

Методы и приложения искусственного интеллекта

УДК 004.032.26

DOI:10.25729/ESI.2024.33.1.001

Быстрые преобразования и самоподобные нейронные сети глубокого обучения. Часть 2. Методы обучения быстрых нейронных сетей

Дорогов Александр Юрьевич

ПАО «Информационные телекоммуникационные технологии» («Интелтех»),

Санкт-Петербургский государственный электротехнический университет,

Россия, Санкт-Петербург, vaksa2006@yandex.ru

Аннотация. В работе отмечено, что быстрые нейронные сети (БНС) автомодельны алгоритму быстрого преобразования Фурье (БПФ). Представлен метод построения матричной формы алгоритма быстрого преобразования. Доказана факторизуемость элементов матрицы БНС по элементам нейронных ядер. Предложен метод мультипликативной факторизации произвольных одномерных образов. Показано, что благодаря своей структуре быстрые нейронные сети обладают особыми алгоритмами обучения, которые принципиально отличаются от классического ErrorBackPropagation отсутствием механизма обратного распространения ошибки. Рассмотренные алгоритмы обучения БНС основаны на предложенных в работе методах мультипликативной факторизации образов и быстрых преобразований. Показаны примеры настройки сети к ортогональному базису Адамара, базису Фурье, а также БНС-реализации квазифракталов Кантора и Серпинского. Описан метод настройки быстрых преобразований на эталонную функцию, основанный на методе фрактальной фильтрации сигналов. Предложен метод настройки ортогональных приспособленных преобразований. Приведены примеры. Данная работа является второй частью обобщающей статьи «Быстрые преобразования и самоподобные нейронные сети глубокого обучения». В первой части рассмотрены стратифицированные модели самоподобных нейронных сетей.

Ключевые слова: быстрые преобразования, быстрые нейронные сети, топологические матрицы, ортогональность, приспособленные преобразования

Цитирование: Дорогов А.Ю. Быстрые преобразования и самоподобные нейронные сети глубокого обучения. Часть 2. Методы обучения быстрых нейронных сетей / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении. – 2024. – № 1(33). – С. 5-19. – DOI:10.25729/ESI.2024.33.1.001.

Введение. Традиционно для обработки сигналов и изображений используются ортогональные преобразования, обладающие быстрыми алгоритмами выполнения (Фурье, Уолша-Адамара, Хаара и др. [1,2]). Как было отмечено в Части 1 настоящей статьи [3], быстрые алгоритмы относятся к классу регулярных самоподобных сетей и могут быть представлены четырёхуровневой стратифицированной моделью. Цель обработки обычно заключается в фильтрации или сжатии образов. В обоих случаях необходимы некоторые априорные знания, которые касаются либо вида образа, либо помехи. В зависимости от этой информации выбирается тип используемого преобразования. Для задач классификации априорная информация задаётся в виде матриц рассеивания точек классов, в этом случае оптимальным является ортогональное преобразование Карунена-Лоэва [4], образованное собственными векторами матриц рассеивания. К сожалению, это преобразование не имеет быстрого алгоритма, и поэтому не используется при больших размерностях данных.

Принцип оптимальности можно распространить и на быстрые перестраиваемые преобразования, поставив перед собой цель – настроить значения коэффициентов базовых операций таким образом, чтобы учесть априорную информацию и сохранить при этом условие ортогональности. Быстрые преобразования имеют меньшее число степеней свободы, чем оптимальное преобразование Карунена-Лоэва, поэтому возможности обучения будут ограничены. Тем

не менее, их достаточно для того, чтобы настроиться, по крайней мере, на одну главную компоненту преобразования Карунена-Лозва, и уже это позволяет решать целый класс задач, связанных с эффективным сжатием, адаптивной фильтрацией и распознаванием образов. Условие ортогональности не является определяющим, во многих случаях задача классификации образов решается обученными нейронными сетями. БНС покрывают оба случая: с одной стороны, они позволяют реализовывать алгоритмы быстрых спектральных преобразований, а с другой, нейронные сети, обученные к эталонным образам.

В Части 1 статьи была представлена стратифицированная модель модульных самоподобных нейронных сетей, где модулями являются нейронные ядра. На параметрическом уровне нейронные ядра можно задать линейными матрицами небольшой размерности. Переход от одного преобразования к другому происходит за счёт изменения параметров нейронных ядер. В этом случае модульная самоподобная нейронная сеть является представлением линейного перестраиваемого преобразования, обладающего быстрым алгоритмом. Исторически процедура обучения перестраиваемого преобразования называется настройкой на заданную систему функций. В статье будет показано, что предложенный метод настройки покрывает известные виды алгоритмов быстрых спектральных преобразований и может быть также использован для порождения фрактальных последовательностей. Метод основан на теореме о мультипликативной факторизации элементов матрицы быстрых преобразований, и матричной факторизации с которой мы и начнём.

1. Матричная форма быстрого алгоритма. В графе быстрого алгоритма модулями являются базовые операции (нейронные ядра), представимые в виде матриц небольшой размерности. Для алгоритма быстрого преобразования в базовой операции z^m слоя m выполняется линейная обработка компонентов входного вектора слоя:

$$y_{z^m}^m(v_m) = \sum_{u_m} x_{z^m}^m(u_m) w_{z^m}^m(u_m, v_m), \quad (1)$$

где $x_{z^m}^m$ и $y_{z^m}^m$ – входной и выходной векторы базовой операции; $w_{z^m}^m$ – матрица весов базовой операции. Для построения алгоритма преобразования необходимо перейти от локальных переменных ядер к глобальным переменным слоя. Этот переход реализуется на основе топологической модели. Параметрическое описание базовых операций вместе с топологической моделью образует параметрическую модель быстрого преобразования. Выберем для определённости топологию Кули-Тьюки «с прореживанием по времени». Топологическая модель этого типа имеет вид (см. Часть 1):

$$\begin{aligned} U^m &= \langle u_{n-1} u_{n-2} \cdots u_{m+1} u_m v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle, \\ V^m &= \langle u_{n-1} u_{n-2} \cdots u_{m+1} v_m v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle, \\ z^m &= \langle u_{n-1} u_{n-2} \cdots u_{m+1} v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle. \end{aligned} \quad (2)$$

Для конкретной сети топологическая модель может быть представлена в виде графа, где к вертикальным контактным площадкам подсоединены нейронные ядра. На рис. 1 показан пример графа топологической реализации для сети с характеристиками $P = [p_0 p_1 p_2] = [3, 2, 2]$ и $G = [g_0 g_1 g_2] = [2, 2, 2]$, здесь числа p_m определяют размерность рецепторных полей нейронного ядра в слое с номером m , а числа g_m – размерности аксоновых полей.

