

ИССЛЕДОВАНИЕ НЕЧЕТКИХ НЕЙРОННЫХ СЕТЕЙ С РАЗЛИЧНЫМИ АЛГОРИТМАМИ ВЫВОДА В ЗАДАЧАХ ПРОГНОЗИРОВАНИЯ НА РЫНКАХ ЦЕННЫХ БУМАГ

Тофик Кязимов, Юрий Зайченко

Аннотация. В статье рассматривается проблема прогнозирования на финансовых рынках с использованием нечетких нейронных сетей (FNN). Для этого предлагаются нейронные сети с различными алгоритмами Мамдани, Цукамото и Сугено. Рассмотрены структура этих сетей и алгоритмы вывода. Проведены экспериментальные исследования точности прогнозов FNN, представлено и проанализировано сравнение их эффективности при прогнозировании цен акций и индексов рынков на фондовых биржах. Были разработаны и исследованы алгоритмы обучения FNN. Чувствительность точности прогноза в зависимости от алгоритма вывода, количества входов, правил и количества итераций была исследована и сравнена для различных FNN.

Key words: financial markets, forecasting, fuzzy neural networks

ITHEA Keywords: I. Computing methodologies I2. Artificial intelligence, I6.5. Model development

Введение

В последние годы нечеткие нейронные сети получили широкие применения в различных задачах вычислительного интеллекта. Их использование определяется их следующими важными свойствами [Зайченко, 2008], [Згуровский, 2013]:

1. Возможность работы в нечеткой и неполной информации, а также качественной информации;

2. Использование экспертной информации в виде нечетких правил вывода;
3. Они являются универсальными аппроксиматорами, т.е. способны аппроксимировать с произвольной точностью непрерывные ограниченные функции от n переменных.

Вместе с тем они обладают рядом недостатков, а именно:

- база нечетких правил может оказаться неполной или противоречивой;
- функции принадлежности и их параметры, которые задаются экспертом могут оказаться неадекватными реальным моделируемым процессам.

Для устранения указанных недостатков их необходимо обучать. Для этого используется целый арсенал алгоритмов обучения многие из которых были разработаны для обычных нейронных сетей [Згуровский, 2013], [Bodyanskiy, 2009], [Зайченко, 2017].

В настоящее время разработаны различные системы нечеткого логического вывода и ННС, отличающиеся алгоритмами нечеткого вывода и обладающие разными прогнозирующими возможностями. Целью настоящей работы является исследование и сравнительный анализ эффективности различных алгоритмов нечеткого вывода в задачах прогнозирования на рынках ценных бумаг.

Основные этапы нечеткого логического вывода

Используемый в разных экспертных и управляющих системах механизм нечетких выводов в своей основе имеет базу знаний, формируемую специалистами предметной области в виде совокупности нечетких предикатных правил вида [Зайченко, 2008]

Π_1 : если x есть A_1 , то y есть B_1 ;

Π_2 : если x есть A_2 , то y есть B_2 ;

...

$$\Pi_n : \text{если } x \text{ есть } A_n, \text{ то } y \text{ есть } B_n,$$

где $x, x \in X$ — входная переменная (имя для известных значений данных); $y, y \in Y$ — переменная вывода (имя для значения данных, которое будет вычислено); A_i и B_i — функции принадлежности, заданные соответственно на множествах X и Y .

Приведем более детальное пояснение. Знания эксперта $A \rightarrow B$ отражают нечеткое причинное отношение предпосылки (антецедент) и следствия (консеквент), поэтому его можно назвать нечетким отношением и обозначить через R :

$$R : A \rightarrow B,$$

где « $\rightarrow$ » называют нечеткой импликацией.

Отношение R можно рассматривать как нечеткое подмножество прямого произведения $X \times Y$ полного множества предпосылок X и выводов Y . Таким образом, процесс получения (нечеткого) результата вывода B' с использованием данного наблюдения A' и знания $A \rightarrow B$ можно представить в виде композиционного правила — нечеткий «modus ponens»:

$$B' = A' \bullet R = A' \bullet (A \rightarrow B),$$

где « $\bullet$ » — операция свертки.

Как операцию композиции, так и операцию импликации в алгебре нечетких множеств можно реализовывать по-разному (при этом будет различаться и получаемый результат), но в любом случае общий логический вывод осуществляется за следующие четыре этапа [Зайченко, 2008],

Введение нечеткости (фаззификация —fuzzification).

Функции принадлежности, определенные на входных переменных, применяются к их фактическим значениям для определения степени истинности каждой предпосылки каждого правила.

2. Логический вывод. Он состоит из двух подэтапов.

2.1. На первом из них определяется степень истинности всей совокупности предпосылок правила. Для этого используется операция пересечения.

2.2. На втором подэтапе определяется выход каждого правила используя нечеткий логический вывод и определяется степень его выполнения (истинности). Для этого используется операция нечеткого пересечения условий и следствия правила.

В качестве правил логического вывода обычно используются только операции $\min$ (МИНИМУМ) или prod (УМНОЖЕНИЕ). В логическом выводе МИНИМУМА функция принадлежности вывода «отсекается» по высоте, соответствующей вычисленной степени истинности предпосылки правила (нечеткая логика «И»). В логическом выводе УМНОЖЕНИЯ функция принадлежности вывода масштабируется при помощи вычисленной степени истинности предпосылки правила.

3. Композиция. Все нечеткие подмножества, назначенные к каждой переменной вывода (во всех правилах), объединяются вместе, чтобы сформировать одно нечеткое подмножество для всех переменных вывода. При подобном объединении обычно используются операции $\max$ (МАКСИМУМ) или sum (СУММА). При композиции МАКСИМУМА комбинированный вывод нечеткого подмножества конструируется как поточечный максимум по всем нечетким подмножествам (нечеткая логика «ИЛИ»). При композиции СУММЫ комбинированный вывод нечеткого подмножества формируется как поточечная сумма по всем нечетким подмножествам, назначенным переменной вывода правилами логического вывода.

4. Приведение к четкости (дефаззификация—*defuzzification*).

Используется, если нужно преобразовать нечеткий набор выводов в четкое число. Существует значительное количество методов приведения к четкости, некоторые из которых рассмотрены ниже.

Пример. Пусть некоторая система описывается следующими нечеткими правилами:

Π_1 : если x есть A , то z есть D ;

Π_2 : если y есть B , то z есть E ;

Π_3 : если w есть C , то z есть F ,

где x, y и w — имена входных переменных; z — имя переменной вывода, а A, B, C, D, E, F — заданные функции принадлежности (треугольной формы).

