

**Методы, технологии и приложения искусственного интеллекта**

УДК 004.93'12:004.032.26

DOI:10.25729/ESI.2024.36.4.001

**Коллективные классификаторы повторного входа и их реализация в классе самоподобных нейронных сетей****Дорогов Александр Юрьевич**

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»,

ПАО «Информационные телекоммуникационные технологии»

Россия, Санкт-Петербург, *vaksa2006@yandex.ru*

**Аннотация.** В статье предложен новый метод обучения частных классификаторов, а также способ агрегации их прогнозов в составе комитета. Обучение основано на гипотезе итерационного повторного входа биологических нейронных сетей и использует для своей реализации метод главных компонент. Для частных классификаторов задаются области компетенций, в совокупности покрывающих обучающее множество примеров. Показано, что процесс итерационного обучения за несколько шагов сходится, обеспечивая точность распознавания 100% в области компетенции частного классификатора. Агрегация прогнозов реализуется по принципу максимальной проекции образа на собственные подпространства классов частных классификаторов. Приведены примеры использования комитета компетентных классификаторов для набора данных MNIST. Предложена модель непрерывного обучения комитета классификаторов, пригодная для построения самообучающихся систем распознавания. Рассмотрена нейросетевая реализация классификаторов в классе самоподобных нейронных сетей.

**Ключевые слова:** частный классификатор, комитет классификаторов, коллективное распознавание, область компетенции, повторный вход, непрерывное обучение, самоподобная нейронная сеть

**Цитирование:** Дорогов А.Ю. Коллективные классификаторы повторного входа и их реализация в классе самоподобных нейронных сетей / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2024. – № 4(36). – С. 5-16. – DOI:10.25729/ESI.2024.36.4.001.

**Введение.** Нобелевский лауреат Джеральд Морис Эдельман в 1977 году выдвинул гипотезу [1, 2] о том, что реентерабельная (reentry - повторно-вводимая) передача сигналов служит общим механизмом для накопления знаний в коре головного мозга. Повторный вход по Эдельману – это динамический процесс постоянной пространственно-временной корреляции, происходящий между функционально разделёнными нейронными областями, который опосредуется повторной передачей сигналов через параллельные нервные волокна. Механизмом этого явления считается ритмическая активность мозга. С точки зрения теории динамических систем повторный вход подобен итерационному процессу, сходящемуся к некоторому устойчивому аттрактору, обобщающему накопленные знания.

Развивая собственную теорию нейродарвинизма (Neural Darwinism) [3], Эдельман отмечал, что в коре головного мозга на один и тот же сенсорный сигнал одновременно реагируют несколько конкурирующих нейронных групп, реакция которых в дальнейшем конвергируется в более глубоких слоях коры, реализуя принцип коллективного распознавания и одновременно стимулируя процесс эволюционного развития нейронных структур коры головного мозга.

В технических приложениях метод коллективного решения задачи классификации заключается в объединении моделей одиночных классификаторов в комитет с некоторым общим принципом агрегации прогнозов частных классификаторов. В настоящей работе будут рассмотрены технические решения, объединяющие методы коллективного распознавания и усиление классификаторов за счет реализации принципа повторного входа. В заключительном разделе будет представлена нейросетевая реализация предлагаемых решений.

**1. Методы коллективного распознавания.** При формировании комитета классификаторов необходимо оптимизировать два критерия – качество обучения частных классификаторов и оптимальность их объединения [4, 5]. Простейшим методом агрегации частных прогнозов может быть простое голосование, которое приводит к увеличению точности коллективного решения за счет снижения степени неопределенности при сложении частных прогнозов. Общее название методов построения коллективных классификаторов выражается термином «boosting» - усиление [6]. Наиболее известными являются методы bagging и adaboost.

Метод bagging, предложенный Лео Брейманом в 1994 году [7], предполагает обучение частных классификаторов на случайных выборках, формируемых из обучающего множества. Агрегация классификаторов выполняется простым голосованием.

Метод adaboost, предложенный Йоавом Фройндом и Робертом Шапире [8] в 1995 году, предполагает обучение частных классификаторов на одной выборке с пошаговой адаптацией по ошибкам обучения как самих классификаторов, так и комитета классификаторов. Агрегация частных прогнозов выполняется взвешенным голосованием. Оптимальные веса для классификаторов рассчитываются на каждом шаге алгоритма. Одновременно видоизменяется обучающая выборка, где для каждого примера также выполняется расчет веса, который используется в процедуре обучения, например, для повторения в выборке примеров, на которых обнаружены ошибки.

В 70-х и 80-х годах советские ученые Л.А. Растрингин и Р.Х. Эренштейн в ряде работ разработали метод коллективного распознавания на основе компетентных классификаторов [9, 10, 11]. Идея таких методов базируется на том, что каждый базовый классификатор может работать хорошо в некоторой области пространства признаков (эта область интерпретируется как область компетентности классификатора), превосходя в этой области остальные классификаторы по точности и достоверности решений. Классификация образов выполняется в два этапа: первый этап связан с определением классификатора, область компетенции которого покрывает распознаваемый образ, и на втором этапе найденный классификатор используется для отнесения образа к некоторому классу. Авторы предложили несколько способов определения областей компетенции классификаторов и доказали применимость используемого подхода на примерах.

