ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2025, № 2, с. 82–97

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

УДК 004.31

АРХИТЕКТУРА БОРТОВОГО НЕЙРОСЕТЕВОГО ВЫЧИСЛИТЕЛЯ НА БАЗЕ КРИСТАЛЛА FPGA С ДИНАМИЧЕСКОЙ ПЕРЕНАСТРОЙКОЙ

© 2025 г. А. М. Соловьев*

ГК "Ростех", АО "Концерн "Созвездие", Воронеж, Россия

**e-mail: darkscribens@gmail.com*

Поступила в редакцию 21.08.2024 г.

После доработки 25.10.2024 г.

Принята к публикации 24.02.2025 г.

Посвящена развитию принципов построения аппаратной архитектуры нейросетевых вычислителей, применяемых в качестве ускорителей и сопроцессоров, для реализации экспертных и интеллектуальных систем в объектах с высокой степенью критичности, в том числе в бортовом оборудовании воздушных судов. Разработана модель нейросетевого вычислителя, построенного на базе кристалла FPGA, которая позволяет динамически развертывать на его основе искусственные нейронные сети без необходимости перепрограммирования самого кристалла FPGA. Описаны принципы развертывания и обучения нейросети с помощью представленного нейросетевого вычислителя. Приведен пример его использования в экспертной системе, осуществляющей анализ состояния бортовой информационно-вычислительной сети воздушного судна на предмет возникновения нештатных ситуаций в реальном масштабе времени и выполняющей парирование возникающих отказов методом динамической реконфигурации.

Ключевые слова: искусственные нейронные сети, нейросетевой вычислитель, авионика, интеллектуальная система, бортовая информационно-вычислительная сеть, динамическая реконфигурация

DOI: 10.31857/S0002338825020065, **EDN:** ASAULL

ARCHITECTURE OF AN ON-BOARD NEURAL NETWORK COMPUTER BASED ON FPGA CRYSTAL WITH DYNAMIC RECONFIGURATION

A. M. Solovyov*

JSC Concern "Sozvezdie", Voronezh, Russia

**e-mail: darkscribens@gmail.com*

The work is devoted to the development of principles for constructing hardware architecture of neural network computers used as accelerators and coprocessors for the implementation of expert and intelligent systems in objects with a high degree of criticality, including on-board equipment of aircraft. A model of a neural network computer built on the basis of an FPGA crystal has been developed, which allows dynamic deployment of artificial neural networks on its basis without the need to reprogram the FPGA crystal itself. The principles of deployment and training of a neural network based on the presented neural network computer are described. An example of its use in an expert system that analyzes the state of an onboard information and computing network of an aircraft for the occurrence of abnormal situations in real time and performs parrying of arising failures by the method of dynamic reconfiguration is given.

Keywords: artificial neural networks, neural network computer, avionics, intelligent system, on-board information-computer network, dynamic reconfiguration

Введение. Активное развитие методов машинного обучения открывает возможность реализации в современных технических системах таких задач, как прогнозирование отказов и обслуживание по состоянию, интеллектуальное управление и оптимизация режимов работы, распознавание образов и обработка сигналов, построение автономных интеллектуальных систем и т.п. [1–10]. Примерами использования данного подхода в авиации являются интеллектуальные системы управления полетом, обеспечивающие повышение безопасности

и оптимизацию летно-технических характеристик самолета. Развитие подобных систем в перспективе позволит внедрить на борту магистральных воздушных судов технологии “виртуальный второй пилот” и принципов беспилотной авиации.

Однако применение методов управления, основанных на традиционных нейронных сетях, реализованных программно, встречает определенные трудности, связанные в первую очередь с вычислительной сложностью алгоритмов машинного обучения. Современные нейронные сети могут насчитывать сотни миллиардов параметров, что делает использование классических вычислительных устройств на базе центральных процессоров неэффективным с точки зрения скорости вычислений, затрачиваемой энергии и в целом при помощи их вычислительного ресурса. Данная проблема ограничивает применение нейросетей, реализуемых программно, в ряде объектов. В связи с этим на практике все большую популярность получают специализированные под нейросетевые алгоритмы вычислительные устройства, архитектура которых учитывает особенности осуществляемых преобразований. Такие вычислительные устройства называют аппаратными ускорителями нейронных сетей или нейросетевыми вычислителями (ASIC for artificial neural networks).

Нейросетевой вычислитель (НСВ) – это специализированная аппаратная разработка (чип или система на кристалле), предназначенная для выполнения операций и задач, связанных с нейронными сетями. Такие нейросетевые чипы могут обеспечивать более эффективное выполнение операций с большим количеством векторов данных, используемых в нейронных сетях, по сравнению с общими центральными процессорами (CPU) или графическими процессорами (GPU). НСВ обладают специализированными устройствами выполнения операций, такими как умножение матриц, активационные функции и управление памятью, оптимизированными для обработки данных в нейронных сетях. Это позволяет им достигать высокой производительности и энергоэффективности по сравнению с универсальными архитектурами.

В рамках настоящей работы рассматривается применение НСВ для широкого класса автономных систем, к которым относятся беспилотные летательные аппараты (БПЛА), мобильные роботы, спутниковые системы, современные системы авионики и мн. др. [11–14]. Для таких систем наиболее характерно применение вычислителей на основе кристалла FPGA в связи с их высокой производительностью при умеренном энергопотреблении [15–20].

Для повышения гибкости и масштабируемости аппаратной нейросети, построенной на базе кристалла FPGA, необходимо разработать модель НСВ, которая позволяла бы динамически изменять свою нейросетевую структуру без необходимости перепрограммирования кристалла FPGA (смены RTL-проекта). Разработке такой модели НСВ посвящена настоящая работа.

1. Математическая модель ядра НСВ. Одним из способов выполнения требований гибкости и масштабируемости НСВ является использование однородной ячеистой структуры, представляющей собой матрицу из однотипных взаимосвязанных вычислительных элементов (нейронов), что хорошо согласуется с принципами функционирования чипов, построенных на основе технологии FPGA. При этом коэффициенты (веса) взаимосвязей, определяющие архитектуру аппаратной нейросети, удобно хранить в оперативном запоминающем устройстве (ОЗУ), внутреннем или внешнем в зависимости от необходимого объема, что позволит динамически перезагружать разные нейросети, не прибегая к перепрограммированию самого кристалла FPGA. Исходя из этого, модель ядра НСВ имеет вид, представленный на рис. 1 [21, 22].

Как видно из рис. 1, ядро НСВ состоит из нейронов N_j, входы которых умножаются на весовые коэффициенты a_{ij}, в каждом нейроне производится преобразование взвешенной суммы входов с помощью функции активации f_j и формируется выход d_j, $i = j = \overline{1,n}$, где n – количество нейронов. Ядро НСВ возможно описать как граф с помощью матриц смежности и инцидентности. Однако использование такого подхода приведет к нерациональному задействованию ресурсов кристалла FPGA (в частности, объема ОЗУ). Учитывая особенность архитектуры нейронов, а именно наличия многих входов и только одного выхода, ядро НСВ можно описать с помощью матрицы взаимодействия нейронов (МВН) $M \in \mathbb{R}^{n\times n}$, элементы которой вне диагонали являются весовыми коэффициентами входов нейрона с номером, соответствующим номеру столбца, а диагональный элемент служит выходом данного нейрона (табл. 1).

