

АППАРАТНАЯ РЕАЛИЗАЦИЯ НЕЧЕТКОЙ СЕТИ СМАС И ЕЕ ПРИМЕНЕНИЕ ДЛЯ ЗАДАЧ СЖАТИЯ ИЗОБРАЖЕНИЙ

Руденко О.Г., Бессонов А.А., Бобух В.А.

Введение. Получающие в последнее время все более широкое распространение искусственные нейронные сети (ИНС) являются особенно эффективными при решении задач идентификации и управления нелинейными динамическими объектами в реальном времени, обработки и фильтрации сигналов и изображений.

Высокая скорость обучения и малый объем требуемой памяти, достигаемые за счет специального кодирования информации, делает сеть СМАС – *Cerebellar Model Articulation Controller* (церебральная модель артикуляционного контроллера) особенно привлекательной для реализации в микроконтроллерных системах управления нелинейными объектами [1].

Дополнительного положительного эффекта можно достичь, переходя от традиционной сети СМАС, использующей прямоугольные активационные функции, к нечеткой или фаззи СМАС (ФСМАС), функции принадлежности которой отличны от прямоугольных [2, 3].

Сеть ФСМАС. Основной особенностью ФСМАС является лингвистическое описание действий, выполняемых в соответствии с текущим состоянием объекта, что позволяет включать в алгоритм ее работы знания эксперта, а сама сеть при этом может рассматриваться как экспертная система реального времени.

Нечеткие правила описывают возможные состояния системы и соответствующие им действия. Правила, реализуемые нечеткой нейросетью СМАС, имеют вид [4]

$$\text{IF } x_1 \text{ IS } X_{1,i} \text{ AND } \dots x_N \text{ IS } X_{N,i} \text{ THEN } f_i = w_m, \quad i = \overline{1, \rho}, \quad (1)$$

где $X_{1,i}, \dots, X_{N,i}$ - лингвистические значения в антецеденте i -го правила; f_i - функция в консеквенте i -го правила; w_m - настраиваемые параметры консеквента i -го правила; ρ – задаваемый априорно параметр, определяющий число активных функций принадлежности; N – размерность входного сигнала.

При использовании в качестве t -нормы произведения степень выполнения i -го правила вычисляется в соответствии с выражением

$$\Phi_i = \prod_{j=1}^N \phi_{ij}(x_j), \quad i = \overline{1, \rho}, \quad (2)$$

где $\phi_{ij}(x_j)$ - функция принадлежности на j -ом входе в антецеденте i -го правила.

Выходной сигнал нечеткой сети СМАС вычисляется следующим образом:

$$y = H(a^T \Phi(x))w, \quad (3)$$

$$\text{где } \Phi(x) = \begin{bmatrix} \Phi_1(x) & 0 & \dots & 0 \\ 0 & \Phi_2(x) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \Phi_n(x) \end{bmatrix}, \quad \Phi_i = \frac{\prod_{j=1}^N \phi_{ij}(x_j)}{\sum_{k=1}^{\rho} \prod_{j=1}^N \phi_{kj}(x_j)}, \quad i = \overline{1, \rho};$$

$H(\bullet)$ - используемый оператор хеширования информации; a – вектор ассоциаций, полученный на этапе кодирования информации; w – весовые параметры сети.

Алгоритмы хеширования информации в ФСМАС. Хеширование заключается в отображении большой требуемой памяти в меньшую, физически реализуемую. Объем

большей памяти определяется размерностью вектора ассоциаций, полученного на этапе кодирования. Если адреса ячеек большей памяти лежат в интервале $[1, M]$, а меньшей – в интервале $[1, m]$, то функция преобразования адреса (хеш-функция) должна удовлетворять требованию

$$1 \leq H(k) < m, \quad \forall 1 \leq k \leq M.$$

Выбор функции $H(k)$ является весьма трудной задачей, однако многочисленные исследования показали, что хорошие результаты обеспечивает применение двух основных типов хеш-функций, один из которых основан на делении, а другой – на умножении [5].

Хеш-функция, реализующая метод, основанный на делении, имеет вид

$$H(k) = 1 + k \bmod m, \quad (4)$$

где $k \bmod m$ означает деление по модулю m .

Адрес ячейки памяти с меньшим объемом при методе хеширования, основанном на умножении, вычисляется по следующей формуле:

$$H(k) = 1 + \left\lfloor m \left[\left(\frac{F}{w} k \right) \bmod 1 \right] \right\rfloor, \quad (5)$$

где w – размер машинного слова; F – некоторая целая константа, взаимно простая с w ; $\lfloor \bullet \rfloor$ – означает округление в сторону ближайшего целого числа.

В качестве F возможно использование чисел Фибоначчи.

Хеш-функция, основанная на делении, равномерно распределяет большую память в меньшую в соответствии с адресом.