В описаниях алгоритмов быстрых спектральных преобразований вместо графа алгоритма часто используется матричная форма, когда алгоритм представляется в виде произведения слабозаполненных матриц, где каждая матрица соответствует одному слою преобразования:

$$H = H_0 H_1 \cdots H_{n-1}.$$

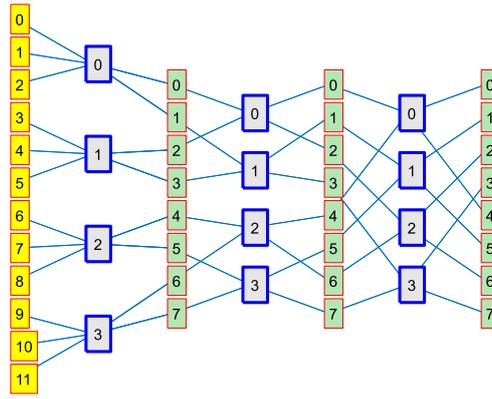


Рис. 1. Граф топологической модели с выделенными базовыми операциями

Такую форму называют *факторизованным представлением* быстрого алгоритма. Структура и параметры слабозаполненных матриц при этом определяются соответствующей теоремой факторизации спектрального преобразования. Таких теорем придумано великое множество, БПФ с топологиями Кули-Тьюки являются типичным примером. Для каждого нового алгоритма быстрого преобразования до недавнего времени доказывали собственную теорему факторизации [1, 2, 5] (другие ссылки приведены в Части 1 статьи). Покажем, что структура слабозаполненных матриц непосредственно следует из топологической модели, и поэтому многообразие топологических моделей покрывает все возможные реализации быстрых алгоритмов, что исключает необходимость какого-либо доказательства новых теорем факторизации.

Для матричного представления алгоритма введём топологические матрицы T_m , состоящие из нулей и единиц, обозначив их элементы $T_m(U^m, V^m)$, где U^m – номер строки матрицы, а V^m – номер столбца. Топологическая матрица по структуре подобна слабо заполненной матрице H_m и отличается от неё только тем, что в позициях ненулевых элементов размещаются единицы. Для каждой топологической матрицы можно задать свои правила нумерации строк и столбцов, для регулярных преобразований удобно использовать поразрядные представления вида:

$$U^m = \langle U_{n-1}^m U_{n-2}^m \dots U_0^m \rangle, \quad V^m = \langle V_{n-1}^m V_{n-2}^m \dots V_0^m \rangle,$$

которые назовём *внешними* в отличие от *внутренних*, заданных топологической моделью (2).

Соответствие между внешними разрядными переменными и внутренними можно задать таблицей, например, для рассматриваемой топологической модели таблица будет иметь следующий вид:

Таблица 1. Соответствия между внешними и внутренними разрядными переменными

$U^m =$	u_{n-1}	u_{n-2}	...	u_{m+1}	u_m	v_{m-1}	v_{m-2}	...	v_1	v_0
$U^m =$	U_{n-1}^m	U_{n-2}^m	...	U_{m+1}^m	U_m^m	U_{m-1}^m	U_{m-2}^m	...	U_1^m	U_0^m
$V^m =$	u_{n-1}	u_{n-2}	...	u_{m+1}	v_m	v_{m-1}	v_{m-2}	...	v_1	v_0
$V^m =$	V_{n-1}^m	V_{n-2}^m	...	V_{m+1}^m	V_m^m	V_{m-1}^m	V_{m-2}^m	...	V_1^m	V_0^m

Единицам топологической матрицы T_m слоя m в слабозаполненной матрице соответствуют элементы ядер (базовых операций) с номерами:

$$z^m = \langle u_{n-1} u_{n-2} \dots u_{m+1} v_{m-1} v_{m-2} \dots v_1 v_0 \rangle.$$

$$h(U, V) = \frac{\partial y^{n-1}(V^{n-1})}{\partial x^0(U^0)}. \quad (3)$$

Дифференцируя (3) по правилу дифференцирования сложной функции получим:

$$h(U, V) = \frac{\partial y^{n-1}(V^{n-1})}{\partial x^0(U^0)} = \frac{\partial y^{n-1}(V^{n-1})}{\partial x^{n-1}(U^{n-1})} \frac{\partial x^{n-1}(U^{n-1})}{\partial y^{n-2}(V^{n-2})} \dots \frac{\partial x^1(U^1)}{\partial y^0(V^0)} \frac{\partial y^0(V^0)}{\partial x^0(U^0)}. \quad (4)$$

Поскольку для межслойного перехода выполнено $x^{m+1}(U^{m+1}) = y^m(V^m)$, то для всех m

$$\frac{\partial x^{m+1}(U^{m+1})}{\partial y^m(V^m)} = 1,$$

а из выражения (1) следует:

$$\frac{\partial y^m(V^m)}{\partial x^m(U^m)} = w_{z^m}^m(u_m, v_m).$$

Подставляя значения частных производных в (4), получим:

$$h(U, V) = w_{z^{n-1}}^{n-1}(u_{n-1}, v_{n-1}) w_{z^{n-2}}^{n-2}(u_{n-2}, v_{n-2}) \dots w_{z^0}^0(u_0, v_0). \quad (5)$$

Таким образом, элементы матрицы быстрого преобразования представимы в виде произведения элементов ядер. Следует отметить, что ещё в 1958 г. в работе Гуда [5] впервые было показано, что элементы матрицы быстрого преобразования Фурье можно представить в виде произведения элементов матриц базовых операций, которые также являются преобразованиями Фурье, но малых размерностей.

