УДК 004.852

DOI: http://doi.org/10.25728/pu.2020.2.1

ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ: ЗАРОЖДЕНИЕ, СТАНОВЛЕНИЕ, СОВРЕМЕННОЕ СОСТОЯНИЕ

А.В. Макаренко

Аннотация. Рассмотрено эволюционное развитие искусственных нейронных сетей: от зарождения в виде нейрона Маккаллока — Питтса до современных глубоких архитектур. Перечислены основные «нейросетевые кризисы» и показаны причины их появления. Основное внимание уделено нейронным архитектурам, обучающимся в режиме «обучения с учителем» по размеченной выборке данных. Приведены ссылки на оригинальные работы и основополагающие математические теоремы, формирующие теоретический фундамент под направлением искусственных нейронных сетей. Проанализированы причины затруднений на пути к формированию эффективных глубоких нейронных архитектур, рассмотрены пути разрешения возникших трудностей, выделены обстоятельства, способствующие успеху. Перечислены основные слои сверточных и рекуррентных нейронных сетей, а также их архитектурные комбинации. Приведены примеры и ссылки на статьи, демонстрирующие эффективность глубоких нейронных сетей не только на данных, имеющих ярко выраженные структурные паттерны (изображения, голос, музыка и др.), но и на сигналах стохастического/хаотического характера. Выделено также одно из основных направлений развития сверточных нейросетей — внедрение в слои обучаемых интегральных преобразований. На базовом уровне рассмотрена современная архитектура «Трансформер» — мейнстрим в задачах обработки последовательностей (в том числе в компьютерной лингвистике). Приведена ключевая проблематика современной теории искусственных нейронных сетей.

Ключевые слова: глубокое обучение, сверточные нейронные сети, рекуррентные нейронные сети.

ВВЕДЕНИЕ

Вычислительный интеллект как одна из ветвей *искусственного интеллекта* опирается на эвристические алгоритмы; в качестве основного математического инструментария применяется *машинное обучение по прецедентам*. Оно основано на выявлении общих закономерностей по частным эмпирическим (экспериментальным) данным и по факту относится к классу *индуктивного обучения*. Формально, задача машинного обучения ставится в следующем общем виде.

Дано: X — описания объектов (характеристики, признаки; англ.: *features*); R — решения алгоритма (ответы, метки; англ.: *patterns, labels*).

Существует, но неизвестна, целевая функция (англ.: *target function*):

$$G': X \to R.$$

На основе анализа набора логических пар $\mathbf{d}_n^* = (\mathbf{x}_n, \mathbf{r}_n)$, где $\mathbf{d}_n^*$ составляет n-й прецедент, необходимо найти алгоритм (решающую функцию; англ.: *decision function*): $G: X \to R$, которая восстанавливает оценку G'.

Минимально выделяют два подмножества прецедентов — *обучающую* (англ.: *train set*) $D^{\mathrm{Tr}} = \{\mathbf{d}_n^*\}_{n=1}^{N_{\mathrm{Tr}}}$ и *тестовую* (англ.: *train set*) $D^{\mathrm{Ts}} = \{\mathbf{d}_n^*\}_{n=1}^{N_{\mathrm{Ts}}}$ *выборки*.

Отметим важное требование — исключение «*протечек данных*» (англ.: *leaked data*): $D^{\mathrm{Tr}} \cap D^{\mathrm{Ts}} \equiv \varnothing$.

Введем в рассмотрение алгоритм $G_i: X \times W_i \to R$, где W_i — множество допустимых значений $\mathbf{w}$ — вектора параметров алгоритма. В этом случае выделяют два основных типа обучения:

параметрический — при фиксированном алгоритме G_i ищется «оптимальное» значение $\tilde{\mathbf{w}}$,

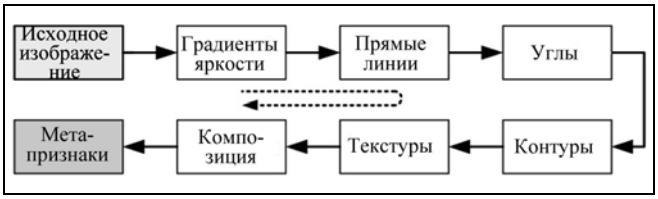

Рис. 1. Иерархические абстракции в данных

доставляющее минимум функционала ошибки $L[G_i(X, \tilde{\mathbf{w}}), R] \to \min$;

структурный — в этом случае вначале осуществляется поиск «оптимального» представления G_i, а затем «оптимального» значения $\tilde{\mathbf{w}}$.

Выделяют три основных типа (режима) обучения:

с учителем — $D^{\mathrm{Tr}} \neq \varnothing$ (это режим обучения по размеченной выборке);

без учителя — $D^{\mathrm{Tr}} \equiv \varnothing$ (в этом режиме, как правило, решаются задачи кластеризации или понижения размерности набора данных X);

с подкреплением (англ.: *reinforcement learning*) — осуществляется поисковое взаимодействие обучаемого агента с внешней средой, обучение управляется системой поощрений и штрафов [1].

Из приведенной формулировки задачи фактически следует, что алгоритм G_i, в зависимости от постановки задачи, может решать различные задачи из области управления: оценивания и прогнозирования процессов, идентификации систем и собственно управления.

Из числа подходов машинного обучения выделяют обширный класс методов *глубокого обучения* (англ.: *Deep Learning*), которые моделируют иерархические абстракции в данных, применяя архитектуры, состоящие из каскадного множества нелинейных преобразований (фильтров). Пример иерархических абстракций в данных (при распознавании изображений) приведен на рис. 1: пунктирная стрелка означает, что метапризнаки того или иного изображения помимо композиции (сцены) включают в себя также и все нижележащие (простые) иерархии, как то: градиенты яркости, прямые линии, углы, контуры, текстуры.

Архитектура, состоящая из каскадного множества нелинейных преобразований (фильтров), в общем виде показана на рис. 2.

Уникальная особенность глубокого обучения заключается в том, что соответствующие алгоритмы работают с исходными данными (низкоуровневыми признаками) и самостоятельно извлекают (формируют) высокоуровневое признаковое описание объектов; т. е. речь идет о *метаобучении* — компьютерная программа самостоятельно учится, как лучше ей учиться. Сравнительные отличия с классическими *статистическими методами* и «плоским» машинным обучением дает диаграмма, представленная на рис. 3.

Мейнстримом технической реализации концепции «Глубокое обучение» в настоящий момент являются глубокие *искусственные нейронные сети* (ИНС) [2]. Важно понимать, что глубокое обучение существенно шире по своей сути, нежели ИНС, и включает в себя также исследования по *глубоким случайным лесам* (англ.: *Deep Random Forest*), по *глубоким байесовым сетям* (англ.: *Deep Bayesian Networks*) и некоторым другим подходам (в том числе и по исторически обусловленному *логическому интеллекту*).

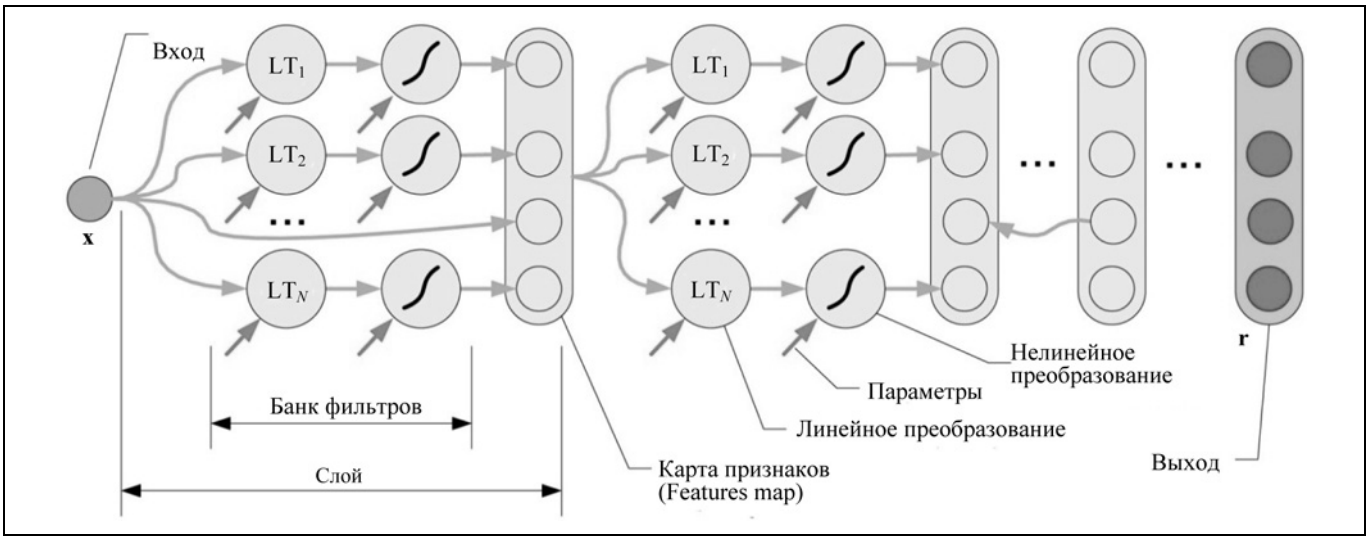

Рис. 2. Обобщенная архитектура алгоритма глубокого обучения на основе каскадного множества нелинейных преобразований

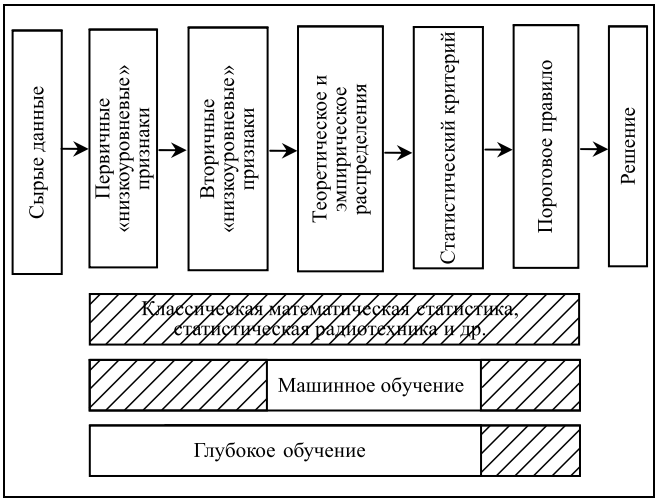

Рис. 3. Кодировка операций: заштрихованные — выполняются человеком; без штриховки — автоматически, в рамках обучения модели

Одна из особенностей глубоких нейронных сетей заключается в возможности реализации ими существенно адаптивного (в какой-то мере даже сверхадаптивного) управления.