Процедура получения логического вывода показана на рис. 1. Предполагается, что заданы конкретные (четкие) значения входных переменных: x_0, y_0, z_0 .

На первом этапе на основании данных значений и исходя из функций принадлежности A, B, C находят степени истинности $\alpha(x_0), \alpha(y_0)$ и $\alpha(w_0)$ для предпосылок каждого из трех приведенных правил. На втором этапе происходит «отсечение» функций принадлежности выводов правил (D, E, F) на уровнях $\alpha(x_0), \alpha(y_0)$ и $\alpha(w_0)$. На третьем этапе рассматриваются функции принадлежности, усеченные на предыдущем этапе, и проводится их объединение с использованием операции $\max$, в результате чего получается комбинированное нечеткое подмножество, описываемое функцией принадлежности $\mu_\Sigma(z)$ и соответствующее логическому выводу для выходной переменной z . Наконец, на четвертом этапе при необходимости находят четкое значение выходной переменной, например, с применением центроидного метода — четкое значение выходной переменной определяется как центр тяжести для кривой $\mu_\Sigma(w)$:

$$z_0 = \frac{\int_{\Omega} z \mu_\Sigma(z) dz}{\int_{\Omega} \mu_\Sigma(z) dz}.$$

Алгоритмы нечеткого вывода

Рассмотрим наиболее применяемые алгоритмы нечеткого вывода, для простоты считая, что база знаний содержит два нечетких правила вида

Π_1 : если x есть A_1 и y есть B_1 , то z есть C_1 ;

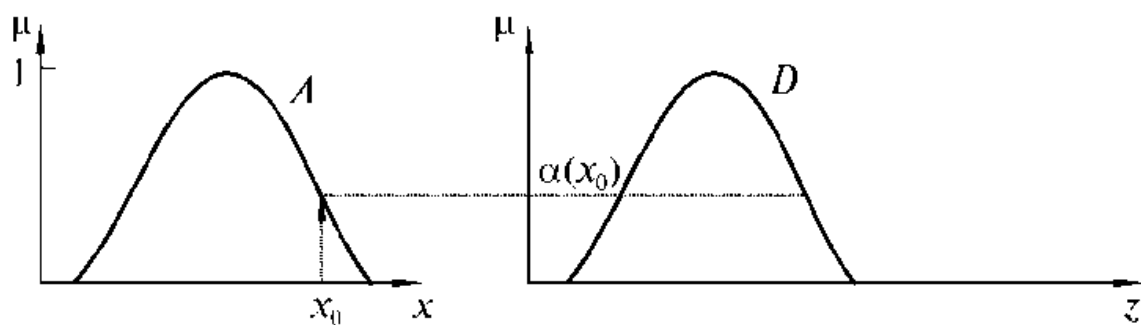
Π_2 : если x есть A_2 и y есть B_2 , то z есть C_2 ,

где x и y — имена входных переменных; z — имя переменной вывода; $A_1, B_1, C_1, A_2, B_2, C_2$ — некоторые заданные функции принадлежности.

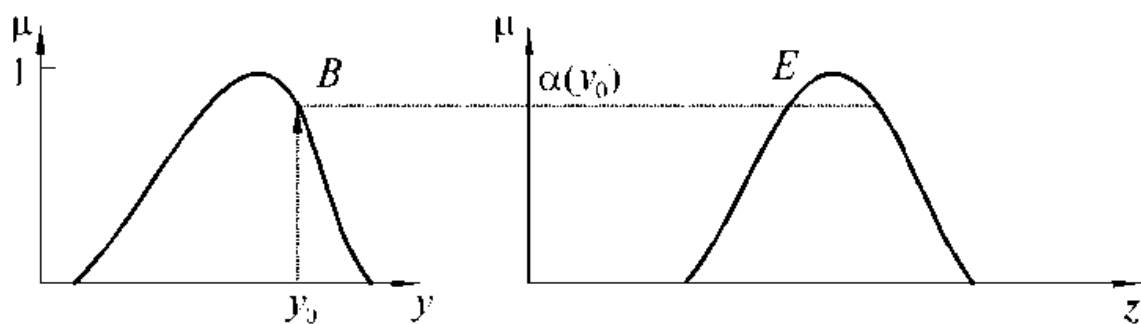
Алгоритм нечеткого вывода Мамдани

Данный алгоритм соответствует рассмотренному примеру на рис. 1.

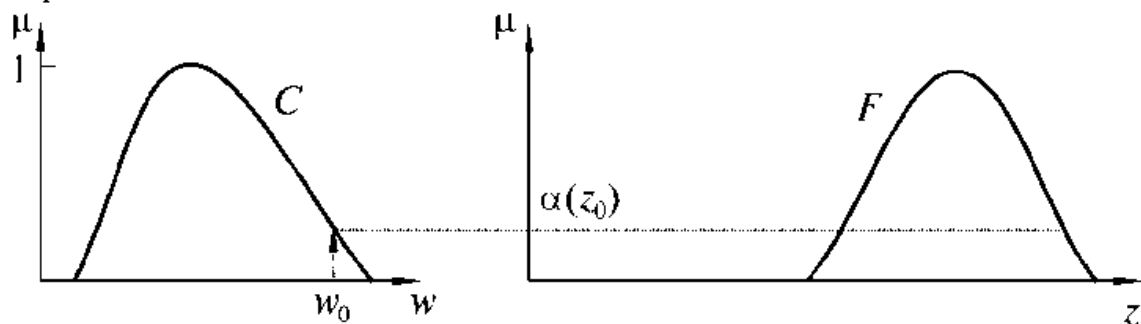
Правило 1:



Правило 2:



Правило 3:



Композиция
и приведение к четкости:

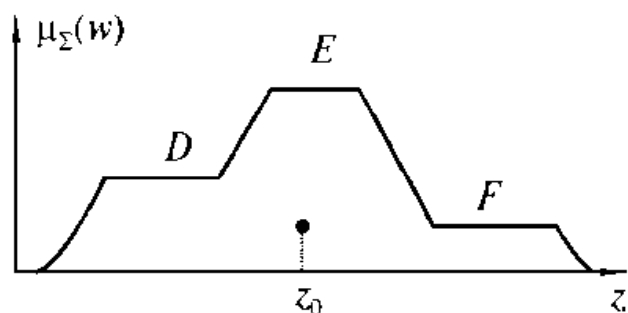


Рис.1. Алгоритм нечеткого вывода Мамдани

При этом он математически может быть описан следующим образом [Зайченко, 2008], [Згуровский, 2013].

