УДК 519.71

С.А. Ляшенко

# ОБ ОДНОМ АЛГОРИТМЕ ОБУЧЕНИЯ НЕЙРОСЕТЕВОЙ МОДЕЛИ НЕЛИНЕЙНОГО ДИНАМИЧЕСКОГО ОБЪЕКТА

Центральными вопросами математического прогнозирования являются выбор и обоснование математической модели прогнозируемого процесса или явления. В свою очередь построение такой модели представляет собой задачу идентификации, успех решения которой зависит от соотношения двух факторов – объёма априорной информации о структуре и параметрах процесса и объёме апостериорной (измеряемой) информации. Априорная информация позволяет определить структуру модели, то есть решить задачу структурной идентификации. На основании же измеряемой или апостериорной информации определяют параметры модели выбранной структуры, то есть решают задачу параметрической идентификации.

Качество решения задач идентификации и прогнозирования во многом определяется характером изучаемого процесса. Так как обычно во всех реальных ситуациях присутствует элемент случайности, для решения данных задач применяют статистические методы. В связи с тем, что априорная информация о свойствах сигналов и помех, как правило, отсутствует или является неполной, предпочтительным оказывается применение теории наименьших квадратов, в частности, метода наименьших квадратов и его рекуррентного аналога. Однако, если исследуемые процессы являются нестационарными, применение этих методов представляется проблематичным. В этих условиях значительно более эффективным оказывается подход, основанный на теории искусственных нейронных сетей [1–4].

## Постановка задачи

Рассмотрим задачу идентификации нелинейного динамического объекта, описываемого уравнением

$$y(k)=f[y(k-1),y(k-2),..,y(k-n),u(k-d),$$
$$u(k-d-1),...,u(k-d-m)]+\xi(k), \qquad (1)$$

где $u(k)\in R^1$, $y(k)\in R^1$, $\xi(k)\in R^1$ — входной, выходной сигналы и помеха соответственно; $m,u,d$ — соответственно порядки запаздывания по входному, выходному сигналам и чистая задержка; $f[\cdot]$ — неизвестная нелинейная функция.

Задача идентификации заключается в определении функции $f[\cdot]$ по измеряемым входным и выходным переменным $\{u(k),\ y(k)\}$.

При использовании нейросетевого подхода осуществляется аппроксимация нелинейной функции $f[\cdot]$ некоторой системой заранее выбранных базисных функций $\{\phi_i[\cdot]\}$, каждая из которых реализуется отдельными нейронами, образующими одно – или многослойную сеть. Точность аппроксимации оценивается с помощью какого-либо функционала. Формально это можно представить следующим образом.

Введём вектор обобщённого входного сигнала

$$X(\kappa)=[y(k-1),y(k-2),...,y(k-n),u(k-d),$$
$$u(k-d-1),...u(k-d-m)]. \qquad (2)$$

В этом случае нейросетевая модель объекта (1) примет вид

$$\hat{y}(k)=\hat{f}[x(k)]=\sum_{i=0}^{p}w_i(k)\Phi_i[x(k)]=\Phi^T[x(k)]w(k), \qquad (3)$$

где $w_i(k)$ — подлежащие определению весовые параметры сети; $p$ — количество используемых в сети нейронов.

Определим ошибку реакции сети как

$$e(k)=y(k)-\hat{y}(k)\ . \qquad (4)$$

Таким образом, при нейросетевом подходе задача идентификации сводится к обучению сети, заключающемуся в определении её параметров на основании минимизации некоторого выпуклого функционала от ошибки, например,

$$J=M\{[y(k)-\hat{y}(k)]^2\}\ , \qquad (5)$$

где $M\{\cdot\}$ — символ математического ожидания.

Применительно к задачам идентификации и прогнозирования наиболее эффективными являются многослойный персептрон и радиально-базисные сети (РБС) [1, 2]. Если в основе обучения персептрона лежит алгоритм обратного распространения ошибки, реализации которого сопутствует целый ряд трудностей, то для обучения РБС используют обычно метод наименьших квадратов (МНК), если обучение проходит в режиме off-line [5–8], или его рекуррентные аналоги в режиме on-line [4, 9].