Функция, описываемая уравнением (5), «тасует» элементы памяти.

Обучение сети. Обучение сети РСМАС состоит в настройке вектора ее весовых параметров w на основании предъявления ей обучающих пар $\{x(k), y(k)\}$ и сравнения реакции сети $y(k)$ с требуемой $y(k)$. Соответствующий алгоритм обучения может быть записан следующим образом:

$$w(k+1) = w(k) + \frac{y(k) - y(k)}{\|\Phi(x(k))a(k)\|^2} \Phi(x(k))a(k). \quad (6)$$

Однако в ряде случаев для улучшения аппроксимирующих свойств сети возникает необходимость настройки не только весовых параметров сети, но и параметров базисных функций. Наиболее эффективным оказывается применение такого подхода для настройки центров μ и дисперсий σ гауссовых функций активации.

Настройка данных параметров может быть осуществлена, например по алгоритму, аналогичному алгоритму Уидроу-Хоффа:

$$m(k+1) = m(k) + \frac{y(k) - y(k)}{\left\| \frac{\partial y(k)}{\partial m(k)} \right\|^2} \frac{\partial y(k)}{\partial m(k)}, \quad (7)$$

где m – вектор настраиваемых параметров.

В случае использования гауссовых функций в качестве m используются центры μ и дисперсии σ , а векторы частных производных $\frac{\partial y(k)}{\partial \mu(k)}$ и $\frac{\partial y(k)}{\partial \sigma(k)}$ вычисляются по формулам

$$\frac{\partial y(k)}{\partial \mu_{ji}(k)} = 2w_i(k) \prod_{s=1}^N e^{-\frac{(x_s - \mu_{si}(k))^2}{\sigma_{si}(k)^2}} \frac{(x_j - \mu_{ji}(k))}{\sigma_{ji}(k)^2},$$

$$\frac{\partial y(k)}{\partial \sigma_{ji}(k)} = 2w_i(k) \prod_{s=1}^N e^{-\frac{(x_s - \mu_{si}(k))^2}{\sigma_{si}(k)^2}} \frac{(x_j - \mu_{ji}(k))^2}{\sigma_{ji}(k)^3},$$

где i – номер возбужденной функции принадлежности; j – номер компонент векторов ассоциаций и весов, соответствующих данной функции принадлежности.

Аппаратная реализация. Рассмотрим кодирование переменной x_i , представляющей собой n -разрядное двоичное число. Так как входной слой сети СМАС содержит ρ ступеней квантования, таких, что для любой ступени существует другая ступень, на которой области квантования сдвинуты относительно рассматриваемой ступени на один уровень квантования входной переменной x_i , являющейся двоичным числом, то такой сдвиг соответствует единице младшего разряда входной переменной. Присвоим ступени квантования индекс $j = \overline{0, \rho - 1}$.

В связи с тем, что область квантования содержит ρ уровней квантования входной переменной, функция принадлежности, соответствующая любой области квантования, определена в ρ точках. Аргументом этой функции будет являться значение

$$u = (x_i + j) \bmod \rho, \quad (8)$$

где \bmod – функция вычисления остатка от целочисленного деления.

Номер активной функции принадлежности на ступени j можно определить как

$$v = (x_i + j) \operatorname{div} \rho, \quad (9)$$

где div – функция целочисленного деления.

Как известно, аппаратная реализация операции деления на произвольное число является либо ресурсоёмкой либо медленной. Поэтому предлагается выбирать значения ρ , равные 2^m . В этом случае операция деления может быть заменена операцией сдвига на m разрядов вправо, а остаток от деления, записанный в m младших разрядах, исчезает.

В классическом варианте сети СМАС активационные функции нейронов областей квантования имеют прямоугольную форму. Однако, как показывают исследования, лучший результат достигается при использовании функций высших порядков (линейных, квадратичных, кубических), а также тригонометрических и степенных функций.

Аппаратная реализация широкого спектра функций принадлежности, отличных от прямоугольных, сложна и ресурсоёмка. Поэтому целесообразно функцию активации задавать таблично в ОЗУ, содержащем $2m$ ячеек. Разрядность ячеек ОЗУ h определяется областью допустимых значений функции принадлежности, а также необходимой точностью задания значений функции принадлежности. Если, например, $m = 8$, то необходимо ОЗУ ёмкостью 256 ячеек. Блоки ОЗУ такого объёма имеются в большинстве современных ПЛИС (например, семейство ACEX1K фирмы ALTERA). Подобная реализация функций активации является простой и гибкой, так как позволяет задать функцию любого вида.