Как будет показано далее, весь объем достижений глубоких нейросетей, с которыми читатель, возможно, сталкивается в повседневной жизни (распознавание номеров автомобилей, перевод текстов, распознавание слитной речи, синтез голоса и др.), объясняется хитроумными комбинациям всего трех типов слоев искусственных нейронов.

Одна из целей настоящего обзора состоит в развенчивании широко распространенного мифа из мира ИНС: нейросети — это какая-то «магия» и сплошная «кустарщина», и «наука не понимает, как это все работает». Также будет показано, что ошибочен стереотип, что глубокие нейронные сети эффективно функционируют только на данных, имеющих ярко выраженные структурные паттерны (изображения, голос, музыка и др.) и не работают со случайными и/или хаотическими процессами.

Для формирования у читателя цельной картины эволюции классических нейросетей в глубокие, далее приводится краткая хронология основных событий: с момента зарождения этого научного направления и до настоящего времени.

1. ЭВОЛЮЦИЯ ПОЛНОСВЯЗНЫХ НЕЙРОСЕТЕЙ ПРЯМОГО РАСПРОСТРАНЕНИЯ

Официально старт нейросетевому направлению работ был дан в 1943 г. в статье У. Маккалока и У. Питтса [3]. Авторы ввели понятие *искусственной нейронной сети* (ИНС) и предложили формальную модель *искусственного нейрона*:

$$s = \mathbf{w} \cdot \mathbf{x} + b, \quad z = g(s), \tag{1}$$

где $\mathbf{x}$ — вектор *входных данных*, $\mathbf{x} \in \mathbb{Z}^N_{\{0,1\}}$; $\mathbf{w}$ — вектор *весов*; b — смещение; «$\cdot$» — операция скалярного умножения; $g(\cdot)$ — *функция активации*; z — *выход*. Отметим, что исходно нейрон оперировал только двухуровневыми сигналами: $x_i = 0$ — логический нуль и $x_i = 1$ — логическая единица, а функция активации строилась по типу пороговой функции Хевисайда:

$$z = g_{01}(s) = \begin{cases} 1, \text{ если } s > a, \\ 0, \text{ если } s \le a, \end{cases}$$

где $a > 0$ — *порог активации*.

При формировании ИНС отдельные нейроны (1) объединяются в *нейросетевой слой*:

$$\mathbf{s} = \mathbf{W}\mathbf{x} + \mathbf{b}, \quad \mathbf{z} = g_{01}(\mathbf{s}),$$

где $\mathbf{W}$ — матрица весов, в общем случае прямоугольная.

В 1949 г. Д. Хебб в книге [4] изложил некоторые гипотезы относительно того, как нейроны человеческого мозга могут обучаться. Одна из основных концепций: обучение происходит в результате усиления связи (*синаптического веса*) между одновременно активными нейронами[1]. Исходя из этого, часто используемые связи усиливаются, что объясняет феномен обучения путем многократного повторения одних и тех же входных стимулов (см. также обучение с подкреплением [1]).

В 1958 г. Ф. Розенблатт изобретает *перцептрон* с одним *скрытым слоем* [5]:

$$\mathbf{s}_1 = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1, \quad \mathbf{y} = g_{01}(\mathbf{s}_1),$$

$$\mathbf{s}_2 = \mathbf{W}_2\mathbf{x} + \mathbf{b}_2, \quad \mathbf{z} = g_{11}(\mathbf{s}_2),$$

где $z = g_{11}(s) = \text{sign}s$, $\mathbf{x} \in \mathbb{Z}^N_{\{0,1\}}$.

Отметим, что это первая ИНС, которая умела решать задачу классификации и активно применялась на практике.

Особенность данной сети состоит в необучаемости скрытого слоя (в терминологии Ф. Розенблатта [5] он именуется А-слоем) — элементы $\mathbf{W}_1$ и $\mathbf{b}_1$ исходно принимают случайные фиксированные значения $\{-1, 0, 1\}$, также фиксируется и пороги a в функции g_{01}. Как было позже показано, смысл этого слоя заключается в приведении несепара-

[1] Впоследствии этот «алгоритм» стал называться «правилом Хебба».

бельной задачи (линейно неразделимой) к сепарабельной (линейно разделимой). Здесь отчасти работает

Теорема 1 (Т. Ковер, 1965 г. [6]). *Нелинейное проецирование в пространство более высокой размерности заданного набора данных, не являющихся сепарабельными, повышает вероятность их линейной разделимости.*

Второй слой (в терминологии Ф. Розенблатта [5] он именуется R-слоем) обучается по *методу коррекции ошибки* [7] — формализованному правилу Хебба — по выходу нейросети $\mathbf{z}$.

В 1960 г. Б. Уидроу и М. Хофф для обучения однослойной сети вида (1) предложили так называемое *дельта-правило* [8] (метод обучения ИНС градиентным спуском по поверхности ошибки) и назвали получившуюся систему ADALINE[2]. Данная ИНС сразу же начала применяться для решения задач *адаптивного управления*. С одной стороны, относительно перцептрона Розенблатта это был шаг назад (отсутствие скрытого слоя и невозможность решения несепарабельных задач). С другой — был применен новый метод обучения на основе минимизации функции стоимости (функционала потерь), который заложил основу для разработки более совершенных алгоритмов машинного обучения[3] и собственно алгоритмов обучения ИНС. Ключевой момент разработанного дельта-правила заключается в вычислении ошибки модели и формировании корректирующих обновлений весов не по дискретному выходу нейросети $\mathbf{z}$, а по непрерывнозначному выходу сумматора $\mathbf{s}$ на основе квадратичной *функции потерь*:

$$L(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N_{\mathrm{Tr}}}(\mathbf{r}_n - \mathbf{s}_n)^2 \to \min_{\mathbf{w}}, \qquad (2)$$

где $\mathbf{r}_n$ — истинное значение n-го обучающего прецедента, N_{Tr} — размер обучающей выборки. В результате сеть ADALINE стало возможно обучать высокоэффективным методом градиентного спуска:

$$\Delta\mathbf{w} = -\eta\nabla L(\mathbf{w}), \qquad (3)$$

где $\Delta\mathbf{w}$ — обновление весов сети (1), $\nabla L(\mathbf{w})$ — градиент функции потерь, η — темп обучения. Как можно заметить из выражения (2), обновление весов вычисляется по всем прецедентам из обучающей выборки (вместо инкрементного обновления веса после каждого образца), поэтому такой подход получил название «пакетный» (англ.: *batch*) градиентный спуск.

В 1969 г. М. Минский и С. Паперт опубликовали книгу [9], в которой содержался целый ряд критических замечаний о функциональных ограничениях перцептронов Розенблатта, тем самым вызвав существенное снижение интереса к тематике ИНС[4]. Отметим, что анализ был сделан для так называемого *элементарного перцептрона*, но название книги и формулировка выводов вызвали у читателей ощущение, что проблемы касаются всего направления ИНС[5]. Началась первая «*нейросетевая зима*», переведя фокус исследований в искусственном интеллекте на символьно-логические системы.

В 1986 г. Д.Е. Румельхарт переоткрывает заново *многослойный персептрон* (англ.: *multilayer perceptron*, MLP) в виде [10]

$$\mathbf{s}_1 = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1, \quad \mathbf{y} = g_{\mathrm{o}}(\mathbf{s}_1),$$
$$\mathbf{s}_2 = \mathbf{W}_2\mathbf{x} + \mathbf{b}_2, \quad \mathbf{z} = g_{\mathrm{o}}(\mathbf{s}_2),$$

где $g_{\mathrm{o}}(s) = g_{\mathrm{sg}}(s) = \dfrac{1}{1+e^{-s}}$ — либо сигмоидальная функция, либо $g_{\mathrm{o}}(s) = g_{\mathrm{th}}(s) = \mathrm{th}s$ — гиперболический тангенс. При этом первый слой становится также обучаемым, а вход сети непрерывнозначным, $\mathbf{x} \in \mathbb{R}^N$. Сеть, как целое, учится по *методу обратного распространения ошибки*, который был впервые описан в 1974 г. в работах А.И. Галушкина [11] и П. Вербоса [12] и существенно развит в последующих работах [13, 14].

Отметим, что Д.Е. Румельхарт при публикации своих результатов по какой-то причине исказил определение перцептрона Розенблатта, представив его как ИНС без скрытого слоя, тем самым породив методическую ошибку[6], что перцептрон Розенблатта не способен решать ряд элементарных несепарабельных задач, например, вычислять булеву функцию XOR (исключающее «или»).

Тем не менее, работа [10] запустила вторую волну массового интереса к ИНС. В 1988 г. Д. Брумхед и Д. Лоу предложили *сеть радиально-базисных функций* (англ.: *Radial Basis Function Network*, RBF) [15]. Это MLP с одним скрытым слоем вида

$$y_m = \exp\left[-\beta\sum_{i=1}^{N}(x_i - r_{im})^2\right], \quad m = \overline{1, M}, \qquad (4)$$

[2] ADAptive LInear NEuron, адаптивный линейный нейрон.

[3] В том числе логистической регрессии, метода опорных векторов и целого семейства регрессионных моделей.

[4] Интересно, что М. Минский был сокурсником Ф. Розенблатта.

[5] Ради справедливости отметим, что в 1987 г. авторы выпустили третье издание книги, где многие критические замечания были учтены.

[6] Искусственное понятие «однослойный перцептрон» стало во главу целого ряда недоразумений, вошло в ряд монографий и учебников, в том числе и современных.

где M — число нейронов скрытого слоя, $\mathbf{r}_m$ — так называемый центральный вектор m-го скрытого нейрона (обучаемый параметр). Выходной слой имеет линейную (тождественную) функцию активации.

В 1989 г. были получены два важных результата. Прежде всего, доказана важная

Теорема 2 (Universal Approximation Theorem FFNN [16], G. Cybenko, 1989). *Искусственная нейронная сеть прямого распространения с одним скрытым слоем может аппроксимировать любую непрерывную функцию многих переменных с любой точностью, при условии, что сеть имеет в скрытом слое достаточное число нейронов N, имеющих сигмоидальную функцию активации* g_{sg}.