3. Настройка на базис Адамара. Функции базиса Адамара задаются на интервале длиной $N = 2^n$ следующим выражением:

$$\text{had}(U, V) = \prod_{m=0}^{n-1} (-1)^{u_m v_m},$$

где $U = \langle u_{n-1} u_{n-2} \dots u_0 \rangle$ и $V = \langle v_{n-1} v_{n-2} \dots v_0 \rangle$. Все разрядные числа принимают значения $\{0, 1\}$. Сравнивая (5) с определением функций Адамара, непосредственно получим правило настройки ядер:

$$w_{z^m}^m(u_m, v_m) = (-1)^{u_m v_m}.$$

Этому выражению соответствует матрица ядра $W = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, очевидно, что все ядра преобразования одинаковы. Выберем для определённости топологию Кули-Тьюки «с прореживанием по времени»:

$$\begin{aligned} U^m &= \langle u_{n-1} u_{n-2} \dots u_{m+1} u_m v_{m-1} v_{m-2} \dots v_1 v_0 \rangle, \\ V^m &= \langle u_{n-1} u_{n-2} \dots u_{m+1} v_m v_{m-1} v_{m-2} \dots v_1 v_0 \rangle, \\ z^m &= \langle u_{n-1} u_{n-2} \dots u_{m+1} v_{m-1} v_{m-2} \dots v_1 v_0 \rangle. \end{aligned}$$

На рис. 3 показано матричное представление быстрого алгоритма преобразования Адамара для размерности $N = 2^3$.

	V_2^m	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1								
	V_1^m	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1								
	V_0^m	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1								
U_2^m	U_1^m	U_0^m																															
0	0	0	1	1																													
0	0	1	1	-1																													
0	1	0		1	1																												
0	1	1		1	-1																												
1	0	0			1	1																											
1	0	1			1	-1																											
1	1	0				1	1																										
1	1	1				1	-1																										

Рис. 3. Матричное представление быстрого преобразования Адамара

Перемножив матрицы факторизованного представления, получим следующую матрицу базисных функций:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

4. **Аппроксимация фракталов.** Рассмотрим быстрое преобразование размерности $N = 3^2$ и выберем все ядра преобразования в виде:

$$W = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Факторизованное представление алгоритма в той же топологии будет иметь вид (рис. 4):

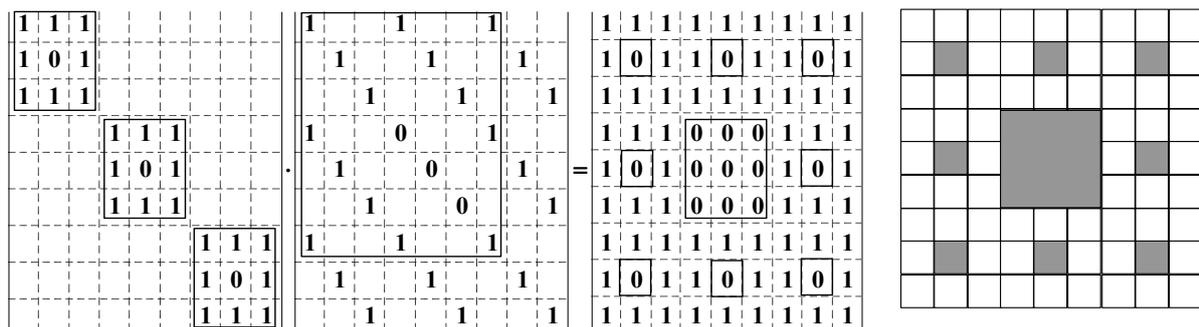


Рис. 4. Аппроксимация фрактала Кантора

Нетрудно видеть, что результирующая матрица представляет собой итерацию фрактала Кантора (см. рис. 4). Рассмотрим быстрое преобразование размерности $N = 2^3$ и выберем все ядра преобразования в виде матрицы:

$$W = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Факторизованное представление быстрого алгоритма показано на рис. 5. В этом варианте параметрической настройки результирующая матрица представляет собой итерацию фрактала Серпинского.

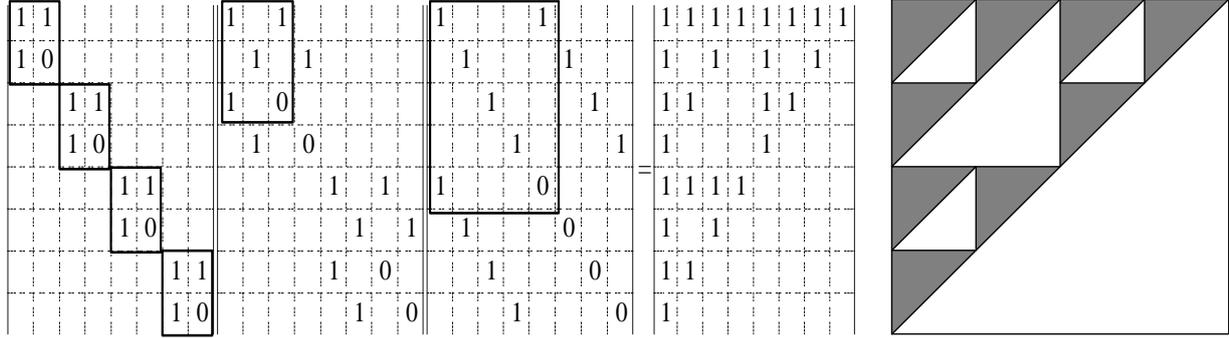


Рис. 5 Аппроксимация фрактала Серпинского

Нарращивание размерности преобразования позволяет получить другие итерации фракталов. Эти примеры подтверждают фрактальные свойства быстрых алгоритмов.

