ISSN 0868–5886 *НАУЧНОЕ ПРИБОРОСТРОЕНИЕ, 2004, том 14, № 3, с. 11–32*

ОБЗОРЫ

УДК 681.51+ 621.391

© Г. Ф. Малыхина, А. В. Меркушева

МЕТОД КОНТРОЛЯ СОСТОЯНИЯ ПОДСИСТЕМЫ (ОБЪЕКТА) ПРИ НЕПОЛНОЙ ИЗМЕРИТЕЛЬНОЙ ИНФОРМАЦИИ О СОВОКУПНОСТИ ПАРАМЕТРОВ, ОПРЕДЕЛЯЮЩИХ ЕЕ ДИНАМИКУ. II. НЕЙРОННЫЕ СЕТИ, ОТРАЖАЮЩИЕ ДИНАМИКУ ВХОДНОЙ ИНФОРМАЦИИ И ПОСТРОЕННЫЕ НА ПРИНЦИПЕ ОБРАТНЫХ СВЯЗЕЙ (РЕКУРРЕНТНЫЕ СЕТИ)

В информационно-измерительных системах (ИИС) и информационно-управляющих системах (ИУС), отражающих состояние контролируемого объекта (подсистемы), существенна проблема отображения в условиях отсутствия воздействия некоторых параметров состояния на датчики измерительной системы, т. е. при неполной измерительной информации. Решение этой проблемы получено на основе анализа уравнений динамики системы объект—ИИС (в пространстве параметров состояния) и использования алгоритмов нейронных сетей (НС), способных отражать изменение входных данных во времени. Вторая (из 3) часть статьи рассматривает структуру и обучение НС, построенных на принципе обратных связей и называемых рекуррентными.

ВВЕДЕНИЕ В РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ

Кроме известной сложности проблемы обработки и анализа нестационарного сигнала в информационно-измерительных системах (ИИС), решению которой посвящен ряд работ [1–4], значительные трудности возникают при интерпретации многомерного сигнала параметров состояния контролируемого объекта в случае, если часть его компонент находится вне зоны чувствительности измерительных датчиков. На решение этой проблемы неполной измерительной информации о состоянии динамического объекта (подсистемы) направлен метод, который основан на представлении уравнений динамики и наблюдения контролируемого объекта в рекуррентной форме и на использовании нейронной сети (НС), отражающей динамику входной информации. В [5] применительно к общей задаче показаны особенности темпоральных НС прямого распространения.

Для метода контроля параметров состояния динамического объекта средствами ИИС при неполной информации здесь анализируются НС, построенные на принципе обратных связей (рекуррентные сети). В отличие от темпоральных НС прямого распространения, в которых реакция на динамику входных данных связана с кратковременной памятью, реализованной на сосредоточенных (в последовательности) или распределенных элементах временно́й задержки (ЭВЗ), НС рекуррентного типа отличаются тем, что в их структуре используются обратные связи, охватывающие всю НС или отдельные ее слои. Такие рекуррентные нейронные сети (РНС) с глобальными или локальными обратными связями (ОС) сравнительно с темпоральными НС прямого распространения имеют несколько меньшие требования к памяти и рассматриваются как адаптивное устройство отображения вход—выход.

В круг задач, решаемых с помощью РНС, входит нелинейное предсказание, адаптивное выравнивание каналов связи (проблема эквализации), обработка речи, контроль производственно-технологических процессов по критерию оптимальной динамики и элементы технической диагностики.

Особенности применения глобальной обратной связи в структуре РНС можно проследить на 4 видах сети. Эти РНС имеют определенную общность: они включают "статический" многослойный персептрон (МСП) и используют способность МСП производить нелинейное отображение. Согласно Тсою и Бэку (Tsoi, Back [6]), структура, характерная для РНС общего вида на основе МСП, включает две линии ЭВЗ на $(q - 1)$ единиц ЭВЗ. Через одну линию ЭВЗ на МСП поступает входной сигнал $u(n)$, а через другую — сигнал обратной связи $y(n + 1)$ с выхода сети (рис. 1) [1)].

[1)] Выход сети $y(n + 1)$ задержан относительно входных данных на единицу ЭВЗ.

Такая структура представляет отображение вход—выход, эквивалентное нелинейной авторегрессии (НАР) с входом на линии ЭВЗ (НАР_ВЗ). На вход НАР_ВЗ подается q-мерный вектор $\mathbf{u}(n) = [u(1), u(2),\ldots, u(n - q + 1)]^T$ и аналогичный q-вектор задержанных значений выхода РНС $\mathbf{y}(n) = [y(1), y(2),\ldots, y(n - q + 1)]^T$. Так что выход РНС $y(n + 1)$ определяется нелинейной авторегрессионной зависимостью вида

$$y(n+1) = F(y(1), y(2),\ldots, y(n-q+1), u(1), u(2),\ldots, u(n-q+1)). \quad (1)$$

Модель НАР_ВЗ отражает особенность общей ОС, однако РНС может использовать и локальные ОС, которые охватывают один или несколько отдельных слоев МСП.

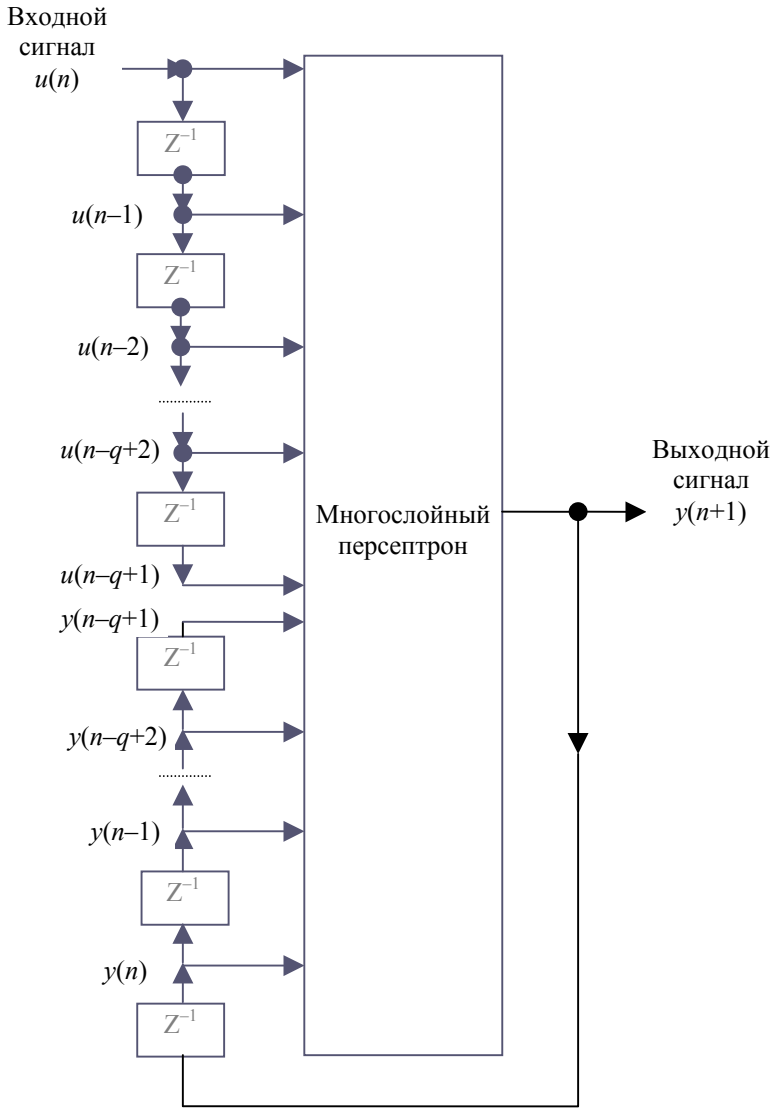


Рис. 1. Модель нелинейной авторегрессии с временно́й задержкой внешнего входа (НАР_ВЗ) на основе РНС

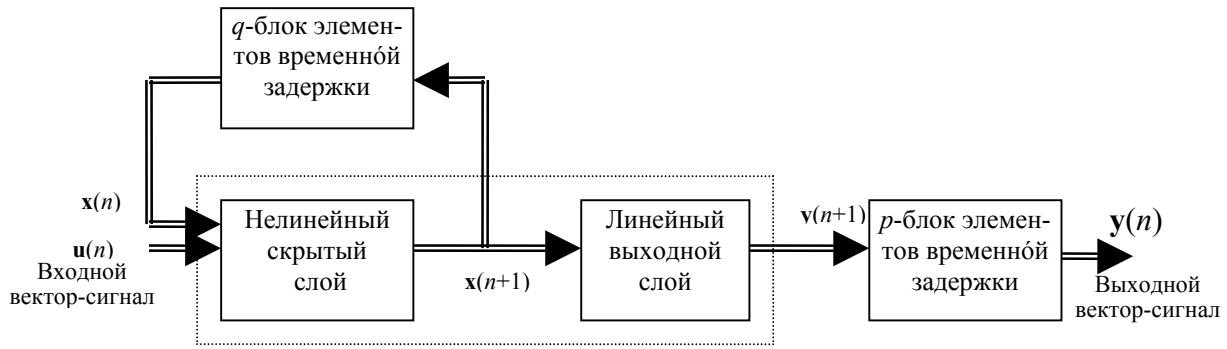


Рис. 2. Модель пространства состояний — прототип РНС Элмана

РНС И МОДЕЛЬ ПРОСТРАНСТВА СОСТОЯНИЙ

Для целей метода интерпретации измерений при недостаточной информации[2] полезно рассмотреть структуру НС (рис. 2), которая служит прототипом РНС Элмана (Elman [7]), является РНС на основе рекуррентного МСП (Р_МСП), а также хорошо моделирует пространства состояний. Скрытые нейроны определяют состояние сети. Выход скрытого слоя $\mathbf{x}(n+1)$ подается через q-блок ЭВЗ [3] на вход РНС. Этот же вектор-сигнал $\mathbf{x}(n+1)$ через выходной слой, состоящий из линейных нейронов, в виде вектора $\mathbf{y}(n+1)$ подается через p-блок ЭВЗ на выход РНС (вектор $\mathbf{y}(n)$). Таким образом, входной слой этой РНС состоит из объединения q узлов обратной связи и m узлов источника внешнего воздействия на сеть (m равно размерности вектора $\mathbf{u}(n)$, $\mathbf{u}(n) = [u_1(n), u_2(n),\ldots, u_m(n)]^{\mathrm{T}}$), а число ЭВЗ в блоке обратной связи (q) равно размерности вектора $\mathbf{x}(n) = [x_1(n), x_2(n),\ldots, x_q(n)]^{\mathrm{T}}$. Величина q определяет порядок РНС.

Описание в дискретном времени динамики модели пространства состояний [4] определяется уравнениями:

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n)), \qquad (2)$$

$$\mathbf{y}(n) = \mathbf{C}\cdot\mathbf{x}(n), \qquad (3)$$

где $\mathbf{f}$ — нелинейная функция, характеризующая скрытый слой, а $\mathbf{C}$ — матрица синаптических весов, определяющая выходной слой.

Принцип такого описания НС вполне соответствует модификации Элмана в виде простой РНС [7] и модификации Пускориуса и Фельдкампа (Puskorius, Feldkamp) в виде рекуррентного многослойного персептрона (Р_МСП) [8]. Структура Элмана (рис. 3, а) содержит рекуррентные связи от скрытых нейронов через блок ЭВЗ (слой контекстных элементов) на входные узлы ОС. На другую часть входных узлов подается (как и в описанной выше общей структуре) вектор-сигнал ("управляющий" вектор).

Основой простой РНС Элмана служит однослойный персептрон. РНС на основе Р_МСП (рис. 3, б) имеет одну или несколько ОС, каждая из которых охватывает один слой Р_МСП, т. е. подает через блок ЭВЗ выходной сигнал слоя на его входные узлы, общее число которых позволяет передавать на вход этого слоя также и выход предыдущего слоя [5].

При использовании индексов I, II и вых. для первого, второго и выходного слоев Р_МСП и тех же индексов для нелинейной функции $\boldsymbol{\varphi}$ преобразования сигнала этими слоями модель пространства состояний для Р_МСП описывается соотношениями:

[2] Этот метод рассмотрен в подготовленной к публикации 3-й части статьи и основан на рекуррентной форме модели динамики объекта в пространстве состояний.

[3] Здесь и далее q-блоком ЭВЗ называется параллельное включение q ЭВЗ. За q-ЛВЗ сохраняется обозначение для линии из q элементов единичной временнóй задержки с равномерно расположенными отводами для выхода сигнала с различной степенью задержки.

[4] Вектор параметров состояния $\mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_q(n)]^{\mathrm{T}}$ в этом случае относится к моменту времени $t = t_0 + nT$, где T — шаг отсчета времени эволюции динамической системы.

[5] Более сложный вариант структуры, не нашедший пока применения, предложен Гайлсом [9]. В нем использован "нейрон второго порядка", возбуждение которого осуществляется за счет произведений компонент входного сигнала и сигнала обратной связи, поступающего на его входные узлы.