Теорема 2 является в определенном смысле специализированным аналогом теоремы А.Н. Колмогорова и В.И. Арнольда о представимости непрерывных функций нескольких переменных суперпозицией непрерывных функций одной переменной и существенно дополнила теорему о сходимости перцептрона [7]. В этом ключе стоит также отметить близкую теорему Хехт — Нильсена [17].

Далее, Дж. Бридли вводит в обиход машинного обучения функцию активации SoftMax [18]:

$$z_i = e^{s_i} / \sum_{i=1}^{M} e^{s_i}, \quad i = \overline{1, M}, \tag{5}$$

где M — число нейронов выходного слоя. Функция (5), в отличие от других «интуитивных» (но, как правило, некорректных) подходов позволила на строгом теоретическом уровне обоснования решать задачу многоклассовой классификации (в режиме «один из многих»). При обучении ИНС с выходным *SoftMax*-слоем, как правило, применяется функция потерь в виде *кросс-энтропии*:

$$L = -\frac{1}{N_{\mathrm{Tr}}} \sum_{n=1}^{N_{\mathrm{Tr}}} \sum_{i=1}^{M} (\mathbf{z}_n^*)_i \ln(\mathbf{z}_n)_i, \tag{6}$$

где N_{Tr} — размер обучающей выборки, M — число нейронов в выходном слое (число классов в решаемой задаче), $\mathbf{z}_n^*$ — вектор меток, ассоциированный с n-м прецедентом.

В 1991 г. К. Хорник обобщает теорему 2 на случай произвольных нелинейных активационных функций [19]. Становится ясно, что универсальные аппроксимационные свойства ИНС — это в большей мере свойство сетевой структуры.

Тем не менее, несмотря на успехи, исследователи очень скоро «упираются» в существенную ограниченность MLP с одним скрытым слоем — удается решать лишь ограниченное число практически важных задач. Такие насущные проблемы, как распознавание изображений, голоса, обработка текста — остаются за гранью приложимости MLP. Попытки добавления числа скрытых слоев не приносят успеха — сети не обучаются, одна из сильнейших проблем — затухание градиента (англ.: *Vanishing Gradients Problem*) $|\nabla L(\mathbf{w})| \to 0$ по мере продвижения обучающих сигналов ко входу сети. Как следствие, во второй половине 1990-х гг. начинается вторая затяжная «нейросетевая зима».

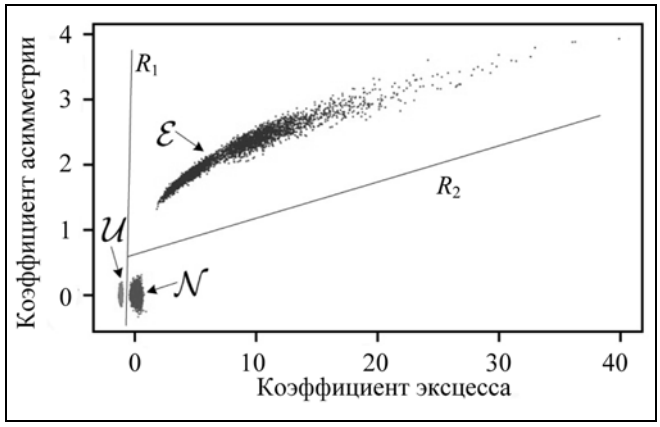

Рис. 4. Демонстрация линейной разделимости изучаемых сигналов x в пространстве третьего и четвертого статистических моментов

Продемонстрируем практическую ограниченность MLP при классификации случайных сигналов[7]. Рассмотрим

Пример 1. Введем в рассмотрение три класса стохастических сигналов **x**, различающихся функциями одноточечной плотности вероятности: $\mathcal{N}$ — нормальное; $\mathcal{U}$ — равномерное; $\mathcal{E}$ — экспоненциальное. Причем все сигналы δ-коррелированные и независимые. Поставим задачу синтеза ИНС по типу MLP для классификации входящих сигналов по принадлежности к одному из классов: $\mathcal{N}$, $\mathcal{U}$ или $\mathcal{E}$. При этом потребуем стандартизацию входящих сигналов: нулевое математическое ожидание и единичная дисперсия. Таким образом, классические энергетические обнаружители функционировать не будут, а проблема классификации сдвигается в область распознавания структурных характеристик случайных процессов. Для определенности положим длину каждого временного ряда в $K = 1024$ отсчета. Поставленная задача, как показано в работе [20], успешно решается MLP, если на вход ИНС подаются высокоуровневые информативные признаки — в данном случае статистические моменты (на рис. 4 приведена диаграмма рассеивания изучаемых сигналов в координатах третьего и четвертого статистических моментов). Если же на вход нейросети подать «сырые» сигналы **x**, то она полностью теряет способность к классификации сигналов: значение меры качества F_1 [2] не поднимается выше 0,417 [21].

[7] Задача по сути постановки близка к прикладной проблематике распознавания сигналов в пассивных акустических пеленгаторах.

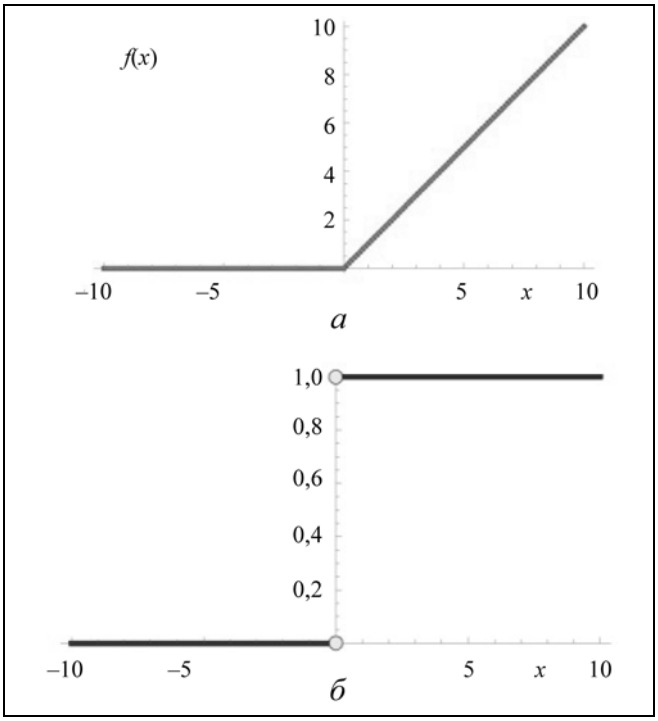

Рис. 5. График функции ReLU (*а*) и ее первой производной (*б*)

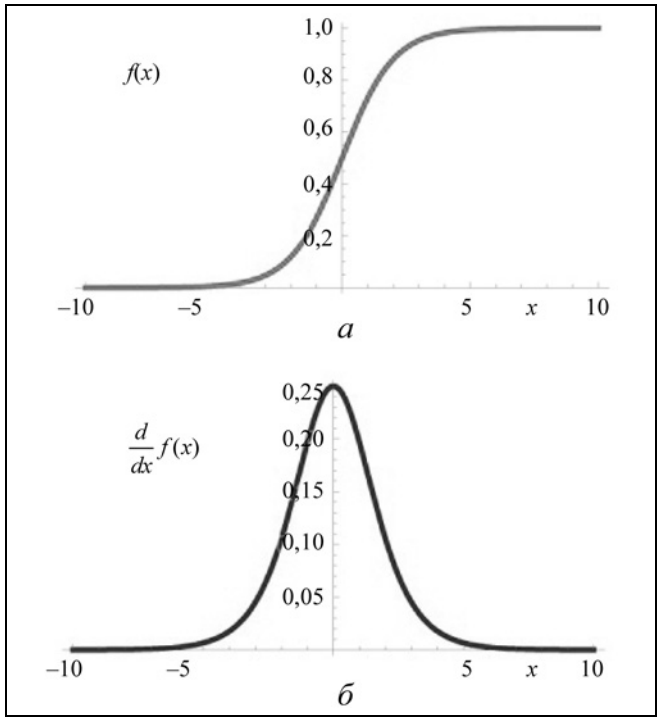

Рис. 6. График сигмоидальной функции (*а*) и ее первой производной (*б*)

Подобные результаты в свое время как раз и вызвали «вторую нейросетевую зиму» и ограничили применимость «плоских» нейросетей к ряду важных прикладных областей, как то: сверхширокополосная радиолокация, гидроакустическая шумопеленгация, инструментальная медицинская диагностика, техническая диагностика и др.

2. ЗАРОЖДЕНИЕ «ГЛУБИНЫ» ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Во второй половине «нулевых» годов (2006—2010 гг.) появляются работы, систематически направленные на разработку конструктивных методов обучения *многослойных нейронных сетей* (с числом скрытых слоев более одного). В 2006 г. Дж. Хинтон и Р. Салахутдинов предлагают *двухфазный подход* к обучению многослойных ИНС [22]. Первая фаза — послойное последовательное обучение *без учителя* скрытых слоев (начиная с первого) внутренним представлениям[8]. На второй фазе выходной слой обучается и скрытые слои дообучаются посредством метода обратного распространения ошибок. Способ оказался работоспособным, но весьма затратным в плане вычислительных ресурсов и, как оказалось в дальнейшем, весьма неустойчивым для сетей, имеющих более 3—5-ти скрытых слоев. Похожая идея тех же авторов на основе ограниченной *машины Больцмана* [2] и *сетей доверия* [2] страдала теми же недостатками. Как выяснилось в дальнейшем, все эти ухищрения — существенно избыточны.

Оказалось, что для решения проблемы обучения глубоких нейронных сетей как единого целого (обучение всех слоев сразу) необходимо было сделать два «простых» шага.

Прежде всего, потребовалось найти адекватную функцию активации, что и сделали Дж. Хинтон с соавтором, предложив в 2010 г. функцию ReLU[9] (Rectified Linear Unit) [23]: $g_{RL}(s) = \max(0, s)$. График этой функции приведен на рис. 5. Для сравнения на рис. 6 приведен график классической функции активации g_{sg}.

Из сравнения графиков функций g_{sg} и g_{RL} видно, что ReLU имеет широкий рабочий отрезок (область, в которой первая производная существенно отлична от нуля). Кроме того, ReLU очень «дешева» в вычислительном плане. Ее недифференцируемость в нуле, как показала практика, никак себя негативно не проявила.

[8] Фактически обучался энкодер в Автоэнкодере.