5. Настройка на базис Фурье. По определению, функции базиса Фурье задаются выражением:

$$F_N(U, V) = \frac{1}{\sqrt{N}} \exp\left(-j \frac{2\pi}{N} UV\right), \quad (6)$$

где U – временной отсчёт; V – частота (или номер) базисной функции; $j = \sqrt{-1}$; $N = p_0 p_1 \dots p_{n-1}$ – размерность преобразования (для быстрых преобразований размерность всегда является составным числом). Выполним настройку быстрого алгоритма для базиса Фурье с естественным упорядочением функций по частотам следования. Для построения алгоритма воспользуемся топологической схемой:

$$\begin{aligned} U^m &= \langle u_m u_{m+1} \dots u_{n-2} u_{n-1} v_{m-1} v_{m-2} \dots v_1 v_0 \rangle, \\ V^m &= \langle u_{m+1} u_{m+2} \dots u_{n-1} v_m v_{m-1} v_{m-2} \dots v_1 v_0 \rangle, \\ z^m &= \langle u_{m+1} \dots u_{n-2} u_{n-1} v_{m-1} v_{m-2} \dots v_1 v_0 \rangle. \end{aligned}$$

Откуда для входного слоя следует:

$$U = U^0 = \langle u_0 u_1 \dots u_{n-2} u_{n-1} \rangle = \sum_{m=0}^{n-1} u_m p_{m+1} p_{m+2} \dots p_{n-1}.$$

Подставив последнее выражение в (6), получим:

$$F(U, V) = \prod_{m=0}^{n-1} \frac{1}{\sqrt{p_m}} \exp\left(-j \frac{2\pi V}{p_{n-1} p_{n-2} \dots p_m} u_m\right). \quad (7)$$

Для выходного слоя из топологической схемы следует:

$$V = V^{n-1} = \langle v_{n-1} v_{n-2} v_{n-3} \dots v_1 v_0 \rangle.$$

Для любого m последнее выражение можно записать в виде:

$$V = \langle v_{n-1} v_{n-2} v_{n-3} \dots v_1 v_0 \rangle = \langle v_{n-1} v_{n-2} \dots v_{m+1} \rangle p_m p_{m-1} \dots p_0 + v_m p_{m-1} p_{m-2} \dots p_0 + \langle v_{m-1} v_{m-2} \dots v_0 \rangle.$$

Подставляя в (7) и учитывая периодичность комплексной экспоненты по периоду 2π после преобразований, получим:

$$F(U, V) = \prod_{m=0}^{n-1} \frac{1}{\sqrt{p_m}} \exp\left(-j 2\pi \left[\frac{u_m v_m}{p_m} + u_m \frac{\langle v_{m-1} v_{m-2} \dots v_0 \rangle}{p_0 p_1 \dots p_m} \right]\right).$$

Каждый сомножитель в этом произведении соответствует элементу базовой операции. Таким образом, базовые операции алгоритма БПФ определяются выражением:

$$w_{z^m}^m(u_m, v_m) = \frac{1}{\sqrt{p_m}} \exp\left(-j \frac{2\pi}{p_m} u_m v_m\right) \exp\left(-j 2\pi u_m \frac{\langle v_{m-1} v_{m-2} \dots v_0 \rangle}{p_0 p_1 \dots p_m}\right).$$

Кроме разрядных переменных $v_{m-1} v_{m-2} \dots v_0$, выражение для номера ядра z^m содержит переменные $u_{m+1} \dots u_{n-2} u_{n-1}$, это означает, что базовые операции в пределах слоя будут повторяться.

На рис. 6 показано факторизованное представление алгоритма БПФ для размерности $N = 8$.

v_0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
u_0	u_1	u_2																						
0	0	0	1	1					1		1						1				1			
0	0	1			1	1				1		1						1				1		
0	1	0					1	1					1	1					1				1	
0	1	1						1	1						1	1				1				1
1	0	0	1	-1					1		-1						1				-1			
1	0	1			1	-1				ω^2		$-\omega^2$					ω				$-\omega$			
1	1	0					1	-1					1		-1			ω^2				$-\omega^2$		
1	1	1						1	-1					ω^2	$-\omega^2$				ω^2				$-\omega^3$	

Рис. 6 Факторизованное представление алгоритма БПФ

На рисунке множители $1/\sqrt{2}$ с целью упрощения не показаны, использовано также обозначение $\omega = \exp\left(-j \frac{2\pi}{N}\right)$. На рис. 7 приведены вещественные и мнимые компоненты функций базиса Фурье, полученные перемножением матриц факторизованного представления.

Следует подчеркнуть, что для данной топологической схемы, в отличие от классической схемы Кули-Тьюки, выходные спектральные коэффициенты упорядочены по возрастанию частот.

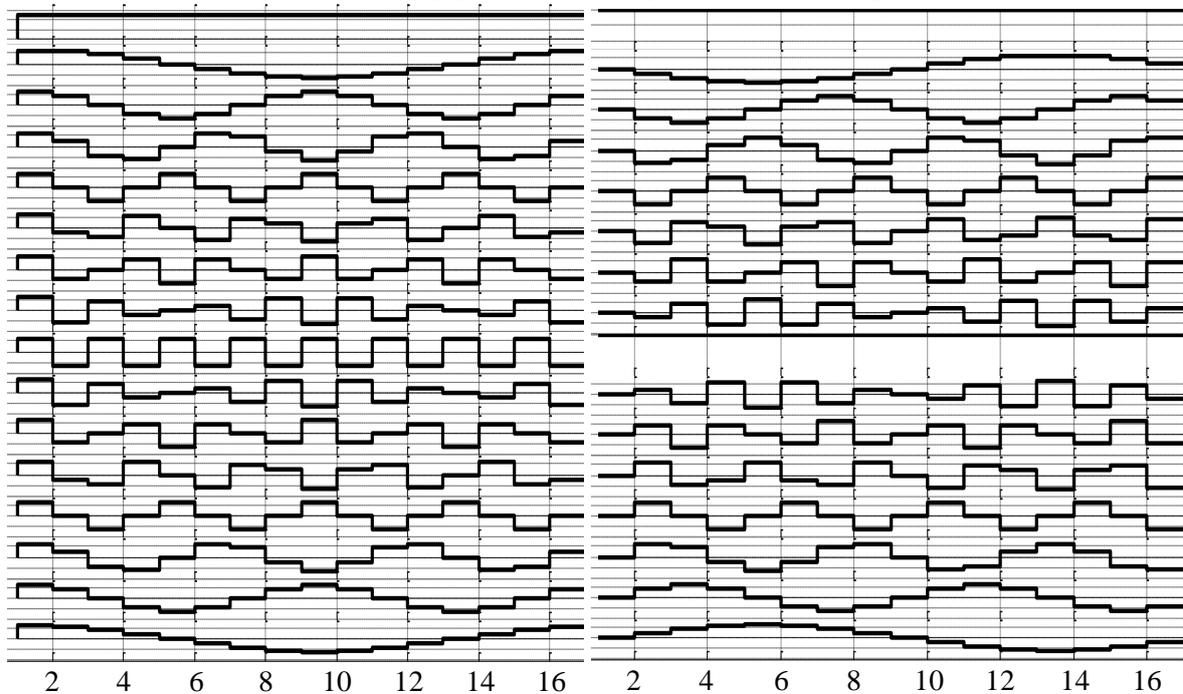


Рис. 7. Функции базиса Фурье с естественным упорядочением по частотам следования

6. Мультипликативное представление сигнальных функций. Будем полагать, что сигнал задан функцией $f(u)$ на дискретном интервале длиной $N = p_0 p_1 \dots p_{n-1}$, где p_m – произвольные целые числа. Представим аргумент функции в позиционной многоосновной системе счисления с основаниями p_0, p_1, \dots, p_{n-1} . Формула перехода, как известно, имеет вид:

$$u = \langle u_{n-1} u_{n-2} \dots u_0 \rangle = u_{n-1} p_{n-2} p_{n-3} \dots p_0 + u_{n-2} p_{n-3} p_{n-4} \dots p_0 + \dots + u_1 p_0 + u_0,$$

где $u_i \in [0, 1, \dots, p_i - 1]$ – разрядные переменные. В результате данного преобразования сигнал представляется, как многомерная функция $f \langle u_{n-1} u_{n-2} \dots u_0 \rangle$. Каждый аргумент функции определяет некоторый масштабный срез сигнала. Зафиксируем все аргументы функции, кроме u_m . Варьируя свободный аргумент u_m , получим выборку S_m (с числом элементов p_m). Фрактальным фильтром [6] частотной локализации m называется произвольный функционал $F(S_m)$, определённый на выборке S_m .

Операцию фрактальной фильтрации можно записать в виде:

$$f_{out} \langle u_{n-1} u_{n-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle = F_{u_m} (f_{inp} \langle u_{n-1} u_{n-2} \dots u_0 \rangle).$$

В простейшем варианте фрактальный фильтр может, например, выполнять суммирование значений функции по аргументу u_m . Если $m = 0$, то такой фильтр генерализует сигнал, сглаживая мелкие детали. Фрактальная фильтрация по аргументу u_m приводит к сокращению интервала определения сигнала в p_m раз. Рассмотрим цепочку фрактальных фильтров, показанную на рис. 8.

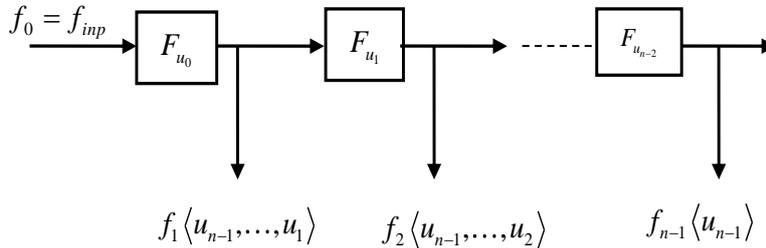


Рис. 8. Цепочка фрактальных фильтров

Выходные сигналы для фильтров цепочки определяются рекуррентным соотношением:

$$f_m \langle u_{n-1} \dots u_m \rangle = F_{u_{m-1}} (f_{m-1} \langle u_{n-1} \dots u_m u_{m-1} \rangle).$$

Введём функции-сомножители:

$$\phi_{i^m} (u_m) = \frac{f_m \langle u_{n-1} u_{n-2} \dots u_m \rangle}{f_{m+1} \langle u_{n-1} u_{n-2} \dots u_{m+1} \rangle}, \quad m = 0, 1, \dots, n-2, \quad (8)$$

где $i^m = \langle u_{n-1} u_{n-2} \dots u_{m+1} \rangle$. Используя данное определение функций, можно записать

$$f_m \langle u_{n-1} u_{n-2} \dots u_m \rangle = \phi_{i^m} (u_m) f_{m+1} \langle u_{n-1} u_{n-2} \dots u_{m+1} \rangle, \quad m = 0, 1, \dots, n-2.$$

Из рекуррентных соотношений (8) непосредственно следует:

$$f(u) = f_0 \langle u_{n-1} u_{n-2} \dots u_0 \rangle = \phi_{i^0} (u_0) \phi_{i^1} (u_1) \dots \phi_{i^{n-2}} (u_{n-2}) \phi_{i^{n-1}} (u_{n-1}). \quad (9)$$

Полученное выражение позволяет утверждать, что любая функция, заданная на дискретном интервале, может быть представлена в мультипликативной форме в виде конечного произведения функций-сомножителей, равного числу множителей в разложении целочисленной длины интервала в произведение целых чисел.

Замечание 1. Как будет показано в Части 3 настоящей статьи, для элементов памяти декомпозицию функций необходимо выполнить по разрядным переменным v_i , начиная со старшего разряда v_{n-1} . В этом случае получим разложение:

$$f(v) = f_0 \langle v_{n-1} v_{n-2} \dots v_0 \rangle = \varphi_{i^{n-1}}(v_{n-1}) \varphi_{i^{n-2}}(v_{n-2}) \dots \varphi_{i^1}(v_1) \varphi_{i^0}(v_0),$$

где $i^m = \langle v_{m-1} v_{m-2} \dots v_1 v_0 \rangle$, $\varphi_{i^{n-m}}(v_{n-m}) = \frac{f_m \langle v_{n-m} v_{n-m-1} \dots v_0 \rangle}{f_{m+1} \langle v_{n-m-1} v_{n-m-2} \dots v_0 \rangle}$, $m = 1, 2, \dots, n$,

Замечание 2. При вычислении функций-сомножителей $\varphi(\)$ в выражении (8), возможно появление нулевого значения в знаменателе, проблема решается за счёт возможности неоднозначного представления произведения сомножителей. Детали процедуры цензурирования нулей рассмотрены в работе [7].

