

УДК 681.324

Б.Б. Нестеренко, М.А. НовотарскийИнститут математики НАН Украины, г. Киев
model@imath.kiev.ua

Имитационное моделирование клеточных нейронных сетей

В работе дан краткий обзор классических принципов организации и функционирования дискретных клеточных нейронных сетей. Рассмотрена постановка задачи решения операторного уравнения локально-асинхронным методом, ориентированным на реализацию в гомогенных вычислительных структурах. Описан подход к созданию имитационных моделей для клеточных нейронных сетей с различными видами межнейронного взаимодействия. Предложены имитационные алгоритмы синхронного, асинхронного и агрегативного взаимодействия в имитационных моделях клеточных нейронных сетей.

Введение

Имитационное моделирование является практически неизбежным этапом при изучении и оценке сложных систем с большим количеством элементов, к которым, безусловно, можно отнести клеточные нейронные сети (КНС). Однако моделирование таких многокомпонентных структур требует существенных затрат времени в случае использования последовательного, пусть даже довольно мощного, компьютера. Очевидный подход к увеличению скорости моделирования состоит в использовании параллельных вычислений, при которых каждый нейрон или группа нейронов клеточной нейронной сети размещаются на отдельном процессорном элементе (ПЭ). Стратегия имитационного моделирования зависит от того, является ли моделируемая КНС синхронной или асинхронной.

В случае моделирования *синхронной* КНС весь жизненный цикл модели состоит из последовательно чередующихся этапов параллельной обработки и обмена. На этапе обработки каждый из процессорных элементов производит вычисления в автономном режиме, а на этапе обмена происходит обмен результатами вычислений. В связи с этим синхронные модели рассматривают как эволюционирующие в дискретные моменты модельного времени $t = 0, 1, 2, \dots$

Принципиальным отличием *асинхронных* моделей является их функционирование в непрерывном времени. Длительность этапа обработки на каждом процессорном элементе (ПЭ) может быть произвольной. Поэтому наступление периода обмена для каждого ПЭ не носит глобального характера. В связи с этим возникает вопрос точного определения типа асинхронности и разработки процедуры имитационного моделирования, обеспечивающей данную асинхронность. Простейшие подходы к решению проблемы, рассмотренные в [1], [2], состоят в

применении различных модификаций исполняемой физической асинхронности, которая базируется на реальном времени работы процессорных элементов. Использование таких подходов полностью оправдано, если моделирование производится с исследовательской целью, когда не требуется строгой повторяемости модельной ситуации. Если же модель используют как инструмент разработки клеточной нейронной сети, то в этом случае важно иметь возможность повторения эксперимента после модификации исходного кода модели. Еще одним существенным требованием к определению асинхронности, кроме повторяемости процесса имитационного моделирования, является сведение к минимуму влияния параметров модели на результаты обработки рабочей нагрузки.

Клеточные нейронные сети

Клеточные нейронные сети относят к одному из перспективных направлений развития теории искусственных нейронных сетей. Интерес к ним, по мнению авторов, обусловлен, прежде всего, возможностью довольно легкой адаптации к их структуре актуальных физических задач, обладающих естественной параллельностью. Подобные структуры широко известны в современной физике как сложные системы, самоорганизующиеся системы, границы хаоса, системы с коллективным поведением и др. В работе [3] они названы клеточными нелинейными сетями, а клеточные нейронные сети (КНС) рассматриваются как их разновидность. Согласно [3], КНС – это массово параллельная вычислительная парадигма, определенная на N -мерном дискретном пространстве. Она состоит из N -мерного гомогенного массива элементов (клеток). Структура типовых связей между клетками КНС показана на рис. 1.