**2. Компетентный классификатор повторного входа.** Рассмотрим классификатор, основанный на оценке длины проекции образа на подпространства классов. Подпространства класса порождаются линейной комбинацией векторов обучающей выборки, принадлежащих одному классу. Проекцию вектора-образа легко определить, если в подпространстве класса задать ортогональный базис. Сумма квадратов коэффициентов разложения вектора образа по этому базису будет определять квадрат проекции образа на подпространство класса. Образы нормируются по энергии к единице. Классификатор относит образ к классу, для которого проекция на подпространство класса оказывается максимальной.

Среди возможных ортогональных базисов можно подобрать оптимальный базис, который обеспечивает наиболее быструю сходимость коэффициентов разложения. Известно, что таким свойством обладает ортогональный базис, полученный методом главных компонент (англ. Principal Component Analysis, PCA) [12] из ковариационной матрицы векторов класса. Эмпирическая или выборочная ковариационная матрица класса определяется выражением:

$$C = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^T (x_i - \bar{x}),$$

где  $m$  - число примеров в классе,  $x_i$  - вектор-строка представителя класса,  $\bar{x}$  - вектор-строка средних значений примеров в классе. Ковариационная матрица является симметричной и положительно определённой. Для такой матрицы можно найти собственные ортогональные вектора  $v$  такие, что

$$Cv = \lambda v.$$

Скаляр  $\lambda$  называется собственным числом, соответствующим собственному вектору  $v$ . Все собственные числа ковариационной матрицы положительные, и чем большее значение они имеют, тем большей значимостью обладает собственный вектор. Для построения модели классификатора отбирают наиболее значимые вектора. При распознавании вектор образа центрируется по среднему образу примеров в классе, и раскладывается по выбранным собственным векторам класса. Далее по коэффициентам разложения вычисляется длина проекции образа на подпространство класса. Вектору образа присваивается метка класса, в котором длина его проекции максимальна.

Воспользуемся теперь принципом повторного входа для обучения PCA-классификатора. В обучающей выборке выберем представительный фрагмент, который далее будем называть областью компетенции классификатора. В области компетенции произвольно выделим базовое множество примеров, которое будем использовать для построения модели PCA-классификатора. На каждом шаге эволюции по результатам классификации выполняется контроль ошибок модели на области компетенции. Примеры, на которых произошли ошибки, добавляются к базовому множеству с дифференциацией по классам и модель PCA-классификатора строится заново. Процесс многократно повторяется до полного устранения ошибок на области компетенции. В процессе эволюции размер базового множества постепенно возрастает. Темп сходимости к нулевой ошибке зависит от размера области компетенции, начального базового множеств и размерности модели (числа удерживаемых собственных векторов). Построенный классификатор будем называть компетентным.

Покажем эффективность метода обучения классификатора на примере набора данных MNIST [13]. Набор данных содержит двумерные образы рукописных цифр от 0 до 9 в виде пиксельных изображений размером  $28 \times 28$ . Объем обучающей выборки равен 60000 образов, и 10000 изображений содержит тестовая выборка. На рис. 1 показана выборка с десятью представителями для каждого класса. При проведении экспериментов все примеры выборок нормировались по энергии к единичному уровню.

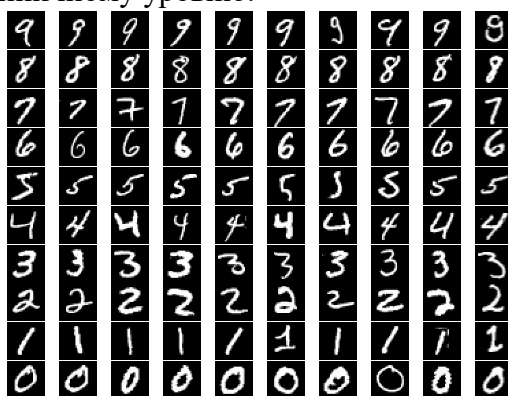
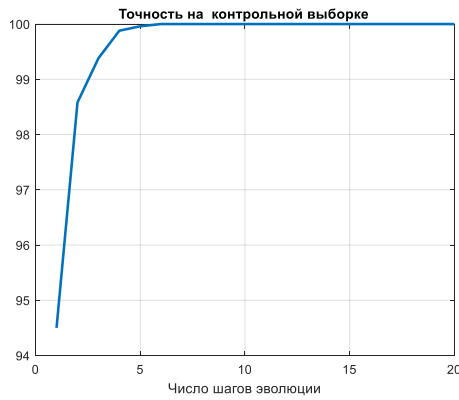


Рис. 1. Выборка из набора рукописных цифр MNIST

Для набора данных максимальный объем области компетенции, на котором эволюционный процесс ещё сходится, составляет примерно 20000 примеров. Наилучшие результаты достигаются на выборках из 5000-10000 примеров. На рис. 2 показана зависимость точности классификатора примеров с областью компетенции из 5000 примеров и начальной базовой выборки 1000 примеров.

Как видно из рисунка, итерационный процесс сходится примерно за 6 шагов. Размер моделей в этом эксперименте составлял 30 собственных векторов. В процессе эволюции базовое множество возросло с 1000 до 1385 примеров. Обученный классификатор проверялся на тестовой выборке размером 10000 примеров. На рис. 3 показана таблица ошибок классификатора. Достигнутая точность на тестовой выборке составила 94.6%.