Таблица 1. Матрица взаимодействия нейронов НСВ

M	**1**	**2**	**...**	***n***
1	d_1	a_{12}	...	a_{1n}
2	a_{21}	d_2	...	a_{2n}
...	...	...	...	...
n	a_{n1}	a_{n2}	...	d_n

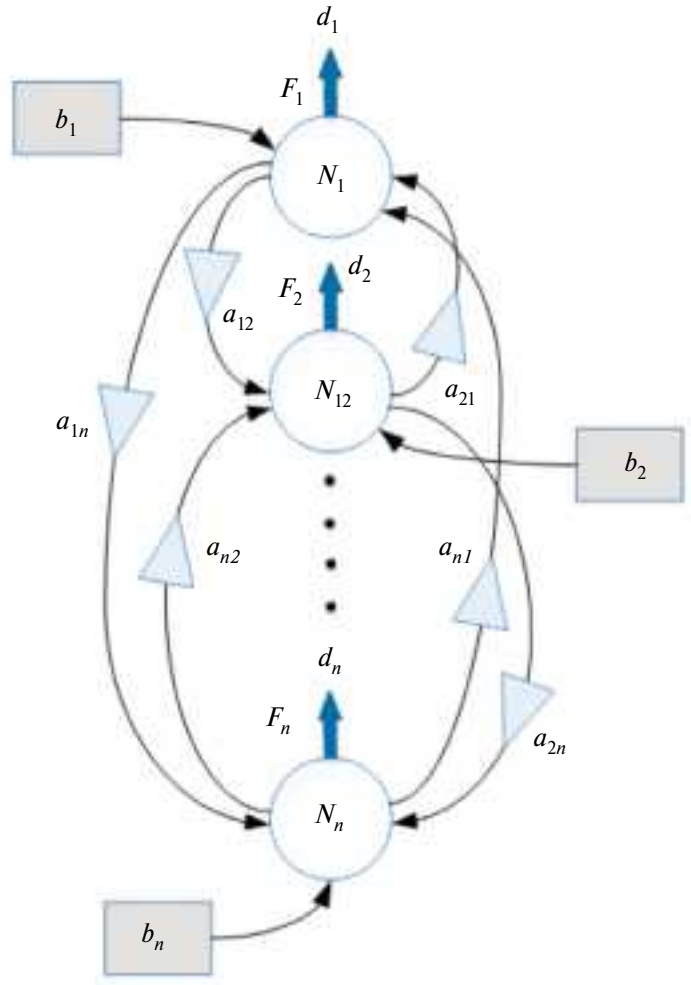

Рис. 1. Модель ядра НСВ.

Помимо МВН необходимо также описать набор функций активации, используемых в НСВ с привязкой к каждому нейрону. С этой целью дополнительно к МВН введем вектор функций активации (ВФА) F, элементы которого f_i указывают на номер функции активации, реализуемой в нейроне N_i. Коэффициенты a_{ij} в МВН влияют на наклон функции активации каждого нейрона, однако для повышения гибкости НСВ и лучшей обучаемости необходимо также ввести коэффициенты смещения b_i, позволяющие переместить функцию активации каждого нейрона влево или вправо. С этой целью введем вектор коэффициентов смещения B, содержащий b_i. Таким образом, НСВ математически однозначно описывается множеством

$$\left\{M_{n\times n}, B_{1\times n}, F_{1\times n}\right\}, \tag{1.1}$$

где M – матрица взаимодействия нейронов, $B=(b_i)_{i=\overline{1,n}}$ – вектор смещения, $F=(f_i)_{i=\overline{1,n}}$ – вектор-функция активации, n – количество нейронов в НСВ.

Динамику НСВ запишем следующим преобразованием:

$$d^t = F\left(d^{t-1}\widehat{M} + B\right), \tag{1.2}$$

где d – вектор выходов нейронов размерностью $1\times n$, принимающий значения диагональных элементов МВН на текущей (индекс t) или предыдущей (индекс t-1) итерациях расчета; $\widehat{M}$ – матрица размерностью $n\times n$, принимающая значения всех элементов МВН, которые лежат вне ее главной диагонали.

Одним из важных аспектов, определяющих возможности НСВ, является выбор функций активации нейронов (ФАН). Спектр существующих ФАН достаточно широк, от ступенчатых и линейных функций до сигмоидальных и гиперболических. Выбор конкретной ФАН влечет

за собой, с одной стороны, использование ресурса вычислителя (в нашем случае – кристалла FPGA), а с другой – определяет поведение НСВ, а также его способность к обобщению и выявлению сложных нелинейных закономерностей. Очевидно, что использование ступенчатых или линейных ФАН выгодно с точки зрения минимизации ресурса вычислителя, но, с другой стороны, лишает НСВ нелинейного поведения. Применение различных нелинейных ФАН требует значительного вычислительного ресурса, однако позволяет обучить НСВ на сложных выборках со скрытыми нелинейными зависимостями.

Составим ВФА таким образом, чтобы обеспечить наиболее разнообразную динамику НСВ. С этой целью введем в ВФА следующие ФАН:

$$f(x)=x, \tag{1.3}$$

$$f(x)=\tanh(x)=\frac{2}{1+e^{-2x}}-1, \tag{1.4}$$

$$\begin{cases} f(x)=x, & x\geq 0, \\ f(x)=\alpha x, & x<0, \end{cases} \tag{1.5}$$

$$\sigma_i(x)=\frac{e^{x_i}}{\sum\limits_{i=1}^{n} e^{x_i}}. \tag{1.6}$$

Линейную функцию (1.3) удобно использовать для решения различных задач аппроксимации данных. Гиперболический тангенс (1.4) по сравнению с сигмоидой имеет больший градиент, что положительно сказывается на процессе обучения по методу обратного распространения ошибки. Функция *ReLu* (1.5) менее требовательна к вычислительным ресурсам, чем гиперболический тангенс или сигмоида, так как производит более простые математические операции, обладает свойством разреженности активации, что более эффективно с точки зрения вычислительного ресурса, при этом также обладая нелинейными свойствами (нелинейную динамику имеет и комбинация функций *ReLu*). Заметим, что коэффициент α может принимать значения из диапазона $[0;R_+]$, однако значение 0 порождает известную проблему умирающего *ReLu* (Dying ReLu problem), а большие значения нивелируют ее нелинейные свойства. Таким образом, целесообразно выбирать коэффициент $\alpha \sim 0.01$. Функция *Softmax* (1.6) – это многопеременная логистическая функция (обобщение логистического уравнения Ферхюльста). Данная функция применяется не к отдельному значению, а к вектору (в выходном слое НСВ). *Softmax* удобно использовать для решения задачи классификации, при этом выходное значение *Softmax* можно интерпретировать как вероятность возникновения каждого из классов объектов.

Отметим, что в выбранных нами функциях входным параметром x является взвешенная сумма входных воздействий каждого нейрона, в котором производится расчет ФАН с учетом дополнительного смещения. Для определенности проиндексируем содержание ВФА:

0 – преобразование f_i отключено (выход всегда равен входу и изменяется только извне НСВ);

1 – в качестве ФАН используется линейная функция;

2 – в качестве ФАН применяется гиперболический тангенс;

3 – в качестве ФАН используется функция *ReLu*;

4 – в качестве ФАН применяется функция *Softmax*.

Учитывая тот факт, что реализация нелинейных функций с помощью кристалла FPGA затруднительна, для решения этой проблемы был выбран следующий подход:

— на этапе разработки производится кусочно-линейная аппроксимация ФАН в диапазоне входов $[x_{\min}, x_{\max}]$ с шагом дискретизации h (количество интервалов m), позволяющим обеспечить необходимую точность работы НСВ;

— в результате аппроксимации формируется таблица значений коэффициентов $\{p_i, q_i\}$, $i=\overline{0,m-1}$, определяющих наклон линейной функции $y=p_i x+q_i$ на участке $[x_{\min}+ih, x_{\min}+(i+1)h]$, где y – прямая, аппроксимирующая функцию ФАН на заданном участке области определения переменной x;

— реализация функции предполагает, что по входному значению x производится поиск соответствующего ему участка, т.е. такого индекса i, что $x\in[x_{\min}+ih, x_{\min}+(i+1)h]$. Если x

выходит из диапазона $[x_{\min}, x_{\max}]$, то в качестве рабочего рассматривается ближайший к x участок. Далее рассчитывается $y = p_i x + q_i$.

Следует отметить, что выбранный подход (кусочно-линейная аппроксимация) является компромиссом между использованием просто выборок $f(x_i)$ с малым шагом дискретизации h (что требует большого объема памяти) и расчетом непосредственно нелинейной функции (что требует значительных вычислительных ресурсов). При этом для повышения точности работы НСВ, реализованной на базе кристалла FPGA, более целесообразным является уменьшение шага дискретизации h, нежели использование точного расчета нелинейной ФАН.

Таким образом, алгоритм работы НСВ с аппроксимированной ФАН имеет вид, представленный на рис. 2.

Существует несколько вариантов реализации НСВ, имеющих свои достоинства и недостатки. Их можно разделить на три основных вида.

Параллельная реализация. Расчет динамики (1.2) производится параллельно с помощью n независимых процессов (по числу столбцов МВН). Требует значительных вычислительных ресурсов, но имеет минимальный временной интервал ожидания отклика НСВ.

