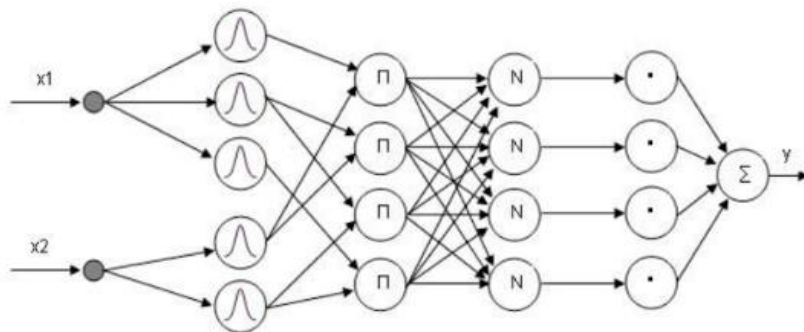


Рис. 1. Пример ANFIS сети

Для работы используем нечеткую сеть TSK. Эта сеть состоит из M правил и N переменных для x_i , представляются в виде.

Правило Мамдани:

ЕСЛИ (x_1 есть A_1) И ... И (x_n есть A_n) ТО ($y = c_0$)

Правило Такаги-Сугено:

ЕСЛИ (x_1 есть A_1) И ... И (x_n есть A_n) ТО ($y = a_0 + a_1x_1 + \dots + a_nx_n$).

ИЛИ THEN $y = a_0 + \sum a_i x_i$.

Условия IF (x_i IS A_i) реализуется функцией фаззификации, которая представлена Гауссовской обобщенной функцией для каждой переменной x_i :

$$\mu_a(x_i) = \frac{1}{1 + \left(\frac{x_i - c_i}{\sigma_i}\right)^{2b_i}} \quad (1)$$

где $\mu(x)$ это оператор A_i . В нечетких сетях условия задаются в алгебраическом произведении, из этого появляются правила k-ого вывода.

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

При M правилах вывода выходного результата сети происходит по формуле:

$$y = \sum_{i=1}^M \frac{w_i}{\sum_{j=1}^N w_j} (p_{i0} + \sum_{j=1}^N p_{ij}x_j) \quad (3)$$

Которую можно представить в виде :

$$y(x) = \frac{1}{\sum_{k=1}^N w_k} \sum_{k=1}^M w_k y_k(x)$$

$$y_k(x) = p_{i0} + \sum_{j=1}^N p_{kj}x_j \quad (4)$$

Как видим в формуле (4;3) присутствуют веса w_k которые определяются по формуле (1) можно увидеть как идут слоя в нечеткой модели нейронной сети (Рисунок 2)