**Рис. 2.** Зависимость точности классификатора на области компетенции от числа шагов эволюции

Матрица ошибок. Точность классификатора=94.5995

1	961		2		2	1	5	2	5	2	961	19
2		1123	4	2		1	1		3	1	1123	12
3	8	4	968	14	4		4	11	14	4	968	63
4	1	1	14	939		28	1	7	14	5	939	71
5		2	3		951		4	8		14	951	31
6	3	2		29	3	825	5	5	17	3	825	67
7	8	2	1		4	15	924			3	924	33
8		10	16	1	2	2		974	1	22	974	54
9	6	6	8	35	5	14	6	8	871	15	871	103
10	4	5	3	9	31	3		23	9	922	922	87
	1	2	3	4	5	6	7	8	9	10		

**Рис. 3.** Таблица ошибок PCA-классификатора повторного входа на тестовой выборке

**3. Комитет компетентных классификаторов.** Для построения комитета классификаторов разделим всю обучающую выборку на представительные области компетенции и на каждой области обучим PCA-классификатор с повторным входом. Для формирования агрегированного результата комитета будем использовать принцип максимума. Фактически за результат комитета будет отвечать классификатор с максимальным значением проекции образа на подпространство класса.

На рис. 4 показана матрица ошибок для полного обучающего множества, состоящего из 60000 примеров. Точность классификации на обучающей выборке составляет чуть более 99 процентов при размере комитета равным 12. На рис. 5 показана матрица ошибок этого же коллективного классификатора на полной тестовой выборке, состоящей из 10000 примеров. Точность классификации составляет примерно 97 процентов.

Матрица ошибок. Точность классификатора=99.0167

1	5913	2			3	2		1	2	5913	10	
2		6737						3	2	6737	5	
3	9	12	5901	10	1		1	9	12	3	5901	57
4	3	1	13	6049		22	2	3	33	5	6049	82
5	2	9	1	1	5782			8		39	5782	60
6	5	1	2	31		5337	18		20	7	5337	84
7	9	4	1		1	4	5897			2	5897	21
8	1	13	8	2	6			6203	1	31	6203	62
9	7	32	8	20	6	14	8	3	5731	21	5731	119
10	10	3	3	11	24	2	1	29	7	5859	5859	90
	1	2	3	4	5	6	7	8	9	10		

**Рис. 4.** Таблица ошибок комитета эволюционных классификаторов для обучающей выборки

Матрица ошибок. Точность классификатора=96.9997

1	972		2			2	1	3		972	8	
2		1130	2	1			2			1130	5	
3	8	6	995	1	2		2	6	10	2	995	37
4	3		7	967		12		6	11	4	967	43
5	1	2	3		962		3	1	2	8	962	20
6	2		1	13		855	7	2	10	2	855	37
7	5	3			2	4	940		3		940	17
8		11	8	1	2	1		989		16	989	39
9	7	3	2	11	3	5	3	5	929	6	929	45
10	6	6	2	6	11	2		11	5	960	960	49
	1	2	3	4	5	6	7	8	9	10		

**Рис. 5.** Таблица ошибок комитета эволюционных классификаторов для тестовой выборки

Для эксперимента использовались классификаторы повторного входа с областью компетенции 5000 примеров и начальным размером базового множества 1000 примеров. На текущий момент мировое достижение [14] по точности классификации тестового набора данных MNIST составляет 99.87%.

**4. Коллективный классификатор непрерывного обучения.** Природа не разделяет данные на обучающую и тестовую выборки, биологический мозг обучается непрерывным потоком поступающих данных. С этой точки зрения имеет смысл отказаться от общепринятого деления выборки на обучающую и тестовую. Принцип разделения классификаторов по областям компетенции, с агрегированием по максимуму, позволяет неограниченно наращивать размер комитета, покрывая всю выборку и обучающую, и тестовую, а также все последующие поступившие данные. На рис. 6 приведена матрица ошибок для объединённой выборки коллективного классификатора для объединённой выборки с числом примеров 70000.

В комитете использовались эволюционные классификаторы с областью компетенции 5000 примеров и начальным базовым множеством 1000 примеров. Число компетентных классификаторов для объединённой выборки из 70000 примеров составило 14. Точность коллективного классификатора составила 99.49%. Модель непрерывного обучения может быть использована для построения самообучающихся систем распознавания реального времени, за счет добавления новых классификаторов в процессе функционирования системы.

Эволюционный принцип повторного входа можно распространить также на весь комитет классификаторов. С этой целью в модели каждого компетентного классификатора запоминаются номера примеров базового множества на момент завершения обучения (т.е. достижения 100% точности на области компетенции). Алгоритм эволюции для комитета на каждом шаге фиксирует номера примеров в объединённой выборке, на которых произошли ошибки. Эти примеры разделяются по областям компетенции частных классификаторов и добавляются в списки примеров базовых множеств. Далее заново выполняется обучение частных классификаторов с обновлёнными базовыми множествами до достижения 100% точности на областях компетенции и вновь контролируются ошибки комитета классификаторов, далее итерации комитета повторяются.

Эволюционный процесс сходится к 100% процентной точности. На рис. 7 показана зависимость точности комитета классификаторов от числа шагов эволюции. Мета-параметры комитета соответствуют предыдущему примеру. Предложенная модель непрерывного эволюционного обучения комитета классификаторов обеспечивает 100 процентную точность распознавания объединённой выборки.

