ОБОБЩЕНИЕ ДИСКРЕТНЫХ МОДЕЛЕЙ ОКРЕСТНОСТНЫМИ СИСТЕМАМИ

Шмырин А.М., Седых И.А., Корниенко Н.А., Шмырина Т.А.
Липецкий государственный технический университет, г. Липецк
amsh@lipetsk.ru, sedykh-irina@yandex.ru, natastas@mail.ru

Ключевые слова: окрестностные системы, сети Петри, нейронные сети.

Введение

В работах [1-4] введены и исследованы окрестностные модели, развивающие общие подходы теории систем и являющиеся обобщением для традиционных дискретных моделей таких, как конечные автоматы, клеточные автоматы, сети Петри и т.д.

Далее показано, что дискретные модели, в частности, сети Петри и нейронные сети являются разновидностями окрестностных систем с некоторыми вариациями.

1. Дискретные модели в классе окрестностных систем

1.1. Обобщенное определение окрестностной модели

Обобщим приведенное в [2] определение окрестностной модели. Окрестностная модель в общем случае описывается набором $NS = (N, X, V, Y, Z, G, F, X[0])$, где:

1). $N = (A, O_x, O_v, O_y)$ – структура окрестностной модели, $A = \{a_1, a_2, ..., a_n\}$ – множество узлов, O_x – окрестности связей узлов по состояниям, O_v – окрестности связей узлов по управлениям, O_y – окрестности связей узлов по выходным воздействиям. Для каждого узла $a_i \in A$ определена своя окрестность по состояниям $O_x[a_i] \subseteq A$, управлениям $O_v[a_i] \subseteq A$ и выходам $O_y[a_i] \subseteq A$; $O_x = \bigcup_{i=1}^{n} O_x[a_i]$, $O_v = \bigcup_{i=1}^{n} O_v[a_i]$; $O_y = \bigcup_{i=1}^{n} O_y[a_i]$;

2). $X \in R^n$ – вектор состояний окрестностной модели в текущий момент времени;

3). $V \in R^m$ – вектор управлений окрестностной модели в текущий момент времени;

4). $Y \in R^l$ – вектор выходов окрестностной модели в текущий момент времени;

5). $Z \in \mathbf{R}_+^n$ – вектор временных задержек в узлах, где $\mathbf{R}_+$ – множество неотрицательных действительных чисел;

6). $G : X_{O_x} \times V_{O_v} \to X$ – функция пересчета состояний окрестностной модели (в общем случае недетерминированная), где X_{O_x} – множество состояний узлов, входящих в окрестность O_x, V_{O_v} – множество управлений узлов, входящих в окрестность O_v;

7). $F : X_{O_x} \times V_{O_v} \to Y$ – функция пересчета выходов окрестностной модели (в общем случае недетерминированная);

8). $X[0]$ – начальное состояние модели.

В частных случаях для различных дискретных моделей отдельные составляющие окрестностной модели могут отсутствовать.

Функции G и F могут быть произвольными, например линейными, билинейными, квадратичными, полиномиальными и т.д. В линейном случае G и F можно представить в виде системы линейных уравнений:

$$(1)\quad \begin{cases} \sum\limits_{x\in O_x[t+1,a_i]} w_x[t+1,a_i,\alpha]x[t+1,\alpha] = \sum\limits_{x\in O_x[t,a_i]} w_x[t,a_i,\alpha]x[t,\alpha] + \sum\limits_{\beta\in O_v[t,a_i]} w_v[t,a_i,\beta]v[t,\beta] \\ \sum\limits_{y\in O_y[t+1,\gamma_i]} w_y[t+1,a_i,\gamma]y[t+1,\gamma] = \sum\limits_{x\in O_x[t,a_i]} w_x[t,a_i,\alpha]x[t,\alpha] + \sum\limits_{\beta\in O_v[t,a_i]} w_v[t,a_i,\beta]v[t,\beta] \end{cases},$$