Последовательная реализация. Расчет динамики (1.2) производится в одном единственном процессе поэлементно (пошагово). Имеет относительно большое время ожидания отклика НСВ, но не требует большого вычислительного ресурса.

Параллельно-последовательная реализация. Часть вычислительных блоков работает параллельно (некоторое количество независимых процессов), а часть – последовательно. Выступает компромиссом между предыдущими подходами и позволяет добиться требуемой скорости реакции НСВ при использовании ограниченного вычислительного ресурса. Таким образом, априорные требования к ресурсам и времени отклика являются определяющими для разработчика реализации НСВ.

Отметим, что с помощью кристалла FPGA возможно осуществить все три подхода к построению НСВ, в то время как применение процессора сводит возможности использования НСВ только в рамках последовательной реализации.

2. Развертывание нейросети в НСВ. Как указывалось ранее, одним из преимуществ представленного НСВ является гибкость процесса развертывания произвольной нейросетевой структуры на его основе. Для наглядности продемонстрируем процесс развертывания простой нейросети прямого распространения.

Пусть нейросеть имеет структуру слоев, приведенную на рис. 3. Для примера будем использовать нейроматрицу, соответствующую архитектуре, в которой содержится 15 нейронов. Тогда МВН M при реализации базовой модели будет иметь вид, представленный в табл. 2.

Как видно из табл. 2, при развертывании нейросети (см. рис. 3) часть МВН остается незадействованной (выходы d_{11}–d_{15} не используются). Вектор смещений в данном примере нулевой (табл. 3), а вектор функций активации рассмотрен в табл. 4. При этом в ВФА будем считать, что:

0 – функция активации не зависит от входов и оставляет состояние нейрона d неизменным;

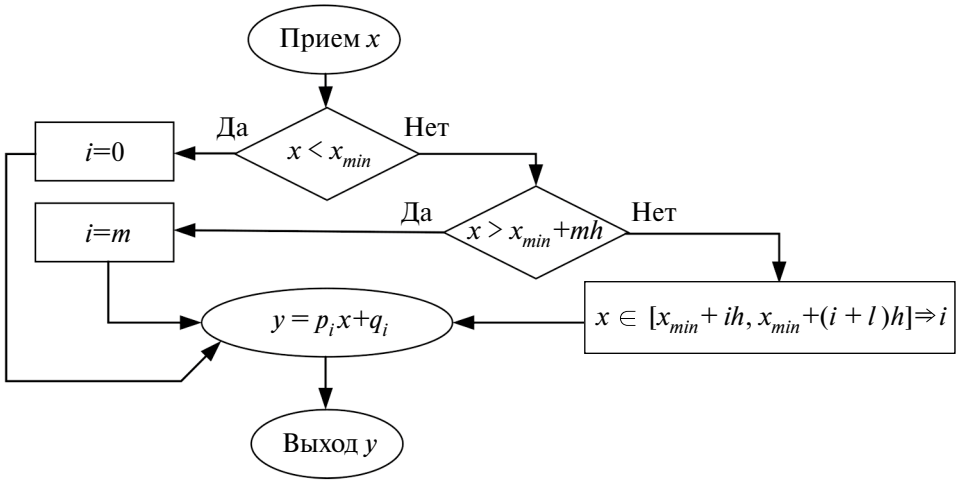

Рис. 2. Алгоритм работы НСВ с аппроксимированной ФАН.

Таблица 2. МВН созданной нейросети

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	d_1	0	0	0	a_{15}	0	0	0	0	0	0	0	0	0	0
2	0	d_2	0	0	0	a_{26}	0	0	0	0	0	0	0	0	0
3	0	0	d_3	0	0	0	a_{37}	0	0	0	0	0	0	0	0
4	0	0	0	d_4	0	0	0	a_{48}	0	0	0	0	0	0	0
5	0	0	0	0	d_5	0	0	0	a_{59}	a_{510}	0	0	0	0	0
6	0	0	0	0	0	d_6	0	0	a_{69}	a_{610}	0	0	0	0	0
7	0	0	0	0	0	0	d_7	0	a_{79}	a_{710}	0	0	0	0	0
8	0	0	0	0	0	0	0	d_8	a_{89}	a_{810}	0	0	0	0	0
9	0	0	0	0	0	0	0	0	d_9	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	d_{10}	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	d_{11}	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	d_{12}	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	d_{13}	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	d_{14}	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d_{15}

Таблица 3. Вектор смещения созданной нейросети

B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Таблица 4. ВФА созданной нейросети

F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0

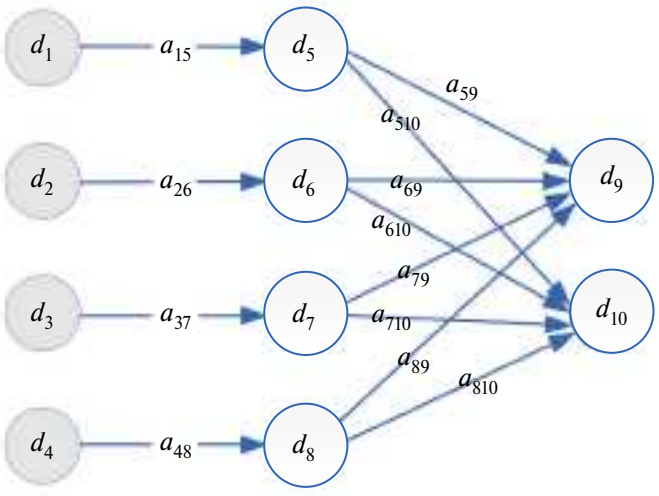

Рис. 3. Архитектура нейросети прямого распространения.

1 – происходит постоянный пересчет выхода по значениям входов и перезапись значения выхода *d*.

При значении $f_j = 1$ работает алгоритм кусочно-линейной аппроксимации нелинейной функции активации. Тогда алгоритм работы нейросети, развернутой на базе НСВ, описывается следующим образом:

1) инициализация матрицы *M* весами *a*;

2) инициализация матрицы *M* входными значениями d_1–d_4;

3) инициализация векторов *B* и *F*;

4) итерационный процесс расчета (здесь $\widehat{M}$ – это матрица *M* с обнуленной диагональю):

$$\operatorname{diag} M = F\left[\left(\operatorname{diag} M\right)\widehat{M} + B\right]; \tag{2.1}$$

5) анализ состояния (значения функции активации) выходов нейронов d_9, d_{10}.

Подобно тому, как в НСВ разворачиваются нейросети прямого распространения, разворачиваются и рекуррентные нейросети. Однако следует отметить, что использование рекуррентных нейросетей целесообразно только в задачах без жестких требований к временному интервалу интеллектуального вывода в силу наличия итерационного процесса поиска решения внутри такого типа нейросетей. Аналогичным образом возможно разворачивать и динамически перезагружать в память НСВ произвольное количество и типов нейросетей.

3. Обучение нейросети на базе НСВ. В силу того, что структура нейросети в НСВ при ее развертывании не изменяется, подходы к процессу обучения остаются стандартными, присущими конкретному типу развертываемой нейросети. Рассмотрим в качестве примера процесс обучения двухслойной нейросети прямого распространения, архитектура которой приведена на рис. 4.

На рис. 4 x – входные воздействия (входной слой нейронов с отключенной функцией активации); y – выходной вектор (слой 2); f_1 – гиперболический тангенс; f_2 – функция *softmax*. Слой 1 является скрытым, а количество его нейронов определяется экспериментально (обычно число, являющееся промежуточным между количеством нейронов входного и выходного слоев). В случае критически малого числа нейронов в скрытом слое, уменьшается эффективность нейросети и повышается уровень ошибочных срабатываний. При слишком большом количестве нейронов в скрытом слое возникает эффект переобучения и нейросеть теряет способность к обобщению.

Математическое описание данной нейросети имеет следующий вид: $X \in \mathbb{R}^{1\times m}$ – вектор входных значений; $W \in \mathbb{R}^{m\times n}$ – матрица весовых коэффициентов скрытого слоя; $U \in \mathbb{R}^{1\times n}$ – вектор воздействий скрытого слоя; $Q \in \mathbb{R}^{1\times n}$ – выходной вектор скрытого слоя; $V \in \mathbb{R}^{n\times p}$ – матрица весовых коэффициентов выходного слоя; $E \in \mathbb{R}^{1\times p}$ – вектор воздействий выходного слоя; $Y \in \mathbb{R}^{1\times p}$ – целевой вектор. Для индексов примем следующие обозначения: входы нумеруются индексом $i = \overline{1,m}$, элементы скрытого слоя – индексом $j = \overline{1,n}$, а выходы – индексом $k = \overline{1,p}$, индекс l – номер итерации работы нейросети.