Матрица ошибок. Точность классификатора=99.4929

1	6900		1		1	1					6900	3	
2		7874			1					2		7874	3
3	7	5	6959	11			1	3	4			6959	31
4	1	3	4	7096		13		1	19	4		7096	45
5	1	3	1		6795		2	4	2	16		6795	29
6	3		1	23		6246	10	1	27	2		6246	67
7	5	1			1	2	6864			3		6864	12
8		5	7		3			7255			23	7255	38
9	4	22	4	9	2	9	5	4	6752	14		6752	73
10	3	1	2	5	16				21	6	6904	54	
	1	2	3	4	5	6	7	8	9	10			
	Предсказанный класс												

Рис. 6. Таблица ошибок непрерывного классификатора на объединённой выборке

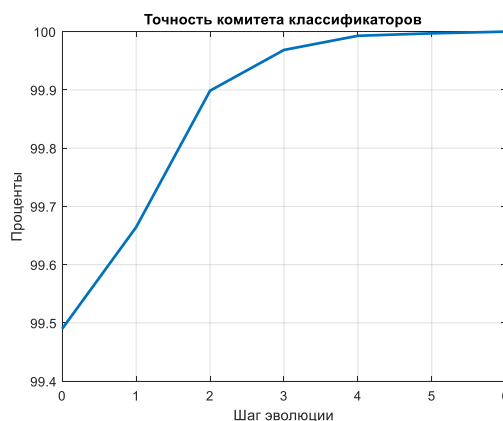
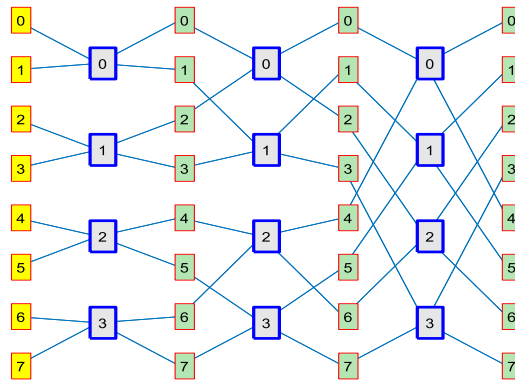


Рис. 7. Зависимость точности комитета классификаторов от числа шагов эволюции

**5. Нейросетевая реализация классификаторов.** Классификатор повторного входа вычисляет проекцию вектора-образа на подпространства классов. Проекция определяется через разложение вектора в усеченных ортогональных базисах. Это линейная матричная операция,

которая реализуется через вычисление скалярных произведений векторов. Переход к нейросетевой реализации связан с факторизацией линейного преобразования в параллельно-последовательную матричную форму.

**5.1. Быстрая нейронная сеть.** В основе построения схемы факторизации лежат классические алгоритмы быстрых преобразований. В работе [4] показано, что они являются частным случаем многослойных самоподобных нейронных сетей. На рис. 8 представлен граф быстрого преобразования Фурье размерности 8 с топологией Кули-Тьюки «с прореживанием по времени». В каждом слое выделены четыре базовых операции типа «Бабочка». Для преобразования Фурье параметры базовых операций полностью определены, однако, если допустить, что их параметры можно изменять, то мы приходим к варианту быстрых нейронных сетей (БНС) [15]. В этом контексте базовые операции уместно назвать нейронными ядрами.



**Рис. 8.** Граф быстрого преобразования Фурье

Сетевая модель данной топологии и описывается набором кортежей [4]:

$$\begin{aligned} U^m &= \langle u_{n-1}u_{n-2} \cdots u_{m+1}u_m v_{m-1}v_{m-2} \cdots v_1v_0 \rangle, \\ V^m &= \langle u_{n-1}u_{n-2} \cdots u_{m+1}v_m v_{m-1}v_{m-2} \cdots v_1v_0 \rangle, \\ z^m &= \langle u_{n-1}u_{n-2} \cdots u_{m+1}v_{m-1}v_{m-2} \cdots v_1v_0 \rangle. \end{aligned} \quad (1)$$

где  $z^m$  – порядковый номер нейронного ядра;  $U^m, V^m$  – порядковые номера рецепторов и аксонов в слое  $m$ ;  $u_m, v_m$  – локальные номера рецепторов и аксонов в пределах нейронного ядра слоя  $m$ . Каждый кортеж представляет собой поразрядную форму представления порядкового номера через разрядные переменные  $u_m, v_m$ . Основания разрядных переменных  $u_m, v_m$  определяются целыми положительными числами  $p_m, g_m$ , и по слоям могут быть заданы соответствиями:

$$\begin{pmatrix} u_0 & u_1 & \cdots & u_{n-2} & u_{n-1} \\ p_0 & p_1 & \cdots & p_{n-2} & p_{n-1} \end{pmatrix}, \quad \begin{pmatrix} v_0 & v_1 & \cdots & v_{n-2} & v_{n-1} \\ g_0 & g_1 & \cdots & g_{n-2} & g_{n-1} \end{pmatrix}.$$