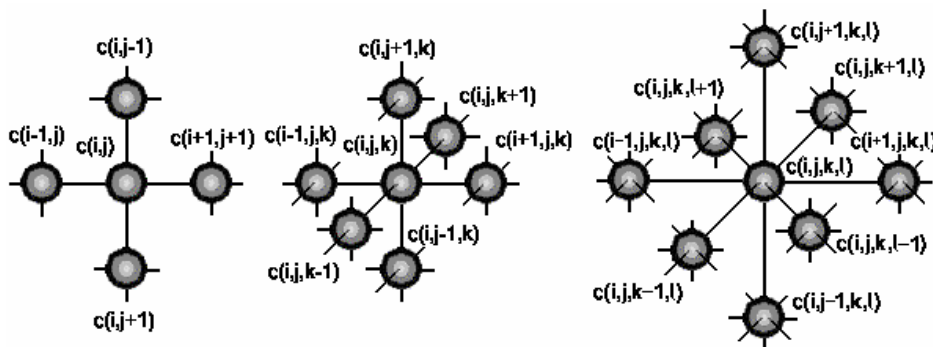


Рисунок 1 – Примеры структурных связей в клеточных нейронных сетях

Каждая из представленных на рис. 1 возможных структур характеризуется количеством связей узлового нейрона (клетки) с соседями, что выражается индексом окружения α . Если величина α превышает количество ближайших соседей, то на такой КНС может быть задано несколько функций соседства:

$$N_r : H \rightarrow H_\alpha, N_r(g) = \{z | d(g, z) \leq r\},$$

где H – множество координат всех клеток, H_α – множество координат соседних клеток, g, z – координатные векторы, r – степень соседства.

Степенью соседства r называют максимальное координатное расстояние между рассматриваемым нейроном и его соседями. Способ определения расстояния задает функция $d(g, z) = |g - z|$.

Узловой нейрон КНС представляет собой специализированный процессор с количеством внешних соединений, соответствующих индексу окружения α . При реализации классического алгоритма КНС его работа сводится в основном к вычислению функционалов, заданных в явной или неявной форме с непрерывным или дискретным временем. Общая форма динамического уравнения для дискретного времени:

$$\begin{aligned} x_g(n+1) &= x_g(n) + \sum_{z \in N_r(g)} A_{g,z} y_z + \sum_{z \in N_r(g)} B_{g,z} u_z + I_g(n), \\ y_g(n) &= f(x_g), \end{aligned} \quad (1)$$

где x_g – внутреннее состояние клетки $c(g)$, y_z – выходной сигнал клетки $c(z)$, u_z – внешний входной вектор клетки $c(z)$, I_g – смещение, $A_{g,z}, B_{g,z}$ – функционалы, соответствующие связям между клетками $c(g)$ и $c(z)$.

В Институте математики НАН Украины предложены локально-асинхронные методы [4], специально ориентированные на реализацию в клеточных нейронных сетях. Задача для решения упомянутыми методами может быть представлена в виде операторного уравнения

$$LX = U, \quad (2)$$

где L – дифференциальный оператор, X – матрица состояния КНС, U – функция правой части.

Для решения данного уравнения следует воспользоваться одним из известных итерационных методов [5], которые предполагают сведение уравнения (2) к системе разностных уравнений на гомогенной сетке Ω с шагом h :

$$L_h X_h = U_h, \quad (3)$$

где L_h – разностный оператор, X_h – разностная матрица состояния, U_h – сетевая функция.

В простейшем случае устанавливают прямое соответствие нейрона клеточной сети узлу сеточной области. Тогда динамика клетки должна обеспечивать решение одного разностного уравнения с помощью итерационной формулы, соответствующей выбранному методу. При выборе подходящей итерационной формулы существенную роль играет вид разностного оператора L_h . Основными критериями выбора разностного оператора являются форма шаблона, на которой он определен, и требования к порядку обновления данных на заданном шаблоне.

Локально-асинхронный метод [4], ориентированный на применение в клеточных нейронных сетях, позволяет сформировать шаблон разностного оператора, ограниченный степенью соседства r , и обеспечить асинхронный режим обмена, снимающий жесткое ограничение на обновление всех данных на шаблоне перед

выполнением очередной итерации. Динамика клетки в данном случае определяется итерационной формулой

$$x_g(n+1) = x_g(n) - \frac{w_g}{h^2} [L_g x_g(n) - u_g], \quad (4)$$

где L_g – компонента разностного оператора, $x_g(n)$ – состояние клетки $c(g)$ на итерационном шаге n , h – величина шага дискретизации, w_g – внутренний параметр, u_g – значение сеточной функции.