В настоящей работе предлагаются рекуррентные алгоритмы обучения РБС, представляющие собой модификацию МНК.

### Обучение сети

В РБС в качестве базисных функций $\phi_i[x]$ могут быть взяты, например, мультиквадратичная, обратная мультиквадратичная, сплайн-функция и получившая наиболее широкое распространение гауссова функция

$$\Phi_i[x] = \exp\left\{-\frac{\|x-\mu_i\|^2}{\sigma_i^2}\right\}. \quad (6)$$

Здесь $\mu_i, \sigma_i^2$ — центры и радиусы базисных функций соответственно; $\|\cdot\|$ — евклидова норма.

Иногда вместо функций (6) в нейронной модели (3) применяют нормализированные базисные функции

$$\tilde{\Phi}_i[x(k)] = \frac{\Phi_i[x(k)]}{\sum_{j=1}^{p}\Phi_j[x(k)]}, \quad (7)$$

обладающие свойством $\sum_{i=1}^{p}\tilde{\Phi}_i[x(k)] = 1$, что является весьма существенным для многих приложений.

Если в качестве базисных выбраны функции вида (6) или (7), то в процессе обучения сети должны быть определены все её параметры: $w_i$, $\mu_i$ и $\sigma_i^2$. При этом возможны следующие варианты обучения [2]:

1. Задаются центры $\mu_i$ и радиусы $\sigma_i^2$ и настраиваются веса $w_i$.

2. Центры $\mu_i$ и радиусы $\sigma_i^2$ определяются в процессе самообучения, а веса вычисляются путём минимизации функционала (5).

3. Все параметры сети определяются путём минимизации функционала (5).

Первый вариант обучения является наиболее простым, но зачастую наименее эффективным. Он используется в РБС с жестко заданными радиусами $\sigma_i^2$. В этом случае оправдывает себя применение нормализованных базисных функций (7), обеспечивающих в отличие от (6) равномерное покрытие всех точек входного пространства и делающих сеть менее чувствительной к неудачному заданию центров $\mu_i$. Однако, как следует из (7), каждая нормализованная базисная функция зависит от всех других базисных функций, и введение в сеть нового нейрона приводит к изменению всех базисных функций. Поэтому при необходимости обучения сети в режиме on-line применение функций вида (7) нецелесообразно. Наиболее эффективным, хотя и наиболее трудоемким, является третий вариант обучения, при котором определяются все компоненты вектора

$$\theta(k) = (w_0(k), w_1(k), \mu_1^T(k), \sigma_1(k), \ldots w_p(k), \mu_p^T(k), \sigma_p(k))^T.$$

Используемый при этом алгоритм обучения рекуррентного МНК имеет вид [1, 9]

$$\theta(k) = \theta(k-1) + \frac{P(k-1)\nabla f(k)}{\lambda + \nabla f^T(k)P(k-1)\nabla f(k)}\cdot e(k), \quad (8)$$

$$P(k) = \frac{1}{\lambda}\cdot\left[P(k-1) - \frac{P(k-1)\nabla f(k)\nabla f^T(k)P(k-1)}{\lambda + \nabla f^T(k)P(k-1)\nabla f(k)}\right], \quad (9)$$

где $\nabla f(k) = \left[1, \Phi_1[x(k)]\right.$,

$$2\Phi_1[x(k)]w_1\sigma_1^{-2}\times(x(k)-\mu_1)^T,$$

$$\left.2\Phi_1[x(k)]w_1\sigma_1^{-3}\times\|x(k)-\mu_1\|^2, \ldots, \Phi_p[x(k)]\right.,$$

$$2\Phi_p[x(k)]w_p\sigma_p^{-2}\times(x(k)-\mu_p)^T,$$

$$\left.2\Phi_p[x(k)]w_p\sigma_p^{-3}\|x(k)-\mu_p\|^2\right]^T;$$

$$e(k) = y(k) - y(\hat{k});\ \lambda \in (0, 1].$$

Как отмечается в [4], с ростом размерности исследуемого объекта и количества обучающих пар $\{u(k),\ y(k)\}$ вычислительные затраты, необходимые для реализации алгоритма (8), (9), существенно возрастают. Поэтому в указанной работе был предложен алгоритм обучения, в котором используется некоторое фиксированное число обучающих пар $S < N$, где $N$ — общее количество настраиваемых параметров. Такой алгоритм обладает лучшими динамическими и худшими по сравнению с (8),(9) фильтрующими свойствами.