1. Введение нечеткости. Находят степени истинности для предпосылок каждого правила:

$$A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0).$$

2. Логический вывод. Находят уровни «отсечения» для предпосылок каждого из правил (с использованием операции МИНИМУМ):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0);$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

где через « $\wedge$ » обозначена операция логического минимума (min). Затем находят «усеченные» функции принадлежности выходов правил:

$$C'_1 = (\alpha_1 \wedge C_1(z)); C'_2 = (\alpha_2 \wedge C_2(z)).$$

3. Композиция. Проводится объединение найденных усеченных функций с использованием операции МАКСИМУМ (max, обозначенная далее как « $\vee$ »), что приводит к получению итогового нечеткого подмножества для переменной выхода с функцией принадлежности:

$$\mu_\Sigma(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)). \quad (1)$$

4. Приведение к четкости. Проводится для нахождения z_0 , например, центроидным методом.

$$Z_0 = \frac{\int_{\Omega} z \cdot \mu_\Sigma(w) dw}{\int_{\Omega} \mu_\Sigma(w) dw} \quad (2)$$

2.2 Алгоритм нечеткого вывода Цукамото

Исходные посылки — как у предыдущего алгоритма, но здесь предполагается, что функции $C_1(z), C_2(z)$ являются монотонными (рис. 2) [Зайченко, 2008], [Згуровский, 2013]:

1. Введение нечеткости (как в алгоритме Мамдани).

2. Нечеткий вывод. Сначала находят уровни «отсечения» α_1 и α_2 (как в алгоритме Мамдани), а затем решают уравнения:

$$\alpha_1 = C_1(z_1) \text{ и } \alpha_2 = C_2(z_2) .$$

При этом определяют четкие значения (z_1 и z_2) для каждого исходного правила.

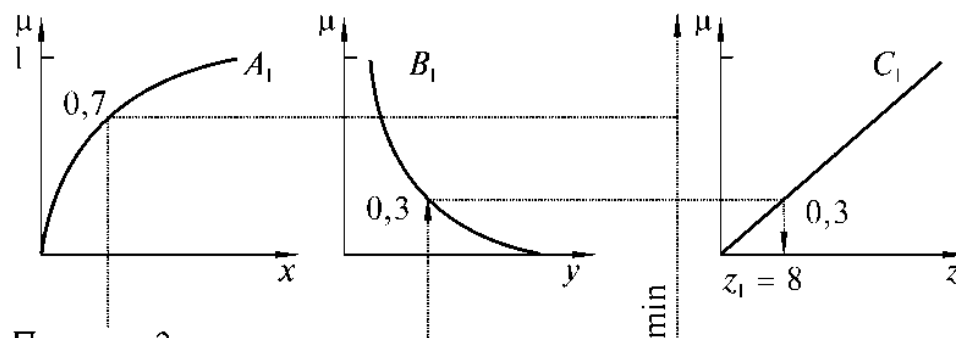
3. Определяют четкое значение переменной вывода (как взвешенное среднее z_1 и z_2):

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} .$$

В общем случае дискретный вариант центроидного метода реализуется так:

$$z_0 = \frac{\sum_{i=1}^n \alpha_i z_i}{\sum_{i=1}^n \alpha_i} . \quad (3)$$

Правило 1:



Правило 2:

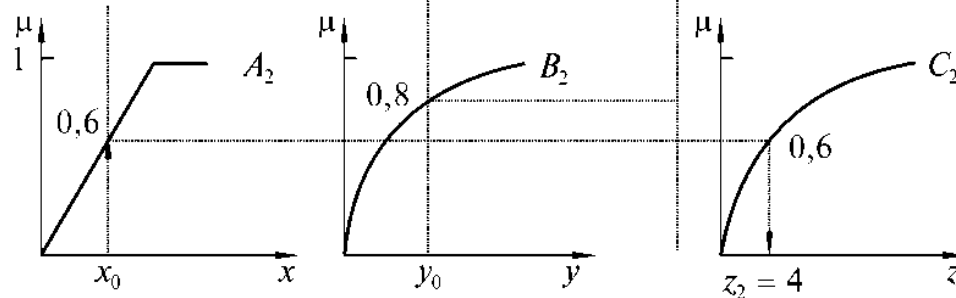


Рис. 2. Алгоритм Цукamoto

2.3 Алгоритм нечеткого вывода Сугено

Сугено и Такаги использовали набор правил в следующей форме (как и ранее, приведем пример двух правил):

Π_1 : если x есть A_1 и y есть B_1 , то $z_1 = a_1x + b_1y$;

Π_2 : если x есть A_2 и y есть B_2 , то $z_2 = a_2x + b_2y$.

Описание алгоритма (рис. 3.):

1. Введение нечеткости (как в алгоритме Мамдани).