Прямой проход через сеть осуществляется следующим образом:

$$\begin{cases} U = XW, \\ Q = f_1(U), \\ E = QV, \\ Y = f_2(E). \end{cases} \tag{3.1}$$

Для обучения представленной нейронной сети удобно использовать алгоритм минимизации целевой функции ошибки, которая находится по формуле

$$E(W,V) = \frac{1}{2}\sum_{k=1}^{p}(y_k - d_k)^2, \tag{3.2}$$

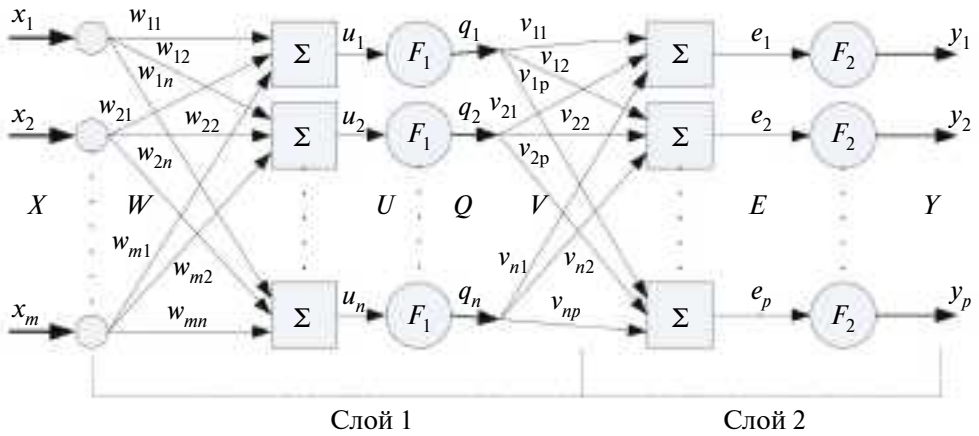

Рис. 4. Двухслойная нейросеть прямого распространения.

где y_k – полученное реальное значение k-го выхода нейросети при подаче на нее одного из входных образов обучающей выборки $\left(X^\alpha, D^\alpha\right)$, $\alpha = \overline{1,L}$; d_k – требуемое (целевое) значение k-го выхода для этого образа. При этом обучающая выборка состоит из L пар векторов X и D, где каждому входному вектору X ставится в соответствие целевой вектор D, содержащий элементы d_k и описывающий требуемую реакцию нейросети на входное воздействие X. Обучение нейросети проводится с помощью процедуры обратного распространения ошибки [23] с применением оптимизационного метода градиентного спуска, обеспечивающего минимизацию $E(W,V) \to \min$. Суть метода заключается в подаче на вход нейросети векторов X из обучающей выборки и подстройки весовых коэффициентов матрицы M (1.1) таким образом, чтобы состояния выходных нейронов НСВ стремились к соответствующим целевым значениям D. Выбор и величина подстройки определяются исходя из условия минимизации целевой функции ошибки (3.2), а процесс обучения является итерационным и завершается при достижении функции $E(W,V)$ заранее определенного минимального значения.

Таким образом, обучение нейросети, развернутой на базе НСВ, возможно производить как до начала процесса развертывания в соответствие с описанием (3.1), (3.2), так и после с учетом расположения нейросетевой структуры в ядре НСВ, произведя необходимый переход от (3.1), (3.2) к (1.1), (1.2).

4. Реализация НСВ на базе кристалла FPGA. Рассмотрим один из подходов, позволяющий реализовать представленную модель ядра НСВ на базе кристалла FPGA фирмы Xilinx. Для оптимизации необходимого вычислительного ресурса и, как следствие, стоимости применяемого кристалла воспользуемся параллельно-последовательной реализацией. В качестве интерфейсов, обеспечивающих информационное сопряжение с ядром НСВ, воспользуемся шинами PCI-Express (для сопряжения с внешними устройствами и вычислителями) и AXI4 (для взаимодействия между основными узлами внутри кристалла FPGA). Исходя из основных параметров данных интерфейсов можно оценить скорость взаимодействия с ядром НСВ как ~5 Гбит/с. В качестве высокоскоростной оперативной памяти для хранения коэффициентов МВН используем блочную память BRAM, встроенную непосредственно в кристалл FPGA. Таким образом, структура проекта кристалла FPGA для развертывания НСВ на ее основе будет иметь вид, представленный на рис. 5.

Как видно из рис. 5, проект построен на базе четырех видов IP-ядер стандартной периферии кристалла FPGA от Xilinx, где XDMA – это пост DMA/Bridge Subsystem for PCIe, AXI Interconnect – маршрутизатор/разветвитель шины AXI4, AXI-Ctrl – блоки контроллеров памяти BRAM, BRAM – блочная оперативная память. Также в состав проекта кристалла FPGA входит блок RTL-Core ANN, написанный на языке HDL Verilog и реализующий непрерывную обработку данных, хранящихся в BRAM в соответствие с логикой работы НСВ (1.2). При этом взаимодействие между блоками XDMA и AXI Interconnect производится посредством высокоскоростной шины AXI4-Bypass, блоки AXI-Ctrl сопряжены с AXI Interconnect через шину AXI4, а обращение к памяти BRAM из AXI-Ctrl осуществляется через специализированный для BRAM интерфейс, представляющий собой параллельную шину данных.

В случае такой реализации НСВ, как представлено на рис. 5, для стороннего вычислительного модуля, сопрягаемого с НСВ посредством интерфейса PCI-Express, все пространство памяти НСВ имеет линейную адресацию и доступно посредством механизма прямого доступа к памяти (DMA). Это позволяет значительно упростить процесс разработки драйверов и программного обеспечения, предназначенных для работы с НСВ, при этом не теряя в скорости интеллектуального вывода на аппаратном уровне.

Оценим временной интервал T_u развертывания произвольной нейросети в НСВ исходя из его структуры (рис. 5). Этот интервал является критически важным для процесса

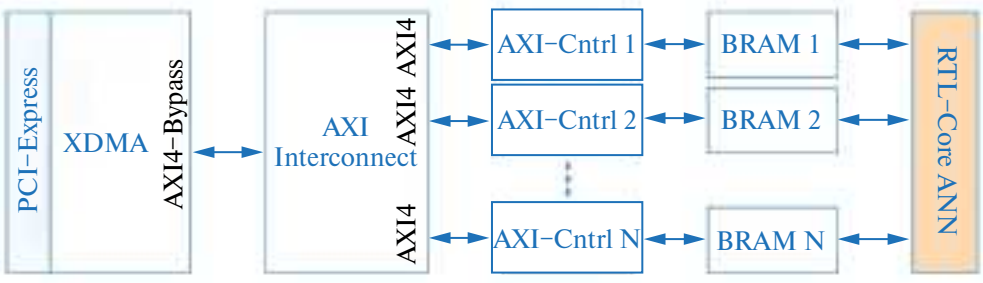

Рис. 5. Структура FPGA-проекта НСВ.

динамической перенастройки НСВ в случае решения интеллектуальных задач, требующих использования нейросетей с принципиально разной архитектурой (последовательное решение разнотипных задач на НСВ). Очевидно, что данный интервал зависит от следующих параметров: F_{PCI} – скорость обмена через внешнюю шину PCI-Express, Б/с, F_{AXI} – скорость обмена через внутреннюю шину AXI4, Б/с, T_{Ctrl} – задержка на обработку в контроллере AXI-Ctrl, с, T_{BRAM} – задержка записи данных в BRAM, с, n – количество нейронов, k – размер ячейки памяти для одного весового коэффициента из МВН, ВФА и вектора смещений B, Б. Таким образом, время развертывания

$$T_u = \left(\frac{1}{F_{\text{PCI}}} + \frac{1}{F_{\text{AXI}}} + T_{\text{Ctrl}} + T_{\text{BRAM}} \right) (2+n) nk. \tag{4.1}$$

Оценим также временной интервал, необходимый для принятия решений T_s НСВ, в случае реализации блока RTL-Core ANN как последовательных вычислений (для минимизации вычислительного ресурса кристалла FPGA). Очевидно, что данный интервал определяется тактовой частотой кристалла FPGA F_{MCLK} и количеством нейронов n:

$$T_s = \frac{kn^2}{F_{\text{MCLK}}}. \tag{4.2}$$

В случае реализации блока RTL-Core ANN в виде параллельных вычислений время принятия решений будет определяться технологией производства кристалла FPGA, которая напрямую влияет на временные задержки распространения сигналов в кристалле.