Порядок обновления данных на шаблоне каждой компоненты разностного оператора L_g определяется хаотической последовательностью $\{J_n\}_{n=1}^{\infty}$ непустых множеств $\{1, 2, \dots, \eta\}$. Используя эту последовательность, строят последовательность итераций $\{x_g(n)\}_{n=1}^{\infty}$ по правилу

$$x_g(n) = \begin{cases} x_g(n-1), & g \notin J_n, \\ L_g [x_1(z_1(n)), x_2(z_2(n)), \dots, x_\eta(z_\eta(n))], & g \in J_n. \end{cases} \quad (5)$$

Максимальная эффективность асинхронного алгоритма достигается при условии

$$\mathfrak{R} = \{g \in \{1, 2, \dots, \eta\} \mid \forall n \in \mathbb{N} \exists m \geq n : g \in J_n\}. \quad (6)$$

В этом случае говорят, что хаотическая последовательность $\{J_n\}_{n=1}^{\infty}$ имеет максимальный осадок, что обусловлено наличием минимальной последовательности множеств $\{m_g\}_{g=0}^{\infty}$.

Возрастающую последовательность множеств называют минимальной при условиях:

- 1) $m_0 = 0$;
- 2) $\bigcup_{i=m_g+1}^{m_{g+1}} J_i = \{1, 2, \dots, n\}$, $g = 0, 1, 2, \dots$;
- 3) $\bigcup_{i=m_g+1}^{m_{g+1}-1} J_i \subset \{1, 2, \dots, n\}$, $g = 0, 1, 2, \dots$.

Поэтому множество $\{z_g(n)\}_{n=1}^{\infty}$, $g = 1, 2, \dots, \eta$ для асинхронного алгоритма (5) должно удовлетворять условиям: $z_g \leq (n-1)$, $z_i = (n) \rightarrow \infty$.

С целью упрощения алгоритма (5) формирования итерационной последовательности $\{x_g(n)\}_{n=1}^{\infty}$ вводят условие паразжимаемости на дифференциальный оператор L в виде неравенства

$$\|LX - \varphi\| < \|X - \varphi\|. \quad (6)$$

В этом случае формирование итерационной последовательности $\{x_g(n)\}_{n=1}^{\infty}$ будет происходить по правилу

$$x_g(n+1) = L_g(x_g(z_1^g(n)), \dots, x_g(z_{k_g(n)}^g(n))), \quad (7)$$

где $\{g(n) | n=1, 2, \dots\}$ – последовательность элементов $g(n) \in H$;

$Z^g = \{z_{k_g}^g(n)\}_{n=0}^{\infty}$, $g=1, \dots, H$, $k_g=1, \dots, k$ – множество целых неотрицательных чисел,

удовлетворяющих условиям

$$\begin{cases} 0 < z_{k_g(n)}^g(n) \leq n, & n > 0, \\ -M \leq z_{k_g(n)}^g(n) \leq 0, & n = 0. \end{cases} \quad (8)$$

Из условия (8) видно, что для начала итерации могут быть использованы данные от ранее стартовавших нейронов с некоторым ограничением M . Нейрон, опередивший своих соседей, может использовать устаревшие данные с глубиной запаздывания с максимальным значением, равным количеству итераций данного нейрона. Отсюда следует, что локально-асинхронный алгоритм вполне работоспособен в условиях физической асинхронности. Однако его максимальная эффективность может быть достигнута только при максимальном осадке хаотической последовательности $\{J_n\}_{n=1}^{\infty}$, т.е. при минимальном запаздывании одного нейрона относительно другого.

Имитационные модели

Имитационная модель межнейронных коммуникаций для упомянутого вычислительного процесса должна отвечать следующим условиям:

- входные потоки, поступающие от нейронов, определяемых функцией соседства $N_r(g)$, независимы;
- основными параметрами моделирования для каждого нейрона являются его состояние $\sigma(c)$ и $\tau(c)$ – время следующей модификации данных, которые потенциально могут привести к изменению состояния.