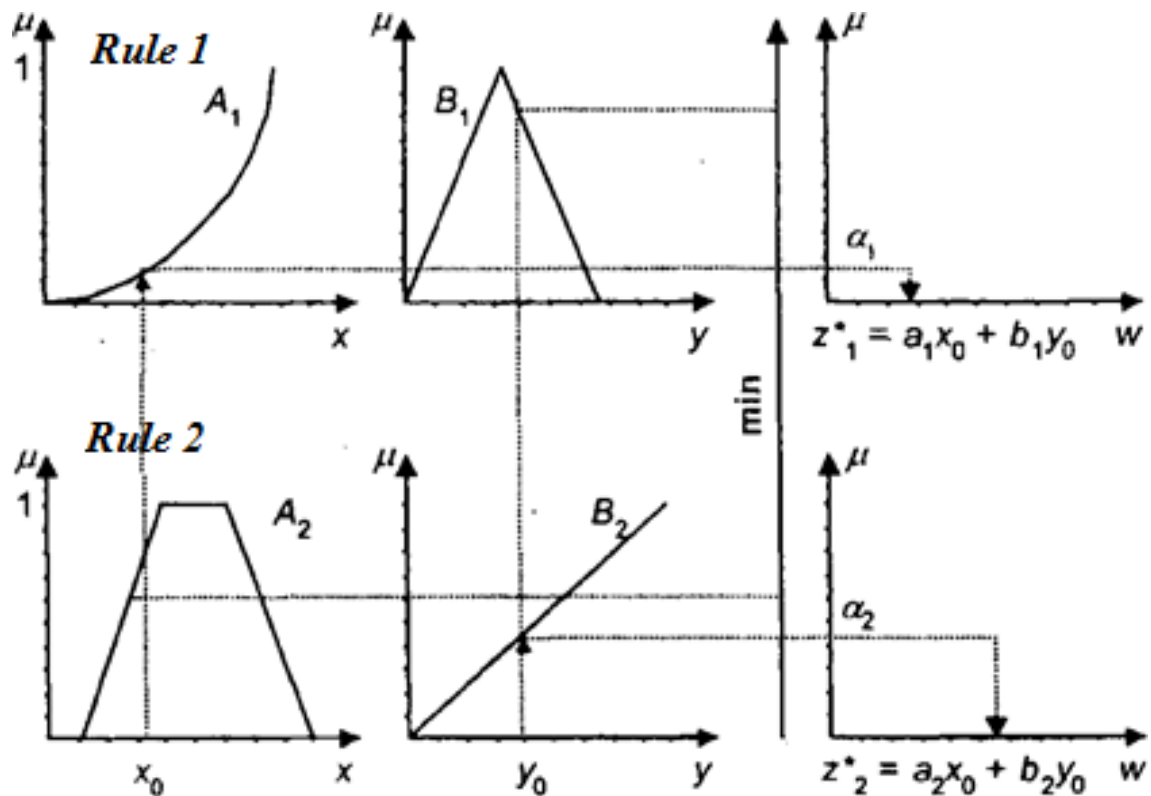


Рис. 3.

2. Нечеткий вывод. Находят $\alpha_1 = A_1(x_0) * B_1(y_0)$, $\alpha_2 = A_2(x_0) * B_2(y_0)$ и индивидуальные выходы правил:

$$\dot{z}_1 = a_1x_0 + b_1y_0;$$

$$\dot{z}_2 = a_2x_0 + b_2y_0.$$

3. Определяют четкое значение переменной вывода центроидным методом:

$$z_0 = \frac{\alpha_1 \dot{z}_1 + \alpha_2 \dot{z}_2}{\alpha_1 + \alpha_2} . \quad (4)$$

2.4. Алгоритм нечеткого вывода Ларсена

В алгоритме Ларсена (Larsen) нечеткая импликация моделируется с использованием оператора умножения.

Описание алгоритма (рис.4):

1. Введение нечеткости (как в алгоритме Мамдани).
2. Нечеткий вывод. Сначала, как в алгоритме Сугено, находят значения:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0) = A_1(x_0) * B_1(y_0)$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0) = A_2(x_0) * B_2(y_0)$$

а затем определяют частные выходные нечеткие подмножества:

$$\alpha_1(C_1(z)) , \alpha_2(C_2(z)) .$$

3. Композиция. Проводится композиция выходов правил, как в алгоритме Мамдани с использованием операции МАКСИМУМ (max, обозначенная далее как « $\vee$ »), что приводит к получению итогового нечеткого подмножества для переменной выхода с функцией принадлежности:

$$\mu_z(z) = C(z) = C_1'(z) \vee C_2'(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)) . \quad (5)$$

4. Приведение к четкости. Проводится для нахождения z_0 , например, центроидным методом, как в алгоритме Мамдани, формула (2).

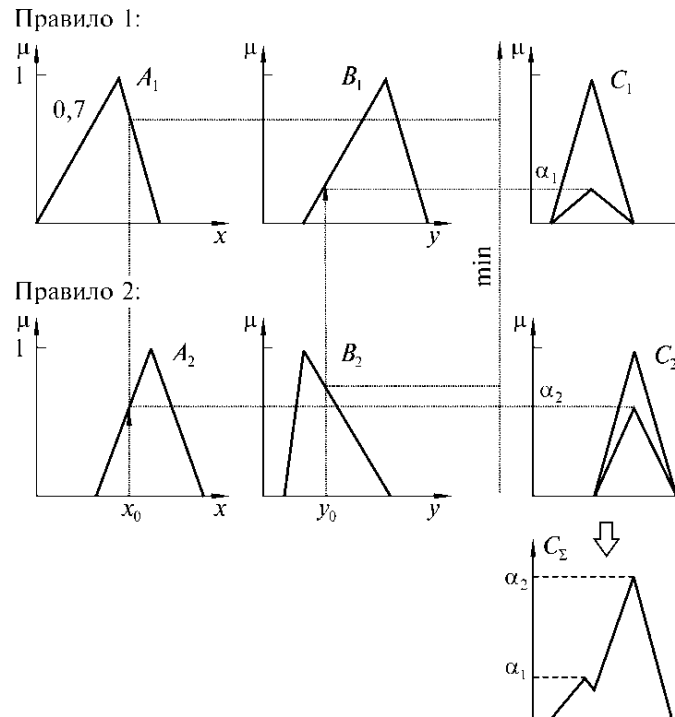


Рис. 4.

Градиентный метод обучения нечетких нейронных сетей

Рассмотрим ННС с алгоритмом вывода Цукамото. Структура этой сети приведена на рис.5.