Для ядер размерности  $2 \times 2$ , имеем  $p_m = 2, g_m = 2$  и все разрядные переменные принимают значения  $\{0,1\}$ . Координатные направления  $U^m$  и  $V^m$  в дальнейшем будем называть входной и выходной плоскостью нейронного слоя. Для терминальных плоскостей (относящихся к начальному и конечному слою сети) будем использовать обозначения  $U$  и  $V$ . Для построения быстрых алгоритмов размерность преобразования должна быть составным числом, и чем больше множителей в разложении размерности, тем выше вычислительная эффективность быстрого алгоритма. Размерности быстрой нейронной сети по входу и выходу вычисляются через произведения оснований разрядных переменных:

$$N = p_{n-1} \cdots p_1 p_0, \quad M = g_{n-1} \cdots g_1 g_0.$$

Число слоёв в быстром преобразовании равно числу сомножителей в этих произведениях. Несмотря на большое разнообразие быстрых алгоритмов, конфигурации их структур удовлетворяют системному инварианту самоподобия [15]. Как известно, таким же свойством самоподобия обладают фракталы, поэтому быстрые алгоритмы можно интерпретировать как квазифракталы. Свойство структурной фрактальности позволяет решить одновременно две задачи: реализовать быструю обработку данных и выполнить быстрое обучение преобразования. Обучение быстрого преобразования заключается в выборе значений элементов нейронных ядер, так, чтобы в столбцах матрицы преобразования содержался заданный набор эталонных функций, это могут быть, например, ортогональные собственные вектора выборочной ковариационной матрицы. В [16] показано, что для самоподобных нейронных сетей элементы матрицы быстрого преобразования могут быть выражены через произведения элементов нейронных ядер:

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

Там же доказано, что произвольная функция, заданная на дискретном интервале длиной  $N = p_{n-1} \cdots p_1 p_0$ , может быть представлена в мультипликативной форме:

$$f(u) = \phi_{i^0}^0(u_0) \phi_{i^1}^1(u_1) \cdots \phi_{i^{n-2}}^{n-2}(u_{n-2}) \phi_{i^{n-1}}^{n-1}(u_{n-1}).$$

где  $i^m = \langle u_{n-1} u_{n-2} \cdots u_{m+1} \rangle$ . Отсюда следует правило настройки нейронных ядер:

$$w_{z^m}^m(u_m, v_m) = \phi_{i^m}^k(u_m). \quad (2)$$

Здесь  $k$  – номер опорной функции. Зададим точку привязки эталонной функции в выходной плоскости числом, представленным в поразрядной форме:

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

тогда номера настраиваемых ядер по слоям будут определяться выражением:

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

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

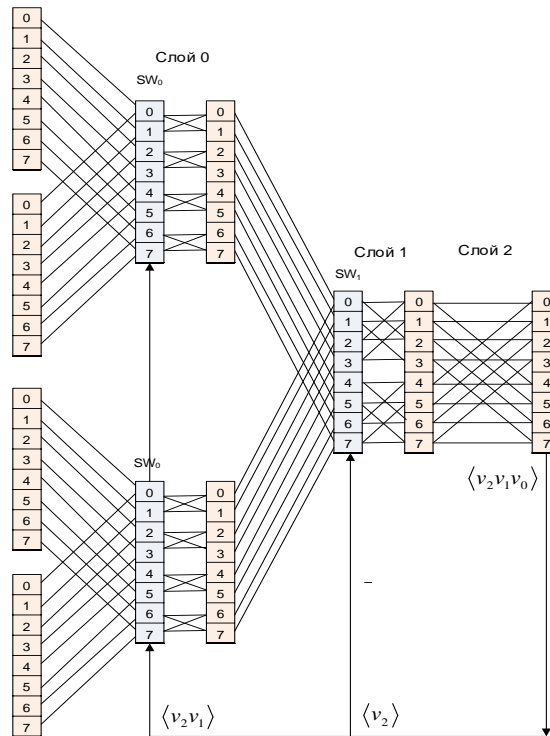
$$w_{z^0}^0(u_0, v_0) = \phi_{i^0}^k(u_0).$$

Очевидно, должно быть задано взаимно-однозначное соответствие  $k \leftrightarrow v_0$  между номером опорной функции  $k$  и разрядной переменной  $v_0$ . Эта разрядная переменная принимает значения  $0, 1, \dots, g_0 - 1$ . Отсюда следует вывод, что число эталонных функций не может быть больше, чем  $g_0$ , а если ещё потребовать выполнения условия ортогональности ядер, то матрица такого преобразования будет содержать только одну произвольную функцию в качестве столбца. Этого явно недостаточно для реализации РСА-классификатора. В следующем разделе будет показана модификация топологии БНС, которая позволяет решить поставленную задачу.

**5.2. Самоподобные нейронные сети с дополнительными плоскостями.** Дополним топологическую модель быстрого преобразования (1) дополнительными плоскостями нейронных ядер [17], номера которых определим переменной  $\pi_m$ , новая топологическая модель в этом случае будет описываться набором кортежей:

$$\begin{aligned}
 U^m &= \langle u_{n-1}u_{n-2} \cdots u_{m+1}u_m v_{m-1}v_{m-2} \cdots v_1v_0 \rangle, \\
 V^m &= \langle u_{n-1}u_{n-2} \cdots u_{m+1}v_m v_{m-1}v_{m-2} \cdots v_1v_0 \rangle, \\
 z^m &= \langle u_{n-1}u_{n-2} \cdots u_{m+1}v_{m-1}v_{m-2} \cdots v_1v_0 \rangle, \\
 \pi_m &= \langle v_{n-1}v_{n-2} \cdots v_{m+2}v_{m+1} \rangle.
 \end{aligned}$$