В соответствии с данным определением коммуникационная модель клеточной нейронной сети для каждого нейрона задает две основные функции:

$Get_state()$ – функция формирования нового состояния;

$Get_time()$ – функция формирования момента времени для нового состояния.

Состояние произвольного нейрона КНС в момент времени t может быть выражено зависимостью

$$\sigma_i(c) = Get_state(c, \{\sigma_x(c_i \in H_\alpha)\}_{i=1}^\alpha, t | x \rightarrow t, x < t).$$

Время следующей модификации

$$\tau(c) = \text{Get_time}(c, \{\sigma_x(c_i \in H_\alpha)\}_{i=1}^\alpha, t | x \rightarrow t, x < t).$$

В обеих формулах используются такие обозначения: x – момент времени, непосредственно предшествующий моменту времени t ; $\{\sigma_x(c_i \in H_\alpha)\}_{i=1}^\alpha$ – состояния нейронов, принадлежащих множеству соседства H_α .

Предположим, что в параллельном компьютере каждый процессорный элемент отвечает за моделирование одного нейрона клеточной сети. Рассмотрим синхронный и асинхронный подходы к реализации межпроцессорных коммуникаций для такой модели. В случае синхронного алгоритма предполагается наличие некоторого дополнительного ресурса, который выполняет функции глобальной синхронизации. Модельное время синхронной модели представим в виде последовательности дискретных приращений: $\tau(c) = k\Delta t$, где $k = 0, 1, 2, \dots$. В случае $\Delta t = \text{const}$ получаем синхронную модель с фиксированными приращениями. Это простейший вариант синхронной модели, требующий предварительного анализа алгоритма с целью выбора оптимального значения Δt . Более гибкий подход предполагает использование условия $\Delta t = \Delta t_{\max}$, где Δt_{\max} – длительность периода вычислений или обмена в самом медленном процессорном элементе. Для формирования начала и конца периода Δt в синхронном алгоритме, показанном на рис. 2, будем использовать глобальные операторы *Start_exchange* и *Stop_exchange*.