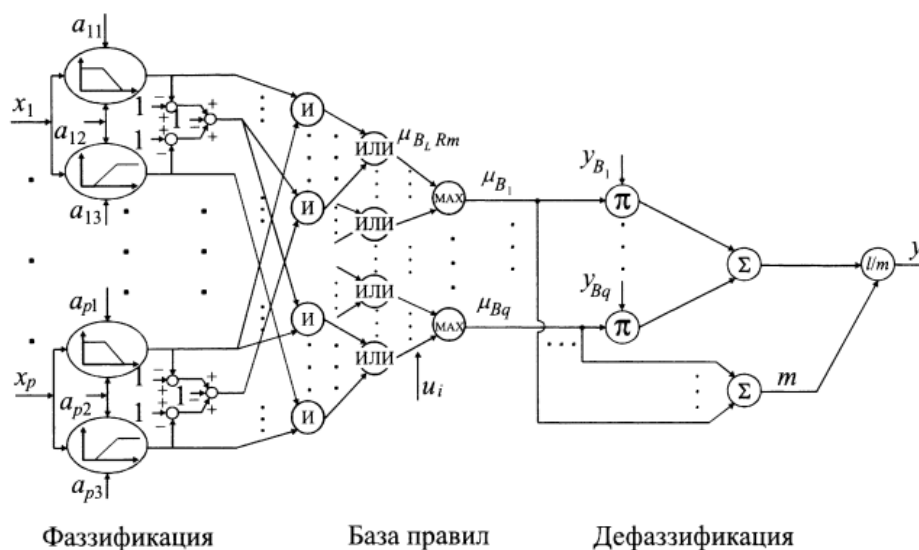


Рис. 2. Нейронная сеть для представления нечеткой модели

Условия $IF (x_i \text{ IS } A_i)$ реализуется функцией фаззификации, которая представлена Сигмоидной функцией для каждой переменной x_i :

$$\mu_a(x_i) = \frac{1}{1 + (e)^{-a(x_i - c_i)}} \quad (5)$$

Гибридная сеть представляет собой многослойную сеть без обратной связи, в которой используются обычные сигналы, веса, функции принадлежности, а суммирование проходит на основе Т-нормы, значения весов данной сети представляют вещественные числа из отрезка от $[0, 1]$.

Идея на которой построены гибридные сети, заключается в использовании имеющихся входных данных для определения параметров функции принадлежности, которые соответствуют системе нечеткого вывода. Для нахождения параметров функций принадлежности используют обучение нейронных сетей.

В стандартном пакете MATLAB имеется пакет Fuzzy Logic Toolbox – это нечеткие сети реализованные в форме системы нейро – нечеткого вывода.

Редактор ANFIS облегчает работу создания и реализацию конкретной модели системы нейро – нечеткого вывода, производить обучение в котором сами можем корректировать число эпох, визуализировать ее структуру, дополнять параметры и использовать определенные правила.

Для реализации описанной модели используется среда MATLAB 2014b. Программные возможности среды MATLAB достаточно обширны и позволяют без труда реализовывать задачи любой сложности, а также создавать дополнительные условия и улучшать программу благодаря большой библиотеки и дополнительных средств, которые позволяют облегчить задачи построения.

Входными данными являются входная матрица данных со значениями в таблице 2. В качестве входных данных была решена задача на лобовое сопротивление. Лобовое сопротивление – это сила, которая направлена против движения объекта в различных средах (например газ, жидкость). Сила сопротивления всегда направлена против вектора скорости объекта в конкретной среде. Лобовое сопротивление является одной из причин обледенения летательных аппаратов (из-за низких температур в слоях атмосферы), а также вызвать нагрев лобной части объекта. Сила лобового сопротивления направлена против вектора скорости движения, её величина пропорциональна характерной площади S , плотности среды ρ и квадрату скорости V :

$$X_0 = C_{x0} \frac{\rho V^2}{2} S \quad (6)$$

$$C_{x0} = \frac{2F}{\rho v^2 S} \quad (7)$$

где X_0 – сила сопротивления,

C_{x0} – безразмерный коэффициент сопротивления формы (КСФ),

F – сила Ньютона полученная экспериментально,

ρ – плотность среды,

v – скорость потока,

S – характерная площадь перпендикулярно потоку.

При помощи КСФ можно точно вычислить силу сопротивления динамическому напору среду тел любой формы, геометрии, для любой скорости.

Таблица 1. Формы при определенных значениях КСФ

Форма	C_{x0}
Сфера	0,47
Куб	1,05
Цилиндр	0,8
Каплевидная тело	0,05

Определение площади зависит от формы тела.

Так же для преодоления силы лобового сопротивления требуется определенная мощность:

$$P = X_0 V = C_{x0} \frac{\rho V^3}{2} S \quad (7)$$

Выходными данными являются промежуточные значения алгоритма и полученные при обучении сети, составленной различными правилами IF THEN. Полученные значения выводятся на график.

Логический вывод можно понимать как процесс, в ходе которого мы получаем некоторые факты из других известных фактов, используя специальные правила, которые подражают правилам человеческого мышления (рассуждения). Они называются правилами вычета. Факты - это некоторые утверждения о реальности, сформулированные на определенном формальном языке. Эти утверждения называются формулами или предложениями.