Рассчитаем данные параметры на примере использования кристалла FPGA XC7K325T-2FFG900C семейства Kintex-7 от фирмы Xilinx. Пусть необходимо развернуть нейросеть из 1000 нейронов (n = 1000), весовые коэффициенты которой имеют размерность 4 Б (k = 4). Ядро XDMA в выбранном кристалле работает со скоростью 5 Гб/с, что соответствует F_{PCI} = 625 МБ/с. Шина AXI4 тактируется частотой 250 МГц, имеет разрядность 128 б, но также имеет служебную информацию и сигналы подтверждения, что примерно соответствует скорости F_{AXI} = 500 МБ/с. Для простоты будем считать, что блоки AXI-Ctrl и BRAM тактируются от шины AXI4 с частотой 250 МГц. Тогда T_{Ctrl} и T_{BRAM} будут примерно равны 4 нс. Можно оценить время развертывания нейросети в НСВ как $T_u \approx 46.5$ мс.

В случае последовательной реализации блока RTL-Core ANN на базе выбранного кристалла FPGA и тактовой частоты F_{MCLK} = 250 МГц время на принятие решения оценивается как $T_s \approx 16$ мс. Для параллельной реализации интервал T_s будет иметь порядок десятков наносекунд.

Отметим основные достоинства и недостатки представленного в настоящей работе НСВ.

Достоинства:

— возможность реализации большого количества различных классов нейросетей (многослойные персептроны, сеть Хопфилда, машина Больцмана и др.) как прямого распространения, так и рекуррентных без изменения структуры FPGA-проекта НСВ;

— возможность динамической смены архитектуры нейросети без изменения архитектуры самого НСВ в силу однородности его структуры (для решения принципиально отличающихся задач на одном НСВ);

— при использовании параллельной реализации НСВ на базе кристалла FPGA время реакции нейросети определяется технологией производства кристалла FPGA и в настоящее время может достигать единиц наносекунд.

Недостатки:

— в случае параллельной реализации НСВ на базе кристалла FPGA необходим значительный вычислительный ресурс (дороговизна используемого кристалла FPGA);

— при реализации классических нейросетей на базе НСВ нейроны входного слоя применяются только как ячейки памяти, а их вычислительный ресурс остается незадействованным.

С помощью аппаратной реализации НСВ, представленной в настоящей работе, невозможно реализовать сверточные нейросети (для этого необходимо усложнение и расширение архитектуры НСВ).

5. Пример использования НСВ. Рассмотрим работу НСВ на примере стандартной задачи классификации объектов. В качестве таких объектов будут выступать изображение чисел от 0 до 9 с разной степенью детализации и разрешением 8 × 8 (64 пиксел). В качестве обучающей выборки используем библиотеку sklearn (Python), содержащую 1797 образцов (изображений), первые 100 из которой приведены на рис. 6.

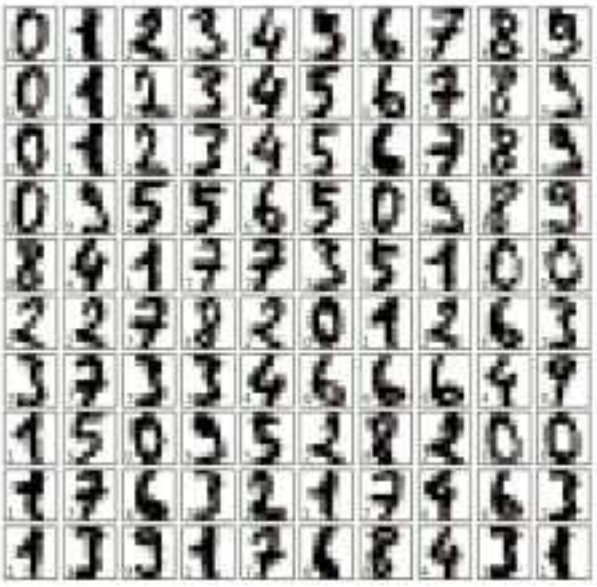

Рис. 6. Первые 100 образцов из базы данных sklearn.

Рассмотрим работу многослойной нейросети прямого распространения, развернутой в НСВ, и решающую задачу классификации объектов. Сравним работу НСВ с эталонной программно реализованной нейросетью. Для определенности будем считать, что аппаратная (в НСВ) и эталонная (программная) нейросети имеют один скрытый слой размером в 30 нейронов и скорость обучения $q = 0.5$ (используется стандартная процедура обучения методом обратного распространения ошибки). С этой целью проведем 10 циклов тестирования, каждый из которых заключается в обучении НСВ и эталона на одинаковых в рамках одного цикла и разных между разными циклами образцах из базы данных sklearn. При этом будем использовать одинаковые в одном цикле и случайные между разными циклами начальные коэффициенты МВН. Проведем 10 эпох обучения в каждом цикле и рассчитаем величину ошибки обучения, усредненную между всеми циклами и зависящую от номера эпохи (рис. 7).

Как видно из рис. 7, процессы обучения НСВ и эталона практически идентичны, ошибка обучения E1 за 10 эпох сходится к одному значению, а максимальное отклонение ошибки E1 для НСВ в сравнении с эталоном составляет 0.31.

Рассмотрим зависимость сходимости процесса обучения для НСВ и эталона в зависимости от скорости обучения q. С этой целью зафиксируем коэффициенты МВН и обучающую выборку, проведем обучение на 10 эпохах для разных скоростей q (рис. 8).

Как видно из рис. 8, сходимость процесса обучения имеет минимум шибки обучения E2 в точке $q = 0.7$ и близка по значениям и форме зависимости между НСВ и эталоном.

6. Пример использования НСВ. Разработанный в настоящей работе подход по построению НСВ может быть применен для широкого класса автономных систем, в том числе интеллектуальных систем авионики и бортовых экспертных систем воздушных судов. Реализация функций интеллектуальной поддержки экипажа воздушных судов прежде всего предполагает идентификацию источников возникновения особых ситуаций, которые создают угрозу безопасности полета. Одним из наиболее значимых источников возникновения особых ситуаций являются отказы или нештатные режимы функционирования бортовых систем.

Перспективным подходом, обеспечивающим парирование отказов, связанных с нештатной работой или отказом функциональных элементов бортовой информационно-вычислительной сети (БИВС), а также идентификации текущей ситуации и формирования рекомендаций экипажу воздушного судна, является использование специализированной экспертной системы поддержки и принятия решений (ЭСППР), разновидностями которой выступают бортовые оперативно-советующие экспертные системы (БОСЭС) [24], экспертные системы реального времени (ЭС РВ) или активные экспертные системы [25]. Указанные системы относят к классу систем, основанных на знаниях (knowledge-based systems), или интеллектуальных систем.

С точки зрения математического описания динамической модели ЭСППР в случае возникновения нештатных ситуаций очевидно, что полноты ее режимов работы возможно достичь в случае, если в качестве базового экспертного материала, на котором строится ЭСППР