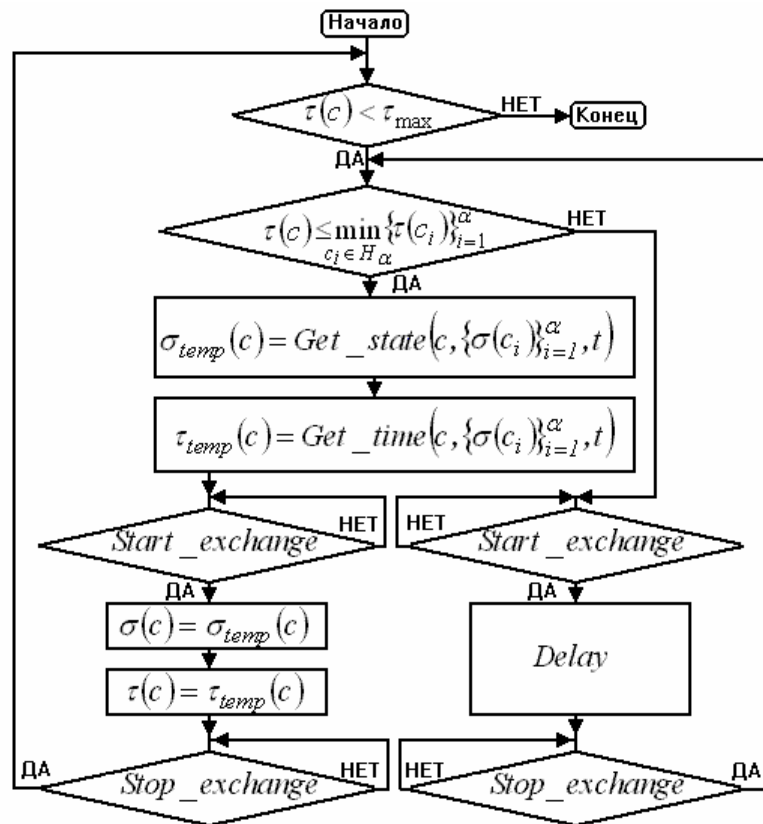


Рисунок 2 – Синхронный имитационный алгоритм

Синхронный алгоритм оперирует двумя экземплярами переменных времени и состояния, которые образуют множество внутренних параметров $\{\sigma_{temp}(c), \tau_{temp}(c)\}$ и множество глобальных параметров $\{\sigma(c), \tau(c)\}$. Предположим, что некоторый ПЭ выполнил операции вычисления внутренних параметров. Тогда он упирается в выполнение глобального оператора начала обмена данными $Start_exchange$. Срабатывание этого оператора определяется следующим логическим условием:

$$Start_exchange = \bigcap_{j=1}^N Start_exchange_j, \text{ где } Start_exchange_j - \text{признак завершения}$$

периода вычислений в j -м ПЭ. Из рис. 2 видно, что существуют две ветки алгоритма, одна из которых актуальна для работающих процессоров, а другая – для завершивших свои вычисления. Глобальное выполнение условия $Start_exchange$ ведет к наступлению периода обмена, в котором значения локальных переменных присваиваются глобальным переменным. Множество $\{\sigma(c), \tau(c)\}$ всегда доступно тем процессорным элементам, множество которых определено функцией соседства N_r и может быть использовано этими ПЭ на следующем этапе вычислений.

Период обмена завершается срабатыванием глобального оператора $Stop_exchange = \bigcap_{j=1}^N Stop_exchange_j$, действие которого аналогично оператору

$Start_exchange$, за исключением того, что этот оператор всегда предшествует наступлению периода вычислений. Условием завершения работы алгоритма является равенство или превышение модельным временем значения некоторой максимально допустимой величины τ_{max} .

Синхронный алгоритм может быть использован для реализации описанного выше локально-асинхронного метода. Положительной стороной его использования является обеспечение более высокой скорости сходимости итерационного процесса по сравнению с физической асинхронностью. К числу отрицательных характеристик можно отнести использование дополнительных ресурсов для обеспечения глобальной синхронизации и возможные существенные простои процессорных элементов в случае неоднородной сходимости вычислительного процесса.

Избавиться от дополнительных ресурсов при сохранении высокой скорости сходимости вычислительного процесса позволяет асинхронный вариант алгоритма межпроцессорных коммуникаций, показанный на рис. 3.

Существенное отличие рассматриваемого подхода от уже описанного синхронного варианта состоит в том, что модельное время $\tau(c)$ каждого из нейронов не разбивается на периоды вычисления и обмена, а носит непрерывный характер. Вследствие этого нейроны клеточной сети могут одновременно находиться на различных этапах вычислений. Ключевыми условиями продвижения локального модельного времени $\tau(c)$ отдельного нейрона в данном случае становятся значения локальных модельных времен $\tau(c_i)$, топологически связанных с ним соседних нейронов. Однако условие $\tau(c) \leq \min_{c_i \in H_\alpha} \{\tau(c_i)\}^\alpha$ не предполагает обнаружение его сразу же после возникновения. Для того чтобы это условие было обнаружено ПЭ,

и *Get_time* безопасно: ни один соседний нейрон не изменит своего состояния или локального времени до момента окончания вычислений нейроном c . Это обстоятельство гарантирует минимальную производительность КНС, которая в среднем значительно выше.

Используя свойства локально-асинхронного метода, обеспечивающие его сходимость в условиях физической асинхронности, в сочетании с описанным асинхронным коммуникационным алгоритмом, можно практически исключить простой вычислительных ресурсов в ходе решения задачи. Признак завершения асинхронного алгоритма носит локальный характер и предусматривает остановку при достижении локального модельного времени некоторой критической величины, выбираемой из условий скорости сходимости локально-асинхронного метода.