Для запуска ANFIS редактора используется команда anfisedit. В результате появится на экране графическое окно (Рисунок 3), так же здесь находятся различные другие области Load Data позволяет загрузить данные в виде файла .dat. Далее используем область генерирования исходной FIS там выбираем функцию Гаусса (Рисунок 4). После генерирования выбираем метод обучения нашей нейронной сети, так как у нас используется гибридная сеть, то выбираем в поле метод свойство hybrid. В области графика выводится два типа графиков: экспериментальные данные, результаты моделирования. Они представляют собой результат скопления множества точек. Данная программа использует свои обозначения в поле визуализации:

- Голубая точка – тестируемая выборка.
- Голубая окружность – обучающая выборка.
- Голубой плюс – контрольная выборка.
- Красная звездочка – результаты моделирования.

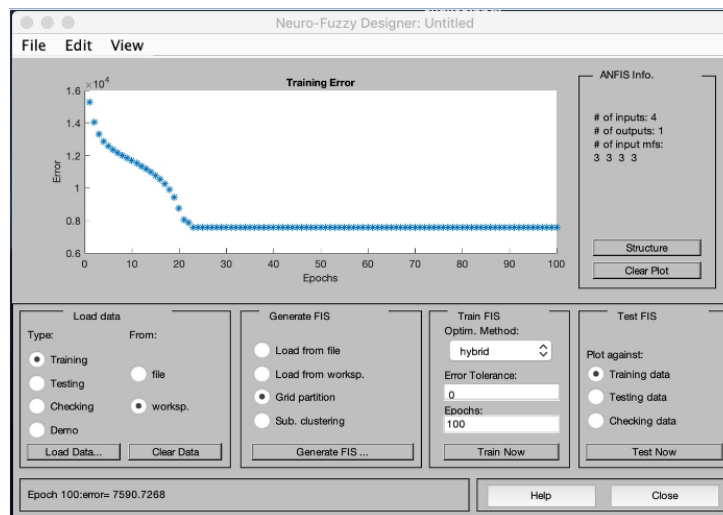


Рис. 3. Окно редактора ANFIS

Таблица 2. Матрица входных данных для обучения нейро-нечеткой сети

Сила	Скорость	КСФ	Лобовое сопротивление	Мощность
1000	11	0,526132	11000	121000
1000	12	0,442097	12000	144000
1000	13	0,376698	13000	169000
1000	14	0,324806	14000	196000
1000	15	0,282942	15000	225000

1000	16	0,24868	16000	256000
1000	17	0,220284	17000	289000
1000	18	0,196488	18000	324000
1000	19	0,176349	19000	361000
1000	20	0,159155	20000	400000
1000	21	0,144358	21000	441000
1000	22	0,131533	22000	484000
1000	23	0,120344	23000	529000
1000	24	0,110524	24000	576000
1000	25	0,101859	25000	625000
1000	26	0,094175	26000	676000
1000	27	0,087328	27000	729000
1000	28	0,081202	28000	784000
1000	29	0,075698	29000	841000
1000	30	0,070736	30000	900000
...	...	...	...	...
1099	99	0,007139	108801	10771299

Рис. 4. Окно для задания количества и выбора функции принадлежности

Обработка ошибок

При работе программы может возникнуть ошибка при вводе неверных данных, в этом случае выводится сообщение об ошибке (см. Рисунок 5).



Рис. 5. Сообщение об ошибке ввода

Все остальные параметры алгоритма не могут быть заданы пользователем некорректно, так как программа не будет работать, пока в полях ввода не будут верные данные.

Для рассмотрения примера система нечеткого вывода содержит 5 входов со своими данными, после чего идут от каждого входа три термы в каждой, 81 правило содержится в следующем слое, которое направлено на один выход, такой результат мы получили при загрузке данных (Рисунок 6).

