# Память образов на пирамидальных нейронных сетях быстрого обучения

А. Ю. Дорогов
*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)*
vaksa2006@yandex.ru

***Аннотация.*** **Рассматривается применение обратно-ориентированных пирамидальных нейронных сетей быстрого обучения для реализации элементов памяти образов. Сети рассматриваемого класса представимы линейными операторами, имеют самоподобную структуру и являются частным случаем алгоритма быстрого преобразования Фурье. Приведены методы топологического построения пирамидальных сетей. Показано что пирамидальная сеть памяти обеспечивает хранение и восстановление образов подобно хранению чисел в компьютерной памяти произвольного доступа. Доказано, что пирамидальные сети относятся к категории сетей с глубокой степенью обучения.**

***Ключевые слова: быстрая нейронная сеть; память образов; пластичность; степени свободы***

## I. Введение

К нейронным сетям, реализующим память, относится большой класс нейронных сетей с обратными связями, включающий в себя сети Элмана [1], Хопфилда [2], Хэмминга [3], АРТ [4], и другие сети рекуррентного типа. Сети данного класса решают задачи восстановления искажённых образов, ассоциативной памяти, кратковременной динамической памяти и другие подобные задачи в контексте распознавания образов, задача сохранения и точного восстановления образа при этом не ставится.

В данной работе будут рассмотрены пирамидальные с сети с регулярной самоподобной структурой. Будет показано, что сети данного типа могут быть эффективно использованы для хранения и точного восстановления образов. Пирамидальные сети имеют глубокую связь с алгоритмом быстрого преобразования Фурье (БПФ), для обозначения нового класса сетей используется термин быстрые нейронные сети (БНС) [5].

Для БНС можно предложить быстрые алгоритмы обучения, абсолютно сходящиеся за конечное число шагов. В основе алгоритмов обучения БНС лежит доказанное свойство структурной фрактальности БПФ, которое можно выразить системным инвариантом морфологического уровня [6]. Идея метода обучения БНС к одной или нескольким функциям образов основана на факторизации каждой функции заданного набора в предфрактальное произведение, соответствующее мультипликативной форме представления элементов матрицы быстрого преобразования.

## II. Построение пирамидальной сети памяти

Рассмотрим для определённости БНС в топологии «Кули-Тьюки с прореживанием во времени». В [7] показано, что топологическая модель сети данной топологии описывается символьными кортежами:

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

Здесь $m$ – номер слоя, $z^m$ – номер нейронного ядра в слое в слое $m$, локальные переменные $u_m, v_m$ определяют позиционные номера рецепторов и аксонов нейронных ядер. Кортежи используются для поразрядного представления чисел в позиционной системе счисления. Например, для системы счисления с основанием 2 имеем:

$$u = \langle u_{n-1}u_{n-2}\ldots u_1 u_0\rangle = u_{n-1}2^{n-1} + u_{n-2}2^{n-2} + \ldots + u_1 2 + u_0 .$$

Нумерация разрядных переменных начинается с нулевого значения, то же самое относится и к нумерации слоёв сети. В общем случае используется многоосновная система счисления, где разрядам $u_i$ соответствуют основания $p_i$, а разрядам $v_j$ основания $g_j$. Значения оснований определяют структурные характеристики БНС.

Нейронное ядро в слое $m$ представляет собой матрицу синаптических весов $W^m_{z^m} = \left\| w^m_{z^m}(u_m, v_m) \right\|$ размерностью $p_m \times g_m$. В [7] показано, что для БНС элементы матрицы преобразования выражаются через элементы матриц нейронных ядер:

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

Для построения пирамидальной сети выберем структурные характеристики следующим образом:

- размерности рецепторных полей сети положим равными $p_0 = p_1 = \cdots = p_{n-2} = 1, \quad p_{n-1} \neq 1$. В этом случае размерность сети по входу будет равна $N = p_{n-1}$;
- размерности аксоновых полей сети зададим произвольными положительными целыми числами $g_0, g_1, \cdots, g_{n-1}$. Размерность сети по выходу в этом случае будет равна $M = g_0 g_1 \cdots g_{n-1}$.