Агрегативные имитационные модели

В рассмотренных выше алгоритмах один ПЭ содержит только один нейрон клеточной сети. Такая структура может оказаться весьма расточительной в случае критичности коммуникационных ресурсов. Вычислительный процесс будет значительно эффективнее в случае размещения на одном ПЭ некоторого подмножества нейронов.

Рассмотрим работу агрегативной имитационной модели на примере двумерной клеточной сети C с размерностью $n \times n$. Пусть $n > m, n:m$ и доступно к использованию $(n/m)^2$ процессорных элементов, каждый из которых содержит подсеть $C_{k,l} | 0 \leq k, l \leq n/m - 1$ из $m \times m$ нейронов (рис. 4).

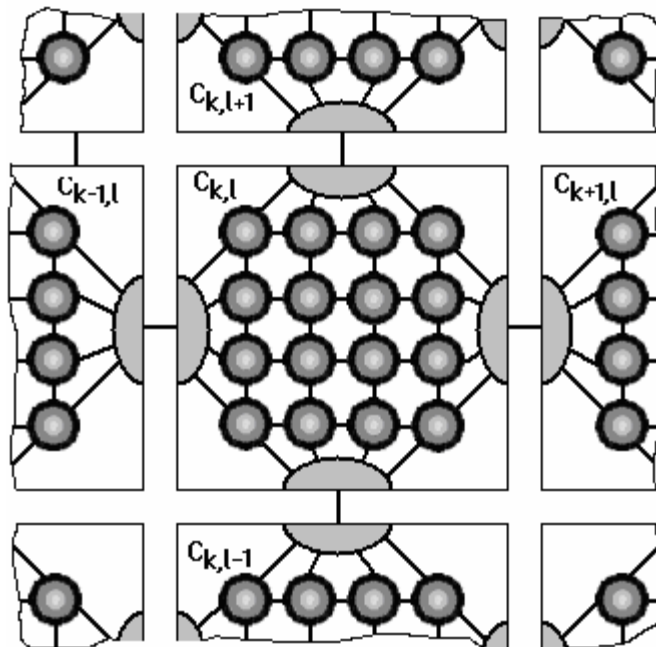


Рисунок 4 – Агрегативная имитационная модель ($m = 4$)

Множество нейронов $C = \{C_{k,l}\}_{k,l=0}^{n/m-1}$, интерпретируемых как агрегативная структура, может быть условно разбито на два подмножества:

- подмножество внешних нейронов $O = \left\{ \left\{ c_{i-m,j}, c_{(i+l)-m-l,j} \right\}_{i=0}^{n/m-1} \right\}_{j=0}^{n-l}$ состоит из нейронов, для которых один или несколько соседей находятся в одной или двух соседних подсетях;
- подмножество внутренних нейронов $I = C \setminus O$, для которых все соседние нейроны принадлежат одной подсети.

Основное отличие структуры данных агрегативного алгоритма состоит в том, что каждый из нейронов дополнительно хранит данные о подсетях, с которыми для этого нейрона существуют коммуникации согласно топологии КНС. Для топологии, представленной на рис. 4, данные могут быть автоматически сгенерированы в виде множеств $w_{i,j} \in W$ исходя из следующих условий:

$$C_{k-1,l} \in \{w_{km,j}\}_{j=lm}^{m(l+1)-1}; C_{k+1,l} \in \{w_{m(k+1)-1,j}\}_{j=lm}^{m(l+1)-1};$$

$$C_{k,l-1} \in \{w_{i,lm}\}_{i=km}^{m(k+1)-1}; C_{k,l+1} \in \{w_{i,m(l+1)-1}\}_{i=km}^{m(k+1)-1}.$$

Агрегативная структура $C_{k,l}$ обладает одним каналом связи с каждой соседней агрегативной структурой. Поэтому возникает вопрос приоритетности доступа к ресурсам этих каналов. Для решения данной задачи введем множество канальных времен $T(C_{k,l}) = \{\tau(C_{k-1,l}), \tau(C_{k+1,l}), \tau(C_{k,l-1}), \tau(C_{k,l+1})\}$. Текущее значение канального времени всегда равно минимальному значению локального времени, определяемому на множестве нейронов, содержащих в своем множестве $w_{i,j}$ соответствующую агрегативную структуру.