7. Алгоритм обучения быстрых приспособленных преобразований. Перестраиваемые спектральные преобразования, настраиваемые к одиночной функции, получили названия приспособленных [2]. Это название мы будем в дальнейшем использовать к любым перестраиваемым преобразованиям, не обязательно спектральным. Выберем для определённости нейронную сеть, заданную топологией Кули-Тьюки «с прореживанием по времени». Топологическая схема данного типа описывается следующей моделью:

$$\begin{aligned} U^m &= \langle u_{n-1} u_{n-2} \dots u_{m+1} u_m v_{m-1} v_{m-2} \dots v_1 v_0 \rangle, \\ V^m &= \langle u_{n-1} u_{n-2} \dots u_{m+1} v_m v_{m-1} v_{m-2} \dots v_1 v_0 \rangle, \\ z^m &= \langle u_{n-1} u_{n-2} \dots u_{m+1} v_{m-1} v_{m-2} \dots v_1 v_0 \rangle. \end{aligned} \tag{10}$$

Обучающим множеством являются одна или несколько функций, представленных в мультипликативной форме, которые назовём опорными функциями. Цель обучения состоит в том, чтобы настроить нейронные ядра таким образом, чтобы результирующая матрица преобразования содержала опорные функции в виде столбцов. Номер столбца, содержащего опорную функцию, назовём точкой приспособления. Точки приспособления относятся к координатному пространству V^{n-1} . Сравнивая выражение (5) с (9), непосредственно получим правило обучения нейронных ядер:

$$w_{z^m}^m(u_m, v_m) = \varphi_{i^m}^k(u_m).$$

Здесь k – номер опорной функции. Зададим точку приспособления числом, представленным в поразрядной форме:

$$x = \langle x_{n-1} x_{n-2} \dots x_0 \rangle,$$

Тогда, в соответствии с (10), номер настраиваемых ядер по слоям будет определяться выражением:

$$z^m = \langle u_{n-1} u_{n-2} \dots u_{m+1} x_{m-1} x_{m-2} \dots x_1 x_0 \rangle.$$

Для $m = 0$ имеем $z^0 = \langle u_{n-1} u_{n-2} \dots u_1 \rangle$ – это означает, что независимо от выбора точки приспособления, все ядра слоя будут настраиваться, причём номер ядра определяется из условия: $z^0 = i^0$. Настройка элементов ядер этого слоя выполняется по правилу:

$$w_{z^0}^0(u_0, v_0) = \varphi_{i^0}^k(u_0), \tag{11}$$

Очевидно, должно быть задано взаимно-однозначное соответствие $k \leftrightarrow v_0$ между номером опорной функции и разрядной переменной v_0 . Эта разрядная переменная принимает значения $0, 1, \dots, g_0 - 1$. Отсюда следует вывод, что число точек приспособления не может быть больше, чем g_0 , и для данной топологии точки приспособления размещаются в соседних столбцах, т.е. принимают значения:

$$\begin{cases} \sum_{u_m} w_{z^m}(u_m, v_m) \varphi_{i^m}(u_m) = 1 & \text{при } v_m = x_m, \\ \sum_{u_m} w_{z^m}(u_m, v_m) \varphi_{i^m}(u_m) = 0 & \text{при } v_m \neq x_m. \end{cases}$$

Соответствие между индексами z^m и i^m определяется выбором точки приспособления. Таким образом, задача настройки быстрого приспособленного спектрального преобразования сводится к построению приспособленных ортогональных ядер. Для ортогональных ядер одна из функций определяется по правилу (11), а остальные находятся через процедуру ортогонализации Грамма-Шмидта [7].

Мультипликативное разложение опорной функции для спектрального преобразования отличается тем, что фрактальные множители должны быть нормированы к единице. Ортогональное приспособленное преобразование не полностью использует степени свободы, которыми оно располагает. Если функция приспособления совпадает с нулевым столбцом результирующей матрицы, то будут настроены только ядра с номерами $z^m = \langle u_{n-1}u_{n-2} \dots u_{m+1}0_{m-1}0_{m-2} \dots 0_0 \rangle$ (для топологии Кули-Тьюки с прореживанием по времени). На рис. 10 приведено факторизованное представление ортогонального преобразования, приспособленного к одной функции, размещённой в нулевом столбце. Символом «*» отмечены элементы незаполненных ядер. Если точка приспособления задана кортежем:

$$x = \langle x_{n-1}x_{n-2} \dots x_0 \rangle,$$

тогда номер настраиваемых ядер по слоям будет определяться выражением:

$$z^m = \langle u_{n-1}u_{n-2} \dots u_{m+1}x_{m-1}x_{m-2} \dots x_1x_0 \rangle.$$

В каждом ортогональном ядре только один столбец совпадает с фрактальным множителем опорной функции, остальные столбцы выбираются ортогональными к нему, следуя процедуре Грамма-Шмидта, поэтому они не могут быть произвольными. Это означает, что ортогональное приспособленное преобразование может быть настроено только на одну опорную функцию (не считая постоянной составляющей).

v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
u_2	u_1	u_0																						
0	0	0	1	1					1	1							1				1			
0	0	1	1	1					*	*							*				*			
0	1	0		1	1				1	1								*					*	
0	1	1		1	1				*	*									*				*	
1	0	0			1	1						1	1			1				1				
1	0	1			1	1							*	*		*				*				
1	1	0				1	1					1	1				*					*		
1	1	1				1	1					*	*					*				*	*	

Рис. 10. Свободные параметры быстрого ортогонального преобразования, приспособленного к одной функции

Для того, чтобы преобразование в целом было ортогональным, достаточно заполнить свободные ядра элементами любых ортогональных матриц соответствующих размеров. В практических алгоритмах можно либо заполнить свободные ядра единичными матрицами

вида: $W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, либо повторить ядра, уже найденные для текущего слоя. В первом случае будут получены вейвлет-подобные ортогональные преобразования с явно выраженными свойствами пространственной локализации, во втором – все столбцы преобразования будут содержать примерно равное число ненулевых элементов.

На рис. 11 приведены базисные функции быстрых ортогональных преобразований, приспособленных к линейно изменяющейся опорной функции.