Для обучения сети нужно воспользоваться вводом параметров, выбор метода обратное распределение или гибридный, метод наименьших квадратов и убывающего градиента.

Количество эпох или циклов обучения 100. Для обучения сети нужно нажать кнопку Train now. После нажатия обучение будет иллюстрироваться в виде графика.

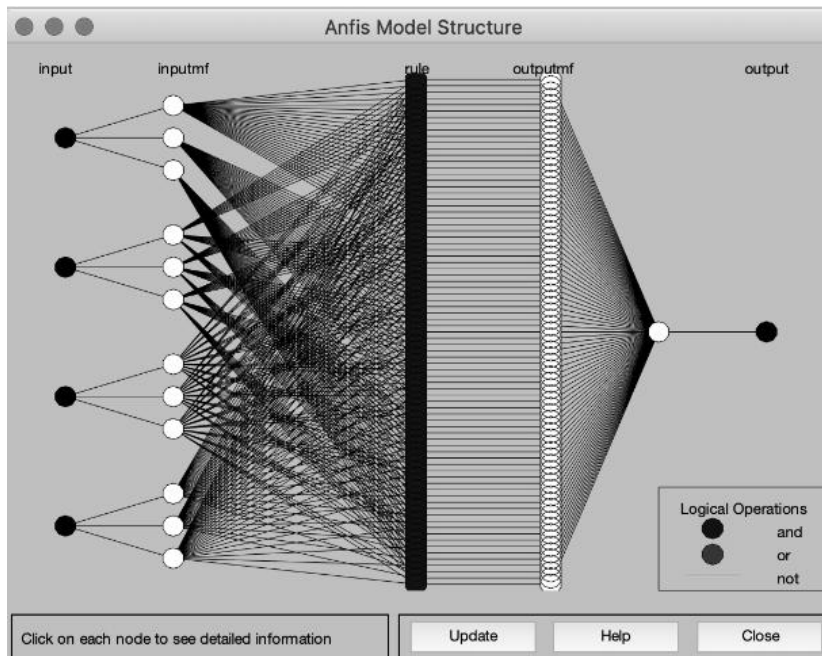


Рис. 6. Структура сгенерированной нейро-нечеткой сети

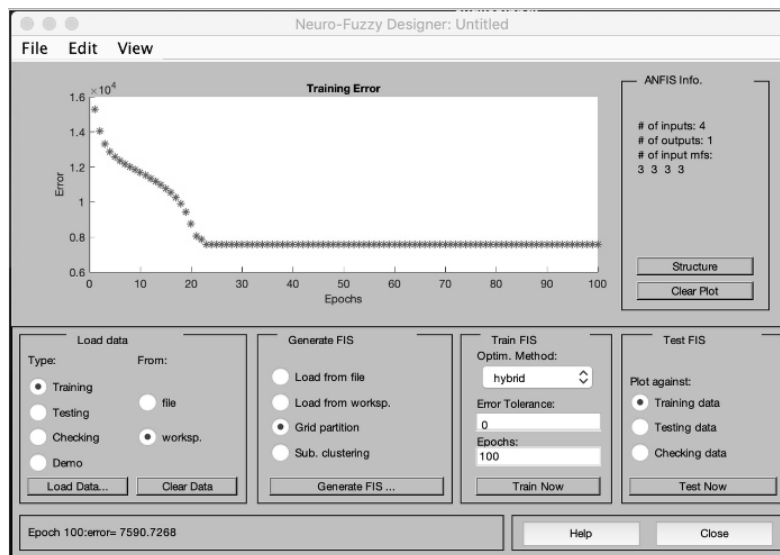


Рис. 7. Процесс обучения сети 100 эпох

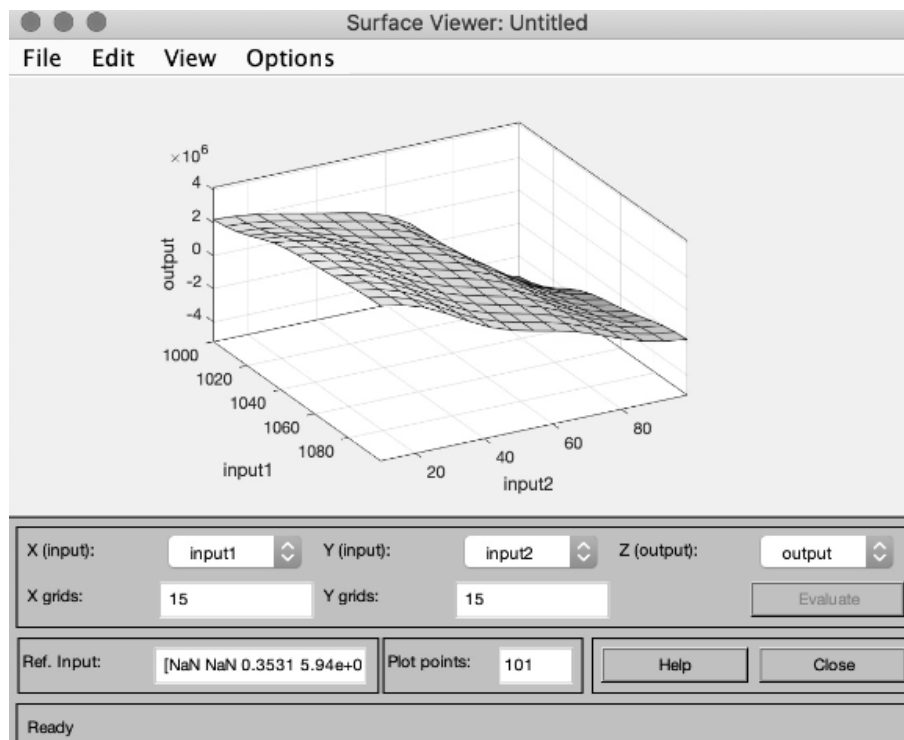


Рис. 8. График полученных результатов

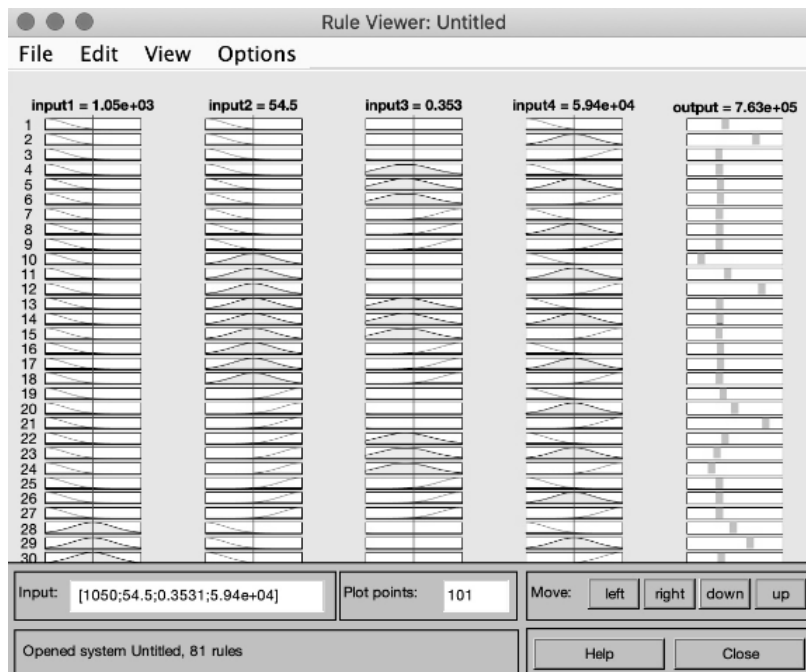


Рис. 9. Полученные результаты - сгенерированные правила системы нечеткого вывода

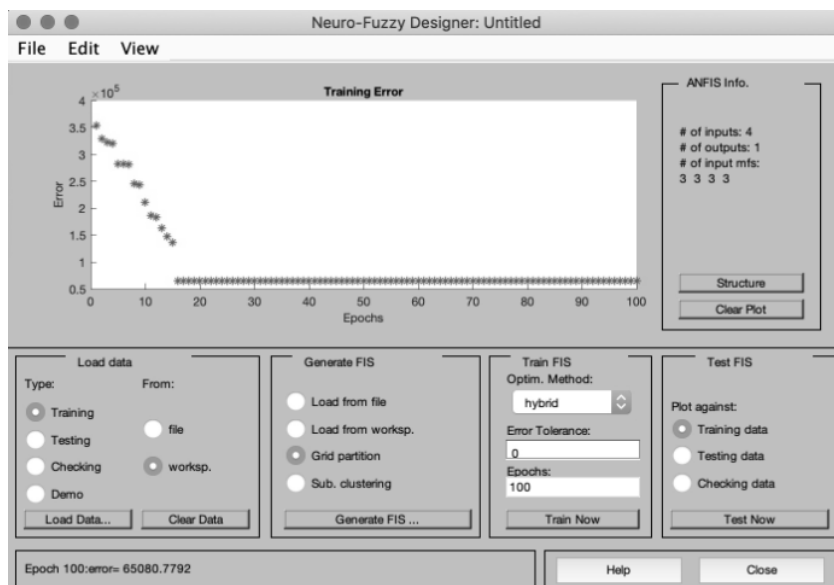


Рис. 10. Процесс обучения сети 100 эпох

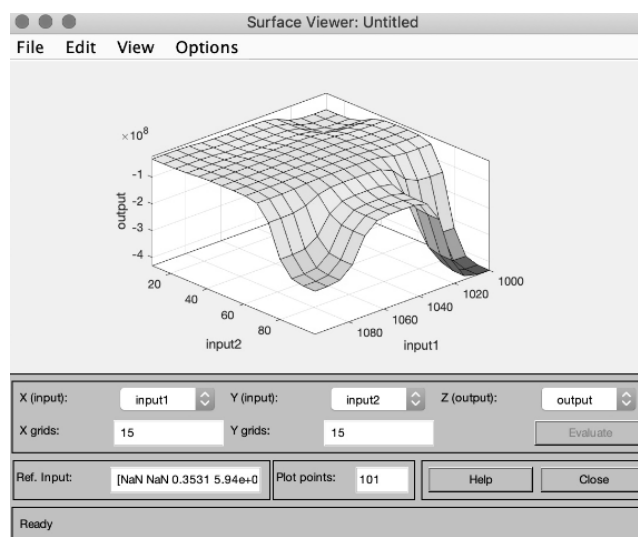


Рис. 11. График полученных результатов при помощи dsigmf