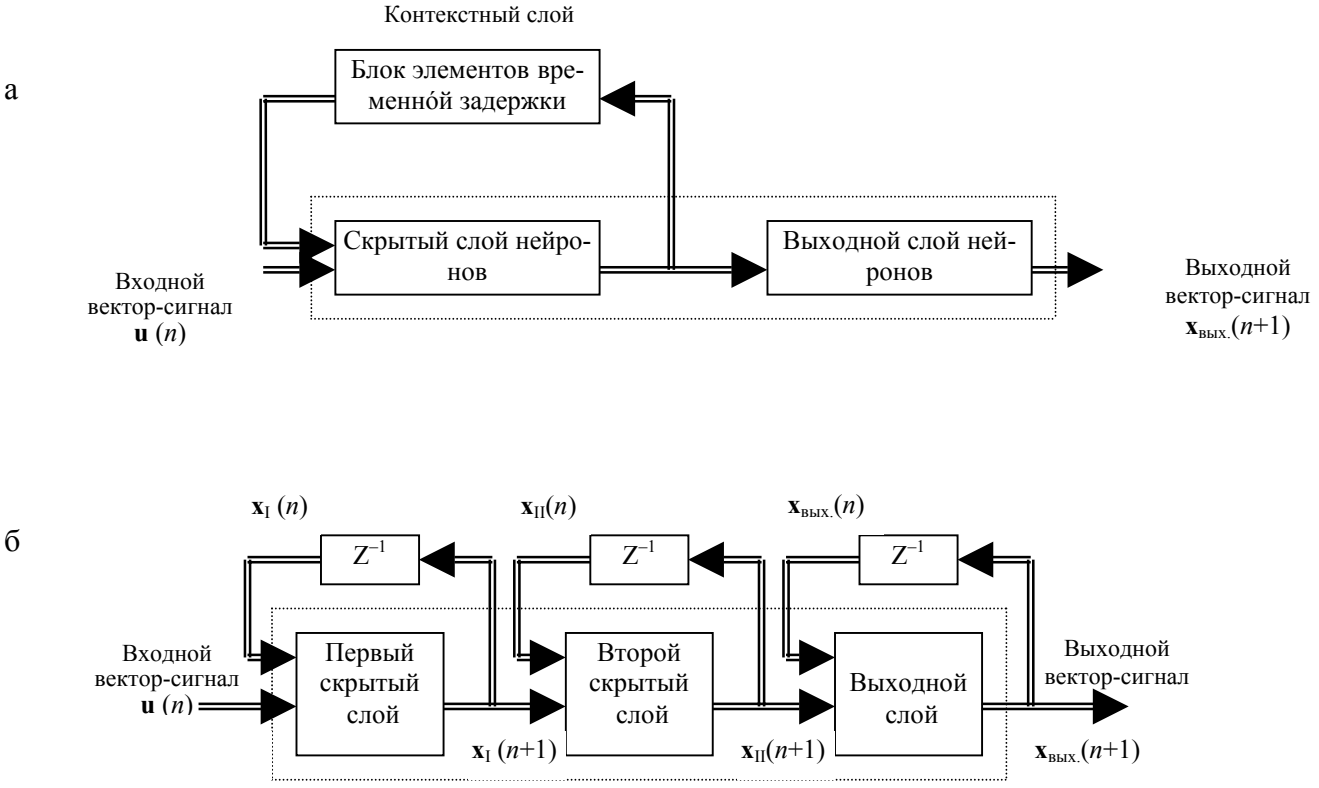


Рис. 3. Простая структура РНС на персептроне с одним скрытым слоем (а) и РНС общего вида на многослойном персептроне с несколькими скрытыми слоями (б)

$$\mathbf{x}_{I}(n+1) = \boldsymbol{\varphi}_{I}(\mathbf{x}_{I}(n), \mathbf{u}(n));$$
$$\mathbf{x}_{II}(n+1) = \boldsymbol{\varphi}_{II}(\mathbf{x}_{II}(n), \mathbf{x}_{I}(n)); \qquad (4)$$
$$\mathbf{x}_{вых.}(n+1) = \boldsymbol{\varphi}_{вых.}(\mathbf{x}_{вых.}(n), \mathbf{x}_{k}(n+1)),$$

где k — число скрытых слоев в НС.

Описанная структура Р_МСП является обобщением НС Элмана и показывает расширенную трактовку модели пространства состояний, поскольку функции преобразования отдельных слоев Р_МСП могут быть различными.

ЭЛЕМЕНТЫ ТЕОРИИ МОДЕЛИ ПРОСТРАНСТВА СОСТОЯНИЙ

Как отмечено выше, метод интерпретации данных ИИС при недостатке измерительной информации существенно использует концепцию модели пространства состояний. Согласно Сонтагу (Sontag [10]), состояние динамической системы (ДС) определяется как совокупность величин, которые, объединяя всю информацию относительно прошлого поведения ДС совместно с данными о внешнем воздействии на систему, позволяют описать будущее ДС. При анализе нелинейной ДС (в форме с дискретным временем), которая контролируется с помощью некоторой ИИС, используется представление ДС уравнениями:

$$\mathbf{x}(n+1) = \boldsymbol{\varphi}(\mathbf{W}_{a} \cdot \mathbf{x}(n) + \mathbf{W}_{b} \cdot \mathbf{u}(n)); \qquad (5)$$

$$\mathbf{y}(n) = \mathbf{C} \cdot \mathbf{x}(n), \qquad (6)$$

где q-мерный вектор $\mathbf{x}(n)$ имеет компонентами q параметров состояния; m-мерный вектор $\mathbf{u}(n) = [u_1(n), u_2(n),\ldots, u_m(n)]^{T}$ — входное воздействие на ДС; p-мерный вектор $\mathbf{y}(n) = [y_1(n), y_2(n),\ldots, y_p(n)]^{T}$ — вектор-сигнал выхода ДС. Вектор $\mathbf{y}(n)$ — это из-

мерительные данные (получаемые в ИИС), которые имеют совокупную эффективность системы датчиков, описываемую матрицей **C** размерности ($p \times q$). В (5), (6) ($q \times q$)-матрица $\mathbf{W}_a$ представляет синаптические веса q нейронов скрытого слоя, которые связаны обратной связью с узлами входного слоя; ($q \times m$)-матрица $\mathbf{W}_b$ представляет веса тех скрытых нейронов, которые связаны с узлами источника воздействия, находящимися во входном слое (считается, что элементы смещения также включены в рассматриваемые матрицы); ($p \times q$)-матрица **C** представляет синаптические веса p линейных нейронов в выходном слое. Нелинейная вектор-функция φ(.) с идентичными компонентами определяет отображение в q-мерном пространстве состояний

$$\mathbf{R}^q \xrightarrow{\varphi} \mathbf{R}^q$$

в форме

$$\varphi \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_q \end{bmatrix} = \varphi(\mathbf{x}) = \begin{bmatrix} \varphi(x_1) \\ \varphi(x_2) \\ \dots \\ \varphi(x_q) \end{bmatrix}. \quad (7)$$

Пространства $\mathbf{R}^m$, $\mathbf{R}^q$, $\mathbf{R}^p$ называют входным пространством, пространством состояний и выходным пространством, а величину q — порядком ДС. Рис. 2 представляет сигнально-потоковый граф динамической системы по рекуррентной модели пространства состояний порядка q с m входами и p выходами. Уравнение (6) является уравнением измерения.

В описанной структуре Р_МСП параметры пространства состояний (скрытого слоя) определяются только теми нейронами, выходы которых идут через блок ЭВЗ на обратную связь; к ним не относятся нейроны выходного слоя. Другой вид рекуррентной структуры на персептроне с одним или несколькими слоями дан на рис. 3, где выход НС или отдельного слоя (как на рис. 3, б) подается в виде обратной связи на вход НС или на вход соответствующего слоя.

УПРАВЛЯЕМОСТЬ И НАБЛЮДАЕМОСТЬ В МОДЕЛИ ПРОСТРАНСТВА СОСТОЯНИЙ ДЛЯ ДИНАМИЧЕСКОЙ СИСТЕМЫ

Как модели пространства состояний ДС, в которых состояние представляется набором компонент вектора-сигнала нейронов скрытого слоя (рис. 2), могут быть представлены многие РНС. Поэтому важны такие признаки модели ДС, как управляемость и наблюдаемость [11]. РНС (как любая ДС) считается управляемой, если любое ее состояние достижимо за несколько временны́х шагов при любом исходном состоянии. Наблюдаемость — это возможность определения текущего состояния НС из конечного набора измерений сигналов входа и выхода.

Наиболее важны локальные формы управляемости и наблюдаемости, в частности в окрестности точки равновесия [12]. Если за точку равновесия принять точку **x**, то для нее будет выполняться условие $\mathbf{x} = \varphi(\mathbf{A}\cdot\mathbf{x} + \mathbf{B}\cdot\mathbf{u})$. Изменением начала отсчета можно привести точку равновесия в точку $\mathbf{x} = \mathbf{0}$, $\mathbf{u} = \mathbf{0}$, для которой справедливо условие $\mathbf{0} = \varphi(0)$. Кроме того, суть условий управляемости и наблюдаемости, согласно [13], достаточно проанализировать для случая одномерных входа и выхода, для которого уравнения (5), (6) принимают вид:

$$\mathbf{x}(n+1) = \varphi(\mathbf{W}_a\cdot\mathbf{x}(n) + \mathbf{w}_b\cdot u(n)), \quad (8)$$

$$y(n) = \mathbf{c}^{\mathrm{T}}\cdot\mathbf{x}(n), \quad (9)$$

где $\mathbf{w}_b$ и **c** — векторы; $y(n)$ и $u(n)$ — скалярные величины.

В окрестности точки равновесия ($\mathbf{x} = \mathbf{0}$, u = 0) уравнения (8), (9) можно линеаризовать, сохраняя один член в разложении Маклорена (разложении Тейлора в окрестности нуля):

$$\delta\mathbf{x}(n+1) = \varphi'(\mathbf{0})\,\mathbf{W}_a\cdot\delta\mathbf{x}(n) + \varphi'(\mathbf{0})\,\mathbf{w}_b\cdot\delta u(n), \quad (10)$$

где $\delta\mathbf{x}(n)$ и $\delta u(n)$ — малые смещения вектора состояния $\mathbf{x}(n)$ и управления $u(n)$; $\varphi'(\mathbf{0})$ — якобиан от $\varphi(\mathbf{v})$ относительно своего аргумента **v** в точке $\mathbf{v} = \mathbf{0}$. Линеаризованная система уравнений имеет вид

$$\delta\mathbf{x}(n+1) = \mathbf{A}\cdot\delta\,\mathbf{x}(n) + \mathbf{b}\cdot\delta u(n); \quad (11)$$

$$\delta y(n) = \mathbf{c}^{\mathrm{T}}\cdot\delta\mathbf{x}(n), \quad (12)$$

где ($q \times q$)-матрица **A** и q-вектор **b** определяются соотношениями:

$$\mathbf{A} = \varphi'(\mathbf{0})\,\mathbf{W}_a, \quad (13)$$

$$\mathbf{b} = \varphi'(\mathbf{0})\,\mathbf{w}_b. \quad (14)$$

Уравнения (11), (12) имеют линеаризованную форму, поэтому для них можно использовать результаты из теории управления [11, 12], которые определяют условия управляемости и наблюдаемости линейной динамической системы.

Локальная управляемость

Повторное использование уравнения (11) дает соотношения

$$\begin{aligned}
&\delta\mathbf{x}(n+1) = \mathbf{A}\cdot\delta\mathbf{x}(n) + \mathbf{b}\cdot\delta u(n);\\
&\delta\mathbf{x}(n+2) = \mathbf{A}\cdot\delta\mathbf{x}(n+1) + \mathbf{b}\cdot\delta u(n+1);\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \qquad (15)\\
&\delta\mathbf{x}(n+q) = \mathbf{A}^q\cdot\mathbf{b}\cdot\delta\mathbf{x}(n) + \mathbf{A}^{q-1}\cdot\mathbf{b}\cdot\delta u(n+q-1) +\\
&\qquad + \dots + \mathbf{A}\cdot\mathbf{b}\cdot\delta u(n+1) + \mathbf{b}\cdot\delta u(n),
\end{aligned}$$

где q — размерность пространства состояний (вектора $\mathbf{x}$).

Таким образом, для ДС, описываемой линеаризованным уравнением (11), управляемость эквивалентна тому, чтобы матрица $\mathbf{M}_{\text{упр.}}$ (называемая матрицей управляемости) имела полный ранг (q), т. е. чтобы ее определитель не был равен нулю:

$$\mathbf{M}_{\text{упр.}} = [\mathbf{A}^{q-1}\cdot\mathbf{b}, \ldots, \mathbf{A}\cdot\mathbf{b}, \mathbf{b}],$$
$$\text{Det}(\mathbf{M}_{\text{упр.}}) \neq 0. \qquad (16)$$

Управляемость применительно к РНС

Рекуррентная нейронная сеть, описываемая уравнениями (8), (9), управляется последовательностью входных воздействий в виде вектора $\mathbf{u}_q$, который определен выражением

$$\mathbf{u}_q(n) = [u(n), u(n+1), \ldots, u(n+q-1)]^{\text{T}}. \qquad (17)$$

Следовательно, можно рассмотреть отображение

$$\text{G}(\mathbf{x}(n), \mathbf{u}_q(n)) = (\mathbf{x}(n), \mathbf{x}(n+q)), \qquad (18)$$

где G — нелинейное преобразование в пространстве размерности $2q$ ($\mathbf{R}^{2q} \xrightarrow{\text{G}} \mathbf{R}^{2q}$).

Из (18) следует, что

1) состояние $\mathbf{x}(n+q)$ является нелинейной функцией своего прошлого значения $\mathbf{x}(n)$ и входных сигналов $u(n)$, $u(n+1)$,…, $u(n+q-1)$;

2) якобиан от $\mathbf{x}(n+q)$ по $\mathbf{u}_q(n)$, оцененный в начале координат (0, 0), равен матрице управляемости $\mathbf{M}_{\text{упр.}}$ из (16).