[9] Впоследствии появилось целое семейство ReLU-подобных функций [12].

Далее, потребовалось изменить схему *начальной инициализации весов* ИНС. Удачная конструкция получилась в том же 2010 г. у З. Глорота с соавтором [24]. Дисперсию инициализирующего шума (равномерного или нормального) было предложено находить по формуле

$$\mathrm{Var}(\mathbf{w}) = \frac{2}{N_{\text{in}} + N_{\text{out}}},$$

где N_{in} и N_{out} — число нейронов в предыдущем и последующем слоях соответственно.

К этому моменту уже более-менее выкристаллизовалось определение глубоких ИНС. К ним формально стали относить нейронные сети с числом скрытых слоев более одного (точнее, более двух — именно такие сети стали успешно извлекать сложные иерархические представления из «сырых» данных) и обучающихся как единое целое (обучение всех слоев сразу).

Таким образом, к концу первой декады XXI столетия все было готово для того, чтобы глубокие нейронные сети продемонстрировали прорывный результат. И его появление не заставило себя долго ждать.

3. ГЛУБОКИЕ СВЕРТОЧНЫЕ НЕЙРОСЕТИ

В 1989 г. Ян Лекун с соавторами публикуют работу [25], в которой описывают реальное приложение ИНС к практической задаче по распознаванию рукописных цифр в почтовом индексе. В статье рассматривается новая архитектура ИНС на основе принципа *разделения весов* (Weight Sharing). В данной работе фактически обобщен и переосмыслен ранний опыт по разработке К. Фукушимой *неокогнитрона* [26] и формализованы идеи *коннекционизма* М. Мозера [27]. К 1998 г. идеи Яна Лекуна окончательно вышлифовываются [28] в так называемые *сверточные нейросети*. Представленная в работе [28] архитектура сети LeNet-5 стала фундаментальной на многие последующие годы, особенно для задач анализа изображений. В сверточной нейросети применялась последовательная комбинация из двух типов слоев[10]. Первый тип — *свертка* (англ.: *Convolution Layer* [2]) — извлекает информативные признаки, имеющие структурную организацию (см. рис. 7, *а*). Второй тип — *субдескритизация* (англ.: *Pooling Layer* [2]) — благодаря пространственному сжатию данных обеспечивает инвариантность отклика слоя к малому смещению паттерна (рис. 7, *б*).

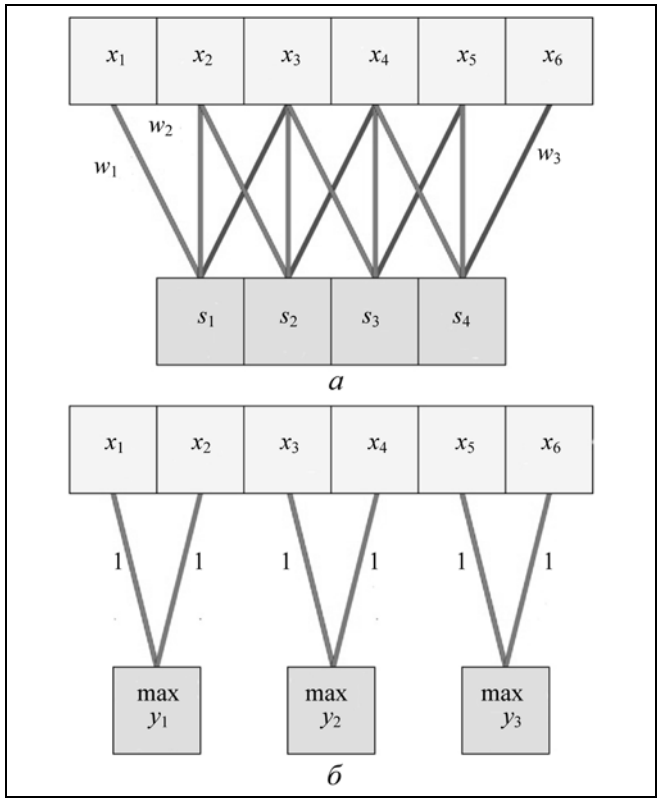

Рис. 7. Базовые слои сверточной нейросети: *а* — 1D-свертка, размер 3, шаг 1; *б* — слой MaxPool, размер 2, шаг 2

На выходе сети LeNet-5 применялся RBF-слой (4), а в качестве функции потерь при обучении применялась квадратичная функция потерь (2).

В связи с отсутствием достаточных по качеству и размеру наборов данных, а также из-за медленного обучения на центральном процессоре (CPU) с 1998 по 2010 г. сверточные нейросети пребывали в состоянии некоторой инкубации.

В 2010 г. были получены два результата, которые впоследствии оказали весьма существенное влияние на всю область глубоких нейросетей.

Прежде всего, Д. Кирешан и Й. Шмидхубер опубликовали одну из первых реализаций сверточной нейросети на графическом ускорителе (GPU) [29]. Реализация содержала 9 скрытых слоев и оба прохода — прямой (расчет) и обратный (обучение).

Далее, М. Цейлер с коллегами предложили новый нейросетевой слой, фактически обратный операции свертки [30], и назвали его «*слой деконволюции*[11]» (англ.: *Deconvolutional Network Layer*):

$$\sum_{m=1}^{M} z_m \oplus f_{m,\,c} = x_c, \qquad (7)$$

[10] Реализация слоев в сети LeNet-5 отличалась от принятой в настоящее время.

[11] Название слоя не совсем верное, поэтому в дальнейшем оно было изменено (примерно с 2015 г, см. далее).

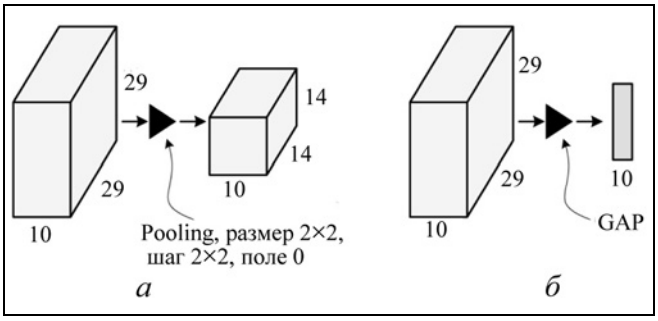

Рис. 8. Структуры слоев субдискретизации: *а* — обычный AveragePooling; *б* — GAP; числа на ребрах параллелепипедов — размеры входных полей и выходных карт-признаков

где x_c — входные данные (изображение); $\oplus$ — операция свертки; z_m — выход слоя (карты признаков, числом M); $f_{m,c}$ — ядра сверток (обучаемые послойно, без учителя, см. § 2), уникальные для каждого c — цветового канала изображения и для каждой карты признаков m. При этом, если входное изображение имеет размер $N_x \times N_y$, а ядро размером $N_k \times N_k$, то выход слоя имеет размер: $(N_k + N_k - 1) \times (N_y + N_k - 1)$.

В 2012 г. А. Крижевский[12] в соревновании по распознаванию изображений ImageNet[13] применил подход на основе глубоких нейронных сетей. Его сверточная сеть AlexNet победила с существенным отрывом от лучших решений, основанных на классических техниках компьютерного зрения и машинного обучения [31]. Эта работа фактически дала исходный толчок к буму Deep Learning, который мы наблюдаем и в настоящее время.

Отметим, что на выходе сети AlexNet стоял уже привычный слой SoftMax (5), в качестве функции потерь при обучении применялась кросс-энтропия (6), а основной функцией активации являлась функция ReLU.

В том же 2012 г. тот же Дж. Хинтон с коллегами ввел в рассмотрение технику *Dropout* [32] для борьбы с *переобучением*: на каждой итерации обучения часть нейронов скрытого слоя вместе с их входящими и исходящими весами исключается, а после завершения итерации — возвращается. После окончания обучения все веса умножаются на нормализующий коэффициент. Как впоследствии было показано, эта процедура эквивалентна порождению экспоненциально большого ансамбля ИНС и усреднения (*ансамблирования*) их решений, что усиливает инвариантность сети к ошибкам в данных.

В 2013 г. М. Лин с коллегами публикуют работу «*Network in Network*» («*Сеть внутри сети*») [33]. Статья содержала две ключевые идеи, которые впоследствии существенно развились. Во-первых, было предложено между слоями сверток вставлять многослойные персептроны, которые усиливали обобщающие свойства сверточных слоев (эта идея в 2014 г легла в основу архитектуры модулей Inception, см. далее). Во-вторых, был предложен новый слой: *глобальная усредняющая субдескритизация* (англ.: *Global Average Pooling*, GAP). Его структура приведена на рис. 8. Применение этого слоя позволило конструировать *полносверточные нейросети* (англ.: *Fully Convolutional Networks*) без полносвязных слоев в концевой части сети (эта идея в будущем привела к разработке целого направления: нейросетей, инвариантных к размеру входных данных, и позволяющих решать задачи локализации объектов и/или сегментации изображений, см. далее).

В 2014 г. К. Симонян и А. Зиссерман публикуют так называемую VGG-сеть [34]. Она содержала 19 скрытых обучаемых слоев. Ее архитектура шла вразрез с рекомендациями авторов ранних сетей: применять во входных слоях крупные свертки размером не менее 5×5 пикселей (LeNet-5) и 11×11 пикселей (AlexNet). Оказалось, что последовательность мелких сверточных ядер 3×3 эффективно эмулирует более крупные рецептивные поля (типа 9×9, 11×11) при явно меньшем числе настраиваемых параметров и с меньшим числом затратных операций умножения.

Но оказалось, что 3×3 — это не предел. Осенью того же 2014 г. Кристиан Жегеди с коллегами публикует так называемую GoogLeNet [35], включающую в свой состав модули Inception (рис. 9), в которых ключевую роль играют ядра размером 1×1. Эта работа во многом является творческим осмыслением ранее предложенного подхода Network-in-Network [33], согласно которому применяются свертки размером 1×1 (фактически пространственно ориентированные слои MLP) для увеличения комбинаторных свойств сверточных слоев.

Буквально одновременно с публикацией архитектуры Inception Л. Сифре[14] защищает кандидатскую диссертацию [36], в которой вводит в рассмотрение так называемый *Depthwise Separable* сверточный слой. Его архитектура приведена на рис. 10, *б*

[12] Кстати, аспирант Дж. Хинтона.