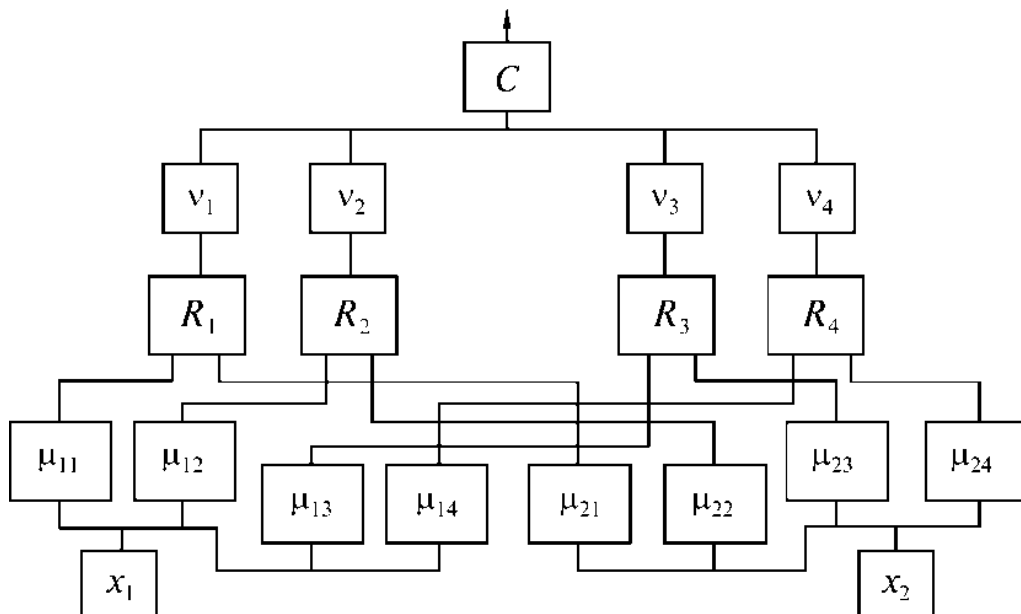


Рис. 5. Структура ННС с выводом Цукамото

Модули x_1 и x_2 здесь представляют входные переменные, и они посылают свои значения в свои μ -модули, которые содержат соответствующие ФП. μ -модули связаны с R -модулями, которые представляют собой нечеткие правила «если, то».

Каждый μ -модуль передает всем связанным с ним R -модулям значения ФП $\mu_{ij}(x_i)$ ее входной величины x_i . R -модуль использует операцию пересечения и находит $\min_i \{\mu_{ij}(x_i)\}$ и передает это значение дальше в v -модуль, который содержит ФП, описывающую выходные значения. v -модуль, используя монотонные функции принадлежности, вычисляет r_i и $v^{-1}(r_i)$ и передает их в C -модуль, который вычисляет итоговую выходную переменную — управляющее воздействие C используя алгоритм центра масс (COA),

где n — число правил вывода; r_i — степень, с которой правило R_i выполняется. Как нетрудно заметить, система на рис. 4 напоминает последовательную многослойную НС, где x -, R - и C -модули выполняют функции нейронов, а μ - и v -модули играют роль адаптируемых весов связей сети.

Пусть ФП i -го μ -модуля, который связан с правилом R_k описывается таким образом

$$\mu_{ik}(x_i) = \exp \left\{ -\frac{1}{2} * \frac{(x_i - a_{ik})^2}{\sigma_{ik}^2} \right\} \quad (6)$$

где a_{ik} , σ_{ik} - параметры ФП, которые определяются в процессе обучения.

v_k - соответствующий вес выходной связи (R_k, C) с следующей ФП

$$\mu_k(y_i) = \exp \left\{ -\frac{1}{2} * \frac{(y_i - a_k)^2}{\sigma_k^2} \right\},$$

Пусть пересечение условий правил определяется в форме произведения:

$$\alpha_k = \prod_{i=1}^n \mu_{ik}(x_i) = \exp \left\{ - \sum_{i=1}^n \frac{1}{2} \cdot \frac{(x_i - a_{ik})^2}{\sigma_{ik}^2} \right\}. \quad (7)$$

Допустим, что мы используем центроидный метод фаззификации. Тогда общий выход сети определяется так:

$$z_0 = \frac{\sum_k z_k \alpha_k}{\sum_k \alpha_k}$$

Допустим, что в выходах правил используется монотонная ФП $C(z)$. Тогда z_k определяется как решение следующего уравнения

$$C_k(z_k) = \alpha_k, \quad (8)$$

где

$$C_k(z_k) = \exp \left\{ - \frac{1}{2} * \frac{(z_k - a_k)^2}{\sigma_k^2} \right\}, \quad (9)$$

Решая уравнение:

$$\exp \left\{ - \frac{1}{2} * \frac{(z_k - a_k)^2}{\sigma_k^2} \right\} = \alpha_k,$$

Находим два корня:

$$z_k = a_k \pm \sqrt{2 \ln \frac{1}{\alpha} * \sigma_k}.$$

Первый корень:

$$z_{1k} = a_k - \sqrt{2 \ln \frac{1}{\alpha} * \sigma_k}.$$

Он находится на части кривой где график $z_r(a_k)$ монотонно возрастает, а второй - z_{2k} , где график монотонно спадает.

Пусть критерий обучения имеет вид $E(z) = \frac{1}{2} (z_0 - z^*)^2 \rightarrow \min$,

где z^* - желаемый выход; z_0 – выход сети,

Тогда мы находим следующие производные критерия по параметрам:

$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial z_0} \frac{\partial z_0}{\partial z_k} \frac{\partial z_k}{\partial a_k} = + (z_0 - z^*) \frac{\alpha_k}{\sum_{k=1}^K \alpha_k}, \quad (10)$$

На монотонно возрастающей части кривой:

$$\frac{\partial E}{\partial \sigma_k} = \frac{\partial E_0}{\partial z_0} \frac{\partial z_0}{\partial z_k} \frac{\partial z_k}{\partial \sigma_k} = - (z_0 - z^*) \frac{\alpha_k}{\sum_{k=1}^K \alpha_k} \cdot \sqrt{2 \ln \frac{1}{\alpha}}, \quad (11)$$

На монотонно спадающей части:

$$\frac{\partial E}{\partial \sigma_k} = + (z_0 - z^*) \frac{\alpha_k}{\sum_{k=1}^K \alpha_k} \sqrt{2 \ln \frac{1}{\alpha}}, \quad (12)$$

Для входных μ - модулей находим

$$\begin{aligned} \frac{\partial E}{\partial \alpha_{ik}} &= \frac{\partial E_0}{\partial z_0} \frac{\partial z_0}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial \alpha_{ik}} = + (z_0 - z^*) \frac{z_k \sum_k \alpha_k - \sum_k z_k \cdot \alpha_k}{\left(\sum_k \alpha_k \right)^2} \cdot \prod_{k=1}^K \exp \left\{ - \frac{(x_i - a_{ik})^2}{2 \cdot \sigma_{ik}^2} \right\} \cdot \frac{(x_i - a_{ik})}{\sigma_{ik}^2} = \\ &= (z_0 - z^*) \cdot \alpha_k \cdot \frac{z_k \sum_k \alpha_k - \sum_k z_k \cdot \alpha_k}{\left(\sum_k \alpha_k \right)^2} \cdot \frac{(x_i - a_{ik})}{\sigma_{ik}^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial E}{\partial \sigma_{ik}} &= \frac{\partial E}{\partial z_0} \frac{\partial z_0}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial \sigma_{ik}} = \\ &= \left((z_0 - z^*) \alpha_k \frac{z_k \cdot \sum_{k=1}^K \alpha_k - \sum_k z_k \cdot \alpha_k}{\left(\sum_k \alpha_k \right)^2} \cdot \frac{(x_i - a_{ik})^2}{\sigma_{ik}^3} \right) \end{aligned} \quad (13)$$