Топологическая модель при данных структурных характеристиках будет иметь вид:

$$U^m = \langle u_{n-1}0_{n-2}\cdots 0_{m+1}0_m v_{m-1}v_{m-2}\cdots v_1 v_0\rangle,$$
$$V^m = \langle u_{n-1}0_{n-2}\cdots 0_{m+1}v_m v_{m-1}v_{m-2}\cdots v_1 v_0\rangle,$$
$$z^m = \langle u_{n-1}0_{n-2}\cdots 0_{m+1}v_{m-1}v_{m-2}\cdots v_1 v_0\rangle.$$

На рис. 1 показана топология сети для структурных характеристик $[p_0 p_1 p_2 p_3] = [1112]$, $[g_0 g_1 g_2 g_3] = [2222]$. Детали построения графа по топологической модели изложены в работе [7]. Пирамида расширяется от входа сети к выходу, поэтому в названии используется термин обратно-ориентированная пирамида.

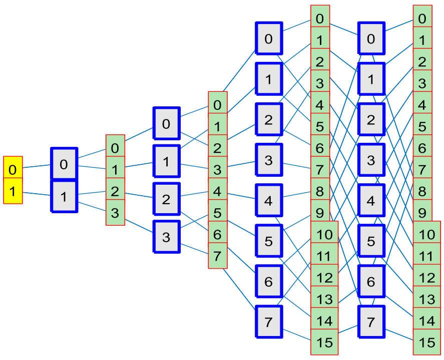

Рис. 1. Обратно-ориентированная пирамидальная сеть для реализации памяти

## III. Факторизация образов

Будем полагать, что образ, задан функцией $f(v)$ на дискретном интервале длиной $M = g_0 g_1 \dots g_{n-1}$, где $g_m$ – произвольные целые числа. Представим аргумент функции в позиционной многоосновной системе счисления с основаниями $g_0 g_1 \dots g_{n-1}$. Формула перехода, как известно, имеет вид:

$$v = \langle v_{n-1}v_{n-2}\dots v_0\rangle = v_{n-1}g_{n-2}g_{n-3}\dots g_0 + v_{n-2}g_{n-3}g_{n-4}\dots g_0 + \dots$$
$$\dots + v_1 g_0 + v_0.$$

где $v_i \in [0,1,\dots g_i - 1]$ – разрядные переменные. В результате данного преобразования образ представляется как многомерная функция $f\langle v_{n-1}v_{n-2}\dots v_0\rangle$. Каждый аргумент функции определяет её некоторый масштабный срез. Зафиксируем все аргументы функции кроме $v_m$. Варьируя свободный аргумент $v_m$, получим выборку $S_m$ (с числом элементов $g_m$). Фрактальным фильтром [7] частотной локализации $m$ называется произвольный функционал $F(S_m)$, определённый на выборке $S_m$. Операцию фрактальной фильтрации можно записать в виде:

$$f_{out}\langle v_{n-1}v_{n-2}\dots v_{m+1}v_{m-1}\dots v_0\rangle = \underset{v_m}{F}\left(f_{inp}\langle v_{n-1}v_{n-2}\dots v_0\rangle\right).$$

В простейшем варианте фрактальный фильтр выполняет, например, суммирование значений функции по аргументу $v_m$. Фрактальная фильтрация по аргументу $v_m$ приводит к сокращению интервала определения функции в $g_m$ раз. Рассмотрим цепочку фрактальных фильтров, показанную на рис. 2.

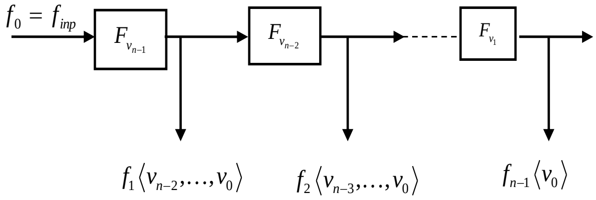


Рис. 2. Цепочка фрактальных фильтров

В этой цепочке фрактальная фильтрация образа начинается со старшего разряда $v_{n-1}$. Выходные сигналы фильтров цепочки определяются рекуррентным соотношением:

$$f_{m+1}\langle v_{n-m-2}\dots v_0\rangle = \underset{v_{n-m-1}}{F}\left(f_m\langle v_{n-m-1}\dots v_1 v_0\rangle\right).$$

Для цепочки фильтров рекуррентно введём функции:

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

где $i^m = \langle v_{m-1}v_{m-2}\dots v_1 v_0\rangle$. Используя определение функций, можно записать.