[13] ImageNet — база данных аннотированных изображений, предназначенная для отработки и тестирования алгоритмов распознавания образов и машинного зрения. Для категоризации объектов на изображениях применяется семантическая сеть WordNet. База данных определяет 1000 классов и по состоянию на 2016 г. содержала около 10 млн. изображений.

[14] Кстати, аспирант Стефана Маллата, который известен теоретическими исследованиями причин эффективности сверточных нейронных сетей и анализом их эквивалентности банкам вейвлет-фильтров.

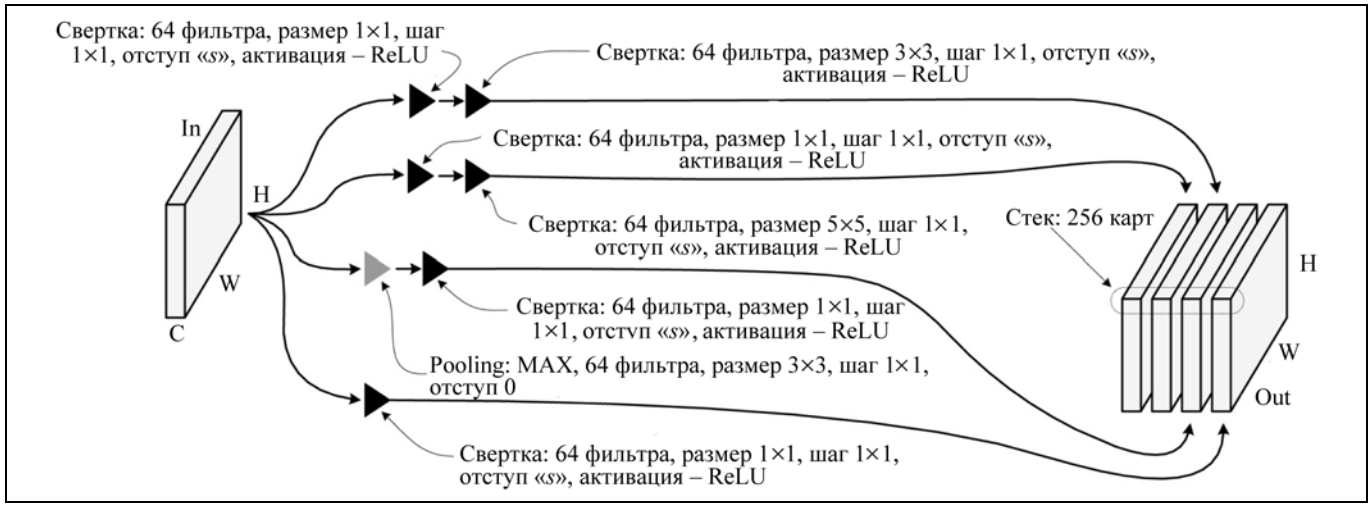

Рис. 9. Структура сверточного модуля Inception: переменные C, W и H на ребрах параллелепипедов — размеры входных (In) полей и выходных (Out) карт-признаков

(в сравнении с архитектурой классического сверточного слоя, изображенного на рис. 10, *а*).

В ноябре 2014 г. Дж. Лонг с соавторами публикуют препринт [37], в котором описывают так называемую «*полносверточную сеть*» (англ.: *Fully Convolutional Networks*), направленную на решение задачи семантической сегментации в режиме «pix2pix[15]». Работа, с одной стороны, развила концепцию статьи [33] в части построения сетей без полносвязных слоев, с другой, ввела в рассмотрение новый слой «*UpSampling*» — операция пространственного расширения карты признаков. Смысл операции иллюстрируют формулы (в случае 2D данных, с размером ядра 2 и шагом 2):

$$\mathbf{Z} = \mathrm{UpS}(\mathbf{X}) : \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix},$$

$$\mathbf{Z'} = \begin{bmatrix} x_{11} & x_{11} & x_{12} & x_{12} \\ x_{11} & x_{11} & x_{12} & x_{12} \\ x_{21} & x_{21} & x_{22} & x_{22} \\ x_{21} & x_{21} & x_{22} & x_{22} \end{bmatrix}, \quad \mathbf{Z} = \mathrm{Flt}(\mathbf{Z'}),$$

где $\mathbf{X}$ — матрица входных данных, $\mathbf{Z}$ — выход слоя, Flt — операция фильтрации (она либо отсутствует — тождественное преобразование $\mathbf{Z} = \mathbf{Z'}$, либо применяется билинейная интерполяция, как в работе [37]). Отметим, что этот слой является упрощенной версией слоя деконволюции (7), который в современной трактовке называется «*транспонированная свертка*» (англ.: *Transposed Convolution*) [38]. Весьма наглядна разница между слоями UpSampling и Transposed convolution продемонстрирована в работе [39].

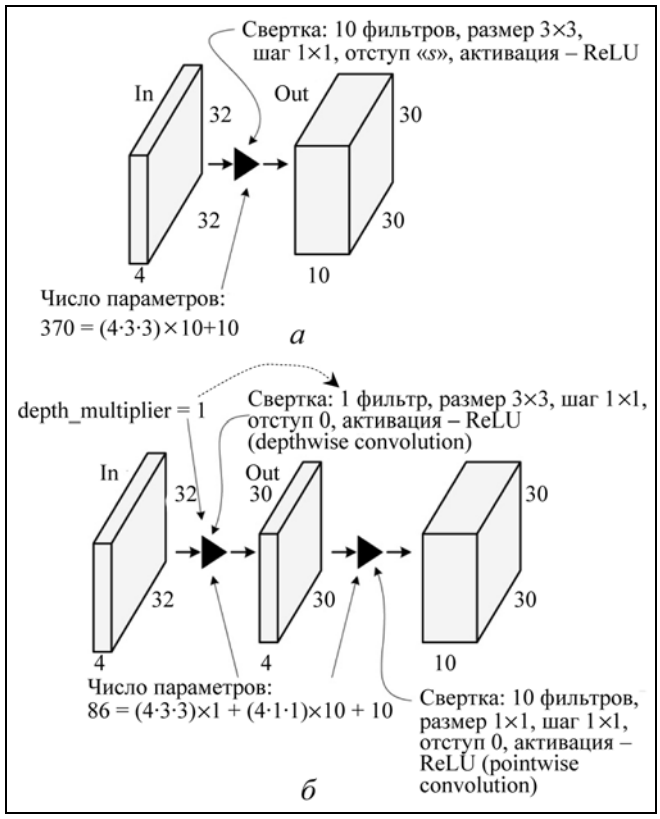

Рис. 10. Структуры сверточных слоев: *а* — обычная регулярная свертка; *б* — свертка Depthwise Separable; числа на ребрах параллелепипедов — размеры входных (In) полей и выходных (Out) карт-признаков

[15] Альтернативное название Image-to-Image, т. е. изображение на входе нейросети преобразуется на ее выходе в некое другое изображение (зависит от задачи), но совпадающее по размеру с исходным.

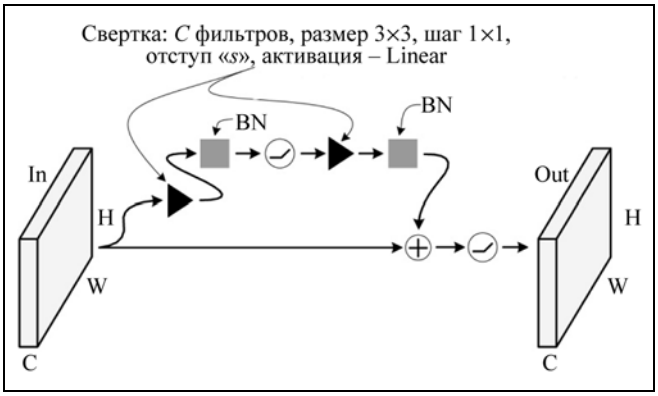

Рис. 11. Структура сверточного модуля ResNet: блоки BN — Batch Normalization, см. выражение (8); переменные C, W и H на ребрах параллелепипедов — размеры входных (In) полей и выходных (Out) карт-признаков

К концу 2014 г. происходит некий «фазовый переход»: интернет-гиганты, в том числе Google, признают высокую эффективность глубоких сверточных ИНС в задачах, связанных с распознаванием изображений и голоса, и начинается активное их внедрение в соответствующие бизнес-процессы.

В 2015 г. С. Иоффе с коллегой предлагают *стандартизовывать* данные внутри нейросети при их передаче между скрытыми слоями [40]. Техника была названа *Batch Normalization*:

$$\mathcal{B} = \{x_1, x_2, ..., x_M\}, \quad y_i = \mathrm{BN}_{\gamma,\beta}(x_i), \quad \mu_{\mathcal{B}} = \mathrm{M}[\mathcal{B}],$$

$$\sigma^2_{\mathcal{B}} = \mathrm{D}[\mathcal{B}], \quad \hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}} + \varepsilon}, \quad \varepsilon \to 0,$$

$$\mathrm{BN}_{\gamma,\beta}: y_i = \gamma \hat{x}_i + \beta, \qquad (8)$$

где $\mathcal{B}$ — мини-батч данных, γ, β — обучаемые параметры.

Предложенный подход существенно облегчил обучение глубоких структур, так как стандартизация приводила к тому, что последующий слой не тратил свои степени свободы на сдвиг и масштабирование входящих данных, а занимался только оцениванием их структурных свойств. Как следствие — ускорение сходимости процесса обучения, работа с более сложными данными, возможность применять более высокие значения Learning Rate — параметра η в формуле (3). Отметим, что в некоторых литературных источниках технику *Batch Normalization* объявляют более эффективной заменой техники *Dropout*, но на самом деле у них разный принцип действия и разное назначение.

Весной 2015 г. О. Роннебергер с коллегами предложил весьма оригинальную архитектуру полносверточной нейросети, решающей задачу сегментации изображений [41]. Сеть состоит из двух частей: входной — сжимающей (сверточные слои и слои субдискретизации) и выходной — расширяющей (в ее основе слои UpSampling, после каждого из них следует сверточный слой). Ключевой момент — наличие прямых связей между сжимающей и расширяющей частями на одинаковых пространственных масштабах. Подобная архитектура, в отличие от, например, ранее рассмотренной [37] требует меньшее число примеров для обучения и при этом порождает более точную сегментацию.