Например, множество $W_{k,l}$ непустых подмножеств $w_{i,j}$ для агрегативной структуры $C_{k,l}$, показанной на рис. 4, содержит следующие подмножества:

$$W_{k,l} = \{w_{4k,4l}, w_{4k+1,4l}, w_{4k+2,4l}, w_{4k+3,4l},$$

$$w_{4k,4l+1}, w_{4k,4l+2}, w_{4k+1,4l+3}, w_{4k+1,4l+3},$$

$$w_{4k+2,4l+3}, w_{4k+3,4l+3}, w_{4k+3,4l+1}, w_{4k+3,4l+2}\}.$$

При формировании канального времени $\tau(C_{k-1,l})$ используют подмножество локальных времен $\{\tau(c_{4k,4l}), \tau(c_{4k,4l+1}), \tau(c_{4k,4l+2}), \tau(c_{4k,4l+3})\}$, поскольку агрегативная структура $C_{k-1,l}$ является элементом соответствующих подмножеств:

$$w_{4k,4l} = \{C_{k-1,l}, C_{k,l-1}\}, w_{4k,4l+1} = \{C_{k-1,l}\},$$

$$w_{4k,4l+2} = \{C_{k-1,l}\}, w_{4k,4l+3} = \{C_{k-1,l}, C_{k,l+1}\}.$$

Следовательно, $\tau(C_{k-1,l}) = \min\{\tau(c_{4k,4l}), \tau(c_{4k,4l+1}), \tau(c_{4k,4l+2}), \tau(c_{4k,4l+3})\}$.

Такой подход к продвижению канального времени освобождает агрегативный алгоритм от клинчевых ситуаций, поскольку нейрон с минимальным временем всегда будет иметь возможность для продвижения вперед.

Блок-схема агрегативного алгоритма показана на рис. 5. Основное отличие этого алгоритма от рассмотренного ранее состоит в наличии ветвления, позволяющего по-разному реализовать продвижение имитационного времени в зависимости от принадлежности нейрона к множеству O или множеству I . В качестве признака $c \in O$ выбрано условие непустоты множества w , ассоциированного с соответствующим нейроном c . Продвижение локального времени для данного типа нейронов носит групповой характер и определяется выражением $\tau(C) = \min_{c_i \in O} \{\tau(c_i)\}$, которое указывает на то, что групповое локальное время $\tau(C)$ равно минимальному локальному времени нейрона, входящего в соответствующую группу агрегативного обмена. Разрешение на дальнейшую эволюцию при этом может быть получено в случае, когда данное групповое локальное время является минимальным среди всех агрегативных групп, подключенных к данному каналу коммуникации.