Тогда градиентный алгоритм обучения ННС Цукамото имеет следующий вид:

Для связей выходного слоя

$$a_k(n+1) = a_k(n) - \gamma_n \frac{\partial E}{\partial \alpha_k} = a_k(n) - \gamma_n (z_0 - z^*) \frac{\alpha_k}{\sum_k \alpha_k}, \quad (14)$$

$$\begin{aligned}\sigma_i(n+1) &= \sigma_k(n) - \gamma'_n \frac{\partial E}{\partial \sigma_k} = \\ &= \sigma_k(n) \pm \gamma_n (z_0 - z_\Sigma^*) \frac{\alpha_k}{\sum_k \alpha_k} * \sqrt{2 \ln \frac{1}{\alpha_k}}\end{aligned}\quad (15)$$

для входных μ - модулей

$$\begin{aligned}a_{ik}(n+1) &= a_{ik}(n) - \gamma_{n_2} \frac{\partial E}{\partial a_{ik}} = a_{ik}(n) - \gamma_n (z_0 - z_\Sigma^*) \cdot \\ &\cdot \frac{\alpha_k \left(z_k \cdot \sum_k \alpha_k - \sum_k z_k \cdot \alpha_k \right)}{\left(\sum_k \alpha_k \right)^2} \cdot \frac{(x_i - a_{ik})}{\sigma_{ik}^2}\end{aligned}\quad (16)$$

$$\sigma_{ik}(n+1) = \sigma_{ik}(n) - \gamma_{n_3} (z_0 - z_\Sigma^*) \frac{\alpha_k \left(z_k \cdot \sum_k \alpha_k - \sum_k z_k \cdot \alpha_k \right)}{\left(\sum_k \alpha_k \right)^2} \cdot \frac{(x_i - a_{ik})^2}{\sigma_{ik}^3}.\quad (17)$$

где $\gamma_n, \gamma_{n_1}, \gamma_{n_2}, \gamma_{n_3}$ - размер шага спуска.

Экспериментальные исследования и анализ результатов

С целью проведения исследований был разработан программный продукт на языке программирования Python в среде JetBrains PyCharm. Он позволяет моделировать работу нечетких нейросетей Мамдани, Цукамото и а Сугено, осуществлять их обучение и выбирать параметры ФП.

В качестве исходных данных для прогнозирования был выбран промышленный индекс Доу Джонса (Dow Jones Industrial Average (DJI)). Данные были взяты из веб-ресурсу [https:// www.investing.com/indices/us-30](https://www.investing.com/indices/us-30). Всего были выбраны 80 значений индекса в период с 2019-01-11 по 2019-05-07.

Прогноз осуществлялся по переменной Price. На этапе подготовки данных значения индекса (Price) нормируются в интервале от 0 до 1. Критериями оценки качества прогноза выбраны MSE (Mean Squared Error) и MAE (Mean Absolute Error).

Анализ и сравнение полученных результатов

Сравнение различных алгоритмов проходило при варьировании следующих параметров:

- соотношение обучающей выборки к проверочной ;
- число входов 6, число итераций: 100.

Результаты прогноза для разных методов представлены на рис.6 и в табл.1. А в табл.2 приведены значения метрик MSE та MAE соответствующих алгоритмов

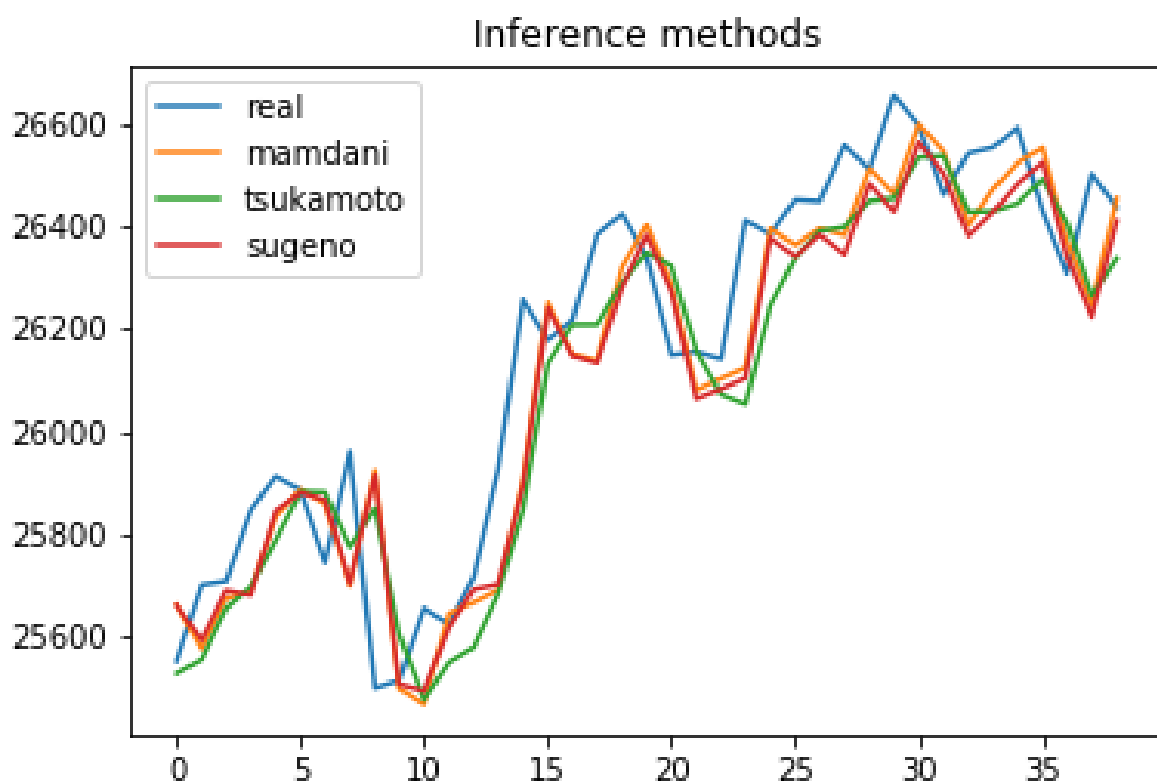


Рис. 6. Сравнение алгоритмов Мамдани, Цукамото та Сугено

Табл. 1. Реальные та прогнозируемые результаты индекса

Реальные значения	25554.66	25702.89	25709.94	25848.87	25914.1	25887.38	25745.67	25962.51
Мамдани	25663.93	25578.39	25676.91	25690.09	25835.78	25888.9	25861.08	25701.35
Цукамото	25530.06	25557.51	25657.21	25699.97	25789.46	25884.25	25883.06	25776.07
Сугено	25663.13	25594.98	25691.98	25682.7	25847.02	25882.16	25868.09	25704.95

Табл. 2. Значения MSE та MAE в зависимости от алгоритма вывода

	Мамдани	Цукамото	Сугено
MSE	0.000944	0.000999	0.001003
MAE	0.023519	0.025614	0.024212

Сравним работу Мамдани при соотношении обучающей к проверочной выборке 80%/20% и различном числе итераций. Число входов равно 6.