Еще одна разработка 2015 г. — *разреженная свертка* (англ.: *Dilation Convolution*), которую предложили Ф. Юу и В. Котлин в работе [42]. Изменяя коэффициент дилатации D, возможно гибко управлять размером рецептивного поля без изменения числа обучаемых параметров (пример 1D свертки размером 3):

$$D = 0: w_0x_0 + w_1x_1 + w_2x_2,$$

$$D = 1: w_0x_0 + w_1x_2 + w_2x_4,$$

$$D = 2: w_0x_0 + w_1x_3 + w_2x_6,$$

где $\mathbf{x}$ — входные данные; $\mathbf{w}$ — ядро свертки. При $D = 0$ разреженная свертка эквивалентна обычной. Отметим, что при комбинировании слоев разреженных сверток с коэффициентом $D > 0$ область видимости подобных нейронов растет весьма быстро, при сохранении малого числа настраиваемых параметров.

В самом конце 2015 г. выходит весьма нетривиальная (и, как оказалось впоследствии, революционная) работа [43] сотрудников одного из исследовательских центров «Microsoft». В ней описываются так называемые Residual Networks (ResNet: *сверточная нейросеть с остаточными блоками*), рис. 11. Основная идея ResNet: неизмененные входные данные суммируются с нелинейно преобразованными. Это сразу же привело к стабильной обучаемости сетей глубиной в 100, а впоследствии и в 1000 слоев[16] на достаточно сложных данных.

На протяжении 2016—2019 гг. исследователи в основном экспериментировали с различными вариациями и комбинациями Inception и ResNet и их приложениями к реальным задачам. Но, помимо этого, были также инициированы исследования по применению различного рода интегральных преобразований в сверточных слоях.

В 2018 г. Х. Кхан с коллегами предложили [44] слой *вейвлет-деконволюции*[17] (англ.: *Wavelet Decon-*

[16] Практического смысла в столь глубокой сети нет никакого (по крайней мере, в исследовавшихся задачах), но есть смысл методический: «можем обучать».

[17] Исходя из математического описания, более корректное название статьи и самой операции: вейвлет-декомпозиция (wavelet decomposition).

volution) в качестве эффективной адаптивной альтернативы предварительного спектрального разложения входных данных (временных рядов):

$$\psi_{s,b}(t) = \frac{1}{\sqrt{s}}\psi^*\left(\frac{t-b}{s}\right), \quad z = x \oplus \psi_{s,b},$$

где x — входные данные (дискретная последовательность), $\oplus$ — операция свертки, z — выход слоя; ψ^* — материнский вейвлет для непрерывного вейвлет-преобразования, b — параметр масштаба, адаптируемая (обучаемая) величина, которая изменяется при обучении на величину $\Delta s = -\eta\frac{\partial \mathrm{L}(s)}{\partial s}$.

В том же 2018 г. С. Фуджиеда с коллегами ввели в рассмотрение [45] вейвлет-аналог слоя субдискретизации (см. рис. 7, *б*), который позволяет эффективно извлекать информацию о характере текстур на 2D данных (изображениях) на разных пространственных масштабах посредством аналога дискретного вейвлет-преобразования. Недостаток реализованного авторами подхода — фиксированное число масштабов разложения, задаваемого посредством числа слоев в нейросети.

В 2019 г. П. Лю с коллегами предложили встроить дискретное прямое и обратное вейвлет-преобразования на основе вейвлета Хаара в сверточные слои [46]. При этом архитектурно полносверточная сеть получилась подобна рассмотренной выше сети U-Net [41].

Подходы, подобные изложенным в работах [44—46], позволили существенно уменьшить число обучаемых параметров сверточных нейросетей, а также улучшить их характеристики на ряде задач относительно типовых архитектур (AlexNet, VGG и т. п.).

Продемонстрируем на задаче из примера 1 богатые функциональные возможности элементарной сверточной нейронной сети при классификации случайных сигналов[18]. Рассмотрим

Пример 2. Сформируем ИНС с одним скрытым слоем из одного сверточного ядра размером 1 (рис. 12) и на вход подадим «сырые данные» **x**. Как показано в работе [21], подобная ИНС весьма успешно решает задачу классификации случайных сигналов, имеющих идентичную энергию и различающихся только функциями плотности вероятности: $\mathcal{N}$, $\mathcal{U}$, $\mathcal{E}$. Мера качества классификации $F_1 = 1$.

В работе [21] также проведен анализ структуры обученных сверточных сетей и механизмов их функционирования при принятии решения в случае представленной задачи. Исследована их устойчивость к загрязнению входных данных по модели запирания канала/сенсора и возможность обнаружения сетью преобладающего сигнала в смеси сигналов $\mathcal{N}$, $\mathcal{U}$, $\mathcal{E}$ в условиях априорной неопределенности. Таким образом, показано, что глубокие сверточные нейронные сети могут эффективно работать не только с сигналами, имеющими ярко выраженные паттерны, но и с реализациями узко- или широкополосных случайных процессов, и таким образом решать ряд задач по обработке сигналов. Решение подобной сложной прикладной задачи изложено в работе [47], посвященной разработке на основе глубокой сверточной ИНС первичного классификатора сигналов для квантовой волоконно-оптической системы охраны магистральных трубопроводов. ♦

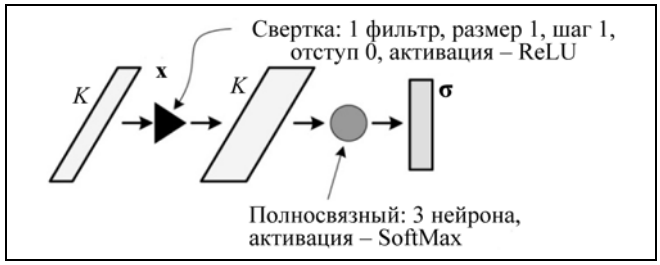

Рис. 12. Структура минимальной сверточной ИНС, успешно решающей задачу классификации случайных сигналов с функциями плотности вероятности $\mathcal{N}$, $\mathcal{U}$, $\mathcal{E}$, K — длина временно́го ряда

Сверточные нейронные сети отлично справляются и с проблематикой идентификации хаоса, а также прямого оценивания показателя Ляпунова в дискретных динамических системах по их наблюдаемым траекториям в расширенном пространстве состояний (см., например, работу [48]).

4. ГЛУБОКИЕ РЕКУРРЕНТНЫЕ НЕЙРОСЕТИ

В 1990 г. Дж. Элман предложил *рекуррентную* ИНС с одним скрытым слоем по типу MLP [49]:

$$\mathbf{h}_k = g_h(\mathbf{W}_h\mathbf{x}_k + \mathbf{U}_h\mathbf{h}_{k-1} + \mathbf{b}_h),$$
$$\mathbf{y}_k = g_y(\mathbf{W}_y\mathbf{h}_k + \mathbf{b}_y), \qquad (9)$$

где k — дискретное время, $\mathbf{h}_k$ — вектор скрытого состояния сети в момент времени k. Как видно из выражения (9), слагаемое $\mathbf{U}_h\mathbf{h}_{k-1}$ задает обратную связь и отвечает за временной контекст. Этот контекст «одношаговый» по времени, подобные сети классифицируют как SimpleRNN (Recurrent Neural Network), в противовес им многошаговые сети называются FullyRNN.

В 1991 г. Х. Зигельманном и Е. Сонтагом доказана

Теорема 3 (о полной тьюринговости RNN [50])**.** *Любые машины Тьюринга могут моделироваться полностью связанными рекуррентными сетями, созданными из нейронов с сигмоидальными функциями активации, при условии, что сеть имеет достаточное число нейронов в скрытом слое M и достаточное число шагов временной памяти K.*

[18] Задача по сути своей постановки близка к прикладной проблематике распознавания сигналов в пассивных акустических пеленгаторах.

В 1992 г. К. Фунахаши и И. Накамурой доказана

Теорема 4 (универсальная аппроксимационная теорема RNN [51]). *Любая нелинейная динамическая система класса*

$$\frac{d}{dt}s(t) = f(s), \quad s(t=0) \in S_0,$$

может быть аппроксимирована рекуррентной нейронной сетью с любой точностью, без ограничений на компактность пространства состояний системы, при условии, что сеть имеет достаточное число нейронов в скрытом слое M.

Из теоремы 4 автоматически вытекает следствие, что любая непрерывная кривая (динамический процесс, временной ряд) может быть аппроксимирована с любой точностью выходом RNN (при соблюдении ряда условий, основными из которых являются достаточное число нейронов рекуррентного слоя M и достаточное число шагов его временной памяти K). Таким образом, открываются возможности применения RNN для высокоэффективного решения ряда задач управления[19], в том числе оценивания и прогнозирования динамических сигналов, идентификации систем управления и др., причем в классе адаптивных и сверхадаптивных систем управления.

В 1997 г. М. Джордан предложил модификацию сети Элмана (9) [52]:

$$\mathbf{h}_k = g_h(\mathbf{W}_h\mathbf{x}_k + \mathbf{U}_h\mathbf{y}_{k-1} + \mathbf{b}_h),$$
$$\mathbf{y}_k = g_y(\mathbf{W}_y\mathbf{h}_k + \mathbf{b}_y). \qquad (10)$$

Из сравнения выражений (9) и (10) видно, что в случае *сети Джордана* контекст решения определяется выходом сети, а не скрытым слоем.

Теоремы 3 и 4 вызвали активные исследования применимости сетей Элмана и Джордана в самых различных областях, но очень скоро выяснились фатальные недостатки SimpleRNN:

— фактически сети оперируют очень короткими динамическими контекстами, забывание «прошлого» идет с экспоненциальной скоростью;

— в рамках одной сети очень сложно совмещать процессы различных масштабов, в том числе «быстрое» и «медленное» время, а также обрабатывать пропуски данных;

— рекуррентные сети, построенные по типу MLP, очень сложно обучать (применяется алгоритм Backpropagation Through Time) при больших значениях K: градиент либо затухает, либо испытывает взрывной рост.

[19] Естественно, что эти возможности относятся к глубоким RNN, в том числе имеющих в своем составе ячейки LSTM (описание см. далее).