Структура блока кодирования переменной x_i представлена на рис.1. Код переменной x_i зашелкивается в регистре RgX, откуда поступает на сумматор $\Sigma 1$. На второй вход этого сумматора поступает m – разрядное значение номера ступени j , на которой в данный момент происходит кодирование. Это значение формируется устройством случайной выборки ступени, служащим для устранения диагональной структуры ассоциативного поля, и, соответственно, улучшения аппроксимирующих свойств сети. С выхода сумматора n – разрядный код поступает на логический сдвигатель (LSh – Logical Shifter), позволяющий изменять значение m . Значение u определяют m младших разрядов выходного кода LSh, а

значение v – соответственно старшие $k = n - m$ разрядов. Значение u через мультиплексор $MX1$ поступает на адресные входы ОЗУ хранения функции принадлежности. Мультиплексор $MX1$, а также буфер $B1$ служат для инициализации ОЗУ значениями функции принадлежности от устройства управления. С выхода ОЗУ h – разрядное значение функции принадлежности поступает на регистр Rg VFS, а с регистра – на выход VFS_i (Value of Function on Stair i) блока кодирования (CB – Coding Block). Значение v поступает на регистр Rg NNS, а оттуда – на выход NNS_i (Number of Neuron on Stair i) блока. Тактирующие импульсы на регистры, а также сигналы управления ОЗУ формируются устройством управления, общим для всей структуры.

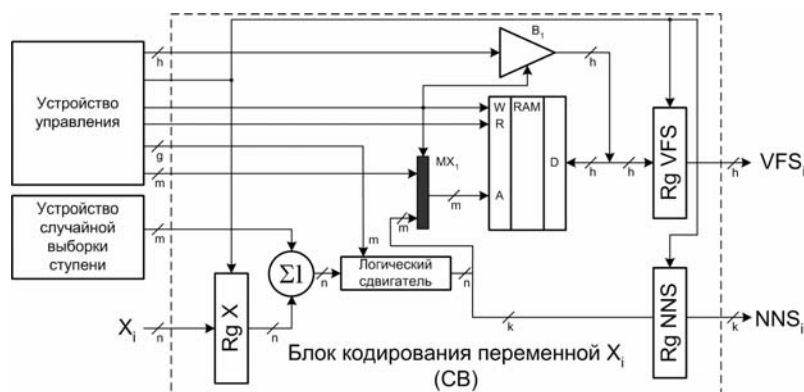


Рис. 1. Блок кодирования переменной x_i

Таким образом, для кодирования переменной x_i с помощью m ступеней квантования необходимо 2^m тактов. При $m < 5$ такое кодирование может осуществляться с частотой 10 – 20 МГц, то есть в реальном времени для обработки телевизионных изображений. В течение этих 2^m тактов значение x_i хранится во входном регистре RgX и обновляется по их истечении.

Устройство случайной выборки ступени может представлять собой m -разрядный линейный рекуррентный регистр с обратными связями, определяемыми простым полиномом. Такой регистр генерирует $2^m - 1$ различных кодовых комбинаций, обладающих статистическими свойствами равномерно распределённой случайной последовательности. Одна комбинация является запрещённой, так как на очередном такте работы регистра приводит к генерации самой себя. Так как работа блока кодирования предусматривает использование 2^m комбинаций, необходимо предусмотреть её аппаратную генерацию и подстановку вместо кода регистра на любом такте (при этом не производится очередной сдвиг регистра). Вид последовательности определяется начальной установкой регистра. Эта операция производится однократно перед первым обучением сети. Регистры, отвечающие за различные входные переменные, должны быть проинициализированы различными значениями. Эти значения, равно как и начальные значения весов в сети, могут быть сформированы аппаратным датчиком случайных чисел [6]. Рассмотрим возможную структуру аппаратной реализации сети СМАС, представленную на рис. 2. Сеть обрабатывает две входные переменные, каждая из которых кодируется блоками CB, содержащими равное число ступеней квантования. Функционирование такой структуры состоит в следующем. На j -м такте блоки CB формируют значения номеров функций принадлежности в текущих ступенях (NNS_1 и NNS_2) и значения их функций активации (VFS_1 и VFS_2). Значения VFS поступают на умножитель $M1$, формирующий произведение функций активации текущих ступеней. Значение этого произведения поступает на умножитель $M2$, вычисляющий произведение веса и произведения функций активации. Адрес веса в ОЗУ весов определяется алгоритмом хеширования, входным параметром которого является комбинация значений NNS. Накапливающий сумматор $\Sigma 2$ вычисляет сумму произведений весов и функций активации и после выполнения m тактов в регистре RgY формирует значение функции, со-

ответствующее входным переменным. С помощью сумматора $\Sigma 3$ формируется код ошибки e_k , определяемый как разность полученного y и требуемого y_k значений функции. Значение ошибки поступает на блок реализации алгоритма обучения, который в течение m тактов вычисления накапливал адреса выборки весов, значения весов и произведений функций активации. Получив значение ошибки, алгоритм обучения модифицирует участвовавшие в вычислении веса и записывает их в ОЗУ весов по соответствующим адресам. Координацию работы узлов сети осуществляет устройство управления.