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

## IV. Сохранение образов в нейросетевой памяти

Пусть запоминаемые образы $y^k(V)$ представляет собой набор, состоящий из $p_{n-1}$ дискретных функций, заданных на интервале длинной $M$, где $k$ – порядковый номер функции. Выполним мультипликативную декомпозицию каждой функции по переменным $v_i$, начиная со старшего разряда, в результате получим:

$$y^k(V) = \varphi^k_{i^{n-1}}(v_{n-1})\varphi^k_{i^{n-2}}(v_{n-2})\cdots\varphi^k_{i^0}(v_0), \qquad (2)$$

где $i^m = \langle v_{m-1}v_{m-2}\cdots v_1 v_0\rangle$. Сравнивая выражения (1) и (2) получим правило обучения нейронной сети для сохранения образов:

$$w^{n-1}_{z^{n-1}}(u_{n-1}, v_{n-1}) = \varphi^k_{i^{n-1}}(v_{n-1}) \quad \textit{для } m = n-1,$$
$$w^m_{z^m}(0_m, v_m) = \varphi^k_{i^m}(v_m) \quad \textit{для } m < n-1,$$
$$z^m = \langle u_{n-1}0_{n-2}\cdots 0_{m+1}v_{m-1}v_{m-2}\cdots v_1 v_0\rangle,$$
$$i^m = \langle v_{m-1}v_{m-2}\cdots v_1 v_0\rangle.$$

Предварительно, для упорядочивания хранимых функций, должно быть установлено взаимно-однозначное соответствие $k \Leftrightarrow u_{n-1}$ между порядковым номером функции и разрядной переменной $u_{n-1}$. Считывание памяти реализуется установкой на входе нейронной сети унитарного кода, в котором только один из разрядов равен 1, а остальные – нулю. Для сети, показанной на рис. 1 это будут комбинации $[0,1]$ или

$[1,0]$. При использовании кодов $[1,1]$ хранимые образы складываются, а при кодах $[1,-1]$ из первого образа вычитается второй. На рис. 3 показаны результаты считывания образов БНС-памяти, представляющих собой дискретные функции синуса, отличающиеся по частоте в два раза.

## V. Пластичность пирамидальной нейронной сети

В исследованиях нейробиологов термин «пластичность» используется, как качественная характеристика способности нейронной сети обучаться при воздействии внешних факторов. Для искусственных нейронных сетей адекватной количественной оценкой может служить число независимых настроек, существующих в сети. Это значение, как правило, меньше полного количества синаптических весов (исключением является однослойный персептрон, для которого соблюдается равенство).

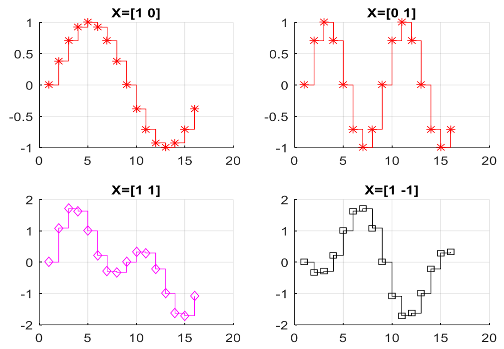


Рис. 3. Результаты считывания памяти, реализованной на пирамидальной БНС

В механике для оценки числа независимых координат используется понятие «число степеней свободы». Близкую аналогию можно провести и для нейронных сетей. В самом деле, нейронную сеть прямого распространения можно представить, как нелинейный оператор, преобразующий входной вектор в выходной. Полное множество операторов образует многообразие в многомерном пространстве, в котором каждый оператор сети можно рассматривать как некоторую материальную точку. Изменение синаптических весов нейронной сети приводит к перемещению точки-оператора в пространстве операторов. Следуя далее механической аналогии, будем называть число независимых координат, необходимое и достаточное для однозначного определения местоположения точки-оператора в пространстве операторов, числом степеней свободы нейронной сети и рассматривать эту величину как количественную оценку пластичности сети.

Расчёт степени пластичности БНС можно выполнить непосредственно по структурной модели сети. На рис. 4 представлена структурная модель для сети с топологией, показанной на рис. 1. Каждая вершина структурной модели является нейронным ядром, пара цифр, записанных в вершине, определяют размерность рецепторного и аксонового поля нейронного ядра, все связи между вершинами имеют ранги равные единице. В [7] получена формула расчёта числа степеней свободы БНС, которая имеет вид:

$$S(H)=\sum_{m=0}^{n-1} p_m g_m q_m - \sum_{m=0}^{n-2} D_m\,, \qquad (3)$$

где $p_m, g_m$ размерности рецепторного и аксонового поля ядра в слое $m$, $q_m$ – число ядер в слое, $D_m$ – количество одноранговых связей в межслойном переходе с номером $m$. Используя (3) получим: $S(H)=(1\cdot 2)\cdot 14+(2\cdot 2)\cdot 8-28=32$. Данная нейронная сеть обеспечивает хранение двух произвольных дискретных функций заданных на интервале длиной $M=16$. Для задания двух функций требуется определить 32 значения. Таким образом, сеть полностью использует свой потенциал для хранения функций и в этом смысле относится к категории сетей с глубокой степенью обучения.