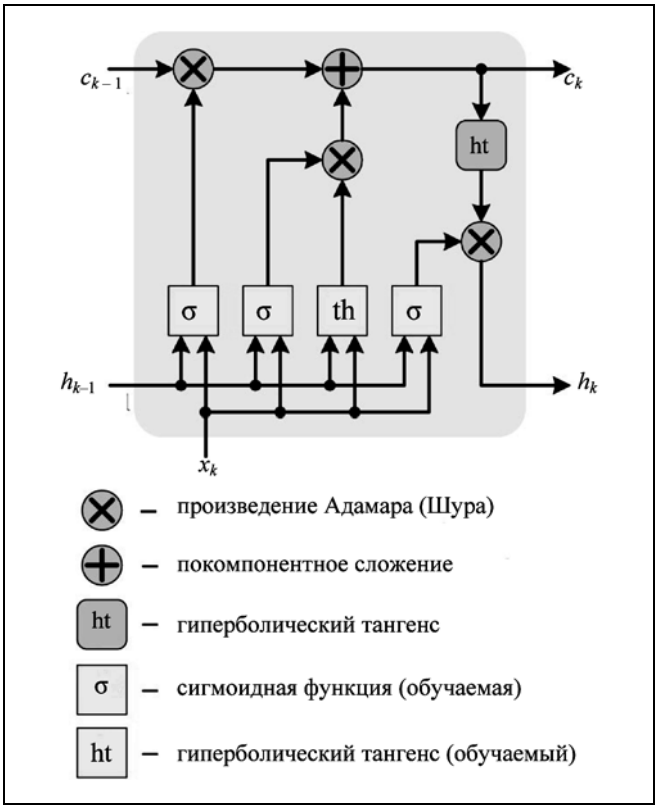

Рис. 13. Элементарная ячейка LSTM

В 1997 г. для решения означенных проблем С. Хохрейтер с коллегами предложили принципиально иную архитектуру RNN, названную LSTM — Long Short-Term Memory (*долгая краткосрочная память*) [53]. Элементарная ячейка скрытого слоя сети приведена на рис. 13.

Ячейка LSTM содержит *конвейер состояния ячейки* $x_{k-1} \to c_k$, который включает в себя только линейные (!) операции. Модификация информации управляется вентилями (англ.: *gates*): $s' = s\sigma(\circ)$, причем $\sigma(\circ) \in [0, 1]$. Стандартная ячейка LSTM состоит из четырех вентилей:

— «forget gate»: $\mathbf{f}_k = \sigma(\mathbf{W}_f\mathbf{x}_k + \mathbf{U}_f\mathbf{h}_{k-1} + \mathbf{b}_f)$ — интерпретация: если тема (сцена) изменяется, то информация о старой теме (сцене) стирается;

— «input gate & activation»: $\mathbf{i}_k = \sigma(\mathbf{W}_i\mathbf{x}_k + \mathbf{U}_i\mathbf{h}_{k-1} + \mathbf{b}_i)$, $\tilde{\mathbf{c}}_k = \text{th}(\mathbf{W}_c\mathbf{x}_k + \mathbf{U}_c\mathbf{h}_{k-1} + \mathbf{b}_c)$ — интерпретация: определяется, какие значения будут обновляться и создается вектор кандидатов на $\tilde{\mathbf{c}}_k$, которые предполагается добавить в состояние ячейки;

— «internal state»: $\mathbf{c}_k = \mathbf{f}_k * \tilde{\mathbf{c}}_{k-1} + \mathbf{i}_k * \tilde{\mathbf{c}}_k$ — интерпретация: формируется новое состояние ячейки $\mathbf{c}_k$;

— «output gate & value»: $\mathbf{o}_k = \sigma(\mathbf{W}_o\mathbf{x}_k + \mathbf{U}_o\mathbf{h}_{k-1} + \mathbf{b}_o)$, $\mathbf{h}_k = \mathbf{o}_k * \text{th}\mathbf{c}_k$ — интерпретация: формируется новый выход ячейки $\mathbf{h}_k$.

Число настраиваемых во время обучения параметров в слое LSTM: $4(MN + M^2 + M)$, где N — число признаков во входном векторе $\mathbf{x}$, M — число нейронов в рекуррентном слое, эту же размерность имеют вектора $\mathbf{c}$ и $\mathbf{h}$.

Последующие исследования LSTM-сетей показали, что для них выполняются теоремы 3 и 4, но при этом LSTM-сети свободны от большинства проблем SimpleRNN.

В 2000 г. Т. Чао и Х. Ли расширили теорему 4 на неавтономные нелинейные обыкновенные дифференциальные уравнения [54]:

$$\frac{d}{dt}s(t) = f(s) + g(t), \quad s(t=0) \in S_0.$$

В 2005 г. Ф. Морин и Б. Йошуа предложили иерархическое обобщение Softmax слоя (5) [55], что сделало возможным устойчивое решение задач классификации (в первую очередь в компьютерной лингвистике) размером свыше 20 тыс. классов.

В этом же году А. Гравес и Ю. Шмидхубер предлагают двунаправленное обобщение LSTM — BiDirectional LSTM [56]. Основная мотивация: «Настоящее зависит не только от прошлого, но и от будущего». Впоследствии это позволило получить более устойчивые и качественные решения ряда задач, так как для формирования выхода z_i сеть использовала информацию не только из левой части временного ряда: $[..., z_{i-2}, z_{i-1}, z_i]$, но также и из правой: $[z_{i+1}, z_{i+2}, ...]$.

Таким образом, к концу 2005 г. у исследователей формируется уверенность в перспективности применения LSTM-сетей в области компьютерной лингвистики, распознавания и синтеза слитной речи, онлайн распознавания слитных рукописных текстов и др. Интенсивность исследований в области рекуррентных ИНС существенно возрастает.

В 2013 г. А. Гравес предлагает первую дифференцируемую реализацию[20] *механизма внимания* (англ.: *Attention Layer*) [57]. Структурная схема предложенного слоя сети приведена на рис. 14.

Выход слоя внимания формируется как взвешенное скользящее среднее от выхода рекуррентного слоя:

$$o_k = \sum_{i=0}^{2n} a_i h_{k+(i-n)}, \quad \sum_{i=0}^{2n} a_i \equiv 1, \qquad (11)$$

где a_i — компоненты вектора обучаемых параметров. Интерпретация выражения (11): при формировании выхода слоя учитываются настраиваемые локальные во времени структурные связи — так называемый контекст.

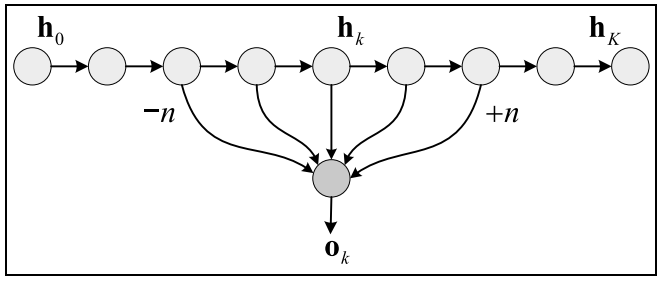

Рис. 14. Схема дифференцируемого слоя внимания

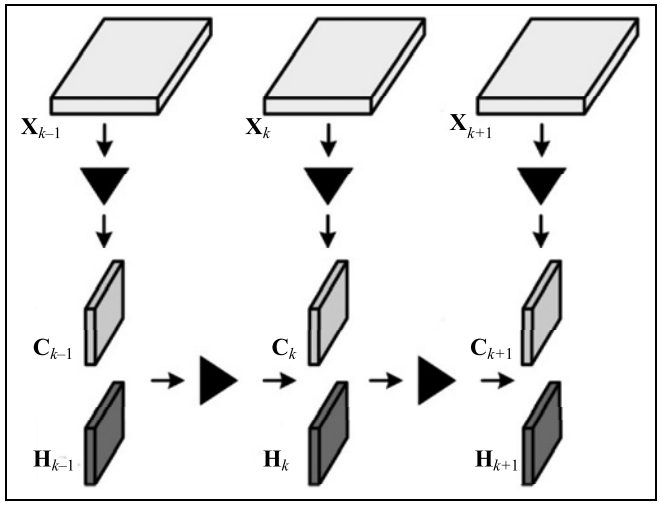

Рис. 15. Схема Convolutional LSTM-сети: треугольник — операция свертки, остальные обозначения аналогичны обозначениям на рис. 13

Наконец, в 2015 г. выходит работа, которую давно ожидали: С. Ши с коллегами предлагают Convolutional LSTM-сеть [58], предназначенную для обработки *пространственно-временных зависимостей*. Обобщенная структура данной сети, демонстрирующая идею, приведена на рис. 15. Основное отличие Convolutional LSTM-сети от обычной LSTM в том, что ее внутренняя MLP-подобная структура (см. рис. 14) заменена на сверточную.

В работе [58] продемонстрирована высокая эффективность ConvLSTM-сети при обработке данных с погодного радара.

Следом за работой [58] в том же году Н. Кальчбреннер с коллегами предлагают решетчатую LSTM-сеть [59]. Основная идея: расширение LSTM функционала с одной «временной оси» на все N осей входных данных, где $N = \dim\mathbf{X}$.

В конце 2015 г. выходят две работы [60, 61], которые демонстрируют противоречивые результаты применения техники Batch Normalization (8) к се-

[20] «Дифференцируемая» здесь означает поддержку нейросетевым слоем режима обучения алгоритмом обратного распространения ошибки.

тям RNN. Так, в работе [60] применение пакетной нормализации фактически никак не повлияло на показатели качества исследуемых рекуррентных нейросетей, но в ряде случаев ускоряло их обучение. В работе [61], напротив, Batch Normalization являлась центральным элементом, применение которого позволило получить рекуррентную нейросеть высокого качества. Такое положение вещей практически моментально привело к появлению альтернативных решений.

Летом 2016 г. Дж. Леи Ба с коллегами предлагают технику «*нормализация слоя*» (англ.: *Layer Normalization*) [62] для применения в RNN вместо пакетной нормализации. Предложенная альтернатива (нормализация слоя) стандартизует каждый сэмпл данных по всем нейронам слоя, в отличие от стандартизации мини-батча в целом, но для каждого нейрона индивидуально (пакетная нормализация). Как показали авторы [62], на их примерах (длинные последовательности и небольшие мини-батчи) техника Layer Normalization, относительно альтернатив, существенно положительно влияла на скорость обучения рекуррентных сетей.

Результаты работ [60—62] относятся к задачам обработки текстов рекуррентными нейросетями, а эти задачи имеют одну примечательную особенность: предложения в наборах данных, как правило, имеют существенную вариативность по длине (числу слов, букв).

На протяжении 2016—2019 гг. исследователи в основном экспериментировали с различными вариациями и комбинациями LSTM-сетей и их приложениями к реальным задачам. Но, помимо этого, активно исследовали и механизм внимания как самостоятельную структурную единицу глубоких нейросетей.