С другой стороны якобиан отображения G относительно $\mathbf{x}(n)$ и $\mathbf{u}_q(n)$ можно выразить по правилу его определения:

$$\mathbf{J}^{\text{упр.}}_{(0,0)} = \begin{bmatrix} \{\partial\mathbf{x}(n)/\partial\mathbf{x}(n)\}_{(0,0)} & \{\partial\mathbf{x}(n+q)/\partial\mathbf{x}(n)\}_{(0,0)} \\ \{\partial\mathbf{x}(n)/\partial\mathbf{u}_q(n)\}_{(0,0)} & \{\partial\mathbf{x}(n+q)/\partial\mathbf{u}_q(n)\}_{(0,0)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{X} \\ \mathbf{0} & \mathbf{M}_{\text{упр.}} \end{bmatrix}, \qquad (19)$$

где $\mathbf{I}$ — единичная матрица; $\mathbf{0}$ — нулевая матрица; $\mathbf{X}$ — некоторая матрица, которая не влияет на величину якобиана.

Определитель $\text{Det}(\mathbf{J}^{\text{упр.}}_{(0,0)}) = \text{Det}(\mathbf{I}) \cdot \text{Det}(\mathbf{M}_{\text{упр.}}) = \text{Det}(\mathbf{M}_{\text{упр.}})$, т. е. если $\mathbf{M}_{\text{упр.}}$ имеет полный ранг, то и якобиан имеет полный ранг. В этом случае будет существовать обращение уравнения (18)

$$(\mathbf{x}(n), \mathbf{x}(n+q)) = \text{G}^{-1}(\mathbf{x}(n), \mathbf{u}_q(n)). \qquad (20)$$

Уравнение (20) устанавливает существование последовательности управлений $\{u_q(n)\}$, которые локально могут перевести состояние ДС $\mathbf{x}(n)$ в состояние $\mathbf{x}(n+q)$ за q временны́х шагов.

Локальная наблюдаемость

Повторное совместное использование уравнений (11) и (12) позволяет получить соотношения:

$$\begin{aligned} &\delta y(n) = \mathbf{c}^{\text{T}}\cdot\delta\mathbf{x}(n); \\ &\delta y(n+1) = \mathbf{c}^{\text{T}}\cdot\delta\mathbf{x}(n+1) = \\ &\qquad = \mathbf{c}^{\text{T}}\cdot\mathbf{A}\cdot\delta\mathbf{x}(n) + \mathbf{c}^{\text{T}}\cdot\mathbf{b}\cdot\delta u(n); \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\delta y(n+q-1) = \mathbf{c}^{\text{T}}\cdot\mathbf{A}^{q-1}\cdot\delta\mathbf{x}(n) + \\ &\qquad + \mathbf{c}^{\text{T}}\cdot\mathbf{A}^{q-2}\cdot\mathbf{b}\cdot\delta u(n) + \\ &\qquad + \ldots + \mathbf{c}^{\text{T}}\cdot\mathbf{A}\cdot\mathbf{b}\cdot\delta u(n+q-3) + \\ &\qquad + \mathbf{c}^{\text{T}}\cdot\mathbf{b}\cdot\delta u(n+q-2). \end{aligned} \qquad (21)$$

Так что линеаризованная система, описываемая уравнениями (11), (12), — наблюдаемая при условии, если матрица $\mathbf{M}_{\text{набл.}}$ имеет ранг q, т. е. полный ранг:

$$\mathbf{M}_{\text{набл.}} = [\mathbf{c}, \mathbf{c}\cdot\mathbf{A}^{\text{T}}, \ldots, \mathbf{c}\cdot(\mathbf{A}^{\text{T}})^{q-1}],$$
$$\text{Det}(\mathbf{M}_{\text{набл.}}) \neq 0. \qquad (22)$$

Теперь положим, что РНС описывается уравнениями (8) и (9) и получает воздействие

$$\mathbf{u}_{q-1}(n) = [u(n), u(n+1),\ldots, u(n+q-1)]^{\text{T}} \qquad (23)$$

и что вектор-сигнал выхода, определяемого начальным состоянием $\mathbf{x}(n)$ и последовательностью управлений (воздействий на НС) $\mathbf{u}_{q-1}(n)$, будет определяться выражением

$$\mathbf{y}_q(n) = [y(n), y(n+1),\ldots, y(n+q-1)]^{\text{T}}. \qquad (24)$$

Тогда можно рассмотреть отображение H

$$\mathbf{R}^{2q-1} \xrightarrow{\text{H}} \mathbf{R}^{2q-1},$$

такое что

$$\text{H}(\mathbf{u}_{q-1}(n), \mathbf{x}(n)) = (\mathbf{u}_{q-1}(n), \mathbf{y}_q(n)). \qquad (25)$$

Показано [13], что якобиан от $\mathbf{y}_q(n)$ по $\mathbf{x}(n)$ равен матрице наблюдаемости $\mathbf{M}_{\text{набл.}}$ из (22). Поэтому если выразить якобиан от H относительно $\mathbf{u}_{q-1}(n)$ и $\mathbf{x}(n)$, то получится

$$\mathbf{J}^{\text{набл}}_{(0,0)} = \begin{bmatrix} \{\partial\mathbf{u}_{q\text{-}1}(n)/\partial\mathbf{u}_{q\text{-}1}(n)\}_{(0,0)} & \{\partial\mathbf{y}_q(n)/\partial\mathbf{u}_{q\text{-}1}(n)\}_{(0,0)} \\ \{\partial\mathbf{u}_{q\text{-}1}(n)/\partial\mathbf{x}(n)\}_{(0,0)} & \{\partial\mathbf{y}_q(n)/\partial\mathbf{x}(n)\}_{(0,0)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{X} \\ \mathbf{0} & \mathbf{M}_{\text{набл.}} \end{bmatrix}. \qquad (26)$$

И следовательно,

$$\text{Det}(\mathbf{J}^{\text{набл.}}_{(0,0)}) = \text{Det}(\mathbf{I})\cdot\text{Det}(\mathbf{M}_{\text{набл.}}) = \text{Det}(\mathbf{M}_{\text{набл.}}) \neq 0.$$

Таким образом, $\mathbf{J}^{\text{набл.}}_{(0,0)}$ имеет полный ранг, и соотношение (25) обратимо:

$$(\mathbf{u}_{q-1}(n), \mathbf{x}(n)) = \mathrm{H}^{-1}(\mathbf{u}_{q-1}(n), \mathbf{y}_q(n)). \quad (27)$$

Следовательно, $\mathbf{x}(n)$ выражается с помощью нелинейной функции H^{-1} через $\mathbf{u}_{q-1}(n)$ и $\mathbf{y}_q(n)$, а обратная ей нелинейная функция H отражает наблюдение РНС в соответствии с (25).

Модель пространства состояний для РНС с одним входом и одним выходом, которая описывается уравнениями (8) и (9), можно преобразовать в эквивалентную ей форму модели типа вход—выход. Уравнения модели пространства состояний позволяют выразить $y(n + q)$ через $\mathbf{x}(n)$ и $\mathbf{u}_q(n)$:

$$y(n + q) = \Phi(\mathbf{x}(n), \mathbf{u}_q(n)), \quad (28)$$

где q — размерность пространства состояний РНС; Ф — отображение

$$\mathbf{R}^{2q} \xrightarrow{\Phi} \mathbf{R}.$$

При условии, что модель РНС обладает свойством наблюдаемости, имеет место соотношение типа

$$\mathbf{x}(n) = \mathbf{\Psi}(\mathbf{y}_q(n), \mathbf{u}_{q-1}(n)), \quad (29)$$

где $\mathbf{\Psi}$ — отображение

$$\mathbf{R}^{2q} \xrightarrow{\mathbf{\Psi}} \mathbf{R}^{2q}.$$

Подстановка (29) в (28) дает выражение

$$y(n + q) = \Phi(\mathbf{\Psi}(\mathbf{y}_q(n), \mathbf{u}_{q-1}(n)) = \mathrm{F}(\mathbf{y}_q(n), \mathbf{u}_q(n)). \quad (30)$$

При этом $\mathbf{u}_{q-1}$ содержится в $\mathbf{u}_q$ в качестве его первых $(q–1)$ элементов, и отображение F

$$\mathbf{R}^{2q} \xrightarrow{\mathrm{F}} \mathbf{R}$$

объединяет отображения Ф и $\mathbf{\Psi}$.

Используя определения $\mathbf{y}_q(n)$ и $\mathbf{u}_q(n)$ по (23) и (24), уравнение (30) можно преобразовать к развернутой форме:

$$y(n + q) =$$

$$= \mathrm{F}(y(n),\ldots, y(n + q–1), u(n),\ldots, u(n – q + 1)). \quad (31)$$

Таким образом, показано существование отображения, которое связывает выход $y(n + 1)$ через свои предшествующие значения $y(n),\ldots, y(n + q–1)$, и текущие, и прошлые значения входного сигнала $u(n),\ldots, u(n – q + 1)$. Чтобы полученное представление вход—выход было эквивалентно модели пространства состояний по (8) и (9), РНС должна быть наблюдаема.

Практическая реализация описанной эквивалентности — это, в частности, то, что описанная выше модель НАР_ВЗ (МСП с двумя ЛЭВЗ на входе и с обратной связью с выходного нейрона (рис. 1)) фактически может имитировать соответствующую полностью рекуррентную модель пространства состояний, изображенную на рис. 2 (если принять $m = 1$ и $p = 1$). При этом поведение эквивалентных НС не будет отличаться с точки зрения преобразования вход—выход.

АЛГОРИТМЫ ОБУЧЕНИЯ

Как известно, применяются две формы обучения: обучение набором данных, в котором корректировка весов НС производится методом предъявления серии обучающих примеров (СОП), и режим последовательного обучения, при котором веса корректируются каждый раз после отдельного предъявления. Аналогичным образом обучаются РНС [14, 15]. Однако для РНС форма обучения СОП имеет смысл, близкий к последовательному обучению обычного МСП: настройка весов РНС производится после получения установившейся реакции выхода на поступающие на вход обучающие сигналы.

Непрерывное обучение применяется, когда нет возможности переустановить начальные состояния и требуется обучение в реальном времени. Отличительная черта непрерывного обучения — это то, что сеть обучается в то время, пока сигнал все еще обрабатывается сетью. Процесс обучения не прерывается, например, при использовании РНС для моделирования нестационарного сигнала. В этой ситуации непрерывное действие РНС не предоставляет удобного времени, чтобы остановить обучение и начать его заново с другими значениями начальных параметров сети.

Имея в виду эти две формы обучения, можно рассмотреть алгоритмы обучения, специфичные для РНС: алгоритм обратного распространения во времени (ОРВ), который предполагает возможность разворачивания темпоральной РНС в цепочку "статических" МСП. Такое разворачивание РНС дает возможность использовать обычный алгоритм обратного распространения (алгоритм ОРО [4]). Алгоритм ОРВ может выполняться способом СОП, последовательным способом для выполнения обучения в реальном времени или с помощью их комбинации. Алгоритм обучения реального времени выводится из модели пространства состояний.

Оба алгоритма обучения основаны на методе спуска по градиенту и используют текущую оценку функции стоимости (усредненного квадрата ошибок) для ее минимизации по набору синаптических весов НС. Оба алгоритма достаточно просты по логике исполнения, но могут медленно сходиться. Кроме того, они похожи в том, что представление сигнально-потоковым графом для алгоритма ОРВ может быть получено из перестановки представления сигнально-потокового графа для определенной формы рекуррентного алгоритма обучения реального времени [16].

Алгоритм обучения реального времени (непрерывного), основанный на градиентном спуске, использует минимальное количество информации — мгновенную оценку градиента функции стоимости (функции риска) по параметрам, которые подставляются в НС. При этом некоторое ускорение процесса обучения может быть достигнуто путем использования теории фильтра Калмана, которая позволяет более эффективно использовать обучающие данные.

Для улучшения процедуры обучения РНС Гайлсом (Giles [17]) предложено несколько эвристических правил, которые могут быть сведены к ряду практических рекомендаций.

- Обучающие примеры должны следовать лексикографическому порядку — наиболее короткие строки символов предъявляются сети в первую очередь.
- Обучение должно начинаться с малых обучающих образцов, а затем в ходе обучения их размер может наращиваться.
- Обновление величины синаптических весов НС должно происходить, только если ошибка отбрасываемого примера больше некоторого порога (выбираемого в контексте решаемой задачи).
- Во время обучения можно использовать некоторое экспоненциальное снижение величины весов ("распад" или релаксацию весов), что своеобразно служит грубым способом регуляризации, направленной на ограничение нормы весов НС.

Первое правило особенно полезно там, где оно применимо, оно смягчает проблему близкого к нулю значения градиента, которая возникает при использовании для обучения РНС метода градиентного спуска.