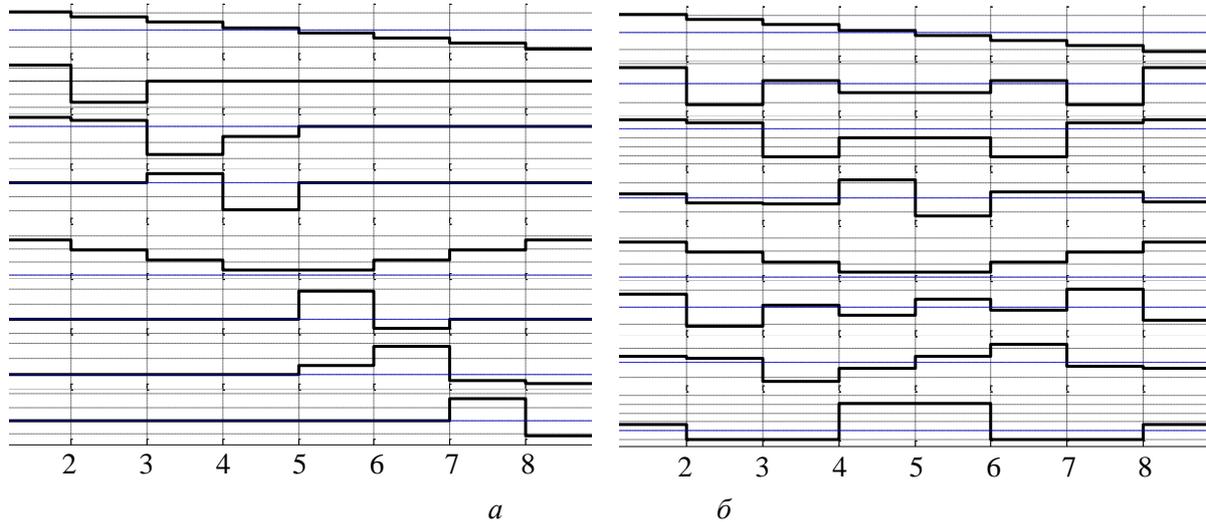


Рис. 11. Базисные функции приспособленных ортогональных преобразований:

а) – с доопределением ядер единичными матрицами;

б) – с приспособлением к фильтрованным образам опорных функций

Заключение. Отличительной особенностью рассмотренного метода обучения БНС является абсолютная устойчивость, поскольку, в отличие от градиентных методов, отсутствует контур параметрической обратной связи по ошибке. Кроме того, алгоритм обучения всегда имеет конечное число шагов, которое кратно числу слоёв быстрого преобразования. Точность обучения не ограничена используемым методом и определяется только разрядной сеткой процессора. Рассмотренный метод обучения не полностью реализует степени свободы перестраиваемого БНС, оставшаяся часть может быть использована для придания преобразованию дополнительных свойств, например, пространственной чувствительности, характерной для вейвлет-преобразований. Алгоритмы обучения не ограничиваются одномерным случаем, в работах [8-11] рассмотрены варианты алгоритмов обучения для двумерных БНС, и подобным образом алгоритмы могут быть построены для многомерных сетей.

Обучаемые быстрые ортогональные преобразования могут быть эффективно использованы для сжатия информации, например, при передаче по каналам связи изображений высокой размерности, с малыми отличиями от эталонного образа. Вместо передачи всего изображения в этом случае достаточно передать всего несколько спектральных коэффициентов приспособленного спектра, несущих отличия от эталонного образа, и восстановить отличия на приёмной стороне, используя быстрый алгоритм.

Список источников

1. Трахтман А.М. Основы теории дискретных сигналов на конечных интервалах / А.М. Трахтман, В.А. Трахтман. – М.: Сов. Радио, 1976. – 208 с.
2. Солодовников А.И. Основы теории и методы спектральной обработки информации / А.И. Солодовников, А.М. Спиваковский. – Л., 1986. – 272 с.
3. Дорогов А.Ю. Быстрые преобразования и самоподобные нейронные сети глубокого обучения. Часть 1. Стратифицированные модели самоподобных нейронных сетей и быстрых преобразований / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2023. – №4 (32). – С. 5-20.

4. Айвазян С.А. Прикладная статистика. Классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1989. – 607 с.
5. Good I.J. The interaction algorithm and practical fourier analysis. Journal of Royal statistical society. Ser.B, 1958, vol. 20, no.2, pp. 361-372.
6. Дорогов А.Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. / А.Ю. Дорогов. – СПб.: «Политехника», 2014. – 328с.
7. Белман Р. Введение в теорию матриц / Р. Белман. – М.: Наука, 1976. – 352 с.
8. Дорогов А.Ю. Программный комплекс фрактальной фильтрации и адаптивной настройки двумерных быстрых преобразований. Свидетельство о государственной регистрации программ для ЭВМ №2013614431 / А.Ю. Дорогов // Заявка №2013610130 дата поступления 15 января 2013. Зарегистрирована в Реестре программ для ЭВМ 07 мая 2013.
9. Дорогов А.Ю. Фрактальный метод настройки быстрых перестраиваемых спектральных преобразований на эталонное изображение / А.Ю. Дорогов // Международная научно-техническая конференция, приуроченная к 50-летию МРТИ–БГУИР (Минск, 18–19 марта 2014 года): материалы конф. В 2 ч. – Ч. 1. – Минск : БГУИР, 2014. – С.284-285. – ISBN 978-985-543-037-8 (ч. 1).
10. Дорогов А.Ю. Фрактальное обучение двумерных ортогональных нейронных сетей. / А.Ю. Дорогов // Материалы международной научно-практической конференции «Информационные технологии и информационная безопасность в науке, технике и образовании «ИНФОТЕХ-2015» Севастополь 07-11 сентября 2015 г., СевГУ. – С. 59-60.
11. Dorogov A.Yu. Two-Dimensional fast orthogonal neural networks. Advances in neural networks – ISSN 2016, 13th International symposium on neural networks, ISSN 2016, St. Petersburg, Russia, July 6-8, 2016, Proceedings, Lecture notes in computer science. Springer international publishing Switzerland 2016, pp. 204-210.

Дорогов Александр Юрьевич. Доктор технических наук, доцент, профессор кафедры «Автоматики и процессов управления» Санкт-Петербургского государственного электротехнического университета (СПбГЭТУ) «ЛЭТИ», главный научный сотрудник ПАО «Информационные телекоммуникационные технологии». Основные направления исследований автора связаны с интеллектуальным анализом данных, цифровой обработкой сигналов, проектированием быстрых преобразований и нейронных сетей быстрого обучения, разработкой аналитических платформ, моделированием радио-телекоммуникационных систем. AuthorID: 17611, SPIN 8645-5873, ORCID 0000-0002-7596-6761, vaks2006@yandex.ru, г. Санкт-Петербург, ул. Попова, 5.