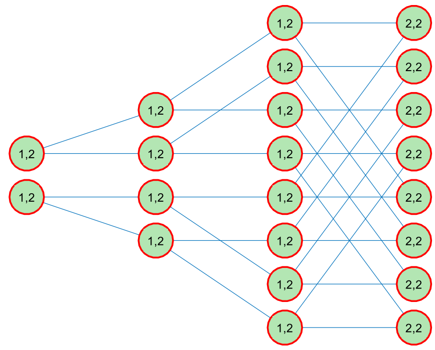


Рис. 4. Структурная модель обратно ориентированной пирамидальной нейронной сети

## VI. Банк памяти на обратно ориентированной сети с коммутацией плоскостей

Дополним топологическую модель Кули-Тьюки «с прореживанием по времени» дополнительными плоскостями нейронных ядер, номера которых определим кортежем $\pi_m$:

$$\begin{aligned} U^m &= \langle u_{n-1}u_{n-2}\cdots u_{m+1}u_m v_m v_{m-1} v_{m-2}\cdots v_1 v_0\rangle, \\ V^m &= \langle u_{n-1}u_{n-2}\cdots u_{m+1} v_m v_{m-1} v_{m-2}\cdots v_1 v_0\rangle, \\ z^m &= \langle u_{n-1}u_{n-2}\cdots u_{m+1} v_{m-1} v_{m-2}\cdots v_1 v_0\rangle, \\ \pi_m &= \langle u_{m-1}u_{m-2}\cdots u_1 u_0\rangle. \end{aligned}$$

Максимальное количество дополнительных плоскостей появится в последнем слое. Номер плоскости в этом слое будет определяться кортежем $\pi_{n-1}=\langle u_{n-2}u_{n-3}\cdots u_1 u_0\rangle$, а число плоскостей будет равно произведению оснований: $p_{n-2}p_{n-3}\cdots p_1 p_0$. По мере движения к входному слою число дополнительных плоскостей будет уменьшаться и для нулевого слоя имеем: $\pi_0=\langle\ \rangle$, т. е. их не будет совсем, останется только одна плоскость базовой топологии. Таким образом, в новой топологии плоскость нулевого слоя останется прежней, а в старших слоях появятся дополнительные плоскости. На рис. 5 показана топологическая модель сети памяти со структурными характеристиками: $[p_0 p_1 p_2 p_3]=[2222]$, $[g_0 g_1 g_2 g_3]=[2222]$.

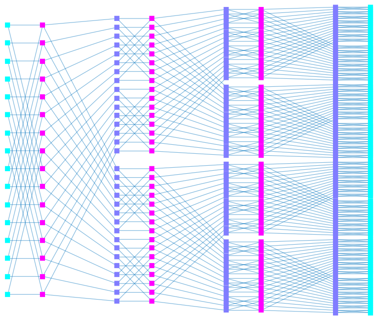

Рис. 5. Топологическая модель нейронной сети банка памяти

Будем полагать, что запоминаемые образы представлены в мультипликативной форме (2). В качестве образов могут использоваться произвольные функции, заданные на дискретном интервале длиной $M = g_0 g_1 \cdots g_{n-1}$. Запись образа в память, соответствует обучению нейронной сети. Для настройки нейронных ядер можно использовать базовое правило, вытекающее из обобщённой теоремы факторизации БНС, дополнив его номером плоскости:

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

здесь $k$ – порядковый номер функции, $V = \langle v_{n-1} v_{n-2} \cdots v_0 \rangle$ – точка выходной плоскости, $z^m = \langle u_{n-1} u_{n-2} \cdots u_{m+1} v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle$ – номер настраиваемых ядер по слоям, $\pi_m = \langle u_{m-1} u_{m-2} \cdots u_1 u_0 \rangle$ – номер плоскости размещения ядер. Для $m = n-1$ имеем $z^{n-1} = i^{n-1} = \langle v_{n-2} v_{n-3} \cdots v_1 v_0 \rangle$, а варьируемыми переменными в левой части являются номер плоскости $\pi_{n-1} = \langle u_{n-2} u_{n-3} \cdots u_1 u_0 \rangle$ и разряд $u_{n-1}$. Вместе они покрывают весь диапазон координат входной плоскости. Этому диапазону отвечают возможные значения индекса $k$ в правой части, отсюда следует, что каждая точка входной плоскости является активирующим адресом для хранимого образа. Т. е. построенная сеть обладает максимально возможной памятью, с адресным пространством, покрывающим всю входную плоскость, и, следовательно, является сетью памяти с глубокой степенью обучения.