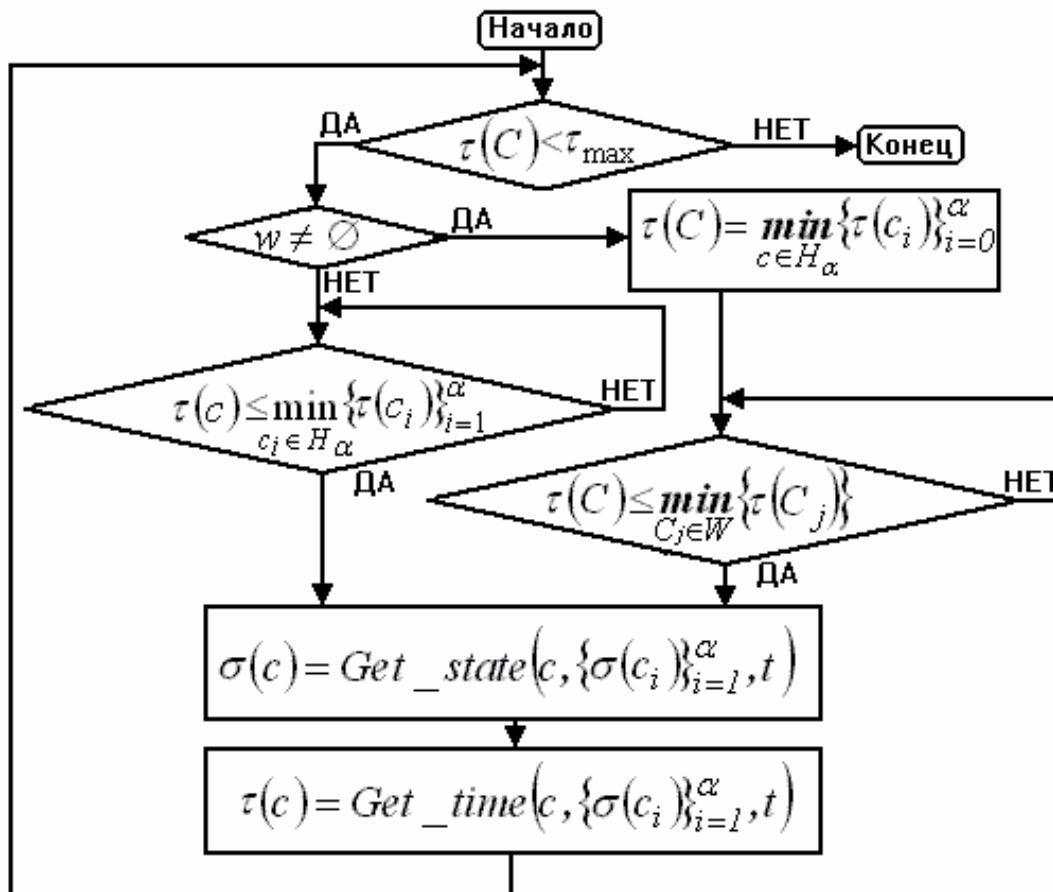


Рисунок 5 – Агрегативный имитационный алгоритм

В случае $w = \emptyset$ получаем условие принадлежности нейрона к множеству I . Ход продвижения локального времени для данного типа нейронов аналогичен рассмотренному ранее при описании работы асинхронного имитационного алгоритма.

Использование данного подхода к организации процессов взаимодействия между нейронами [4] позволило реализовать целый ряд асинхронных моделей клеточных нейронных сетей для решения краевых задач математической физики локально-асинхронным методом.

Литература

1. Geman S., Geman D. Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images // IEEE Transactions on pattern analysis and machine intelligence. – 1984. – Vol. 6, № 6. – P. 721-741.
2. Нестеренко Б.Б., Новотарский М.А. Мультипроцессорные системы. – Киев: Ин-т математики, 1995. – 408 с.
3. Chua L.O. CNN: A Paradigm for Complexity. – World Scientific, Singapore, 1998. – 380 p.
4. Новотарський М.А., Нестеренко Б.Б. Штучні нейронні мережі: обчислення. – Київ: Ін-т математики, 2004. – 408 с.
5. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. – М.: Наука, 1978. – 592 с.

Б.Б. Нестеренко, М.А. Новотарський

Імітаційне моделювання клітинних нейронних мереж

У роботі дано короткий огляд класичних принципів організації та функціонування дискретних клітинних нейронних мереж. Розглянута постановка задачі розв'язання операторного рівняння локально-асинхронним методом, орієнтованим на реалізацію в гомогенних обчислювальних структурах. Описаний підхід до створення імітаційних моделей для клітинних нейронних мереж з різними видами міжнейронної взаємодії. Запропоновані імітаційні алгоритми синхронної, асинхронної та агрегативної взаємодії в імітаційних моделях клітинних нейронних мереж.

B.B. Nesterenko, M.A. Novotarsky

Imitation Modelling Cellulate Networks

In this paper the brief review of classical principles of discrete cellular neural network organization and functioning is given. The problem of operational equation solving by the local – asynchronous method focused on realization in homogeneous computing structures is considered. The approach to cellular neural networks simulations with some kinds of interneural interaction is described. Synchronous, asynchronous and aggregate interaction algorithms for cellular neural networks simulations are offered.

Статья поступила в редакцию 25.06.2005.