UDC 004.032.26

DOI:10.25729/ESI.2024.33.1.001

Fast transformations and self-similar deep learning neural networks.

Part 2. Methods of training fast neural networks

Alexander Yu. Dorogov

PJSC “Information Telecommunication Technologies” (“Inteltech”),

St. Petersburg State Electrotechnical University,

Russia, St. Petersburg, vaks2006@yandex.ru

Abstract. It is noted in the paper that fast neural networks (FNN) are self-similar to the fast Fourier transform (FFT) algorithm. A method for constructing a matrix form of a fast transformation algorithm is presented. The factorizability of the elements of the FNN matrix by the elements of neural cores is proved. A method of multiplicative factorization of arbitrary one-dimensional images is proposed. It is shown that due to their structure, fast neural networks have special learning algorithms that are fundamentally different from the classic ErrorBackPropagation by the absence of a mechanism for error back propagation. The considered algorithms for teaching FNN are based on the methods of multiplicative factorization of images and fast transformations proposed in the work. Examples of network tuning to the orthogonal Hadamard basis, Fourier basis, as well as the FNN implementation of the Cantor and Sierpinski quasifractals are shown. A method of tuning fast transformations to a reference function based on the method of fractal filtering of signals is described. A method for tuning orthogonal adapted transformations is proposed. Examples are given. This work is the second part of the generalizing article "Fast transformations and self-similar neural networks of deep learning". In the first part, stratified models of self-similar neural networks are considered.

Keywords: fast transformations, fast neural networks, topological matrices, orthogonality, adapted transformations.

References

1. Trahtman A.M., Trahtman V.A. Osnovy teorii diskretnykh signalov na konechnykh intervalakh [Fundamentals of the theory of discrete signals at finite intervals]. M., "Sov. Radio", 1976, 208 p.
2. Solodovnikov A.I., Spivakovskij A.M. Osnovy teorii i metody spektral'noj obrabotki informacii [Fundamentals of the theory and methods of spectral information processing]. Leningrad, 1986. 272 p.
3. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye nejronnye seti glubokogo obuchenija. Chast' 1. Stratifirovannye modeli samopodobnykh nejronnykh setej i bystrykh preobrazovanij [Fast transformations and self-similar deep learning neural networks. Part 1. Stratified models of self-similar neural networks and fast transformations]. Informacionnye i matematicheskie tehnologii v nauke i upravlenii [Information and mathematical technologies in science and management], 2023, no. 4 (32), pp. 5-20.
4. Ajvazjan S. A., Buhshtaber V. M., Enjukov I. S., Meshalkin L. D. Prikladnaja statistika. Klassifikacija i snizhenie razmernosti [Applied statistics. Classification and dimensionality reduction]. M., Finansy i statistika [Finance and Statistics], 1989, 607 p.
5. Good I.J. The interaction algorithm and practical fourier analysis. Journal of Royal statistical society. Ser.B, 1958, vol. 20, no.2, pp. 361-372.
6. Dorogov A.Yu. Teorija i proektirovanie bystrykh perestraivaemykh preobrazovanij i slabosvjazannykh nejronnykh setej [Theory and design of fast tunable transformations and loosely coupled neural networks]. Spb. Publ "Politehnika" [Polytechnic], 2014, 328p.
7. Bellman R. Introduction to matrix analysis. Moscow, Nauka [Science], 1976. – 352 c.
8. Dorogov A.Yu. Programmnyj kompleks fraktal'noj fil'tracii i adaptivnoj nastrojki dvumernykh bystrykh preobrazovanij [A software package for fractal filtering and adaptive adjustment of two-dimensional fast transformations]. Certificate of state registration of computer programs No №2013614431, 07 May 3013.
9. Dorogov A.Yu. Fraktal'nyy metod nastrojki bystrykh perestraivayemykh spektral'nykh preobrazovanij na etalonnoye izobrazheniye [Fractal method for setting fast tunable spectral transformations on a reference image]. Mezhdunarodnaya nauchno-tehnicheskaya konferentsiya, priurochennaya k 50-letiyu MRTI-BGUIR (Minsk, 18–19 marta 2014 goda): materialy konf. V 2 ch. Ch. 1. [International scientific and technical conference dedicated to the 50th anniversary of MRTI-BSUIR (Minsk, March 18–19, 2014): materials of the conference. In 2 parts. Part 1], Minsk, BSUIR, 2014, pp. 284-285, ISBN 978-985-543-037-8 (part 1).
10. Dorogov A.Yu. Fraktal'noe obuchenie dvumernykh ortogonal'nykh nejronnykh setej [Fractal training of two-dimensional orthogonal neural networks]. Materialy mezhdunarodnoj nauchno-prakticheskoy konferencii "Informacionnye tehnologii i informacionnaja bezopasnost' v nauke, tehnike i obrazovanii "INFOTEH-2015" [Information technologies and Information security in science, technology and education "INFOTECH-2015"], Sevastopol 07-11, September 2015, SevSU, pp. 59-60.
11. Dorogov A.Yu. Two-Dimensional fast orthogonal neural networks. Advances in neural networks – ISSN 2016, 13th International symposium on neural networks, ISSN 2016, St. Petersburg, Russia, July 6-8, 2016, Proceedings, Lecture notes in computer science. Springer international publishing Switzerland 2016, pp. 204-210.

Dorogov Alexander Yurievich. Doctor of technical sciences, associate professor, professor of the department of automation and control Processes of St. Petersburg state electrotechnical university (SPbETU) "LETI", Chief researcher of PJSC "Information telecommunication technologies". The main directions of the author's research are related to data mining, digital signal processing, the design of fast transformations and neural networks of fast learning, the development of analytical platforms, modeling of radio and telecommunications systems. AuthorID: 17611, SPIN 8645-5873, ORCID 0000-0002-7596-6761, vaksa2006@yandex.ru, St. Petersburg, st. Popova, 5.

Статья поступила в редакцию 13.06.2023; одобрена после рецензирования 13.03.2024; принята к публикации 13.03.2024.

The article was submitted 06/13/2023; approved after reviewing 03/13/2024; accepted for publication 03/13/2024.