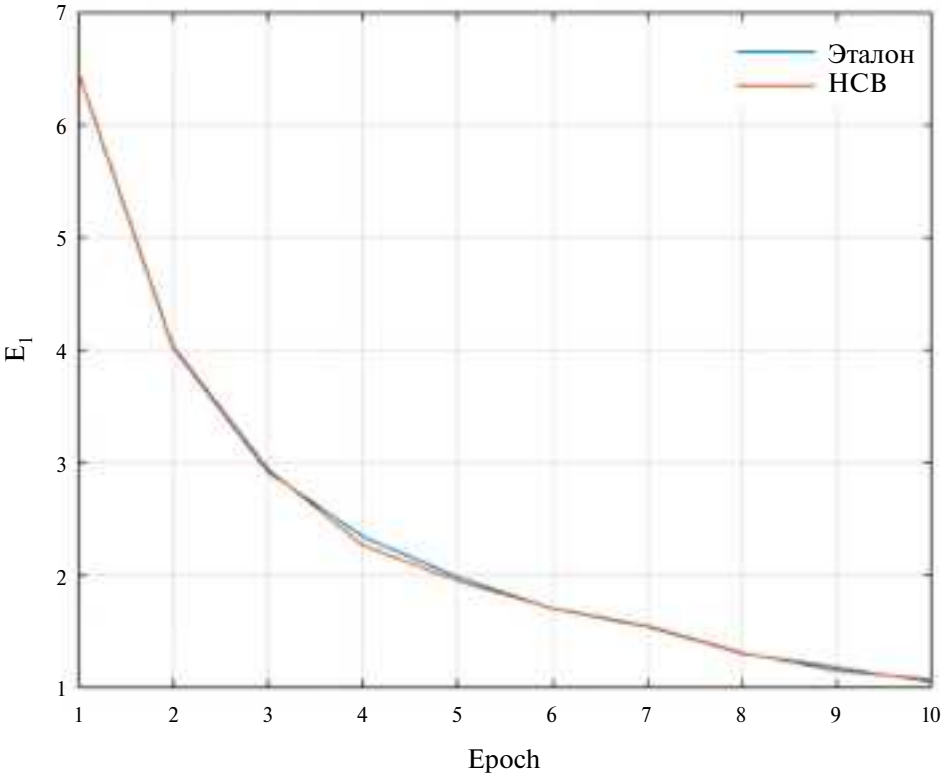

Рис. 7. Процесс обучения НСВ и эталона.

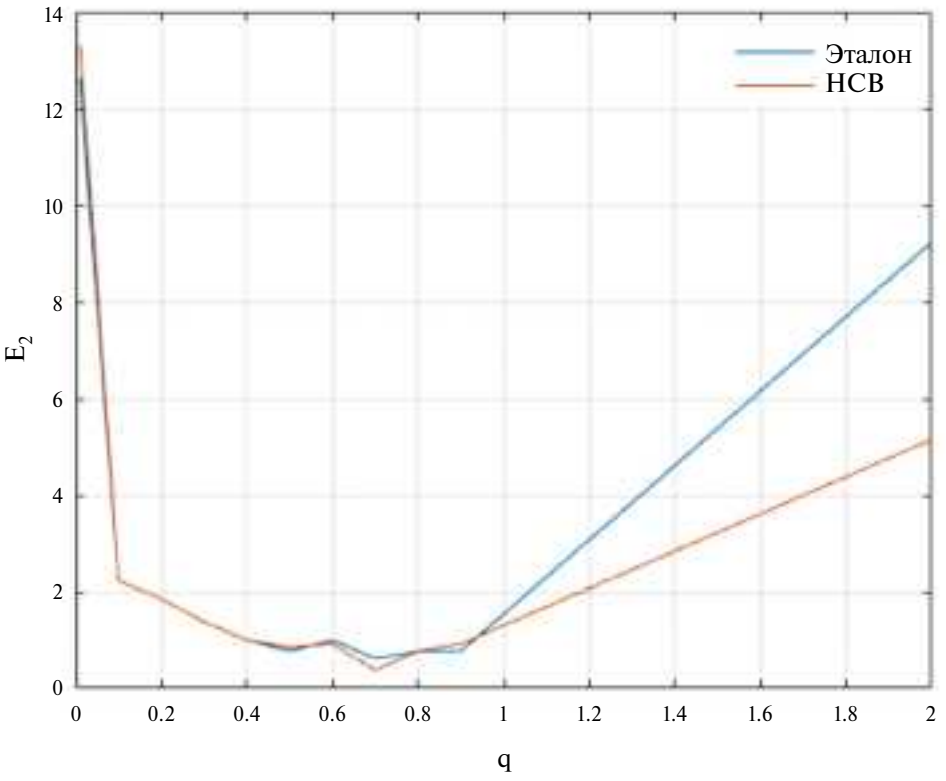

Рис. 8. Сходимость процесса обучения НСВ и эталона в зависимости от q.

использовать как формализуемые, так и неформализуемые данные (паттерны действий, приводящие к парированию возникших отказов). Поэтому для описания формализуемых данных применяем метод логического вывода из теории экспертных систем, а для описания не формализуемых данных – нейросетевой подход.

Метод логического вывода является детерминированным и опирается на пространство состояний, переход между которыми осуществляется посредством взаимодействия со специализированной базой знаний (БЗ). При этом БЗ представляет собой множество правил (паттернов поведения), заранее формулируемых экспертом по знаниям. Недостаток метода логического вывода – его ограниченность и неспособность обобщить знания эксперта таким образом, чтобы появилась возможность синтеза нового знания, отсутствующего непосредственно в БЗ, но вытекающего из ее содержания. Данного недостатка лишены системы, построенные на нейросетевых принципах, которые обеспечивают:

— возможность воспроизведения достаточно сложных нелинейных зависимостей;

— высокое быстродействие (особенно в случае аппаратной реализации с использованием параллельной обработки, например на базе кристалла FPGA);

— универсальность применения;

— возможность решения плохо формализуемых задач (распознавание изображений и речи, кластерный анализ, идентификация, прогноз и т.п.).

Недостатком систем, построенных на базе нейросетевого подхода, является необходимость наличия большой выборки данных, предназначенной для успешного процесса обучения нейросети.

Таким образом, комбинируя и объединяя эти подходы, возможно построить эффективную интеллектуальную систему, способную решать широкий спектр разнородных задач и осуществляющую гибкий анализ текущей ситуации с последующим интеллектуальным выводом рекомендаций экипажу воздушного судна.

Одной из наиболее важных функций ЭСППР выступает парирование отказов методом динамической реконфигурации БИВС [26–29]. Очевидно, что при разработке подсистемы, реализующей данную функцию, особое внимание уделяется минимизации временного интервала принятия решения (поиска решения в БЗ). Применение НСВ с параллельной реализацией на кристалле FPGA позволяет значительно минимизировать данный интервал, а также повысить гибкость и глубину интеллектуального вывода (за счет увеличения информации в БЗ без увеличения времени выборки). Рассмотрим простой пример реализации на НСВ поиска оптимальной конфигурации БИВС в БЗ по входному вектору отказов. Пусть БЗ содержит 14 резервных конфигураций БИВС, каждая из которых имеет 10 элементов, подверженных отказу. Тогда БЗ можно описать с помощью табл. 5.

В представленной БЗ количество строк соответствует набору резервных конфигураций БИВС, количество столбцов – набору функциональных элементов, входящих в каждую конфигурацию. Значение 1 указывает на то, что данный элемент БИВС активен (используется) в данной конфигурации (0 – находится в горячем резерве).

Проанализируем отказы элементов бортового радиоэлектронного оборудования (БРЭО) и найдем оптимальную резервную конфигурацию, позволяющую парировать возникший

Таблица 5. Конфигурации БИВС в БЗ

Конфигурация	***Активные элементы***									
1	0	1	1	1	0	1	0	0	0	0
2	0	1	1	0	0	1	0	1	1	0
3	1	0	1	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	1	1
5	1	0	0	0	0	0	1	1	0	1
6	0	0	1	1	1	0	1	0	0	1
7	0	0	0	0	1	0	1	0	0	1
8	0	0	0	1	0	1	0	1	0	0
9	1	1	1	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	1	0	0
11	1	1	0	0	0	1	0	0	0	0
12	0	1	1	1	0	1	0	0	1	0
13	0	1	0	0	0	1	1	1	0	1
14	0	0	0	0	1	0	0	1	0	1

отказ. При этом будем ориентироваться на вектор отказов вида, приведенного в табл. 6. Здесь значение 1 указывает на отказ соответствующего элемента БИВС (по номеру столбца), а 0 – на его штатную работу.

Развернем в НСВ нейросеть прямого распространения, при этом входной слой будет содержать 10 нейронов (по количеству элементов вектора отказа), а выходной слой – 14 нейронов (по количеству резервных конфигураций БРЭО в БЗ). Создадим также скрытый слой, содержащий 7 нейронов (для повышения эффекта обобщения информации в процессе обучения). Обучим данную нейросеть на следующей логике работы: подходящей конфигурацией является та, при которой парируется максимальное количество отказов, т.е. после динамической реконфигурации остается минимальное количество активных элементов БИВС, находящиеся в состоянии 1 в векторе отказов. Обученную таким образом нейросеть загрузим в НСВ и протестируем ее работу. Рассмотрим реакцию НСВ на следующие векторы отказов:

$$\begin{aligned} FLT_1 &= [1\,1\,1\,0\,0\,0\,0\,0\,0\,0], \\ FLT_2 &= [1\,1\,1\,0\,0\,0\,0\,1\,0\,0], \\ FLT_3 &= [1\,0\,0\,0\,0\,0\,0\,1\,0\,1]. \end{aligned} \tag{6.1}$$

Состояния P выходных нейронов НСВ как реакция на данные векторы отказов приведены на рис. 9–11.