Улучшение фильтрующих свойств может быть достигнуто путём увеличения числа обучающих пар, то есть путём выбора $S > N$. Подобный рекуррентный алгоритм может быть построен по аналогии с алгоритмом текущего регрессионного анализа [10]. Использование блочного представления векторов и матриц позволяет получить следующие рекуррентные соотношения, представляющие собой такой алгоритм обучения:

$$\theta_{S+1}(k) = \theta_S(k-1) + \frac{P_S(k-1)\nabla f(k)}{\lambda + \nabla f^T(k)P_S(k-1)\nabla f(k)}\times$$

$$\times(y(k) - \theta_S^T(k-1)\nabla f[x(k)]); \quad (10)$$

$$P_{S+1}(k) = \frac{1}{\lambda}\cdot\left[P_S(k-1) - \frac{P_S(k-1)\nabla f(k)\nabla f^T(k)P_S(k-1)}{\lambda + \nabla f^T(k)P_S(k-1)\nabla f(k)}\right]; \quad (11)$$

$$\theta_S(k) = \theta_{S+1}(k) - \frac{\lambda^S P_{S+1}(k)\nabla f(k-S+1)}{1-\lambda^S + \nabla f^T(k-S+1)P_{S+1}(k)\nabla f(k-S+1)}\times$$

$$\times(y(k-S+1) - \theta_{S+1}^T(k)\nabla f[(k-S+1)]); \quad (12)$$

$$P_S(k) = P_{S+1}(k) + \frac{\lambda^S P_{S+1}(k)\nabla f(k-S+1)\nabla f^T(k-S+1)P_{S+1}(k)}{1-\lambda^S + \nabla f^T(k-S+1)P_{S+1}(k)\nabla f(k-S+1)}, \quad (13)$$

где $\lambda \in (0,1]$.

Соотношения (10), (11) описывают коррекцию вектора оценок $\theta$ и информационной матрицы $P$ при поступлении новой информации, а (12), (13) — при удалении устаревшей. Алгоритм (10)–(13) является более динамичным по сравнению с (8), (9), приближаясь ростом $S$ к нему по фильтрующим свойствам.

Если же при обучении сети сначала происходит удаление устаревшей информации, а потом учитывается вновь поступившая, то алгоритм примет следующий вид:

$$\theta_{S-1}(k) = \theta_S(k-1) - \frac{\lambda^{S-1}P_S(k-1)\nabla f(k-S+1)}{1-\lambda^{S-1} + \nabla f^T(k-S+1)P_S(k-1)\nabla f(k-S+1)} \times$$

$$\times (y(k-S+1) - \theta_S^T(k-1)\nabla f(k-S+1)); \quad (14)$$

$$P_{S-1}(k) = P_S(k-1) + \frac{\lambda^{S-1}P_S(k-1)\nabla f(k-S+1)\nabla f^T(k-S+1)P_S(k-1)}{1-\lambda^{S-1} + \nabla f^T(k-S+1)P_S(k)\nabla f(k-S+1)}; \quad (15)$$

$$\theta_S(k) = \theta_{S-1}(k) + \frac{P_{S-1}(k)\nabla f(k)}{\lambda + \nabla f^T(k)P_{S-1}(k)\nabla f(k)} \times$$

$$\times (y(k) - \theta_{S-1}^T(k)\nabla f(k)); \quad (16)$$

$$P_S(k) = \frac{1}{\lambda}\cdot\left[ P_{S-1}(k-1) - \frac{P_{S-1}(k-1)\nabla f(k)\nabla f^T(k)P_{S-1}(k-1)}{\lambda + \nabla f^T(k)P_{S-1}(k-1)\nabla f(k)}\right]. \quad (17)$$

В принципе оценки, получаемые с помощью алгоритмов (10)–(13) и (14)–(17), совпадают. Однако применение алгоритма (14)–(17) оказывается более предпочтительным с точки зрения удобства вычислений (например удаление устаревшей информации в (16) приводит к меньшему количеству арифметических операций при вычислении (17)).