где $O_x[t+1,a_i]$, $O_x[t,a_i]$ – окрестности узла a_i по x соответственно в моменты времени $t+1$ и t, $O_v[t,a_i]$ – окрестность узла a_i по v в момент времени t, $O_y[t+1,a_i]$ – окрестность узла a_i по y в момент времени $t+1$, $a_i\in A$, $x[t+1,a_i]\in R^n$, $x[t,a_i]\in R^n$ – состояния в узле a_i модели соответственно в моменты времени $t+1$ и t, $v[t,a_i]\in R^m$ – вход в узле a_i модели в момент времени t, $y[t+1,a_i]\in R^l$ – выход в узле a_i модели в момент времени $t+1$, $w_x[t+1,a_i,\alpha]\in R^{c\times n}$, $w_x[t,a_i,\alpha]\in R^{c\times n}$, $w_v[t,a_i,\beta]\in R^{c\times m}$, $w_y[t+1,a_i,\gamma]\in R^{c\times l}$ – матрицы-параметры, $\alpha,\beta,\gamma\in A$.

Представим модель (1) в матричном виде. Для этого определим матрицы $W_x[t+1]$, $W_x[t]$ коэффициентов по состояниям в моменты времени $t+1$ и t соответственно, матрицу $W_v[t]$ коэффициентов по входам в момент времени t, матрицу $W_y[t+1]$ коэффициентов по выходам в момент времени $t+1$:

$$W_x[t+1] = \begin{bmatrix} w_x[t+1,a_1,a_1] & w_x[t+1,a_1,a_2] & \dots & w_x[t+1,a_1,a_n] \\ w_x[t+1,a_2,a_1] & w_x[t+1,a_2,a_2] & \dots & w_x[t+1,a_2,a_n] \\ \dots & \dots & \dots & \dots \\ w_x[t+1,a_n,a_1] & w_x[t+1,a_n,a_2] & \dots & w_x[t+1,a_n,a_n] \end{bmatrix};$$

$$W_x[t] = \begin{bmatrix} w_x[t,a_1,a_1] & w_x[t,a_1,a_2] & \dots & w_x[t,a_1,a_n] \\ w_x[t,a_2,a_1] & w_x[t,a_2,a_2] & \dots & w_x[t,a_2,a_n] \\ \dots & \dots & \dots & \dots \\ w_x[t,a_n,a_1] & w_x[t,a_n,a_2] & \dots & w_x[t,a_n,a_n] \end{bmatrix};$$

$$W_v[t] = \begin{bmatrix} w_v[t,a_1,a_1] & w_v[t,a_1,a_2] & \dots & w_v[t,a_1,a_m] \\ w_v[t,a_2,a_1] & w_v[t,a_2,a_2] & \dots & w_v[t,a_2,a_v] \\ \dots & \dots & \dots & \dots \\ w_v[t,a_n,a_1] & w_v[t,a_n,a_2] & \dots & w_v[t,a_n,a_m] \end{bmatrix};$$

$$W_y[t+1] = \begin{bmatrix} w_y[t+1,a_1,a_1] & w_y[t+1,a_1,a_2] & \dots & w_y[t+1,a_1,a_n] \\ w_y[t+1,a_2,a_1] & w_y[t+1,a_2,a_2] & \dots & w_y[t+1,a_2,a_n] \\ \dots & \dots & \dots & \dots \\ w_y[t+1,a_n,a_1] & w_y[t+1,a_n,a_2] & \dots & w_y[t+1,a_n,a_n] \end{bmatrix}.$$

Тогда модель (1) будет иметь вид:

$$(2)\quad \begin{cases} W_x[t+1]\cdot X[t+1] = W_x[t]\cdot X[t] + W_v[t]\cdot V[t] \\ W_y[t+1]\cdot X[t+1] = W_x[t]\cdot X[t] + W_v[t]\cdot V[t] \end{cases}.$$