Как видно из представленного эксперимента, НСВ производит выборку из БЗ набора конфигураций, позволяющих парировать возникшие отказы, указанные во входном векторе. При этом процесс поиска будет полностью параллельным, а время, необходимое на реакцию НСВ, не зависит от размера БЗ. Отметим важную особенность описанного выше подхода. Несмотря на тот факт, что динамика нейросети по своей природе слабо детерминирована, результат

Таблица 6. Вектор отказов элементов БИВС

Отказы	0	1	0	0	0	1	1	0	0	0

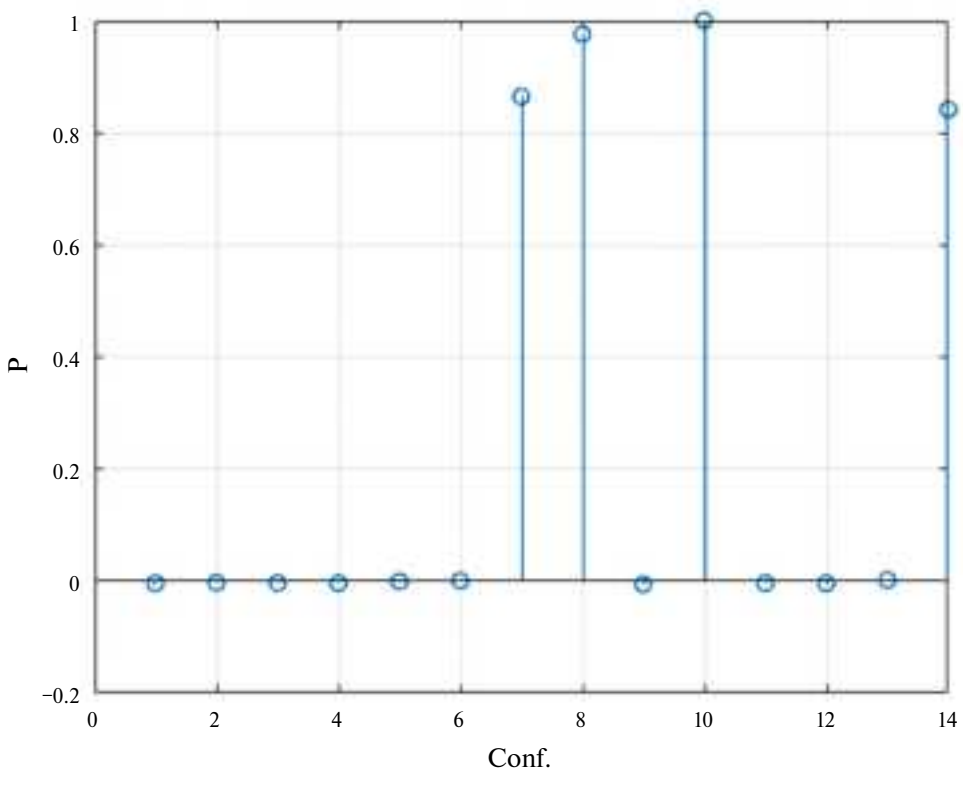

Рис. 9. Реакция НСВ на FLT_1.

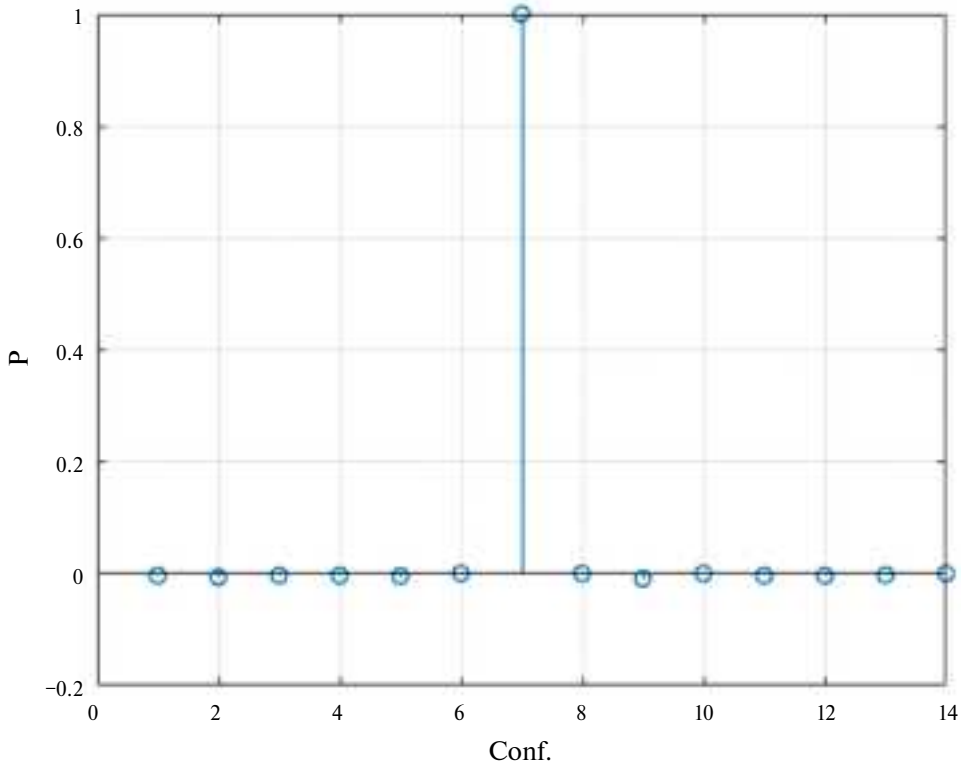

Рис. 10. Реакция НСВ на FLT_2.

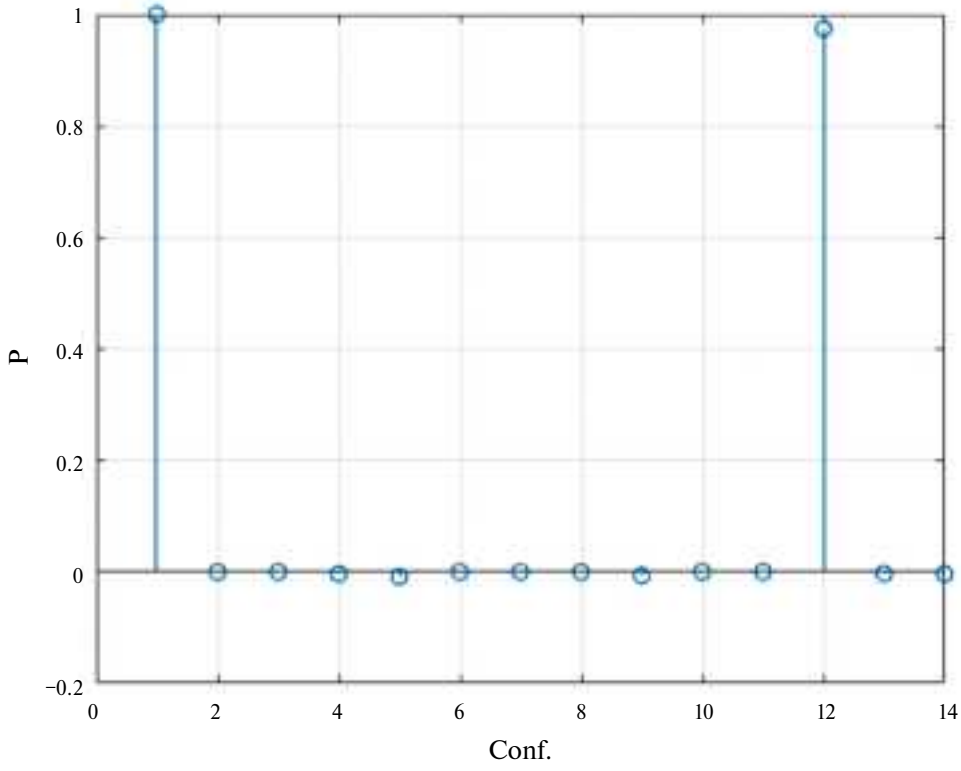

Рис. 11. Реакция НСВ на FLT_3.