Максимальное количество дополнительных плоскостей появится в нулевом слое. Номер плоскости в нулевом слое будет определяться кортежем  $\pi_0 = \langle v_{n-1}v_{n-2} \cdots v_2v_1 \rangle$ . Плоскость с номером  $\pi_0 = 0$  будем считать плоскостью исходной базовой топологической структуры. Число плоскостей в нулевом слое будет равно произведению оснований:  $g_{n-1}g_{n-2} \cdots g_2g_1$ . По мере движения к выходному слою число дополнительных плоскостей будет уменьшаться и для последнего слоя  $\pi_{n-1} = \langle \rangle$ , т.е. их не будет совсем, останется только одна плоскость базовой топологии. Таким образом, в новой топологии плоскость последнего слоя останется прежней, а в младших слоях появятся дополнительные плоскости. Номер ядра теперь следует уточнять его размещением в дополнительной плоскости. На рис. 9 показана новая топология, построенная на базе трёхслойной БНС с основаниями 2.



**Рис. 9.** Топология самоподобной нейронной сети с дополнительными плоскостями

В сети условно показаны коммутаторы (SW), которые служат для пояснения принципа работы сети. Коммутаторы управляются разрядными переменными точек выходной плоскости. Фактически коммутаторов нет, они реализуются в составе алгоритма и не препятствуют параллельной обработке входных данных. В контексте спектрального преобразования точки выходной плоскости точки ассоциируются со спектральными коэффициентами, поэтому привязка спектрального коэффициента к координатам выходной плоскости предопределяет выбор дополнительных плоскостей, которые используются для обработки входных образов. При построении полного спектрального анализатора все векторные входы сети параллельно объединяются.



Поскольку правило порождения новых плоскостей не противоречит базовой топологической модели, то для настройки ядер преобразования к эталону можно использовать прежнее правило (2), расширив его аргументом  $\pi_m$  для дополнительных плоскостей:

$$w_{z^m}^m \langle \pi_m \rangle (u_m, x_m) = \phi_{\pi_m}^k (u_m).$$

здесь  $k$  – номер эталонной функции,  $x = \langle x_{n-1} x_{n-2} \dots x_0 \rangle$  – точка привязки,

$z^m = \langle u_{n-1} u_{n-2} \dots u_{m+1} x_{m-1} x_{m-2} \dots x_1 x_0 \rangle$  – номер настраиваемых ядер по слоям,

$\pi_m = \langle x_{n-1} x_{n-2} \dots x_{m-1} \rangle$  – номер плоскости размещения ядер.

Индекс  $k$  в правой части нумерует точку приспособления. Для  $m = 0$  имеем  $z^0 = i^0 = \langle u_{n-1} u_{n-2} \dots u_1 \rangle$ , а варьируемыми переменными в левой части являются номер плоскости  $\pi_0 = \langle x_{n-1} x_{n-2} \dots x_1 \rangle$  и разряд  $x_0$ .

Вместе они покрывают весь диапазон координат выходной плоскости. Этому диапазону отвечают возможные значения индекса  $k$  в правой части, отсюда следует, что каждая точка выходной плоскости может быть связана с собственной эталонной функцией, т.е. построенная сеть обладает максимально возможным числом точек привязки, покрывающих всю выходную плоскость и, таким образом, может быть использована для реализации произвольного линейного преобразования размерности  $N \times M$ .

В модели РСА-классификатора для каждого класса сохраняется только часть ортогональных векторов, которые обладают наибольшей значимостью. Размерность выходной плоскости следует выбрать так, чтобы значение  $M$  было не меньше, чем общее число всех векторов модели классификатора. Подобным образом можно построить все частные классификаторы комитета. Нейросетевая реализация обеспечивает максимальное распараллеливание вычислительных операций, что позволяет получить высокое быстродействие классификатора на специализированных процессорах.

**Заключение.** Рассмотренный метод построения коллективных классификаторов отличается от методов bagging и adaboost использованием классификаторов с заданными областями компетенции и способом получения агрегативного решения комитета.

В отличие от метода коллективного распознавания Л.А. Растригина и Р.Х. Эренштейна, области компетенции классификаторов не локализуются расчётным путём, а назначаются, исходя из условия достаточной представительности классов в базовых множествах и областях компетенции. Допустимо также частичное пересечение областей компетенции. Благодаря принципу повторного входа, частные классификаторы идеально обучаются в пределах областей их компетенций. Это позволяет сделать процедуру классификации одноступенчатой, без предварительного определения области компетенции. Предложенный способ агрегации решений частных классификаторов по принципу максимума позволяет наращивать классификатор в процессе функционирования и реализовать принцип непрерывного обучения.

Для построения ортогонального базиса подпространства класса можно использовать также сингулярное разложение матрицы, составленной из векторов обучающей выборки [18]. Эти классификаторы также обучаются по методу повторного входа, причём скорость обучения оказывается несколько выше, чем у РСА-классификаторов. Точность комитета классификаторов данного типа примерно соответствует точности комитета с РСА- классификаторами.