К этому результату можно прийти, с другой стороны. С каждой выходной плоскостью $\pi_{n-1} = \langle u_{n-2} u_{n-3} \cdots u_1 u_0 \rangle$ связаны область ответственности во входной плоскости, определяемая схемой $U^0 = \langle u_{n-1} u_{n-2} \cdots u_1 u_0 \rangle$. Размер области определяется основанием разрядной переменной $u_{n-1}$. Нетрудно видеть, что выделенный сегмент вход-выход представляет собой пирамидальную обратно-ориентированную сеть памяти. Таким образом, вся сеть с коммутацией плоскостей представляет собой лес непересекающихся пирамидальных сетей памяти, упакованных в матричные структуры. Как было показано ранее пирамидальная сеть является сетью глубокого обучения, а значит и вся сеть с коммутацией плоскостей также является сетью глубокого обучения. Считывание памяти реализуется установкой на входе нейронной сети унитарного кода в области ответственности, каждой выходной плоскости. Хранимые образы воспроизводятся в выходных плоскостях сети. В любой выходной плоскости может воспроизводиться $p_{n-1}$ образов, независимо от воспроизведения хранимых образов в других областях. Сеть памяти, показанная на рис. 5 имеет 8 выходных плоскостей, с каждой плоскости можно считывать две функции. Таким образом, нейронная сеть обеспечивает хранение 16 произвольных функций заданных на интервале длинной $M = 16$. Данная сеть имеет 256 степеней свободы.

При программной реализации нейронной сети и ограничении числа одновременно отображаемых образов до одного, можно реализовать считывание всех образов с одного выхода. Это достигается за счёт неявного переключения плоскостей по значению адреса считывания $U$. Адрес считывания в поразрядном представлении имеет вид $U = \langle u_{n-1} u_{n-2} \cdots u_1 u_0 \rangle$, поэтому при считывании образа по данному адресу должны быть выбраны плоскости $\pi_m = \langle u_{m-1} u_{m-2} \cdots u_1 u_0 \rangle$, где $m$ – номер слоя. Такой приём выделяет из сети только те нейронные цепи, которые ответственны за хранение данного образа, сокращая до минимума число вычислительных операций при его восстановлении. Альтернативно, образы можно считывать параллельно и независимо с каждой выходной плоскости.

## VII. Заключение

БНС являются частным случаем алгоритма быстрого преобразования Фурье (БПФ). Топологическая модель БПФ описывается символьными кортежами, которые позволяют реализовать выбор произвольных размерностей для базовых операций (нейронных ядер в терминах БНС). Это позволяет реализовать богатое разнообразие БНС для одной и той же топологической схемы. Среди возможных вариантов БНС, пирамидальная сеть отличается полным использованием степеней свободы, что делает её эффективной в системах машинного обучения.

В памяти, реализованной на основе пирамидальных БНС, могут храниться двумерные и многомерные образы. В отличие от кристаллов цифровой памяти с последовательным хранением данных, где восстановление образа происходит за счет последовательного опроса ячеек, в памяти на БНС все пикселы образа восстанавливаются одновременно, что потенциально обеспечивает сверхвысокое быстродействие для памяти образов.

## Список литературы


[1] Elman J.L. Finding structure in time. Cognitive Science. 1990. pp. 179–211.

[2] Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proceedings of National Academy of Sciences, vol. 79 no. 8 pp. 2554–2558, April 1982.

[3] Richard Lippmann. 1987. An introduction to computing with neural nets. IEEE Assp magazine.

[4] Carpenter, G.A. & Grossberg, S. (1987), ART 2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics volume 26, number 23, December, 1987.

[5] Дорогов А.Ю. Быстрые нейронные сети: Проектирование, настройка, приложения // Лекции по нейроинформатике Ч.1. В тр. школы-семинара «Современные проблемы нейроинформатики», науч.-техн. конф. "Нейроинформатика-2004" 28-30 января 2004г. Москва: МИФИ, 2004. С. 69-135.

[6] Дорогов А.Ю. Системные инварианты быстрых преобразований // Труды Четвертой межрегиональной школы семинара «Распределенные и кластерные вычисления», 14 –16 сентября 2004 года, г. Красноярск.

[7] Дорогов А.Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб.: «Политехника», 2014. 328с.