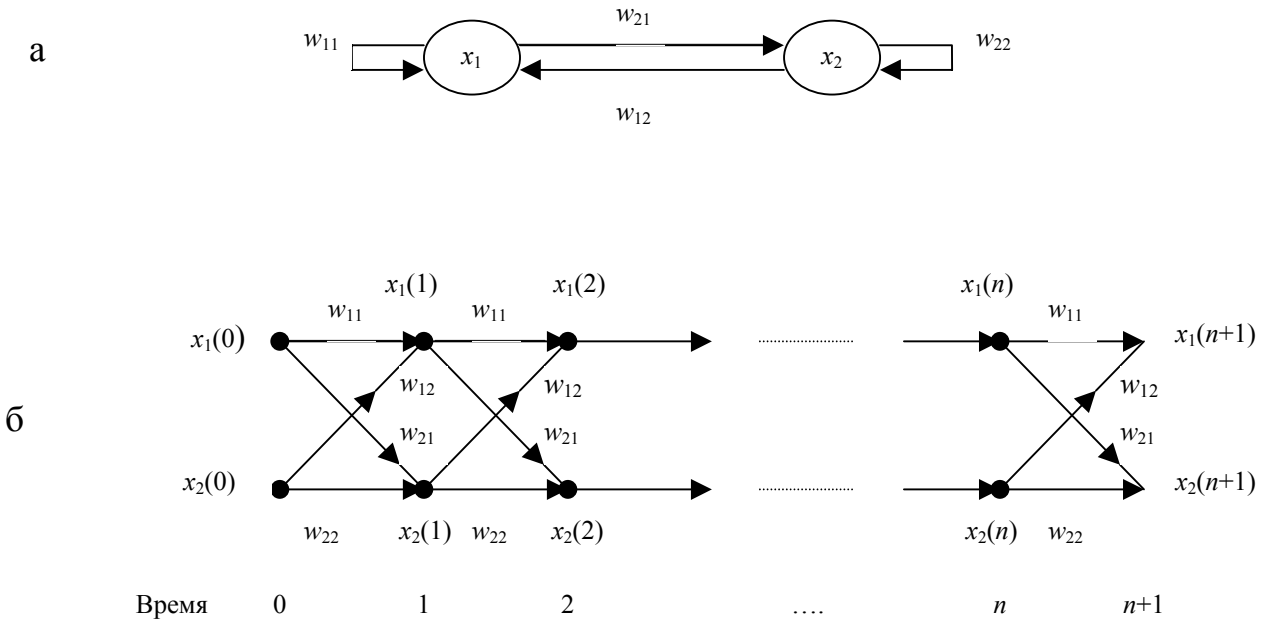


Рис. 4. Граф структуры РНС с двумя нейронами (а) и ее сигнально-потоковый граф (б). РНС на (б) развернута во времени

Обучение по алгоритму обратного распространения во времени

Алгоритм обратного распространение во времени (ОРВ) — расширение алгоритма ОРО [4] — является основным принципом реализации способа обучения РНС. Алгоритм ОРВ строится путем так называемого разворачивания время-зависимого функционирования РНС в многослойную (однородную) нейронную сеть с распространением вперед, структура которой наращивается на 1 слой за каждый шаг по времени работы исходной РНС. Так, если N — РНС, которую требуется обучать темпоральной задаче, начинающейся со времени n_0 и длящейся до момента n, а N* — это нейронная сеть с распространением вперед, получающаяся при разворачивании время-зависимого (темпорального) действия РНС N, то N* соотносится с исходной РНС N, как показано на рис. 4.

- Для каждого временно́го шага интервала (n_0, n), сеть N* имеет слой с k нейронами, где k — число нейронов, содержащихся в РНС N.
- В каждом слое сети N* содержится копия каждого нейрона сети N.
- Для каждого временно́го шага l ($l \in [n_0, n]$) синаптическая связь нейрона i в слое l (в развернутой сети) к нейрону j в слое $l + 1$ в сети N* является копией связи нейрона i к нейрону j в сети N.

На рис. 4 отображена упрощенная модель РНС (без ЭВЗ) из двух нейронов с обратными связями (эта сеть N представлена графом ее структуры) и преобразованная форма РНС в виде эквивалентной сети прямого распространения (сеть N* показана сигнально-потоковым графом на рис. 4, б). Эквивалентная форма сети прямого распространения удобна для выполнения обычного алгоритма ОРО, который служит эмуляцией алгоритма ОРВ для исходной рекуррентной сети N.

Обучение по алгоритму ОРВ в формате использования СОП

В этой модификации алгоритма ОРВ множество данных разделяется на независимые серии, каждая из которых представляет один из анализируемых видов временно́го (темпорального) образца (ТО). Если начало серии (ТО) — n_0, а конец ее — n_1, то общая функция риска (стоимости) определяется выражением:

$$E_{\text{общ.}}(n_0, n_1) = \frac{1}{2} \sum_{n=n_0}^{n_1} \sum_{j \in A} e_j^2(n), \qquad (32)$$

где A — множество индексов j, принадлежащих тем нейронам сети, для которых известны желаемые отклики выхода; $e_j(n)$ — сигнал ошибки выхода нейрона j, измеренный относительно желаемого значения выхода.

Чувствительность сети определяется как частные производные функции риска $E_{\text{общ.}}$ относительно весов сети. Чтобы определить чувствительность, можно использовать развитый Вильямсом и Пенгом (Williams, Peng [18]) алгоритм ОРВ для ОСП, который строится на обычном алгоритме ОРО с ОСП (или с обучением эпохами [4, 13]). Алгоритм основывается на следующих процедурах.

- Выполняется проход в прямом направлении через сеть для интервала $[n_0, n_1]$. Запоминается полный список входных данных, состояния сети (ее синаптических весов) и желаемого отклика за этот временно́й интервал.
- Выполняется обратный проход, в ходе которого на основе зарезервированных данных осуществляется вычисление компонент локального градиента

$$\delta_j(n) = -\{\partial E_{\text{общ.}}(n_0, n_1)/\partial v_j(n)\} \qquad (33)$$

по формуле

$$\delta_j(n) = \begin{cases} \varphi'(v_j(n))\, e_j(n) & \text{для } n = n_1; \\ \varphi'(v_j(n)) \times & \\ \times \left[e_j(n) + \right. & \\ \left. + \sum_{k \in A} w_{jk} \cdot \delta_k(n+1) \right] & \text{для } n_0 < n < n_1, \end{cases} \qquad (34)$$

где $\varphi'(.)$ — производная функции активации по своему аргументу; $n_0 < n < n_1$; $v_j(n)$ — индуцированный потенциал нейрона j, который с помощью функции φ преобразуется в его выходной сигнал; w_{jk} — синаптический вес связи от k к j-му нейрону.

Использование (34) повторяется, начиная с n_1, шаг за шагом в обратной последовательности по n (n_1, $n_1 - 1$,…, $n_0 + 1$). Число шагов равно длине временно́го интервала (n_0, n_1).

- После завершения ОРВ до момента $n_0 + 1$ применяется корректировка синаптичекого веса w_{ji} нейрона j

$$\Delta w_{ji} = -\eta \frac{\partial E_{\text{общ.}}(n_0, n_1)}{\partial w_{ji}} = \eta \sum_{n=n_0+1}^{n_1} \{\delta_j(n) x_i(n-1)\}, \qquad (35)$$

где η — параметр скорости обучения; $x_i(n-1)$ — входной сигнал, приложенный к синапсу i нейрона j в момент времени $n - 1$.

Процедура ОРВ для ОСП существенно отличается от процедуры ОРО тем, что желаемые отклики специфицируются для нейронов во многих слоях НС, потому что выходные нейроны многократно дублируются, когда динамическое поведение РНС разворачивается в НС прямого распространения.

Сокращенный алгоритм ОРВ

Для того чтобы применять алгоритм ОРВ для процедуры реального времени, при минимизации функции риска приходится использовать текущую величину суммы квадратов ошибок

$$E(n) = \frac{1}{2}\sum_{j\in A} e_j^2(n). \qquad (36)$$

Для последовательного способа обучения по алгоритму ОРО используется градиент $E(n)$ для подстройки синаптических весов на каждом шаге времени n. Подстройка производится постоянно в процессе работы сети. Однако, чтобы сделать процесс возможно гибким, производится запоминание входных данных и состояния сети только на конечном интервале, называемом глубиной использования данных (ГИД). Практически это значит, что при обозначении ГИД через h информация старше h единиц времени не принимается во внимание. Если бы этого не делалось, то длительность вычисления (так же как и объем запоминаемой информации) линейно возрастал бы от времени, достигая таких значений, что продолжение обучения могло бы стать неэффективным. Эту вторую форму алгоритма называют усеченным ОРВ или алгоритмом УОРВ [18].

Изменение ОРВ при переходе к УОРВ достаточно очевидно. Локальное значение градиента определяется теперь на глубину h (по времени), поэтому у нейрона j для времени l значение этого градиента дает соотношение

$$\delta_j(l) = -\partial E(l)/\partial v_j(l), \quad \forall : j\in A, \quad n-h < l \le n. \qquad (37)$$

Тогда общий вид выражений для локального градиента в алгоритме УОРВ дает выражение

$$\begin{aligned} &\delta_j(l) = \varphi'(v_j(l))\, e_j(l) \qquad \text{для } l = n, \\ &\delta_j(l) = \varphi'(v_j(l))\times \\ &\quad \times \sum_{k\in A}\left[w_{kj}\delta_k(l+1)\right] \text{ для } n-h < l < n. \end{aligned} \qquad (38)$$

После вычисления УОРВ в обратном направлении (до момента времени $n - h + 1$) производится подстройка синаптических весов w_{ji} нейрона j по соотношению

$$\Delta w_{ji}(n) = \eta \cdot \sum_{l=n-h+1} (\delta_j(l)\cdot x_i(l-1)), \qquad (39)$$

где η и $x_i(l-1)$ определены ранее, а $\delta_j(l)$ определяется по (38).

Использование w_{kj} в (38) требует, чтобы сохранялись предшествующие значения весов. Задача несколько облегчается при очень маленьком значении параметра η, когда на каждом шаге веса меняются несущественно.

Сравнение (38) и (34) показывает, что в отличие от алгоритма ОРВ с ОСП в алгоритме ОРВ реального времени используется только текущий сигнал ошибки (в момент n). Поэтому не приходится хранить предшествующие значения желаемого отклика. В результате алгоритм УОРВ проводит вычисления для всех предшествующих отсчетов времени практически аналогично способу, каким алгоритм ОРО осуществляет вычисления для скрытых нейронов МСП.

С точки зрения практических аспектов реализации алгоритма ОРВ усечение его — это не слишком искусственный способ. Если РНС устойчива, то должны сходиться производные $\partial E(l)/\partial v_j(l)$, т. к. вычисления назад по времени слишком далеко соответствуют более сильной обратной связи (которая, грубо говоря, равна крутизне сигмоида, умноженной на вес). В любом случае глубина усечения h должна быть достаточно велика, чтобы вычисленные производные хорошо аппроксимировали фактические их значения. Это требование создает нижнюю границу для величины h.

Процедура разворачивания для алгоритма ОРВ обеспечивает полезное средство для изображения РНС в виде последовательного (каскадного) соединения идентичных слоев с прямым (вперед) распространением сигнала. Это помогает понять, как реализуется процедура алгоритма ОРВ для РНС. Но это преимущество имеет обратную сторону. Процедура хорошо работает для сравнительно простых РНС, состоящих из небольшого количества нейронов. В то же время выражения, связанные с алгоритмом ОРВ, в частности (38), становятся громоздкими, когда процедура ОРВ применяется к общим видам структур, которые встречаются в практике. Для этих ситуаций Вербосом (Werbos [19]) рекомендован более общий подход, согласно которому каждое выражение для слоя прямого распространения порождает соответствующий набор выражений обратного распространения. Преимущество этого подхода состоит в его однородной трактовке прямых и рекуррентных (обратных) связей. Однако следует отметить, что изложение этого метода в [19] не ориентировано на непосредственное использование и пока не нашло прикладной адаптации.

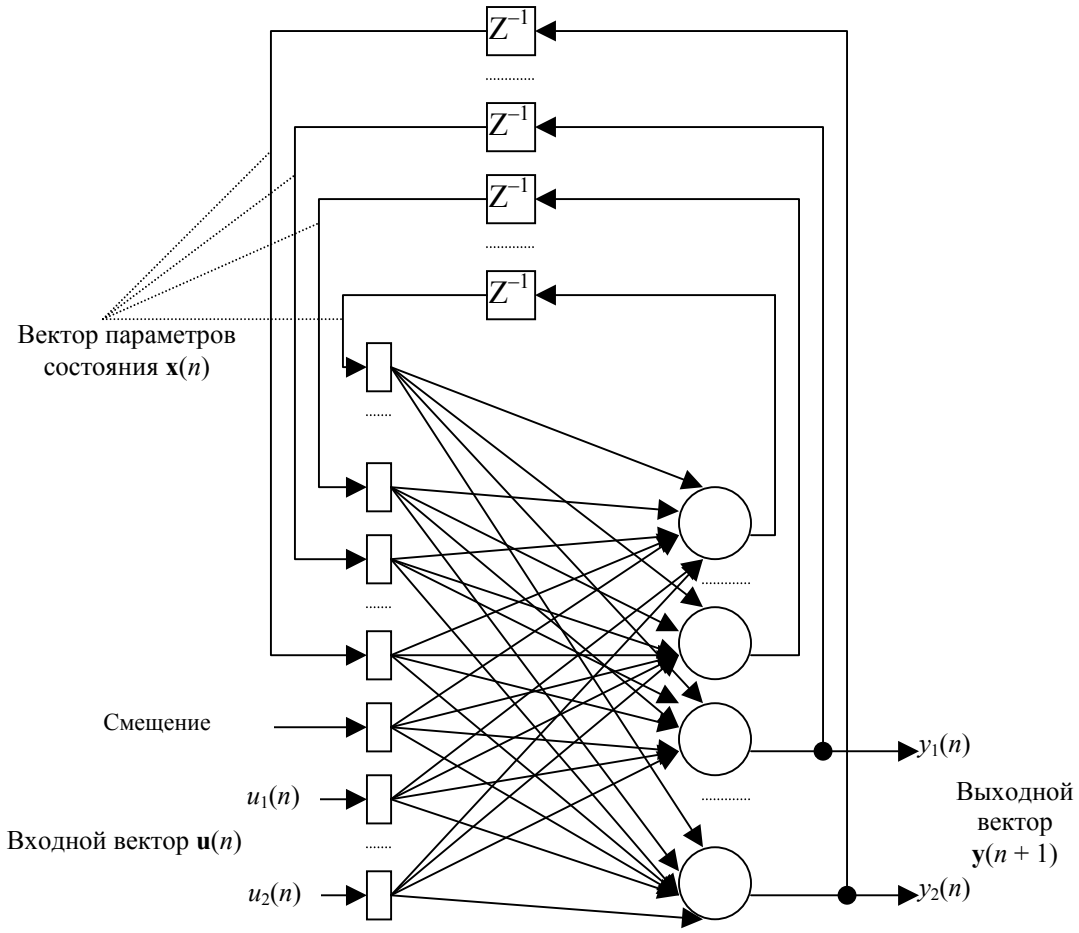


Рис. 5. Сигнально-потоковый граф полносвязной РНС с четырьмя нейронами ($q = 4$), двумя входами ($m = 2$) и двумя выходами ($p = 2$)

Рекуррентный алгоритм обучения НС в реальном времени

Название алгоритма рекуррентного обучения реального времени (РОРВ) отражает его назначение для полносвязной РНС и для реализации подстройки весов в реальном времени, т. е. в процессе выполнения сетью обработки сигналов [14]. Полносвязная РНС простой структуры может состоять из q нейронов и иметь m внешних входов (внешний сигнал с m компонентами). Сеть имеет две функционально различные (но объединенные конкатенацией в один слой[6)]) группы нейронов. На входные узлы одной группы поступают сигналы обратной связи с выхода всех нейронов слоя (т. е. вектор $\mathbf{x}(n)$ параметров состояния), а на входные узлы другой группы нейронов поступает m-вектор входного воздействия $\mathbf{u}(n)$.