В статье показано, что РСА-классификаторы можно реализовать в классе линейных самоподобных нейронных сетей с дополнительными плоскостями, при этом число реализуемых эталонных функций по сравнению с БНС кардинально возрастает и покрывает все элементы выходной плоскости нейронной сети. Такое расширение топологии не нарушает принципа по-

строения обучающего алгоритма и позволяет строить и обучать РСА- классификаторы с произвольной размерностью модели. Более того, можно показать, что нейросетевые реализации сегментируются в лес независимых пирамидальных сетей [19]. Это порождает уникальное качество – возможность дообучения классификаторов к новым образам без изменения или потери ранее накопленных знаний. Алгоритмы обучения к модельным векторам являются абсолютно устойчивыми и завершаются за конечное число шагов.

### Список источников

1. Edelman G. M. Group selection and phasic reentrant signaling: a theory of higher brain function. The mindful brain: cortical organization and the group-selective theory of higher brain function, Eds. Edelman G.M., Mountcastle V.B. Boston, MIT Press, 1978, pp 51–98.
2. Mountcastle V.B. An organizing principle for cerebral function: the unit model and the distributed system. The mindful brain: cortical organization and the group-selective theory of higher brain function, Eds. Edelman G.M., Mountcastle V.B. Boston, MIT Press, 1978, pp. 7-50, ISBN 0-262-05020-X.
3. Edelman G.M. Neural Darwinism: The theory of neuronal group selection. New York, Basic Books, 1987, 240 p.
4. Городецкий В.И. Методы и алгоритмы коллективного распознавания: обзор / В.И. Городецкий, С.В. Серебряков // Труды СПИИРАН. – СПб.: Наука, 2006. – Вып. 3. – т. 1.
5. Терехов С.А. Гениальные комитеты умных машин / С.А. Терехов. Научная сессия МИФИ 2007. IX Всероссийская научно-техническая конференция «НЕЙРОИНФОРМАТИКА–2007»: Лекции по нейроинформатике. Часть 2.– М.: МИФИ, 2007. – С. 11-43.
6. Eric Bauer, Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning, 1999, vol. 36, pp. 105–139.
7. Breiman L. Bagging predictors. Department of Statistics University of California Berkeley, California. Technical Report No.421, 1994.
8. Yoav Freund, Robert E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. Second european conference on computational learning theory, 1995, pp 23-37, DOI:10.1007/3-540-59119-2\_166.
9. Растринин Л. А. Обучение коллектива решающих правил / Л.А. Растринин, Р.Х. Эренштейн // Адаптивные системы, 1974. – Вып. 4. – С. 8–20.
10. Растринин Л.А. Принятие решений коллективом решающих правил в задачах распознавания образов / Л.А. Растринин, Р.Х. Эренштейн // Автоматика и телемеханика, 1975. – №. 9.– С. 133–144.
11. Растринин Л.А., Метод коллективного распознавания / Л.А. Растринин, Р.Х. Эренштейн – Москва: Энергоиздат, 1981.
12. Лагутин М.В. Наглядная математическая статистика. Учебное пособие / М.В. Лагутин. – М.: БИНОМ. Лаборатория знаний, 2007. – 472с.
13. THE MNIST DATABASE of handwritten digits. Available at: <http://yann.lecun.com/exdb/mnist/>
14. Adam Beyerly, Tatiana Kalganova, Ian Dear No routing needed between capsules. ArXiv:2001.09136, DOI:10.48550/arXiv.2001.09136.
15. 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. – DOI:10.25729/ESI.2023.32.4.001.
16. 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.
17. Дорогов А.Ю. Быстрые нейронные сети глубокого обучения / А.Ю. Дорогов. III Международная научная конференция по проблемам управления в технических системах (CTS'2019). Сборник докладов. Санкт-Петербург. СПб.: СПбГЭТУ «ЛЭТИ», 2019. – С. 275-280.
18. Воеводин В.В. Матрицы и вычисления / В.В. Воеводин, Ю.А. Кузнецов // М.: Наука, 1984. – 320с.
19. Дорогов А.Ю. Пластичность самоподобных нейронных сетей / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2024. – № 3(35). – С. 33–43. – DOI:10.25729/ESI.2024.35.3.003.

*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 об-*

работкой 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, vaksa2006@yandex.ru, г. Санкт-Петербург, ул. Попова, 5.

UDC 004.93'12:004.032.26

DOI:10.25729/ESI.2024.36.4.001

## Collective re-entry classifiers and their implementation in a class of self-similar neural networks

Alexander Yu. Dorogov

St. Petersburg State Electrotechnical University

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

Russia, St. Petersburg, vaksa2006@yandex.ru

**Abstract.** The article proposes a new method for teaching private classifiers, as well as a way to aggregate their forecasts as part of a committee. The training is based on the hypothesis of iterative re-entry of biological neural networks and uses the principal component method for its implementation. For private classifiers, the areas of competence are defined in the aggregate covering the training set of examples. It is shown that the iterative learning process converges in several steps, ensuring 100% recognition accuracy in the area of competence of the private classifier. The aggregation of forecasts is implemented according to the principle of maximum projection of the image onto its own subspaces of classes of private classifiers. Examples of the use of the committee of competent classifiers for the MNIST dataset are given. A continuous learning model of the classifier committee is proposed that is suitable for building self-learning recognition systems. The neural network implementation of classifiers in the class of self-similar neural networks is considered.