интеллектуального вывода при этом строго детерминирован в силу дизайна процесса обучения, опирающегося на заранее определенный набор проверенных конфигураций.

Заключение. Представленный в настоящей работе НСВ является универсальным вычислителем, способным выполнять широкий спектр задач как детерминированных (аппроксимация, поиск оптимального пути, поиск по большому массиву данных), так и недетерминированных, формализация которых осложнена неполнотой информации и нелинейностью внутренних зависимостей (классификация, кластеризация, принятие решений на основе человеческого опыта). При этом применение представленного НСВ как одного из блоков сложной системы (например, бортовой интеллектуальной системы поддержки принятия решений) позволяет увеличить область решаемых ею задач, сделать процесс поиска решений более гибким и унифицированным с точки зрения аппаратной реализации, а также обеспечить необходимую скорость реакции на входные воздействия (например, возникшие нештатные ситуации), т.е. минимизировать время принятия критически важных решений.

СПИСОК ЛИТЕРАТУРЫ

1. *Визильтер Ю.В., Вишняков Б.В., Желтов С.Ю.* Современные технологии искусственного интеллекта и их применение в авиационных комплексах // XVI Всероссийск. мультиконф. по проблемам управления (МКПУ-2023). Волгоград, 2023. С. 13–16.
2. *Аксентов В.А.* Эволюция развития нейронных сетей: прошлое, настоящее, будущее // Вестник науки. 2023. Т. 3. № 8 (65). С. 89–91.
3. *Капустин А., Бунас К.* Искусственный интеллект в авиации // Наука и инновации. 2019. № 9 (199).
4. *Кочнев А.А.* Концептуальные основы практического использования нейронных сетей: проблемы и перспективы // Общество и инновации. 2023. Т. 4. № 1. С. 1–10.
5. *Кочнев А.А.* Применение искусственных нейронных сетей в прогнозировании // Международный научный журнал “Научные горизонты”. 2023. № 1 (65). С. 48.
6. *Дроговоз П.А., Шиболденков В., Вакунов С.* Перспективы применения гибридных нейросетевых систем для создания цифровых двойников производственных процессов // Десятые Чарновские чтения. М., 2021. С. 54–60.
7. *Краковский И.В., Асалханов П.Г.* Текстовые нейронные сети: возможности, применение и перспективы развития // ББК 40 Н 347. 2023. С. 262.
8. *Виноградова Е.П., Нестеренко А.В.* Применение нейронных сетей для решения задач цифровой обработки сигналов // Радиотехнические, оптические и биотехнические системы. Устройства и методы обработки информации. 2023. С. 90–92.
9. *Мишин Д.В., Холопов Д.А.* Разработка подавителя сигналов беспроводной связи на базе нейросетевого модуля // Информационные технологии в науке и образовании. Проблемы и перспективы. 2023. С. 261–263.
10. *Визильтер Ю.В., Горбацевич В.С., Моисеенко А.С.* Однопроходный алгоритм обнаружения и распознавания лиц на основе сверточных нейронных сетей // Вестник компьютерных и информационных технологий. 2021. Т. 18. № 4. С. 11–20.
11. *Деев М.И., Ковалева О.А., Ковалев С.В.* Тестирование и анализ нейросетевых моделей для распознавания объектов на изображении в режиме реального времени // Российская наука, инновации, образование (РОСНИ-О-II-2023). Красноярск, 2023. С. 274–280.
12. *Шатравин В., Шашев Д.В.* Построение нейросетевого классификатора на основе перестраиваемых вычислительных сред // Системы управления, информационные технологии и математическое моделирование. Матер. V Всероссийск. научно-практической конф. с международным участием. Омск, 2023. С. 542.
13. *Бондарчук А.С., Шидловский С.В.* Использование перестраиваемой вычислительной среды для вычисления местоположения регионов интереса на бинарном изображении // Системы управления, информационные технологии и математическое моделирование. Матер. V Всероссийск. научно-практической конф. с международным участием. Омск, 2023. С. 161–165.
14. *Шатравин В., Шашаев Д.В.* Применение вычислительных сред для ускорения рекуррентных нейронных сетей // Цифровая экономика. 2023. № 1 (22). С. 27–35.
15. *Хорошайлова М.В.* Реализации нейронной сети на ПЛИС с использованием аппаратных ресурсов // Вестн. Воронежского гос. технического ун-та. 2021. Т. 17. № 3. С. 127–134.
16. *Бахчевников В.В., Деркачев В.А., Бакуменко А.Н.* Способ использования средств быстрого прототипирования для реализации сверточной нейронной сети на ПЛИС // Изв. Южного федерального ун-та. Технические науки. 2020. № 3 (213). С. 146–156.
17. *Бахтин А.А., Волков А.С., Солодков А.В., Свиридов И.А.* Система распознавания модуляции сигналов на основе нейронной сети с использованием ПЛИС // Тр. МАИ. 2021. № 121. С. 13.
18. *Бахтин В.В.* Математическая модель искусственной нейронной сети для устройств на ПЛИС и микроконтроллерах, ориентированных на туманные вычисления // Вестн. Пермск. национального исследовательского политехнического ун-та. Электротехника, информационные технологии, системы управления. 2021. № 40. С. 109–129.
19. *Носкова Е.С., Захаров И.Е., Шкандыбин Ю.Н., Рыкованов С.Г.* Повышение энергоэффективности нейросетевых вычислений с использованием NVDLA на ПЛИС // Компьютерная оптика. 2022. Т. 46. № 1. С. 160–166.

20. *Некрасов В.М.* Модификация нейронной сети YOLOV3-TINY для выполнения на ПЛИС // Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности. 2023. С. 131–134.
21. *Solovyov, A.M., Selvesyuk, N.I., Kosyanchuk, V.V., Zybin, E.Y.* A Model of a Universal Neural Computer with Hysteresis Dynamics for Avionics Problems // Mathematics 2022. V. 10. P. 2390. https://doi.org/10.3390/math10142390
22. *Соловьев А.М., Семенов М.Е., Косьянчук В.В., Новиков В.М., Сельвесюк Н.И.* Аппаратная реализация вычислительных процедур СППР в задачах авионики // XV Всероссийск. мультиконф. по проблемам управления (МКПУ-2021). Геленджик, 2021.
23. *Rumelhart, D., Hinton, G., Williams, R.* Learning Representations by Back-propagating Errors // Nature. 1986. V. 323. P. 533–536. https://doi.org/10.1038/323533a0
24. *Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е.* Интеллектуальное управление динамическими системами. М.: Физматлит, 2000. 352 с.
25. *Лохин В.М., Захаров В.Н.* Интеллектуальные системы управления: понятия, определения, принципы построения // Интеллектуальные системы автоматического управления / Под ред. И.М. Макарова, В.М. Лохина. М.: Физматлит, 2001. 576 с.
26. *Соловьев А.М., Семенов М.Е., Сельвесюк Н.И., Зыбин Е.Ю., Новиков В.М., Солоделов Ю.А.* Развертывание интеллектуальной системы динамической реконфигурации КБО с оптической коммуникационной средой на платформе JetOS // Всероссийский форум с международным участием “Академические Жуковские чтения”. Воронеж: ВУНЦ ВВС “ВВА”, 2021.
27. *Соловьев А.М., Семенов М.Е., Сельвесюк Н.И., Новиков В.М., Карпов Е.А.* Динамическая реконфигурация бортовой распределенной информационно-вычислительной среды на базе ОСРВ JetOS // Международная научно-практическая конф. им. Э.К. Алгазинова “Информатика: проблемы, методы, технологии” (ІРМТ). Воронеж: ВГУ, 2022.
28. *Solovyov A.M., Semenov M.E., Sel’vesyuk N.I., Zybin E.Yu., Novikov V.M.* Control of Dynamic Reconfiguration of the Fully Optical On-board Network of the Aircraft // Intern. Conf. for Information Systems and Design (ICID-2021). Геленджик, 2021.
29. *Kosyanchuk V., Selvesyuk N., Zybin E., Novikov V., Olenev V., Solovyov A., Semyonov M.* Application of a Deterministic Optical Network Model for the Implementation of an Expert System Knowledge Base for Information Transmission Failure Management // Engineering Proceedings. 2023. V. 33 (1). P. 51. https://doi.org/10.3390/engproc2023033051