В случае, когда функции W и F являются нелинейными, модель (2) преобразуется к виду:

$$(3)\quad \begin{cases} W_x[t+1]\cdot X[t+1] = G(X[t],V[t]) \\ W_y[t+1]\cdot Y[t+1] = F(X[t],V[t]) \end{cases}.$$

1.2. Сеть Петри как частный случай окрестностной модели

В [2] было показано, что сеть Петри является динамической недетерминированной окрестностной моделью $NS_{PN} = (N,X,V,W,X[0])$, причем система (2) в случае сети Петри принимает вид:

(4) $$\begin{aligned}&\left[W_x^1[t+1]\ \ W_x^2[t+1]..W_x^m[t+1]\right]\cdot D\cdot X[t+1]=\\&=\left[W_x^1[t]\ \ W_x^2[t]..W_x^m[t]\right]\cdot D\cdot X[t]+\left[W_v^1[t]\ \ W_v^2[t]..W_v^m[t]\right]\cdot D\cdot V[t]\end{aligned},$$

где $W_x^k[t+1]\in R^{n\times n}$, $W_x^k[t]\in R^{n\times n}$ – матрицы коэффициентов k -го слоя по состояниям в моменты времени $t+1$ и t соответственно, $W_v^k[t]\in R^{n\times n}$ – матрица коэффициентов k -го слоя по входам в момент времени t; $X[t+1]\in R^n$, $X[t]\in R^n$ – вектор состояний окрестностной системы в моменты времени $t+1$ и t соответственно; $V[t]\in R^n$ – вектор входов в момент времени t, $D\in R^m$ – случайный вектор, состоящий из нулей и одной единицы в позиции, соответствующей выбираемому слою k, по уравнениям которого происходит пересчет состояний узлов окрестностной модели в следующий момент времени $t+1$.

1.3.Нейронная сеть как частный случай окрестностной модели

Рассмотрим представление нейронной сети в виде окрестностной модели. Пусть сеть состоит из *M* слоев, в *j*-м слое которой находится *Nj* нейронов. Поставим в соответствие нейронам нейронной сети узлы окрестностной модели.

Обозначим a_{ij} – *j*–ый нейрон (узел в окрестностной модели) слоя *i*, *i*=0,…,*M*, *j*=1,…,*Nj*. Тогда окрестностная модель нейронной сети представляется формулой:

(5) $$\sum_{\beta\in O_y[a_{ij}]} w_y[a_{ij},\gamma]y[\gamma]=F_{ij}\left(\sum_{\beta\in O_v[a_{ij}]} w_v[a_{ij},\beta]v[\beta]\right),$$

где $O_v[a_{ij}]$ – окрестность узла a_{ij} по V, $O_y[a_{ij}]$ – окрестность узла a_{ij} по y, состоящая из одного узла a_{ij}, $a_{ij}\in A$, $v[a_{ij}]\in R^m$ – вход в узле a_{ij} модели, $y[a_{ij}]\in R^l$ – выход в узле a_{ij} модели, $w_v[a_{ij},\beta]\in R^{c\times m}$, $w_y[a_{ij},\gamma]\in R^{c\times l}$ – матрицы-параметры., причем матрица $w_y[a_{ij},\gamma]$ является единичной матрицей, $\beta,\gamma\in A$, $F_{ij}:V\to Y$ – некоторая функция.

Модель (5) в общем виде:

(6) $W_y\cdot Y=F(V)$.

Таким образом, нейронная сеть в виде окрестностной модели задается набором $NS_{NN}=(N,V,Y,F)$.

2. Алгоритмы идентификации окрестностных моделей нейронных сетей

В данном разделе рассмотрим некоторые вопросы идентификации подкласса окрестностных моделей. Идентификация окрестностной модели нейронной сети соответствует задаче обучения нейронной сети и заключается в нахождении функции F. В работе приведены результаты идентификации нейронных сетей с линейной и нелинейной функциями с помощью двух алгоритмов: стандартного алгоритма обратного распространения ошибки и алгоритма Качмажа [5].