**Keywords:** private classifier, committee of classifiers, collective recognition, area of competence, re-entry, continuous learning, self-similar neural network

### References

1. Edelman G. M. Group selection and phasic reentrant signaling: a theory of higher brain function. The mindful brain: cortical organization and the group-selective theory of higher brain function, Eds. Edelman G.M., Mountcastle V.B. Boston, MIT Press, 1978, pp 51–98.
2. Mountcastle V.B. An organizing principle for cerebral function: the unit model and the distributed system. The mindful brain: cortical organization and the group-selective theory of higher brain function, Eds. Edelman G.M., Mountcastle V.B. Boston, MIT Press, 1978, pp. 7-50, ISBN 0-262-05020-X.
3. Edelman G.M. Neural Darwinism: The theory of neuronal group selection. New York, Basic Books, 1987, 240 p.
4. Gorodeckij V.I., Serebrjakov S.V. Metody i algoritmy kollektivnogo raspoznaniya: obzor [Methods and algorithms of collective recognition: an overview]. Trudy SPIIRAN [Proceedings of SPIIRAN], 2006, no. 3, vol.
5. Terekhov S.A. Genial'nyye komitety umnykh mashin [Brilliant committees of smart machines]. Nauchnaya sessiya MIFI 2007. IX Vserossiyskaya nauchno-tehnicheskaya konferentsiya «NEUROINFORMATIKA–2007», Lektsii po neyroinformatik [Scientific session of MEFPhI 2007. IX All-Russian Scientific and Technical Conference "NEUROINFORMATICS–2007": Lectures on neuroinformatics], 2008, part 2, Moscow, MIFI, pp. 11–43.
6. Eric Bauer, Ron Kohavi An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning, 1999, vol. 36, pp. 105–139.
7. Breiman L. Bagging predictors. Department of Statistics University of California Berkeley, California. Technical Report No. 421, 1994.
8. Yoav Freund, Robert E. Schapire A decision-theoretic generalization of online learning and an application to boosting. Second european conference on computational learning theory, 1995, pp 23-37, DOI:10.1007/3-540-59119-2\_166.
9. Rastrigin L. A., Erenshiteyn R. Kh. Obucheniye kollektiva reshayushchikh pravil [Teaching a team of crucial rules]. Adaptivnye sistemy [Adaptive systems], 1974, no. 4, c. 8–20.

10. Rastrigin L.A., Erenshteyn R.Kh. Prinyatiye resheniy kollektivom reshayushchikh pravil v zadachakh raspoznaniya obrazov [Decision-making by a team of decisive rules in image recognition tasks]. *Avtomatika i telemekhanika* [Automation and telemechanics], 1975, no.9, pp. 133–144.
11. Rastrigin L.A., Erenshteyn R.Kh. Metod kollektivnogo raspoznaniya [The method of collective recognition]. Moskva, Energoizdat [Moscow, Energoizdat], 1981.
12. Lagutin M.V. Naglyadnaya matematicheskaya statistika [Visual mathematical statistics]. M., BINOM. Laboratoriya znaniy [Moscow, BINOM. Laboratory of knowledge], 2007, 472 p.
13. THE MNIST DATABASE of handwritten digits. Available at: <http://yann.lecun.com/exdb/mnist/>
14. Adam Byerly, Tatiana Kalganova, Ian Dear No routing needed between capsules. ArXiv:2001.09136, DOI:10.48550/arXiv.2001.09136.
15. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye neyronnye seti glubokogo obucheniya. Chast' 1. Stratifitsirovannyye modeli samopodobnykh neyronnykh setey i bystrykh preobrazovaniy [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, DOI:10.25729/ESI.2023.32.4.001.
16. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye neyronnye seti glubokogo obucheniya. Chast' 2. Metody obucheniya bystrykh neyronnykh setey [Fast transformations and self-similar neural networks of deep learning. Part 2. Methods of training fast neural networks]. *Informacionnye i matematicheskie tehnologii v nauke i upravlenii* [Information and mathematical technologies in science and management], 2024, no. 1(33), pp. 5-19. DOI:10.25729/ESI.2024.33.1.001.
17. Dorogov A.Yu. Bystrye neyronnye seti glubokogo obucheniya [Fast deep learning neural networks]. III Mezhdunarodnaya nauchnaya konferentsiya po problemam upravleniya v tekhnicheskikh sistemakh (STS'2019). Sbornik dokladov. Sankt-Peterburg. SPb.: SPbGETU "LETI" [Proceedings of 2019 3rd international conference on control in technical systems [CTS 2019. Saint Petersburg, LETI], 2019, pp. 275-280.
18. Voevodin V.V., Kuznetsov Yu.A. Matritsy i vychisleniya [Matrices and calculations]. M., Nauka [Moscow, Science], 1984, p.320.
19. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye neyronnye seti glubokogo obucheniya Chast' 3. Piramidal'nye neyronnye seti s glubokoy stepen'yu obucheniya [Fast transformations and self-similar deep learning neural networks Part 3. Pyramid neural networks with a deep learning degree]. *Informacionnye i matematicheskie tehnologii v nauke i upravlenii* [Information and mathematical technologies in science and management], 2024, no. 3(35), pp. 33–43, DOI:10.25729/ESI.2024.35.3.003.

**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](mailto:vaksa2006@yandex.ru), St. Petersburg, st. Popova, 5.

Статья поступила в редакцию 12.05.2024; одобрена после рецензирования 21.11.2024; принята к публикации 17.12.2024.

The article was submitted 05/12/2024; approved after reviewing 11/21/2024; accepted for publication 12/17/2024.