### Моделирование

Изучалась эффективность работы алгоритма обучения(10)–(13) при построении нейронных моделей различных нелинейных динамических объектов. В частности, на рис. 1–3 представлены результаты моделирования объекта, описываемого уравнением [4, 11]

$$y(k) = 0{,}725\beta \cdot \left(\frac{16u(k-1) + 8y(k-1)}{\beta(3+4u(k-1)) + 4y^2(k-1)}\right) + 0{,}2u(k-1) + 0{,}2y(k-1), \quad (18)$$

при отсутствии и наличии помехи измерений $\xi(k)$ (1).

Условия эксперимента совпадали с теми, которые описаны в [4], то есть входной сигнал $u(k)$ и помеха $\xi(k)$, получаемые с помощью датчика случайных чисел, представляли собой стационарные $p$-равномерно распределенные в интервале [–1; 1] и [–0,25; 0,25] соответственно случайные последовательности. Построение модели осуществлялось на основании 5000 обучаемых пар. Все кривые отражают зависимость $y(k)$ от $u(k-1)$ при $y(k-1) = 0$.

На рис. 1, 2 сплошные линии соответствуют выходному сигналу объекта $y(k)$, линии с крестиками и кружками — выходному синтезу модели, обучаемой по алгоритму (10)–(13) при выборе $\lambda = 1$ и $\lambda = 0{,}995$ соответственно.

На рис. 1 приведены результаты идентификации объекта (18) при различных значениях параметров $\beta$ и отсутствии помех измерений.

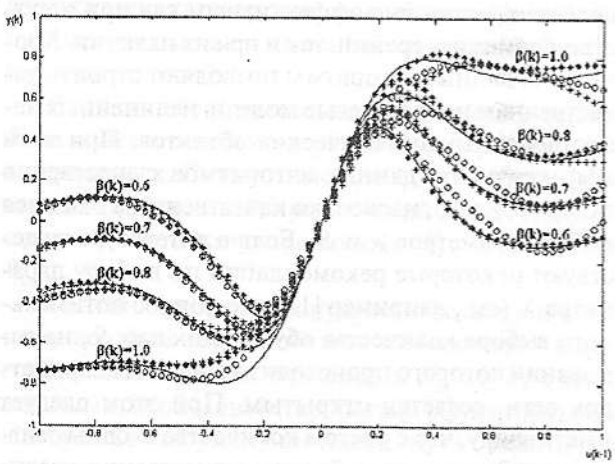


Рис. 1

На рис. 2, 3 показаны результаты идентификации при наличии помех измерения $\xi(k)$. Рис. 2 отражает работу алгоритма обучения (10)–(13), а рис. 3 — алгоритма (14)–(17).