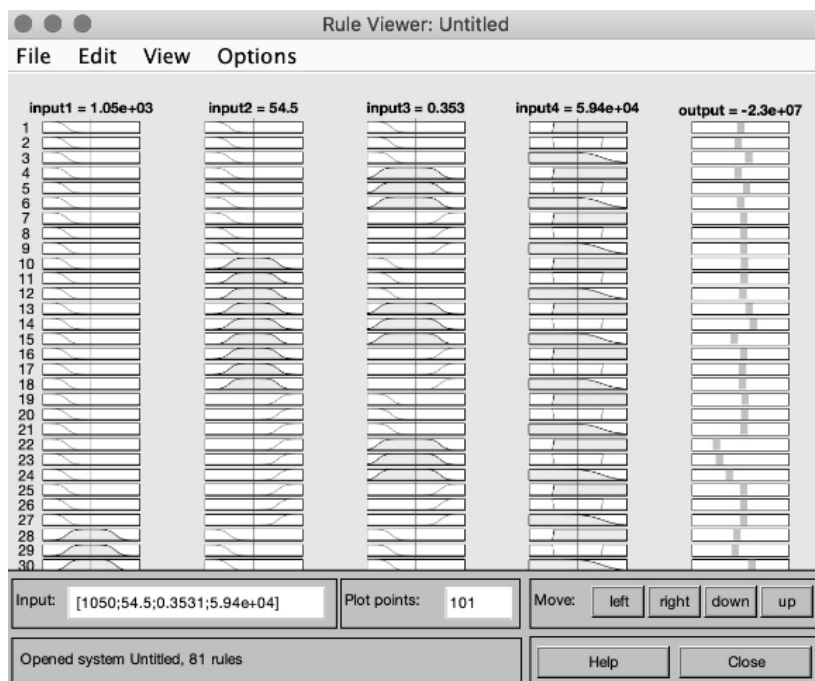


Рис. 12. Полученные результаты нечеткого вывода с помощью dsigmf

В результате для сравнения результатов, полученных при помощи различных функций принадлежности, делаем выводы.

Таблица 3. Сравнение результатов различных функций принадлежности

Функция	Сила	Скорость	КСФ	Лобовое сопротивление	Мощность
Dsigmf	1050	54,5	0,3531	59400	2300000
Gauss2mf	1050	54,5	0,3531	59400	5240000
gaussmf	1050	54,5	0,3531	59400	763000
Pimf	1050	54,5	0,3531	59400	9480000
Psigmf	1050	54,5	0,3531	59400	2100000
Trapmf	1050	54,5	0,3531	59400	3040000
Trimf	1050	54,5	0,3531	59400	660000
Данные	1050	54	0,0229	56700	3061800

Использованы методы моделирования и принципы функционирования нейро–нечетких сетей, в том числе при решении задачи, построенные на прогнозировании, а также приобретены навыки в работе с MATLAB. Данная программа MATLAB позволяет быстро смоделировать нейро-нечеткую