2.1. Алгоритм обратного распространения

Наиболее популярным алгоритмом обучения нейронной сети, является алгоритм обратного распространения ошибки. Это итеративный алгоритм обучения, который используется с целью минимизации среднеквадратического отклонения текущих от требуемых выходов многослойных нейронных сетей с последовательными связями [1].

Имеется множество функций, которые можно использовать в качестве активационных. Одной из наиболее распространенных функций, является сигмоидальная.

Несомненным достоинством алгоритмов обучения нейронных сетей по типу обратного распространения ошибки является возможность достаточно точной аппроксимации целевой функции без точного знания внутренней структуры данных, возможность самостоятельного приобретения знаний в процессе обучения, а так же возможность выявления скрытых закономерностей в наборе данных.

Недостатком подобных систем является невысокая скорость сходимости и значительное число необходимых итераций до достижения удовлетворительной точности, а также склонность к увязанию в локальных минимумах, значительные трудности в оптимальном подборе длины шага и поиске глобальных минимумов.

Кроме того, для этих алгоритмов характерна возможность паралича сети, при котором большинство нейронов функционируют при очень больших значениях аргумента функций активации, т.е. на её пологом участке. Поскольку градиентный алгоритм обратного распространения оперирует с производными функции ошибки, которая на пологих участках мала, то процесс обучения практически замирает. Кроме того существует опасность слишком точной аппроксимации данных, что приводит к резким скачкообразным изменениям выходного вектора и невозможности получить приемлемый уровень обобщения. Это состояние так называемого переобучения сети.

2.2.Алгоритм Качмажа

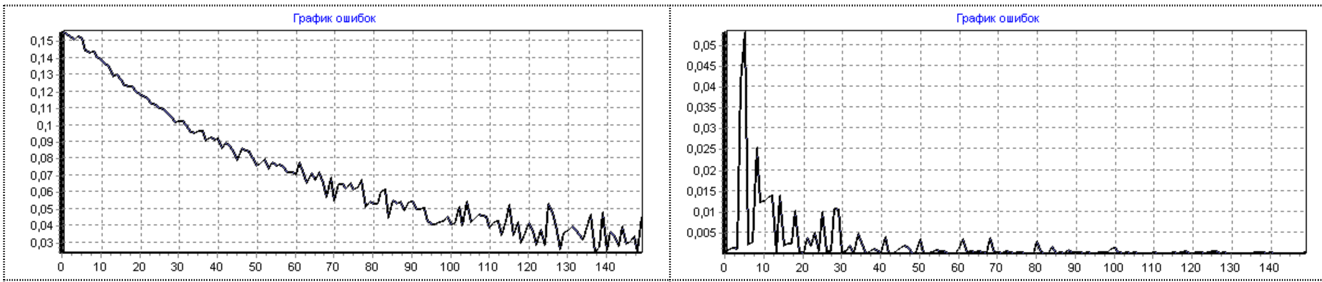
Алгоритм, описанный Качмажем в 1937 г., представляет собой классический метод решения системы линейных уравнений, относительно неизвестных значений весовых коэффициентов. Обучение сети описывается итерационной последовательностью по следующим рекуррентным соотношениям [5]:

$$c_{ij} = c_{ij} + \frac{\Delta_i x_{ij}}{\sum_{j=1}^{n} x_{ij}^2}, \quad \text{где} \quad \Delta_i = y_i - v_i \text{ - ошибка идентификации.}$$

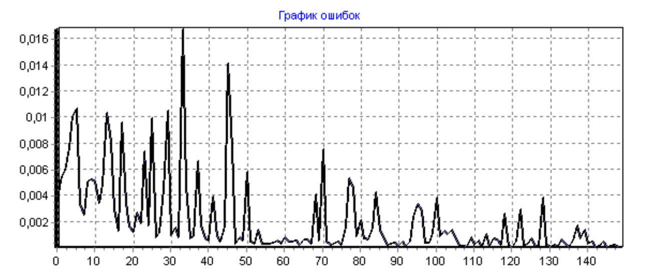
В рамках изучения вопросов идентификации указанных подклассов окрестностных систем было проведено исследование классического алгоритма обучения Качмажа и сравнение его с алгоритмом обратного распространения ошибки на примере аппроксимации линейной и нелинейной функций.