Принципы построения алгоритма РОРВ удобно проанализировать на несложной структуре полносвязной рекуррентной НС (рис. 5), которая имеет

[6)] Образующие отдельный слой с двумя последовательно расположенными группами узлов, имеющих в каждой из групп различное функциональное назначение.

в своем составе $q = 4$ нейрона, вектор из параметров состояния $\mathbf{x}(n)$ размерности 4 и $m = 2$.

Описание модели пространства состояний, определяемое уравнением (8), в развернутом виде выражается соотношением

$$\begin{aligned}&\mathbf{x}(n+1) = \\ &= \left[\varphi(\mathbf{w}_1^{\mathrm{T}} \cdot \xi(n)), \ldots, \varphi(\mathbf{w}_j^{\mathrm{T}} \cdot \xi(n)),\right. \\ &\left.\ldots, \varphi(\mathbf{w}_q^{\mathrm{T}} \cdot \xi(n))\right]^{\mathrm{T}},\end{aligned} \tag{40}$$

где φ — активационная функция нейронов; $\mathbf{w}_j$ — $(q + m + 1)$-вектор синаптических весов нейрона j в РНС и $\xi(n)$ — $(q + m + 1)$-вектор, описанный ниже.

Векторы $\mathbf{w}_1, \ldots, \mathbf{w}_q$ определяются соотношением

$$\mathbf{w}_j = [\mathbf{w}_{\mathrm{a},j} \;\; \mathbf{w}_{\mathrm{b},j}]^{\mathrm{T}}, \quad j = 1, \ldots, q, \tag{41}$$

в котором $\mathbf{w}_{\mathrm{a},j}$ и $\mathbf{w}_{\mathrm{b},j}$ — это j-е столбцы транспонированных полных матриц весов $\mathbf{W}_{\mathrm{a}}^{\mathrm{T}}$ и $\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}$.

$(q + m + 1)$-вектор $\xi(n)$ определяется выражением

$$\xi(n) = [\mathbf{x}(n) \;\; \mathbf{u}(n)]^{\mathrm{T}}, \tag{42}$$

где $\mathbf{x}(n)$ — q-мерный вектор параметров состояния; $\mathbf{u}(n)$ — $(m + 1)$-мерный входной вектор (m компонент внешнего воздействия и 1 компонента постоянного смещения на нейрон, причем первая компонента $\mathbf{u}(n)$ равна 1 и соответственно первый элемент в $\mathbf{w}_{\mathrm{b},j}$ равен смещению b_j, приложенному к нейрону j).

Для некоторого упрощения удобно ввести матрицы $\Lambda_j(n)$, $\mathbf{U}_j(n)$ и $\Phi(n)$, определяя их следующим образом.

- $\Lambda_j(n)$ — $(q \times q + m + 1)$-матрица, элементы которой суть частные производные от $\mathbf{x}(n)$ по вектору веса $\mathbf{w}_j$:

$$\Lambda_j(n) = \partial \mathbf{x}(n)/\partial \mathbf{w}_j, \quad j = 1, 2, \ldots, q. \tag{43}$$

- $\mathbf{U}_j(n)$ — это $(q \times q + m + 1)$-матрица, у которой все строки нулевые, кроме строки j, которая равна транспонированному вектору $\xi(n)$:

$$\mathbf{U}_j(n) = \begin{bmatrix} \mathbf{0} \\ \cdots \\ \xi^{\mathrm{T}}(n) \\ \mathbf{0} \end{bmatrix} \leftarrow \text{строка } j, \quad j = 1,2,\ldots,q. \tag{44}$$

- $\Phi(n)$ — это диагональная $(q \times q)$-матрица, у которой k-й диагональный элемент является частной производной активационной функции по своему аргументу, оцененной в точке $\mathbf{w}_j^{\mathrm{T}} \cdot \xi(n)$:

$$\begin{aligned}\Phi(n) = \operatorname{diag}\{&\varphi'(\mathbf{w}_1^{\mathrm{T}} \cdot \xi(n)), \ldots, \varphi'(\mathbf{w}_j^{\mathrm{T}} \cdot \xi(n)), \\ &\ldots, \varphi'(\mathbf{w}_q^{\mathrm{T}} \cdot \xi(n))\}.\end{aligned} \tag{45}$$

При этих обозначениях можно произвести дифференцирование уравнения (40) по $\mathbf{w}_j$, и применение правила вычисления производной сложной функции позволяет получить рекурсивное уравнение

$$\begin{aligned}&\Lambda_j(n+1) = \Phi(n) \cdot [\mathbf{W}_{\mathrm{a}}(n)\, \Lambda_j(n) + \mathbf{U}_j(n)], \\ &j = 1, 2, \ldots, q.\end{aligned} \tag{46}$$

Это рекурсивное уравнение описывает нелинейную динамику состояния (т. е. эволюцию состояния) процесса обучения в реальном времени. Оно дает математическую форму реализации алгоритма РОРВ.

Теперь для завершения алгоритма необходимо соотнести матрицу $\Lambda_j(n)$ с градиентом поверхности ошибки (градиентом функции риска) относительно $\mathbf{w}_j$. Для этого вначале с помощью уравнения измерения (9) определяется p-вектор сигнала ошибки

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n) = \mathbf{d}(n) - \mathbf{C} \cdot \mathbf{x}(n).$$

Сумма квадратов ошибки (или, точнее, квадрат нормы вектора-сигнала ошибки)

$$E(n) = E_n = (1/2)\, \mathbf{e}(n)^{\mathrm{T}} \cdot \mathbf{e}(n)$$

определяет функцию риска (ФР) — суммарную величину текущего значения ошибки

$$\text{ФР} = E_{\text{общ.}} = \sum_{(n)} E_n = (1/2) \sum_{(n)} \left(\mathbf{e}(n)^{\mathrm{T}} \cdot \mathbf{e}(n)\right). \tag{47}$$

В свою очередь критерий минимизации ФР служит для настройки весов РНС. Алгоритм крутого спуска осуществляет движение к минимуму по поверхности ошибок в направлении, противоположном градиенту ФР ($E_{\text{общ.}}$).

Таким образом, для выполнения алгоритма РОРВ необходимо получение выражения для градиента (символ ∇) от $E_{\text{общ.}}$:

$$\begin{aligned}&\nabla_{\mathbf{w}}(E_{\text{общ.}}) = \partial E_{\text{общ.}} / \partial \mathbf{w} = \\ &= \sum_{(n)} \{\partial E_n / \partial \mathbf{w}\} = \sum_{(n)} \nabla_{\mathbf{w}}(E_n),\end{aligned} \tag{48}$$

где $\nabla_{\mathbf{w}}(E_n)$ — градиент от $E(n)$ относительно матрицы весов $\mathbf{W} = \{\mathbf{w}_k\}$ (т. е. относительно каждого элемента этой матрицы).

Непосредственное вычисление градиента предоставляет способ получения уравнения обновления весов РНС, не делая аппроксимаций. Однако, чтобы получить алгоритм обучения РНС в реальном времени, необходимо использовать текущую оценку градиента, а именно оценку мгновенного значения $\nabla_{\mathbf{w}}(E_n)$, которая могла бы служить аппроксимацией для метода крутого спуска.

Для этой цели требуется использовать квадратичную ошибку E_n в текущем времени n и, мини-

мизируя ее, двигаться против градиента этой функции ошибок:

$$\partial E_n / \partial \mathbf{w}_j = \{\partial \mathbf{e}(n) / \partial \mathbf{w}_j\} \mathbf{e}(n) =$$
$$= -\mathbf{C} \cdot \{\partial \mathbf{x}(n) / \partial \mathbf{w}_j\} \mathbf{e}(n) =$$
$$= -\mathbf{C} \cdot \mathbf{\Lambda}_j(n) \cdot \mathbf{e}(n), \quad j = 1, \ldots, q. \qquad (49)$$

Подстройка вектора синаптических весов $\mathbf{w}_j$ нейрона j осуществляется по соотношению

$$\Delta \mathbf{w}_j(n) = -\eta \cdot \partial E_n / \partial \mathbf{w} =$$
$$= \eta \cdot \mathbf{C} \cdot \mathbf{\Lambda}_j(n) \cdot \mathbf{e}(n), \quad j = 1, \ldots, q \qquad (50)$$

где η — параметр скорости обучения; $\mathbf{\Lambda}_j(n)$ — определяется выражением (43).

В начале алгоритма обучения проводится инициализация матрицы $\mathbf{\Lambda}_j(0)$ в виде $\mathbf{\Lambda}_j(0) = \mathbf{0}$ для всех j, т. е. предполагается, что в начальный момент РНС пребывает в некотором постоянном состоянии.

В Приложении дана сводка соотношений, определяющих алгоритм РОРВ. Представленный там алгоритм применим к РНС с произвольной (дифференцируемой) функцией активации нейронов $\varphi(.)$. Для случая сигмоидальной нелинейности нейронов в виде функции гиперболического тангенса следует использовать выражение (51), которое показывает связь выхода нейрона j с потенциалом его возбуждения (v_j):

$$x_j(n+1) = \varphi(v_j(n)) = \tanh(v_j(n)),$$
$$\varphi'(v_j(n)) = \partial \varphi(v_j(n)) / \partial v_j(n) = \operatorname{sech}^2(v_j(n)) =$$
$$= 1 - [x_j(n+1)]^2. \qquad (51)$$

В этом же Приложении для полносвязной РНС (рис. 1П) дан пример применения алгоритма РОРВ. Показаны порядок и обозначения параметров модели пространства состояний (матриц $\mathbf{W}_a$, $\mathbf{W}_b$ и вектора $\mathbf{C}$); построение вектора $\boldsymbol{\xi}(n)$ и матрицы $\mathbf{\Lambda}_j$. Приведены выражения для подстановки параметров анализируемой РНС и граф чувствительности рассматриваемой полносвязной РНС.

Использование в алгоритме РОРВ текущего ("мгновенного") градиента $\nabla_{\mathbf{w}}(E(n))$ означает, что этот описанный здесь алгоритм отличается от алгоритма ОРВ (не выполняющегося в реальном времени), основанного на $\nabla_{\mathbf{w}}(E_{\text{общ.}})$. Однако это отличие аналогично отличию стандартного алгоритма обратного распространения (алгоритма ОРО [4]) при двух вариантах обучения: с настройкой параметров сети после каждого предъявления примера (sample-метод) и с настройкой после предъявления серии образцов (batch-метод). Хотя алгоритм РОРВ не гарантирует движения по поверхности ошибок ($E_{\text{общ.}}(\mathbf{W})$) в направлении, противоположном градиенту, различие между РВ и не-РВ вариантами алгоритма часто очень незначительны, и методы становятся почти идентичными, когда параметр скорости обучения выбирается очень малым. Самым неприятным потенциальным последствием этого отклонения от модификации, построенной на истинном градиенте ($\nabla_{\mathbf{w}}(E_{\text{общ.}})$), является то, что наблюдаемая траектория (полученная как зависимость точек эволюции весов при градиентном спуске по поверхности $E(n)$) может стать зависимой от темпа изменения веса, производимого алгоритмом РОРВ. Последняя ситуация может трактоваться как дополнительный источник обратной связи, вносящий элемент неустойчивости в систему. Этого эффекта можно избежать, используя достаточно малое значение η, — малое настолько, чтобы масштаб изменения веса был намного меньше масштаба весов функционирующей сети.

Одной из стратегий, часто используемых при обучении РНС, является усиливающее влияние учителя (УВУ) [14, 15]. В области адаптивной фильтрации эта стратегия известна как метод выровненной ошибки [20]. В основном УВУ состоит в замене действительного выхода нейрона (во время обучения сети) на соответствующий желаемый отклик. Эта замена действует в последующих за ней вычислениях динамического поведения сети. Стратегия УВУ, кроме алгоритма РОРВ, используется и в некоторых других алгоритмах обучения, но для ее применения необходимо, чтобы нейрон, подвергнутый УВУ, блокировал свой выход в обратную связь сети. По данным Вильямса и Зипсера (Williams, Zipser [15]) использование при обучении стратегии УВУ имеет определенные преимущества.