сеть, структура которой сгенерируется с получением входных данных на входе, а также программа сама генерирует набор нечетких правил. Так же можно быстро получить результаты (Рисунок 9).

В рамках данной работы был изучен базовый алгоритм создания нейронной сети, на основе которого была разработана в среде MATLAB ANFIS сеть, осуществляющий получение в зависимости от входных данных на входе.

Список литературы

1. Тэрано Т., Асаи К., Сугено М. Прикладные нечеткие системы перевод с японского канд. техн. наук Ю.Н. Чернышова. Москва. «Мир», 1993. 363 с.
2. Норвиг А.М., Турсон И.Б. Построение функций принадлежности // Нечеткие множества и теория возможностей. Последние достижения: пер. с англ./ под ред. Р.Р. Ягера. М.: Радио и связь, 1986. 408 с.
3. Рыжов А.П. Элементы теории нечетких множеств и измерения нечеткости. М.: Диалог. МГУ, 1998.
4. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. [Электронный ресурс]. Режим доступа: <http://www.matlab.ru/fuzzylogic/book1/index.asp/> (дата обращения: 14.01.2019).
5. Ярушкіна Н.Г. Основы теории нечетких и гибридных систем. М.: Финансы и статистика, 2004.
6. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Радио и связь, 2000.

7. Катковник В.Я. Непараметрическая идентификация и сглаживание данных: метод локальной аппроксимации. М.: Наука, 1985.
8. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. Мир. М., 1976.
9. Алексеев А.Н., Волков Н.И., Кочевский А.Н. Элементы нечёткой логики при программном контроле знаний // Открытое образование. 2004. Гроп Д. Методы идентификации. М.: Наука, 1979.
10. Борисов В.В., Круглов В.В., Федюлов А.С. Нечеткие модели и сети. М.: Горячая линия – Телеком, 2007.
11. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2003.
12. Дворак А., Перфильева И. Введение в нечеткое моделирование. Анг.: 2016. 272 с.