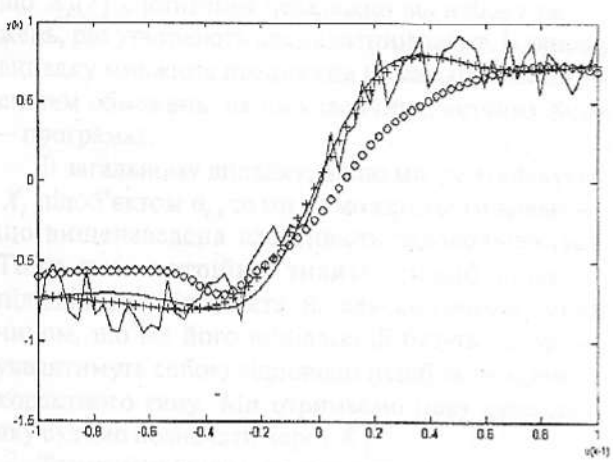

Рис. 2

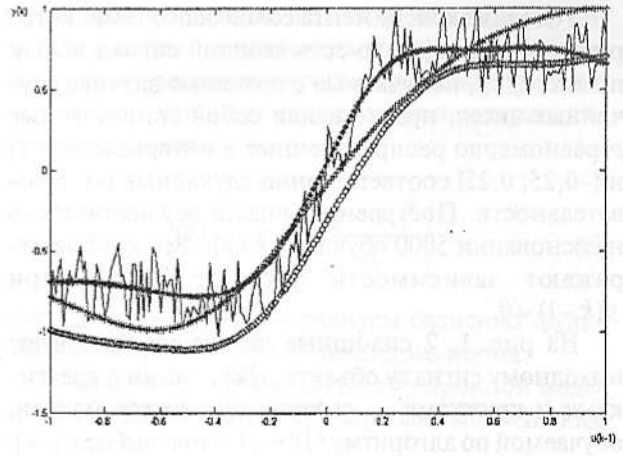

Рис. 3

**Выводы**

Как следует из приведенных результатов, использование алгоритмов обучения (10)–(13) и (14)–(17) является достаточно эффективным как при отсутствии помех измерений, так и при их наличии. Кроме того, данные алгоритмы позволяют строить качественные нейросетевые модели нелинейных нестационарных динамических объектов. При этом эффективность данных алгоритмов существенно зависит от того, насколько качественным оказался выбор параметров $\lambda$ и $S$. Если в литературе существуют некоторые рекомендации по выбору параметра $\lambda$ (см., например [12]), то вопрос оптимального выбора количества обучающих пар $S$, на основании которого происходит коррекция параметров сети, остаётся открытым. При этом следует иметь ввиду, что с ростом количества входных сигналов исследуемого объекта, существенно увеличивается число нейронов сети, приводящее к резкому возрастанию вычислительной сложности реализации нейросетевой модели. Поэтому алгоритмы типа (10)–(13) и (14)–(17) являются наиболее эффективными при построении моделей нестационарных нелинейных объектов с небольшим количеством входных сигналов.

**Список литературы:** 1. *Nelles O.* Nonlinear system identification. Berlin: Springer, 2001. 785p. 2. *Руденко О.Г., Бодянский Е.В.* Основы теории искусственных нейронных сетей. Харьков: Телетех, 2002. 317с. 3. *Sohilling R.J., Carroll J.J., Al-Ajlouni A.F.* Approximation of nonlinear systems with radial basis function neural networks //IEEE Trans. on Neural Networks. 2001. V.12 - №1. P. 1-15. 4. *Руденко О.Г., Бессонов А.А.* Идентификация нелинейных нестационарных объектов в реальном времени с помощью радиально-базисных сетей //Кибернетика и системный анализ. 2003. №6. С.177-185. 5. *Chen S., Billings S.A.* Neural networks for nonlinear dynamic system modeling and identification // Int. J. Control. 1992. V.56. №2. P. 319-346. 6. *Zhu S., Billings S.A.* Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks // Int. J. Control. 1996. V.64. №5. P. 871-886. 7. *Chen S., Chng E.S., Alkadhimi K.A.* Regularized orthogonal least squares algorithm for constructing radial basis function networks // Int. J. Control. 1996. V.64. №5. P. 829-837. 8. *Chng E.S., Chen S., Mulgrew B.* Gradient radial basis function networks for nonlinear and nonstationary time series prediction // IEEE Trans. on Neural Networks. 1996. V.7 №1. P. 190-194. 9. *Fung Ch.F., Billings S.A., Luo W.* On-line supervised adaptiv training using radial basis function networks // Neural Networks. 1996. V.9 №3. P. 1597-1617. 10. *Руденко О.Г., Теренковский И.Д., Штефан А., Ода Г.А.* Модернизированный алгоритм текущего регрессивного анализа в задачах идентификации и прогнозирования // Радиоэлектроника и информатика. 1998. №4. С. 58-61. 11. *Nerendra K.S., Parthasarathy K.* Identification and control of dynamical system using neural networks // IEEE Trans. on Neural Networks. 1990. V.1 №1. P. 4-26. 12. *Куник Е.Г., Коваленко А.Н., Ляшенко С.А.,* Аль Сади Ф.М. О выборе параметра взвешивания информации при построении нестационарных регрессионных моделей // АСУ и приборы автоматики. 2002. Вып.116. С. 109-114.

*Поступила в редколлегию 19.11.2004*