2.3. Результаты идентификации

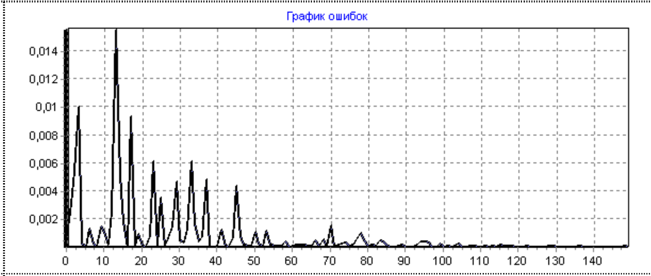
На примере аппроксимации линейной и нелинейной функций были получены графики ошибок обучения. Для тестирования были взяты нормированные в Excel данные, случайно искаженные шумом. Одновременно, алгоритм обратного распространения ошибки исследовался при различных активационных функциях (АФ). Нейронная сеть состояла из 3-х слоев. На входном слое и скрытом слоях по 3 нейрона, на выходном – 1 нейрон (рис. 1). Результаты для обучения за 150 шагов представлены на рис. 1, за 1000 шагов – на рис. 2.



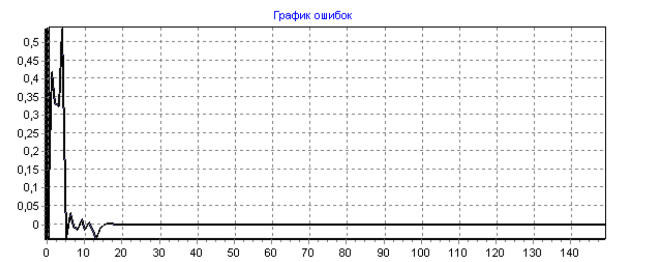
а). Сигмовидная АФ. б). Линейная АФ.



в). Радиальная базисная

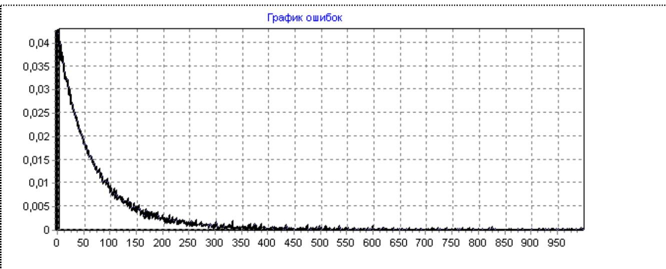


г). Гиперболический тангенс

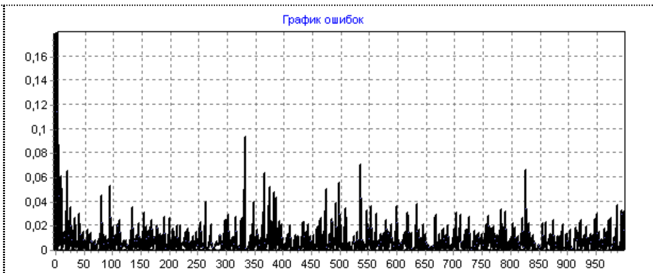


д). Алгоритм Качмажа

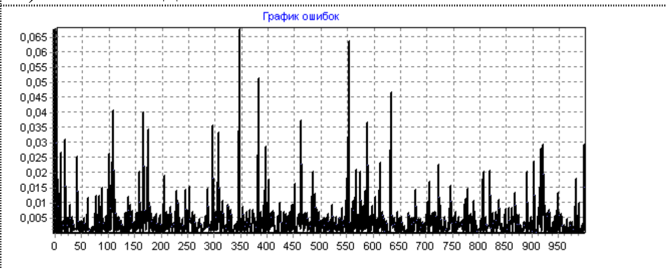
Рис. 1. Ошибка аппроксимации линейной функции $y(x_1, x_2, x_3)$ (150 шагов)