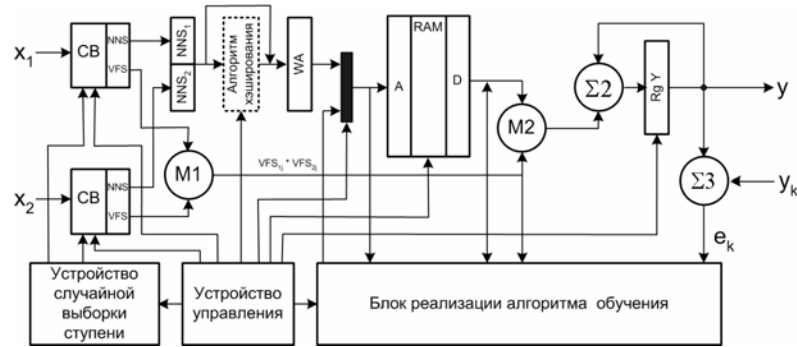
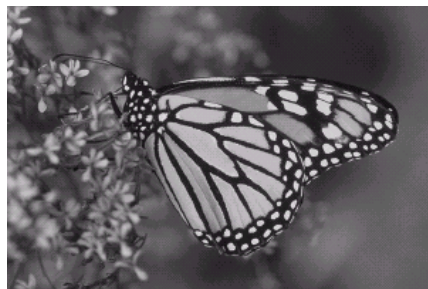


Рис. 2. Вариант аппаратной реализации сети СМАС

Практические результаты. В качестве тестовой рассматривалась задача кодирования изображения, приведенного на рис.3 а). Размерность данного изображения составляет 300x200 пикселей, т.е. для его хранения в формате bmp необходимо 60 Кб памяти, а для хранения весовых параметров сети, с помощью которой осуществлялось кодирование, требуется 8 Кб. Таким образом, коэффициент сжатия составил 7,5 раза. Результаты кодирования данного изображения с помощью сети FCMAC представлены на рис.3 б)-д). Изображение на рис.3 б) получено с помощью сети с косинусоидальными функциями принадлежности, на рис.3 в) – с гауссовыми, а на рис.3 г) – с гауссовыми, параметры которых настраивались по алгоритму (7).



а)



б)



в)



г)

Рис. 3 Результат сжатия изображения

Как видно из результатов, настройка параметров функций принадлежности позволяет более точно восстанавливать кодируемые изображения.

Основные результаты и выводы. Как показывают результаты моделирования, нейронная сеть FCMAC является достаточно эффективной и удобной при решении задач обработки, в частности сжатия изображений. Существенная экономия требуемой памяти в сочетании с простыми алгоритмами обучения делают ее особенно привлекательной для реализации в микроконтроллерных системах. Следует отметить, что приведенная в статье аппаратная реализация сети FCMAC является универсальной и может использоваться не только в задачах обработки изображений, но и для аппроксимации функций, идентификации нелинейных объектов и управления ими, т.к. во-первых, при этом возможно использование любых базисных функций, а во-вторых – в схеме предусмотрен блок хеширования информации. Дело в том, что вопрос выбора базисных функций нейронной сети FCMAC остается открытым и определяется характером решаемой прикладной задачи. Хеширование же необходимо при обработке больших объемов информации (изображения высокого качества, объекты большой размерности).

In this paper some questions of FCMAC hardware realization and its application for an image compression are considered. The results of some simulated examples performed with assistance of Matlab are presented. It testified the effectiveness of FCMAC for problem solving of image processing in real time.

1. Albus J.S. A new approach to manipulator control: the cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control, 1975. – Vol. 97. – №3. – P. 220-227.
2. Руденко О.Г., Бессонов А.А. О выборе базисных функций в нейронной сети CMAC // Проблемы управления и информатики. –2004. –№2. – С. 143-154.
3. Руденко О.Г., Бессонов А.А. Адаптивное управление нелинейными объектами с помощью нейронной сети CMAC // Проблемы управления и информатики. – 2004. –№5. – С. 16-30.
4. K. S. Hwang, C. S. Lin and C. L. Chang. Fuzzy Cerebellar Model Articulation Controller // Proc. of the Second Chinese World Congress on Intelligent Control and Intelligent Automation. - 1997. - Pp. 1536-1541.
5. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. - М.: Вильямс, 2000. С. 549-597.
6. Торба А.А., Елаков С.Г., Степченко А.З. Генерация равновероятных случайных последовательностей на основе физических датчиков // «Радиотехника»: Всеукр. межвед. науч.-техн. сб. - Харьков. ХТУРЭ. - 2001. - Вып. 119. - С. 108-113.