В табл.3 показаны результаты прогнозирования, а в табл.4 соответствующие значения критериев MSE и MAE.

Табл. 3. Сравнение реальных и прогнозных значений

Реальные значения	25554.6	25702.8	25709.9	25848.8	25914.1	25887.3	25745.6	25962.5
10 итераций	25463.4	25504.3	25531.8	25646.8	25676.3	25807.5	25797.4	25718.1
50 итераций	25662.6	25565.4	25684.3	25685.7	25837.4	25886.1	25859.1	25702.0
100 итераций	25663.9	25578.3	25676.6	25690.0	25835.7	25888.9	25861.0	25701.3

Табл. 4. Значения MSE и MAE при соотношении 80/20

	10 итераций	50 итераций	100 итераций
MSE	0.001653	0.000947	0.000953
MAE	0.035366	0.023586	0.023841

Следующий эксперимент имел аналогичные входные данные, а соотношения обучающей к проверочной выборке 50/50. В табл.5 приведены результаты прогнозирования, а в табл.6 значения MSE и MAE.

Табл. 5. Результаты прогнозирования

Реальные значения	24100.51	23592.98	23675.64	23323.66	22859.6	22445.37	21792.2	22878.45
10 итераций	25456.96	24456.96	24066.91	23686.48	23391.21	23227.27	22795.13	22185.08
50 итераций	24692.04	24160.58	23578.21	23768.85	23577.18	23017.44	22575.91	22028.06
100 итераций	24687.94	24139.37	23584.02	23790.99	23548.44	23011.34	22578.6	22012.83

Табл. 6. Значения MAE та MSE при соотношении 50/50

	10 итераций	50 итераций	100 итераций
MSE	0.003073	0.002744	0.002720
MAE	0.039513	0.035193	0.035167

В следующем эксперименте соотношение обучающей к проверочной выборке было: 30/70. Все остальные параметры оставались прежними.

В табл.7 показаны результаты прогнозирования ННС Мамдани, а в табл.8 соответствующие значения MSE та MAE.

Табл. 7. Результати прогнозування

Реальные значения	25379.45	25444.34	25317.41	25191.43	24583.42	24984.55	24688.31	24442.92
10 итераций	25577.77	25520.29	25399.68	25466.77	25358.33	25012.06	24939.12	24909.9
50 итераций	25712.71	25303.38	25437.78	25400.67	25252.11	24592.77	25139.71	24846.86
100 итераций	25710.87	25356.81	25352.43	25408.75	25294.8	24616.37	25091.68	24875.04

Табл. 8. Значения MSE и MAE при соотношении выборок 30/70

	10 итераций	50 итераций	100 итераций
MSE	0.004609	0.003246	0.003428
MAE	0.052707	0.040476	0.040608

В последней серии экспериментов проведено варьировании числа входов при различных соотношениях обучающей и проверочной выборок.. В таблице 9 приведены значения критериев MSE та MAE, а их зависимость показана на Рис.7 и Рис.8.

Табл. 9. Значения MSE та MAE при разных соотношениях выборок и разном числе входов

%	80/20			50/50			30/70		
Входи	2	5	7	2	5	7	2	5	7
MSE	0.000967	0.000943	0.000934	0.002499	0.003059	0.002958	0.003139	0.003779	0.003561
MAE	0.023229	0.023613	0.023613	0.033441	0.038011	0.037819	0.039286	0.043735	0.042632

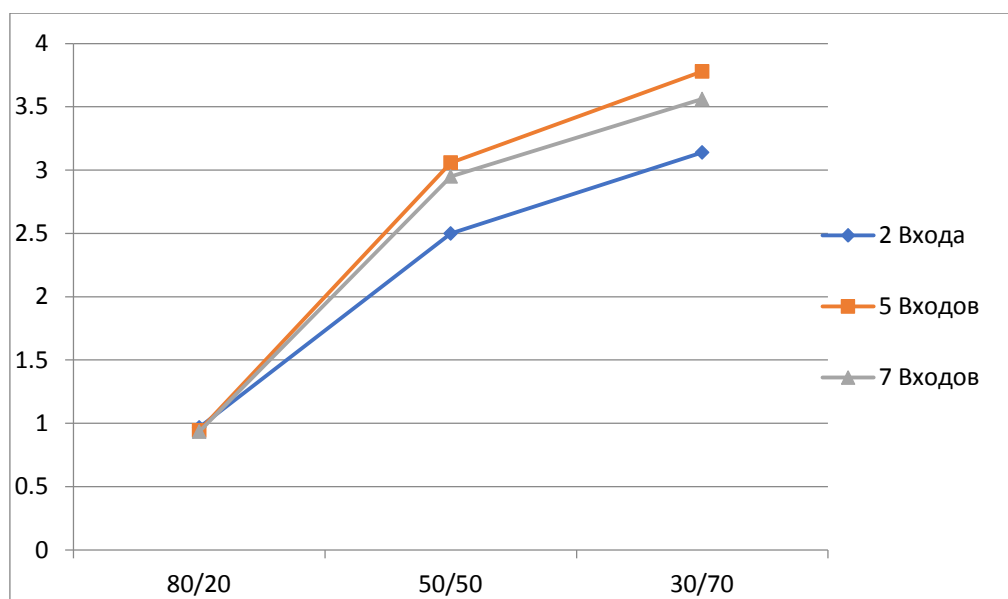


Рис.7. Значения MSE при разных входах и соотношениях выборок (x1000)