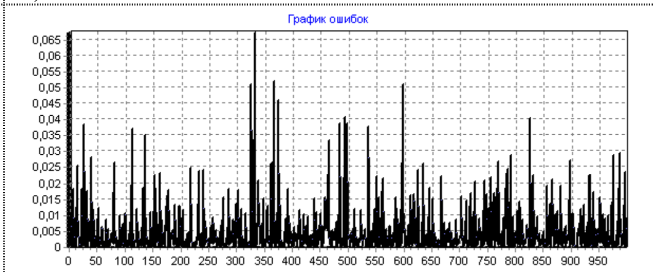
а). Сигмовидная АФ.



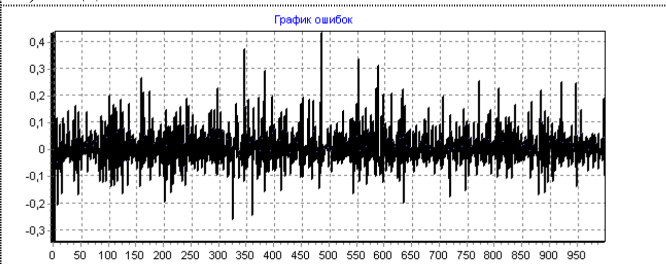
б). Линейная АФ.



в). Радиальная базисная



г). Гиперболический тангенс



д). Алгоритм Качмажа

Рис. 2. Ошибка аппроксимации нелинейной функции $y(x_1, x_2, x_3)$ (1000 шагов)

Таблица 1. Результаты идентификации

Функция	Ошибка

Сигмовидная	0,0077
Линейная	0,0509
Радиальная базисная	0,0477
Гиперболический тангенс	0,0762
Алгоритм Качмажа	0,0901

Таким образом, на одинаковых данных погрешность аппроксимации линейной функции с применением алгоритма Качмажа несколько меньше, чем с применением алгоритма обратного распространения. Алгоритм Качмажа с вычислительной точки зрения реализуется значительно проще, чем алгоритм обратного распространения, и время счета, для достижения заданной точности меньше. Для аппроксимации же нелинейной функции хороший результат показала только сигмовидная функция активации, что не исключает реабилитацию алгоритма Качмажа для известных из печати его модификаций.

3. Заключение

Таким образом, в работе приведено обобщенное определение окрестностных моделей. Показано, что сети Петри и нейронные сети являются разновидностями окрестностных систем.

Окрестностные модели сетей Петри $NS_{PN} = (N, X, V, W, X[0])$ отличаются от окрестностных моделей нейронных сетей $NS_{NN} = (N, V, Y, F)$ следующими характеристиками: послойной структурой, недетерминированностью функционирования, динамикой развития системы, наличием состояний и отсутствием выходных воздействий.

В работе также приведены результаты идентификации нейронных сетей с линейной и нелинейной функциями с помощью двух алгоритмов: стандартного алгоритма обратного распространения ошибки и алгоритма Качмажа.

Литература

1. *Блюмин С.Л., Шмырин А.М. Окрестностные системы. – Липецк: Липецкий эколого-*гуманитарный институт, 2005. – 132 с.
2. *Блюмин С.Л., Шмырин А.М., Седых И.А., Филоненко В.Ю. Окрестностное моделирование сетей Петри. – Липецк: ЛЭГИ, 2010. – 124 с.*
3. *Блюмин С.Л., Шмырин А.М., Шмырина О.А..* Билинейные окрестностные системы– Липецк: ЛГТУ, 2006. – 130 с.
4. *Карабутов Н.Н., Шмырин А.М.* Окрестностные системы: Идентификация и оценка состояния. – Липецк: ЛГТУ, 2005. – 132 с.
5. *Хайкин С.* Нейронные сети: полный курс. 2-е изд. М., "Вильямс", 2006 – 995 с.