- УВУ может приводить к более быстрому обучению. Причина состоит в предположении, что сеть правильно обучена на всей предшествующей части задачи, относящейся к нейрону, для которого реализовано УВУ.
- УВУ может служить как некоторый корректирующий механизм во время процесса обучения. Например, синаптические веса НС могут иметь корректные значения, но тем не менее сеть функционирует где-то в неверной области пространства состояний. В этом случае простая подстройка параметров (без УВУ) является неправильной стратегией.
- Алгоритм обучения, основанный на градиенте функции риска (ФР) и использующий УВУ, оптимизирует ФР иначе, чем аналогичный алгоритм без применения УВУ. Поэтому обе версии алгоритма могут давать несколько отличающиеся по комплекту весов НС при условии, что соответствующий сигнал ошибки не равен нулю. Но в последнем случае и в самом обучении нет необходимости.

Описанный алгоритм РОРВ работает недостаточно быстро, что связано главным образом с необходимостью многократно вычислять текущее (на каждом временно́м шаге) значение градиента. Совершенствование алгоритма основано на трактовке супервизорного (с учителем) обучения РНС как оптимальной фильтрации. Применение такого подхода позволяет рекуррентно использовать информацию, содержащуюся в обучающих данных, путем своеобразного возвращения к первой итерации процесса обучения. Основой усовершенствованного алгоритма РОРВ для рекуррентной нейронной сети служат элементы теории фильтров Калмана [7)].

Исчезающий градиент у РНС

В практическом приложении РНС при обучении с использованием алгоритма, основанного на градиентном спуске, встречается ситуация исчезающего градиента [21, 22]. Ситуация связана со стремлением получить желаемый отклик НС в текущий момент времени и с тем, что этот отклик зависит не только от текущего входного сигнала, но и от его прошлых значений. Суть в том, что за счет комбинации нелинейных нейронов предельно малое изменение отдаленного по времени входного сигнала почти не изменяет обучение сети. Проблема может возникнуть даже в том случае, если большое изменение в отдаленных по времени входных данных влияет на переменные параметры состояния РНС, но это изменение почти не отражается на величине градиента. Это явление, называемое проблемой *исчезающего градиента*, при применении алгоритма, основанного на градиентном спуске, делает трудным обучение РНС при редко проявляющихся изменениях входных данных. В связи с этим Бенгио (Bangio [22]) для многих прикладных задач полагает необходимым сохранять информацию о состоянии РНС в течение возможно длительного времени, особенно при анализе зашумленной информации.

ИСПОЛЬЗОВАНИЕ РНС В ПРИКЛАДНЫХ ЗАДАЧАХ

Экспериментальный подход к моделированию ДС с неизвестными параметрами используется в задаче идентификации [11, 23, 25]. Процедура идентификации включает этапы планирования и отбора структуры модели, оценку параметров, проверку моделей и имеет интерактивную форму вычисления, в которой можно возвращаться или двигаться вперед (относительно описанных этапов) до построения удовлетворительной модели. При анализе ДС для построения удобно параметризованной модели и ее идентификации можно использовать процедуры, основанные на модели пространства состояний или на модели вход—выход. Выбор этих двух схем идентификации зависит от информации о входных и наблюдаемых переменных ДС.

Система идентификации на основе пространства состояний

Система идентификации использует модель ДС в виде

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n)), \tag{52}$$

$$\mathbf{y}(n) = \mathbf{h}(\mathbf{x}(n)), \tag{53}$$

где $\mathbf{f}(.,.)$ и $\mathbf{y}(.)$ — нелинейные вектор-функции, вид каждой из которых неизвестен. (53) является обобщением (9).

Для идентификации ДС могут быть использованы две НС: первая НС-I отражает уравнение процесса (52), вторая НС-II — уравнение измерения (53). Учитывая, что $\mathbf{x}(n)$ является задержанной на единицу времени переменной $\mathbf{x}(n+1)$ и что $\hat{\mathbf{x}}(n+1)$ обозначает оценку для $\mathbf{x}(n+1)$, структуру первой НС можно представить рис. 6, а. Эта НС включает конкатенацию двух групп входных узлов, на одну из которых поступает вектор-сигнал внешнего воздействия $\mathbf{u}(n)$, а на другую — вектор параметров состояния $\mathbf{x}(n)$. Совместное действие $\mathbf{u}(n)$ и $\mathbf{x}(n)$ определяет на выходе сети оценку $\hat{\mathbf{x}}(n+1)$. Вектор сигнала ошибки:

$$\mathbf{e}_{\mathrm{I}}(n+1) = \mathbf{x}(n+1) - \hat{\mathbf{x}}(n+1). \tag{54}$$

В (54) $\mathbf{x}(n+1)$ играет роль желаемого отклика, и предполагается, что локальное состояние доступно для получения величины ошибки. Ошибка $\mathbf{e}_{\mathrm{I}}(n+1)$ служит для подстройки синаптических весов РНС на основе минимизации функции риска.

Вторая НС (НС-II, рис. 6, б) имитирует систему измерения и работает с реальным состоянием $\mathbf{x}(n)$ (в целом неизвестной) ДС, получая оценку $\hat{\mathbf{y}}(n)$ выхода НС. Вектор-сигнал ошибки $\mathbf{e}_{\mathrm{II}}(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n)$, в котором $\mathbf{y}(n)$ играет роль желаемого выхода, служит для образования ФР и подстройки синаптических весов РНС на основе минимизации этой ФР.

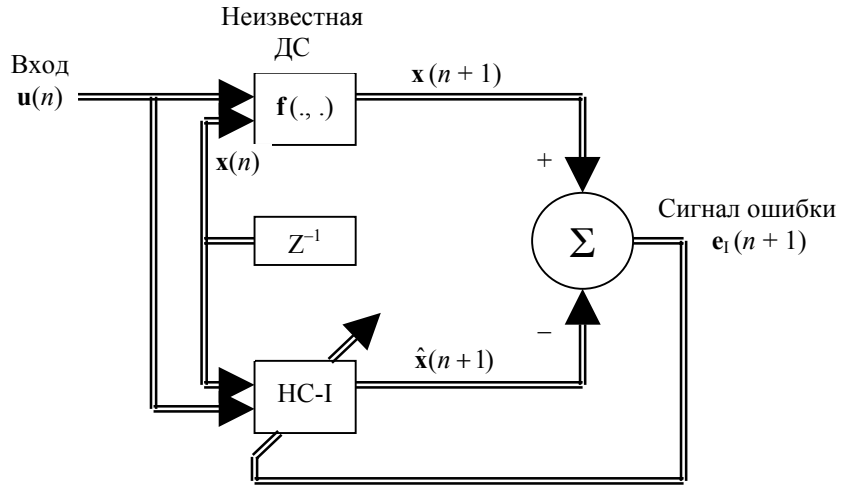
Обе НС (рис. 6) работают синхронно и дают решение модели пространства состояний, необходимое для обеспечения идентификации ДС. Такой способ идентификации называют последовательно-параллельной моделью идентификации в распознавании для обозначения того факта, что реальное состояние неизвестной ДС подается на модель идентификации так, как это показано

[7)] Рассмотрению фильтров Калмана в приложении к РНС будет посвящена отдельная статья.

на рис. 6, а. Используемая здесь форма обучения относится к УВУ.

Последовательно-параллельную модель следует отличать от параллельной модели идентификации, в которой $\mathbf{x}(n)$, поступающее на НС-I, заменяется на $\hat{\mathbf{x}}(n)$ (получается из $\hat{\mathbf{x}}(n+1)$ после ЭВЗ Z^{-1}).

а



б

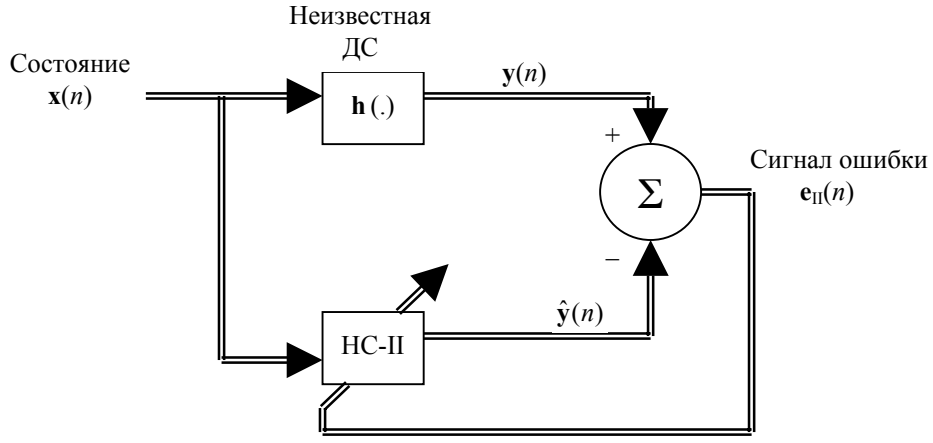


Рис. 6. Решение задачи идентификации на основе двух РНС.
а — РНС-I, отражающая процесс ДС (объекта, подсистемы);
б — РНС-II для измерительной системы

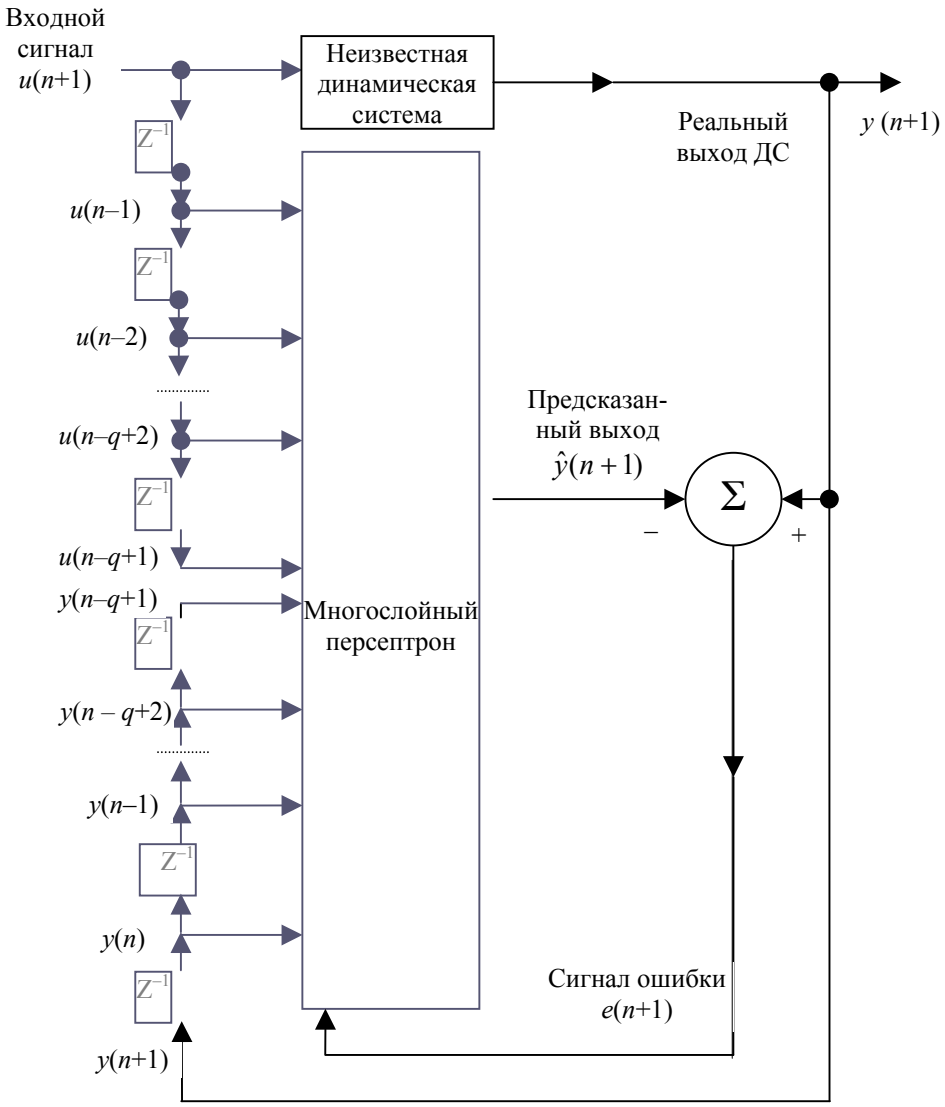


Рис. 7. РНС при решении задачи идентификации в форме модели нелинейной авторегрессии с временно́й задержкой внешнего входа (НАР_ВЗ)

Практическое преимущество этой альтернативной модели обучения состоит в том, что модель НС функционирует точно таким образом, как неизвестная ДС, т. е. таким образом, каким модель будет использоваться после окончания обучения. Поэтому вполне вероятно, что модель, полученная за счет параллельного способа обучения, может показывать автономное поведение, которое превосходит поведение модели, полученной за счет последовательно-параллельного типа обучения. Но параллельный способ обучения дольше, чем последовательно-параллельный. В частности, в рассматриваемой ситуации оценка $\hat{\mathbf{x}}(n)$, используемая в параллельном способе, обычно не так точна, как фактическое состояние $\mathbf{x}(n)$, введенное

в структуру последовательно-параллельного способа обучения

Модель вход—выход

Эта модель в идентификации используется, когда имеется доступ только к выходному вектору ДС. Для упрощения анализа достаточно рассмотреть ДС с одним входом и одним выходом (т. е. скалярные $u(n)$ и $y(n)$) и структуру модели НАР_ВЗ, при которой соотношения вход—выход имеют вид

$$\hat{y}(n+1) = \varphi(y(n),\ldots, y(n-q+1), u(n),\ldots, u(n-q+1)), \quad (55)$$

где q — порядок ДС.

Таким образом, оценка $\hat{y}(n+1)$ реального выходного сигнала РНС $y(n+1)$ используется для получения сигнала ошибки

$$e(n+1) = y(n+1) - \hat{y}(n+1),$$

где $y(n+1)$ играет роль желаемого отклика сети. Как обычно, подстройка весов РНС осуществляется, исходя из минимизации ФР, в качестве которой принимается квадрат сигнала ошибки.