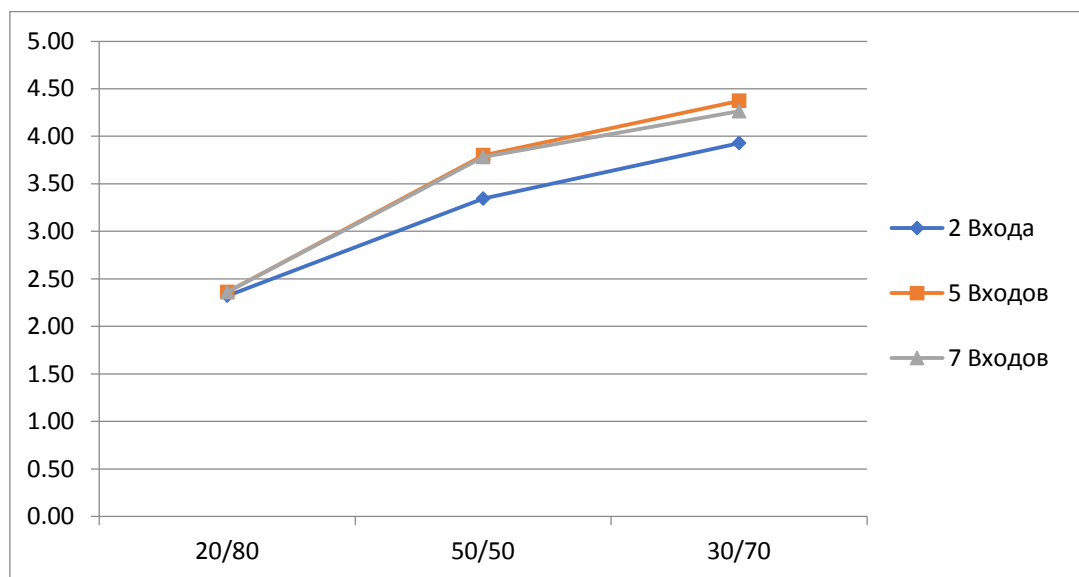


Рис.8. Значения MAE при разном числе входов и соотношениях выборок

Анализируя эти результаты на рисунках можно сделать вывод, что при уменьшении доли обучающей выборки значения критериев растут (точность падает).

Кроме того, наилучшие результаты показали нейросети с наименьшим числом входов. Этот факт можно объяснить тем, что при увеличении числа входов сложность сети возрастает и для ее обучения требуется большее число итераций.

Анализируя результаты прогнозирования различных нейронных сетей можно сделать вывод, что наиболее высокую точность прогнозирования имеет ННС с выводом Мамдани, далее с примерно одинаковыми результатами следуют ННС Цукамото и Сугено.

Выводы

В работе рассмотрены ННС с выводами Мамдани, Цукамото и Сугено в задаче прогнозирования на рынке ценных бумаг. Рассмотрены алгоритмы их обучения.

Проведены экспериментальные исследования нечетких нейросетей в задаче прогнозирования индекса Доу-Джонса на рынке ценных бумаг NYSE. В процессе экспериментов варьировалось соотношение обучающей к проверочной выборке, число входов. В качестве критериев точности прогнозов использовались MAPE и MAE. Исследования показали достаточно высокую точность прогнозов для всех типов сетей.

При этом наиболее высокую точность прогноза показала ННС с выводом Мамдани.

В целом по результатам исследований можно заключить, что нечеткие нейронные сети являются хорошим инструментом прогнозирования курсов акций и биржевых индексов на рынках ценных бумаг.

Список литературы

- [Зайченко, 2008] Зайченко Ю.П. Нечеткие модели и методы в интеллектуальных системах. — К.: Изд. дом «Слово», 2008. — 354 с.
- [Згуровский, 2013] Згуровский М.З., Зайченко Ю.П. Основы вычислительного интеллекта. — К.: Наукова думка, 2013. — 406 с.
- [Bodyanskiy, 2009] Ye. Bodyanskiy, Yu. Zaychenko, E. Pavlikovskaya, M. Samarina and Ye. Viktorov, The neo-fuzzy neural network structure optimization using the GMDH for the solving forecasting and classification problems, Proc. Int. Workshop on Inductive Modeling, Krynica, Poland, 2009, pp. 77-89.
- [Bodyanskiy, 2013] Ye. Bodyanskiy, O. Vynokurova, A. Dolotov and O. Kharchenko, Wavelet-neuro-fuzzy network structure optimization using GMDH for the solving forecasting tasks, Proc. 4th Int. Conf. on Inductive Modelling ICIM 2013, Kyiv, 2013, pp. 61-67.
- [Зайченко, 2017] Ю.П. Зайченко, Галиб Гамидов. Каскадные нейронечеткие сети в задачах прогнозирования на рынках ценных бумаг. Системні дослідження та інформаційні технології, 2017. №2,- с.92-102.

Информация об авторах



Yuri Zaychenko – Professor, doctor of technical sciences, Institute for applied system analysis, NTUU “KPI”, 03056, Ukraine, Kyiv, Peremogi pr. 37, Corpus 35; e-mail: baskervil@voliacable.com, zaychenkoyuri@ukr.net

Fields of Scientific Research: Information systems, Fuzzy logic, Decision making theory



Tofig Kazimov– Professor, Doctor of Physics and Mathematics of sciences, Institute of Information Technology of ANAS, AZ1141, Azerbaijan Republic, Baku, B.Vahabzade str., 9A, e-mail: tofig@mail.ru

Major Fields of Scientific Research: Information systems, Fuzzy logic, Data Mining, software engineering

Fuzzy Neural Networks Investigations in Forecasting Problems at Stock Exchanges

Tofiq Kazimov, Yuriy Zaychenko

Abstract. *The problem of forecasting at financial markets with application of fuzzy neural networks (FNN) is considered in this paper. For this aim neural networks with different algorithms Mamdani, Tsukamoto and Sugeno are suggested. The structure of these networks and inference algorithms are considered. The experimental investigations of forecasting accuracy of FNN were carried out and comparison of their efficiency in predicting of share prices and markets indices at stock exchanges is presented and analyzed. The training algorithms for FNN were developed and explored. The sensitivity of forecasting accuracy in dependence on inference algorithm, number of inputs, rules and iterations number was investigated and compared for different FNN.*

Key words: *financial markets, forecasting, fuzzy neural networks*

ITHEA Keywords: *1. Computing methodologies 12. Artificial intelligence, 16.5. Model development*