Модель идентификации (рис. 7) относится к последовательно-параллельной форме с УВУ, т. к. реальный выход ДС $y(n+1)$ (а не модели идентификации) подается через обратную связь на вход.

Система адаптивного управления с опорной моделью (САУ_ОМ)

Другой областью применения РНС является адаптивное управление [8, 24, 25]. В структуру управления входит базовая опорная модель (имитатор желаемого и оптимального поведения ДС), РНС — как управляющее устройство (контроллер) и объект управления (ОУ), в качестве которого может быть некоторый производственно-технологический комплекс (рис. 8, а).

Стратегия адаптивного управления, ориентированная на опорную модель, предполагает, что при создании системы управления на РНС имеется достаточно информации относительно анализируемой ДС. От контроллера и через объект управления (ОУ) подается обратная связь на вход системы, образуя сеть с внешней рекуррентностью. ОУ получает входной сигнал $\mathbf{u}_c(n)$ от контроллера и внешнее управляющее воздействие $\mathbf{u}_d$. Эволюция ОУ во времени является функцией приложенных воздействий и своего собственного состояния $\mathbf{x}_{ОУ}(n)$. Выход ОУ $\mathbf{y}_{ОУ}(n+1)$ определятся вектором параметров его состояния $\mathbf{x}_{ОУ}$, а также возможными шумами. Контроллер получает на входе внешне определенный опорный сигнал $\mathbf{r}(n)$ и выход $\mathbf{y}_{ОУ}(n+1)$, преобразованный за счет ЭВЗ в $\mathbf{y}_{ОУ}(n)$. Контроллер производит сигнал управления

$$\mathbf{u}_c(n) = \mathbf{f}_1(\mathbf{x}_c(n), \mathbf{y}_{ОУ}(n), \mathbf{r}(n), \mathbf{w}), \quad (56)$$

где $\mathbf{x}_c$ — соответственное состояние контроллера; $\mathbf{w}$ — вектор параметров НС, доступный для подстройки; вектор-функция $\mathbf{f}_1(.\ ,\ .\ ,\ .\ ,\ .\ )$ определяет соотношение вход—выход контроллера.

Желаемый отклик $\mathbf{d}(n+1)$ объекта управления обеспечивается выходом опорной модели (ОМ), которая дает этот отклик в качестве реакции на сигнал $\mathbf{r}(n)$. Поэтому $\mathbf{d}(n+1)$ является функцией сигнала $\mathbf{r}(n)$ и собственного состояния $\mathbf{x}_{ом}(n)$ опорной модели

$$\mathbf{d}(n+1) = \mathbf{f}_{ом}(\mathbf{x}_{ом}(n), \mathbf{r}(n)). \quad (57)$$

Вектор-функция $\mathbf{f}_{ом}$ определяет соотношение вход— выход опорной модели. Ориентиром для подстройки параметров сети служит движение в направлении антиградиента поля ФР, которое определяется квадратом нормы вектора $\mathbf{e}_c(n+1) = \mathbf{d}(n+1) - \mathbf{y}_{ОУ}(n+1)$. И цель подстройки весов сети $\mathbf{w}$ определяется минимизацией функции риска.

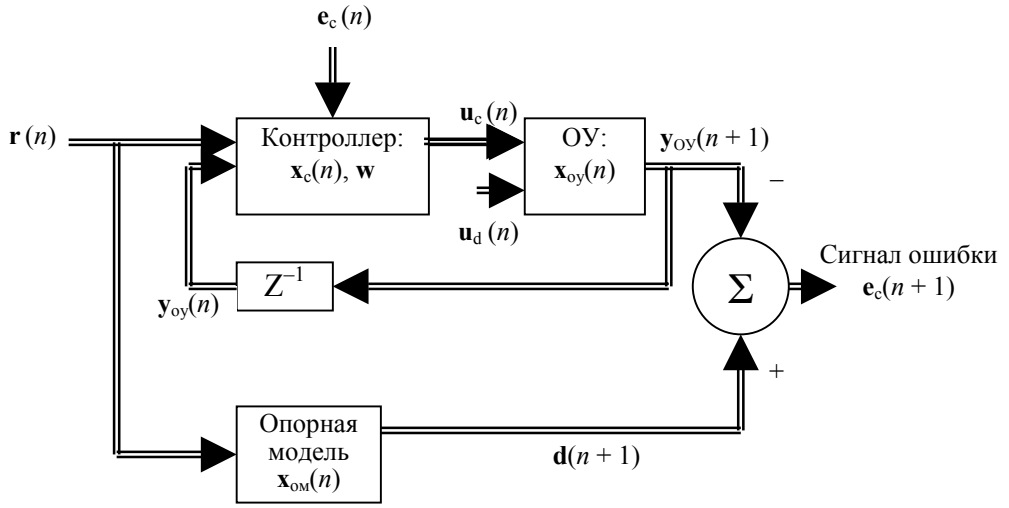
Метод управления в САУ_ОМ (рис. 8, а) можно назвать прямым, т. к. он не требует идентификации параметров объекта управления и параметры контроллера непосредственно подстраиваются, чтобы улучшить функционирование ОУ. Однако пока для подстройки параметров контроллера (на основе сигнала ошибки) нет достаточно точных и достоверных методов. Поэтому неизвестный объект управления структурно включается между контроллером и ошибкой выхода. В связи с эти предложена более гибкая структура САУ_ОМ, реализующая непрямое управление (НПУ) (рис. 8, б). В этой структуре используется двухступенчатая процедура:

1. Модель объекта управления ОУ (обозначаемая $\hat{ОУ}$) служит для обеспечения оценок, необходимых для получения разностных соотношений выхода ОУ со значениями входа ОУ, которые предшествуют выходу ОУ и его внутренним состояниям. Описанная процедура использована для обучения НС идентификации ОУ, и модель $\hat{ОУ}$ считается моделью идентификации.

2. Модель идентификации $\hat{ОУ}$ используется (после обучения) вместо ОУ для вывода оценок динамических производных ОУ по компонентам вектора параметров контроллера, которые доступны для подстройки.

Таким образом, в непрямом управлении, нейронная сеть с внешней рекуррентностью составлена из контроллера и представления ОУ в форме вход—выход, которое эмулируется моделью идентификации $\hat{ОУ}$.

а



б

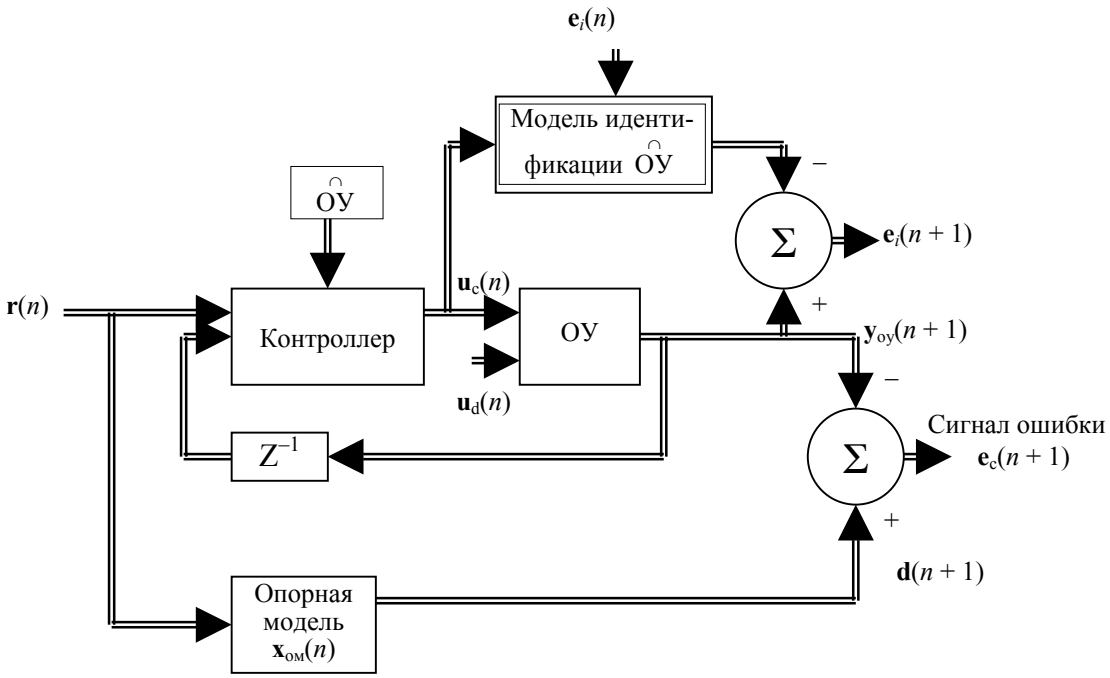


Рис. 8. Структуры РНС для системы адаптивного управления с опорной моделью (САУ_ОМ):
а — схема прямого управления; б — схема непрямого управления

Приложение РНС для создания контроллеров с общей структурой, показанной на рис. 8, б, занимает значительное место в сфере систем адаптивного управления (от двигателей и биореакторов до автоматических подсистем) [8, 24]. Часто эти контроллеры включают рекуррентные МСП, а для обучения начинают использовать довольно трудоемкие алгоритмы на основе фильтров Калмана.

ЗАКЛЮЧЕНИЕ

В целях применения в ИИС для анализа данных при неполной измерительной информации рассмотрены разновидности структуры, элементы теории и алгоритмы обучения рекуррентных нейронных сетей (РНС), а также их приложения — идентификация динамического объекта (подсистемы) и адаптивное управление по "эталонной" модели. Проанализированы модель пространства состояний динамического объекта, понятия и условия управляемости и наблюдаемости. В терминах пространства состояний представлена модель нелинейной авторегрессии, построенной на РНС с внешним входом на линии элементов временно́й задержки.

Проанализированы два способа обучения РНС: на основе алгоритма обратного распространения во времени (алгоритм ОРВ) и на основе рекуррентного алгоритма обучения реального времени (РАО_РВ). Намечен путь усовершенствования РАО_РВ на основе фильтров Калмана. Приведены некоторые рекомендации (эвристического типа), позволяющие противодействовать ухудшению характеристик обучения РНС при значениях градиента, близких к нулю.

Таким образом, в предшествующей [5] и в этой частях статьи проанализированы структуры темпоральных (с прямым распространением сигнала) и рекуррентных (с обратными связями) нейронных сетей, реагирующих на изменение входной информации изменением реализуемого сетью отображения. Эти виды нейронных сетей используются в методе интерпретации данных, получаемых и обрабатываемых ИИС в условиях, когда не весь набор параметров состояния контролируемого динамического объекта (подсистемы) воздействует на датчики измерительной системы. Метод трактовки измерительных данных при неполной информации [8)] основан на рекуррентном представлении уравнений динамики контролируемого объекта и уравнения наблюдения (определяющего преобразование информации в системе измерения) и на использовании темпоральной или рекуррентной нейронной сети.

ПРИЛОЖЕНИЕ. *Сводка соотношений, определяющих алгоритм рекуррентного обучения в реальном времени (алгоритм РОРВ). Пример*

Параметры

m — размерность входного сигнала; q — размерность пространства состояний; p — размерность выходного сигнала; $\mathbf{w}_j$ — вектор синаптических весов нейрона, j =1, 2, …, q.

Инициализация

1. Множеству синаптических весов (компонентам векторов $\mathbf{w}_j$, j = 1, 2, …, q) присваиваются малые значения, выбираемые из равномерного распределения.
2. Начальная величина всех компонент вектора состояния полагается равной нулю, $\mathbf{x}(0) = \mathbf{0}$.
3. Начальная величина всех векторов $\mathbf{\Lambda}$ также полагается равной нулю, $\mathbf{\Lambda}_j(0) = \mathbf{0}$ для j = 1, 2,…, q.

Процедура вычислений

Для n = 0, 1, 2, … вычисляются значения векторов (в указанной ниже последовательности):

$$\mathbf{\Lambda}_j(n+1) = \mathbf{\Phi}(n)\{\mathbf{W}_a(n)\,\mathbf{\Lambda}_j(n) + \mathbf{U}_j(n)\},$$
$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{C}\cdot\mathbf{x}(n),$$
$$\Delta\mathbf{w}_j(n) = \eta\cdot\mathbf{C}\,\mathbf{\Lambda}_j(n)\,\mathbf{e}(n).$$

Определения для $\mathbf{x}(n+1)$, $\mathbf{\Lambda}_j(n)$, $\mathbf{U}_j(n)$ и $\mathbf{\Phi}(n)$ приведены в основном тексте статьи выражениями (40), (43), (44) и (45).

Формирование алгоритма РОРВ

Формирование алгоритма выполнено для полносвязной рекуррентной сети (рис. 1П), которая включает 3 нейрона, два входных узла и один выход. Сеть имеет двухкомпонентный вход (входной вектор $\mathbf{u}(n)$, m = 2), характеризуется 3-компонентным вектором $\mathbf{x}$ параметров состояния нейронного слоя (q = 3) и имеет скалярный выход (p = = 1). Матрицы $\mathbf{W}_a$, $\mathbf{W}_b$ имеют вид:

$$\mathbf{W}_a = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix},\quad \mathbf{W}_b = \begin{bmatrix} b_1 & w_{14} & w_{15} \\ b_2 & w_{24} & w_{25} \\ b_3 & w_{34} & w_{35} \end{bmatrix}.$$

Первый столбец матрицы $\mathbf{W}_b$ представляет смещения (b_1, b_2, b_3), приложенные к нейронам 1, 2 и 3. Матрица $\mathbf{C}$ в данном случае является вектором (поскольку мы рассматриваем скалярный выход сети) $\mathbf{C}$ = [1, 0, 0].

[8)] Метод предполагается опубликовать в виде третьей части настоящей статьи.

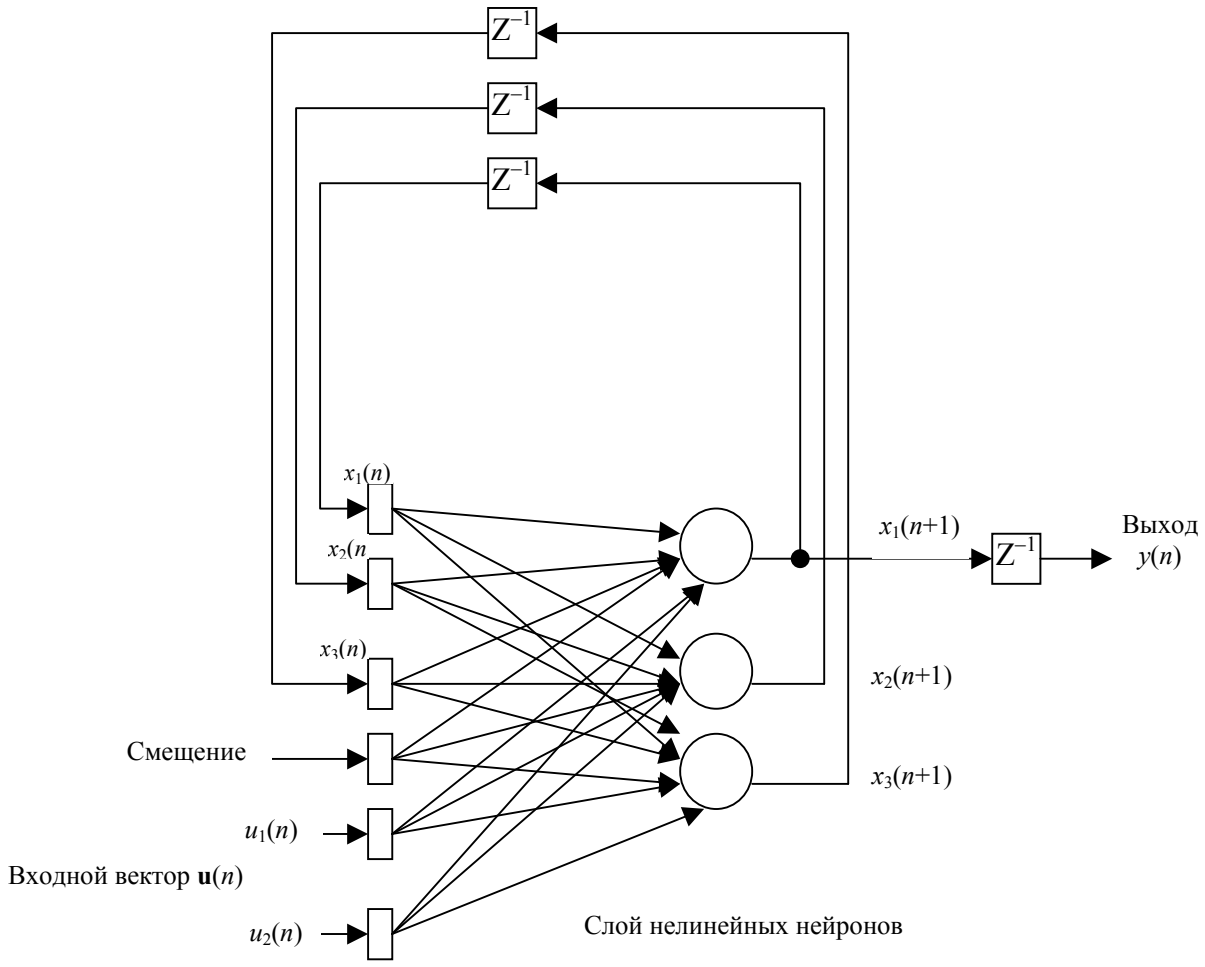


Рис. 1П. Полносвязная РНС двумя входами, двумя скрытыми нейронами и одним выходным нейроном

Имея в виду, что $m = 2$, $q = 3$, можно с помощью выражения (42) (основного текста статьи) определить $\boldsymbol{\xi}(n)$ и kl-элементы ($\lambda_{j,kl}$) матрицы $\boldsymbol{\Lambda}_j(n)$:

$$\boldsymbol{\xi}(n) = [x_1(n)\ \ x_2(n)\ \ x_3(n)\ \ 1\ \ u_1(n)\ \ u_2(n)]^{\mathrm{T}},$$

$$\lambda_{j,kl}(n+1) = \varphi'(v_j(n))\cdot\left[\sum_{(i)}\left\{w_{ji}(n)\lambda_{j,kl}(n) + \delta_{kj}\xi_l(n)\right\}\right],$$

где δ_{kj} — символ Кронекера ($k, j = 1, 2, 3$); индекс $l = 1, 2, \ldots, 6$.

Подстройка параметров сети по алгоритму рекуррентного обучения реального времени (РОРВ) осуществляется на основе выражения

$$\Delta w_{k\,l}(n) = \eta(d_1(n)) - x_1(n)\lambda_{1,k\,l}(n).$$

Отметим, что индексы в этом выражении принимают свои значения в соответствии с определением матриц: $\mathbf{W}_{\mathrm{a}} = \{w_{ji}\}$, $(j, i) = 1, 2, 3$ и $\mathbf{W}_{\mathrm{b}} = \{w_{jl}\}$, $j = 1, 2, 3$, $l = 4, 5, 6$.

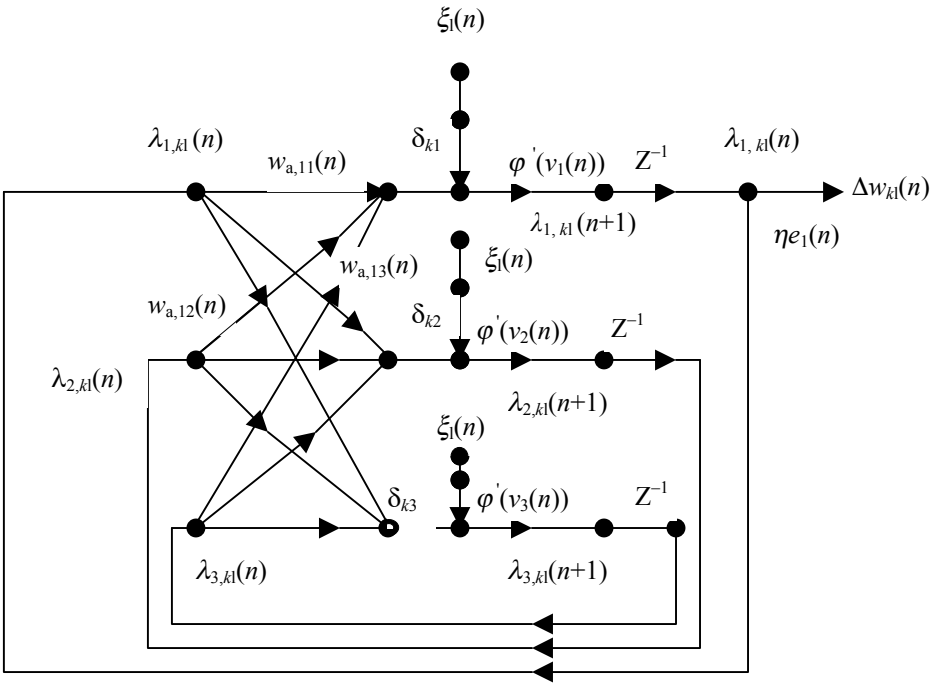


Рис. 2П. Граф чувствительности полносвязной РНС, показанной на рис. 1П

Эволюция величины подстройки весов $\Delta w_{kl}(n)$ рассматриваемой нейронной сети может быть представлена графом чувствительности, который показан на рис. 2П.

СПИСОК ЛИТЕРАТУРЫ

1. *Gaunaurd G.C., Strifors H.C.* Signal analysis by mean of time-frequency transformation of Wigner type // Proceedings of IEEE. 1996. V. 84, N 9. P. 1231–1247.
2. *Исмаилов Ш.Ю., Меркушева А.В.* Нейросетевой алгоритм на вейвлет-преобразовании нестационарного сигнала в ИИС // Труды Международной научной конференции по мягким вычислениям и измерениям SCM'2001. СПб.: Изд. ГЭТУ (ЛЭТИ), 2001. Т. 1. С. 251–256.
3. *Малыхина Г.Ф., Меркушева А.В.* Вейвлет-фильтрация нестационарного сигнала с адаптацией на основе нейронной сети // Труды Международной научной конференции по мягким вычислениям и измерениям SCM'2001. СПб.: Изд. ГЭТУ (ЛЭТИ), 2001. Т. 1. С. 239–242.
4. *Меркушева А.В.* Применение нейронной сети для текущего анализа нестационарного сигнала (речи), представленного его вейвлет-отображением. I. Основные принципы // Научное приборостроение. 2003. Т. 13, № 1. С. 64–70.
5. *Малыхина Г.Ф., Меркушева А.В.* Метод контроля состояния подсистемы (объекта) при неполной измерительной информации о совокупности параметров, определяющих ее динамику. I. Анализ структуры нейронной сети, приспособленной к динамическому характеру анализируемой информации // Научное приборостроение. 2004. Т. 14, № 1. С. 57–67.
6. *Tsoi A.C., Back A.D.* Locally recurrent globally feet-forward networks: A critical review // IEEE Transactions on Neural Networks. 1994. V. 5. P. 222–239.
7. *Elman J.L.* Finding structure in time // Cognitive Science. 1990. V. 14. P. 179–211.
8. *Puskorius G.V., Feldkamp L.A.* Dynamic neural networks methods applied to on-vehicle idle speed control // Proceedings of IEEE. 1996.

V. 84. P. 1407–1420.
9. *Giles C.L., Sun G.Z., Lee Y.C., Chen D.* Higher order recurrent networks and grammatical interference // Advances in Neural Information Processing Systems. San Mateo, CA: Morgan Kaufmann, 1990. V. 2. P. 382–387.
10. *Sontag E.D.* Mathematical control theory: Dynamic finite-dimension systems. N.Y.: Springer Verlag, 1990. 360 p.
11. *Сейдж Э.П., Уайт Ч.С.* Оптимальное управление системами. М.: Радио и связь, 1982. 391 с.
12. *Levin A.V., Narendra K.S.* Control of nonlinear dynamic systems using neural networks: Reliability, identification and control // IEEE Transactions on Neural Networks. 1996. V. 7. P. 30–42.
13. *Haykin S.* Neural networks. N.Y.: Prentice-Hall, 1999. 842 p.
14. *Williams R.J., Zipser D.* Learning algorithms for continually running fully recurrent neural networks // Neural Computation. 1989. V. 1. P. 271–280.
15. *Williams R.J., Zipser D.* Gradient-based learning algorithms for recurrent networks and their computational complexity // Back-propagation: Theory, Architecture and Applications / Eds. Chauvin Y., Rumelhart D.E. Hillsdale, N.Y.: Lawrence Erlbaum, 1995. P. 433–484.
16. *Beaufays F., Wan E.A.* Relating real-time backpropagation through time: Application to flow-graph inter-reciprocity // Neural Computation. 1994. V. 6. P. 396–406.
17. *Giles C.L. et al.* Constructive learning of recurrent neural networks: Limitation of recurrent cascade correlation with simple solution // IEEE Transactions on Neural Networks. 1995. V. 6. P. 829–836.
18. *Williams R.J., Pang J.* An efficient gradient-based algorithm for on-line training the recurrent - network trajectories // Neural Computation. 1990. V. 2. P. 4090–5100.
19. *Werbos P.J.* Back-propagation through time: What it does and how do it // Proceedings of IEEE. 1990. V. 78. P. 1550–1560.
20. *Mendel J.M.* Lessons in estimation theory for signal processing. Communication and control. Englewood Cliffs, N.Y.: Prentice-Hall, 1995. 360 p.
21. *Hochreiter S., Schmithuber J.* LSTM can solve hard long time lag problems // Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 1997. V. 9. P. 473–479.
22. *Bengio Y., Simard P., Frasconi P.* Learning long time dependences with gradient descent is difficult // IEEE Transactions on Neural Networks. 1994. V. 5. P. 157–166.
23. *Цыпкин Я.З.* Информационная теория идентификации. М.: Наука, Физматлит, 1995. 336 с.
24. *Puskorius G.V., Feldkamp L.A.* Neuro control of nonlinear dynamic systems on the basis of recurrent neural networks // IEEE Transactions on Neural Networks. 1994. V. 5. P. 279–297.
25. *Уидроу Б., Стирнз С.* Адаптивная обработка сигналов. М.: Радио и связь, 1989. 439 с.

Санкт-Петербург

Материал поступил в редакцию 20.01.2004.

PLANT (SUBSYSTEM) STATE CONTROL AT INCOMPLETE MEASUREMENT INFORMATION ON THE PARAMETER SET DETERMINING ITS DYNAMICS. II. FEEDBACK-BASED NEURAL NETWORKS (RECURRENT NETWORKS) REPRESENTING THE INPUT INFORMATION DYNAMICS

G. F. Malykhina, A. V. Merkusheva

Saint-Petersburg

In information-measurement systems (IMS) and information-control systems representing the state of the plant (subsystem) being controlled, there exist problems that arise in conditions when some state parameters have no effect on subsystem measuring sensors, i. e. in conditions of incomplete information. This problem is solved based on analysis of the plant-IMS system dynamics equation (in the state parameters space), and neural network (NN) algorithms. The second (of three) paper parts considers the structure and learning algorithms for feedback-based NN called recurrent NN, which adequately simulate the input data dynamics.