Заметим, что многие задачи компьютерной лингвистики (перевод, аннотирование текста, распознавание слитной речи и др.) в указанный период начали решаться в парадигме *sequence-to-sequence*, т. е. предложение (фрагмент текста) целиком поступает на вход нейросети (например, на английском языке), предложение (фрагмент текста) формируется на ее выходе (к примеру, с переводом на русский). Входная часть нейросети называется *энкодер* (она обычно реализуется либо сверточными, либо рекуррентными слоями), выходная часть — это *декодер* (как правило, реализуется рекуррентными слоями). Между этими двумя частями включается слой внимания (11). Это была классическая — достаточно эффективная конструкция[21].

Но в 2017 г. А. Васвани с коллегами[22] предложили так называемую архитектуру *Transformer* [63] — целиком и полностью состоящую из слоев иерархически организованных нелинейных банков ячеек внимания[23] (!), названных в оригинальной работе «*Multi-head attention*». Основная операция Att(∘) — *attention* (*внимание*) выражается в виде:

$$\mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{SM}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V},$$

где SM(∘) — функции активации SoftMax (9), d — число столбцов матриц $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ (фактически размерность *эмбеддинга* (англ.: *embeddings*) — вложения), $\mathbf{Q}$ — запрос, $\mathbf{K}$ — ключ, $\mathbf{V}$ — значения эмбеддинга (как правило, векторное представление (кодирование) обрабатываемого токена).

Предложенная конструкция существенно улучшила качество машинного перевода[24], став ведущей моделью компьютерной лингвистики. Ее развитие и улучшение не заставили себя долго ждать, ибо базовая архитектура обладает существенным недостатком — ограниченная длина операционного контекста (не более нескольких десятков токенов).

Летом 2018 г. М. Дегани с коллегами обобщают архитектуру *Transformer*, включая в ее состав рекуррентные цепочки [64]. Осенью того же года Я. Девлин с коллегами предложили архитектуру BERT (Bidirectional Encoder Representations from Transformers) [65]. Существенное преимущество BERT перед LSTM-сетями — это длина операционного контекста: десятки токенов у LSTM, против двух — трех сотен у BERT.

Наконец, в 2019 г. З. Даи с коллегами предлагают новую архитектуру — так называемый Transformer-XL [66]. Длина генерируемых последовательностей согласованных токенов достигла нескольких тысяч[25].

ЗАКЛЮЧЕНИЕ

Итак, «естественный отбор» эффективных в прикладном плане архитектур ИНС привел к тому, что к началу 2019 г. мейнстримом в глубоких нейросетях являются всего три качественно различающихся типа слоев:

— *полносвязные* (по типу персептрона Румельхарта);

[21] Весьма схожая по архитектуре с автоэнкодерами [2].

[22] Все — сотрудники различных подразделений Google.

[23] В состав сети дополнительно входят аналоги полносвязных слоев, ResNet блоков и функции активации SoftMax и, что примечательно, Layer Normalization [62].

[24] На паре «английский → немецкий» метрика качества BLEU превысила значение 28, что более чем на два с лишним пункта лучше предыдущего результата (сеть SliceNet /CNN/).

[25] Порождается весьма согласованный и достаточно осмысленный текст.

— *сверточные* (со всем многообразием их модификаций);

— *рекуррентные* (в основном LSTM и GRU [2]).

При этом полносвязные слои почти потеряли всякую самостоятельность. Они стоят, как правило, на выходе сети, представляя собой линейный классификатор (регрессор) сепарабельной задачи (см. теорему 1), которую формируют более выразительные (имеющие лучшие обобщающие свойства) сверточные и рекуррентные слои.

Мощность (выразительность, обобщающая способность) сверточных и рекуррентных слоев по большей части объясняется тем, что они, в отличие от полносвязных архитектур, построенных по типу MLP, максимально задействуют при построении информативных признаков явную и/или скрытую структуру данных. Наглядный пример: распознавание изображений, голоса, построение языковых моделей. Примеры 1 и 2, приведенные выше, также демонстрируют этот аспект.

Колоссальная «обобщающая способность» глубоких нейросетей формируется в основном благодаря широкой структурно-статистической вариативности и, как следствие, большому объему обучающих данных. В этом вопросе, к сожалению, прогресс с 1960-х гг. пока весьма слаб [7]. Поэтому такая процедура как *data augmentation* [2], весьма востребована в практике обучения глубоких ИНС.

Еще раз подчеркнем, что все приведенные в настоящем обзоре теоремы формируют строгий математический фундамент теории ИНС — гарантируют решение задачи, но не являются конструктивными: они не дают путей решения задачи. И именно здесь начинается «*нейросетевое искусство*». Попытки формализовать этот процесс активно предпринимаются в рамках такой научной дисциплины как «Теория статистического обучения», ряд существенных результатов в которой был получен в свое время учеными Института проблем управления РАН В.Н. Вапником и А.Я. Червоненкинсом при разработке ими теории так называемой VC-размерности [67]. Правда, эти результаты к оцениванию характеристик глубоких нейронных сетей имеют очень ограниченную применимость.

Из рассмотренного материала становится отчетливо видно, что успех глубоких ИНС объясняется не только алгоритмическими и математическими прорывами[26]. Есть еще кое-что.

- Экспоненциальный рост вычислительных мощностей (наглядно демонстрируемый сравнением двух вариантов: top-1 суперкомпьютера начала XXI в. и массивно-параллельного ускорителя (GPU) наших дней), обеспечил возможность приемлемой длительности обучения весьма сложных нейросетей (см. таблицу).

Характеристики top-1 суперкомпьютера начала XXI в. и современного массивно-параллельного ускорителя (GPU)

Модель	ASCI White (октябрь 2000 г. — июнь 2002 г.)	GPU NVIDIA V100 (3-й квартал 2017 г.)
Производительность, стоимость	12,3 Тфлопс, 110 млн $ США	15,0 Тфлопс, 1700 $ США
Потребляемая мощность, масса	6 МВт, 106 т	300 Вт, 370 г

- Существенное снижение стоимости хранения цифровых данных и широкое распространение социальных сетей обеспечило экспоненциальный рост размеров датасетов различной природы (текст, фото, видео, звук, музыка и др.), на которых возможно тренировать нейросети.
- С программно-алгоритмическим обеспечением в области Deep Learning сложилась уникальная ситуация, кардинально отличающаяся от принятых «правил игры» в других научно-технических областях: подавляющее большинство библиотек и фреймворков — бесплатно; исходный код основных библиотек и фреймворков — открыт; обучающие материалы — бесплатны и свободно доступны; функционируют широкие и отзывчивые группы поддержки — от уровня новичка и до топовой проблематики.

В данном обзоре вне рассмотрения остался ряд важных вопросов, фактически формирующих ключевую проблематику современной теории искусственных нейронных сетей:

— два мощных и современных направления глубоких ИНС: *автоэнкодеры* — осуществляющие сжатие входных данных для представления их в *Latent-Space* (*скрытое пространство признаков и состояний*) и GAN — Generative Adversarial Network (генеративно-состязательные сети), осуществляющие порождение данных посредством комбинации двух сетей: генератора и дискриминатора (осуществляет оценку правильности генерации). Заинтересованный читатель сможет найти о них подробную информацию в книге [2];

— обработка глубокими нейросетями нерегулярных сложных данных, к примеру, взвешенных графов произвольной структуры;

— обучение сетей, эффективные функции потерь и оптимизаторы (здесь можно порекомендовать читателю достаточно регулярно обновляемый пост С. Рудера [68]);

— машинный синтез эффективных нейроструктур, согласованных с решаемой задачей (в том числе подходы AutoML);

— интерпретация решений ИНС;

[26] Действительно, большинство теоретических результатов получено в конце XX в., а активный научной рост начался после 2012 г.

— сохранение эффективности нейросети вне домена обучающих данных и защита от имитационных атак (adversarial attack).

ЛИТЕРАТУРА

1. *Sutton, R.S., Barto, A.G.* Reinforcement Learning: An Introduction. — Cambridge: The MIT Press, 1998.
2. *Бенджио И., Гудфеллоу Я., Курвилль А.* Глубокое обучение. — М.: ДМК-Пресс, 2018. — 652 с. [*Goodfellow, I., Bengio, Y., Courville, A.* Deep Learning. — Cambridge: The MIT Press, 2016.]
3. *McCulloch, W., Pitts, W.* A Logical Calculus of the Ideas Immanent in Nervous Activity // Bull. Math. Biophys. — 1943. — Vol. 5. — P. 115—133.
4. *Hebb, D.* The Organization of Behavior. — N.-Y.: John Wiley & Sons, 1949. — 335 p.
5. *Rosenblatt, F.* The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // Cornell Aeronautical Laboratory / Psychological Review. — 1958. — Vol. 65, no. 6. — P. 386—408.
6. *Cover, T.* Geometrical and Statistical properties of systems of linear inequalities with applications in pattern recognition // IEEE Trans. on Electronic Computers. — 1965. — EC-14. — P. 326—334.
7. *Розенблатт Ф.* Принципы нейродинамики: Перцептроны и теория механизмов мозга. — М.: Мир, 1965. — 478 с. [*Rosenblatt, F.* Principles of neurodynamics: perceptrons and the theory of brain mechanisms. — Spartan Books, 1962. — 616 p.]
8. *Widrow, B.* Pattern Recognition and Adaptive Control // Proc. of the IRE-AIEE Joint Automatic Control Conference. — August 1962. — P. 19—26.
9. *Minsky, M., Papert, S.* Perceptrons: An Introduction to Computational Geometry. — Cambridge: The MIT Press, 1969.
10. *Parallel* Distributed Processing: Explorations in the Microstructures of Cognition / ed. by D.E. Rumelhart, J.L. Mcclelland. — Cambridge: MIT Press, 1986.
11. *Галушкин А.* Синтез многослойных систем распознавания образов. — М.: Энергия, 1974. — 368 с. [*Galushkin, A.* Synthesis of multilayer pattern recognition systems. — Moscow: Energy, 1974. (In Russian)]
12. *Werbos, P.* Beyond regression: New tools for prediction and analysis in the behavioral sciences / PhD thesis, Harward University, 1974.
13. *Барцев С., Охонин В.* Адаптивные сети обработки информации / Препринт № 59Б. — Красноярск: Ин-т физики СО АН СССР, 1986. — 20 с. [*Bartsev, S., Okhonin, V.* Adaptive Information Processing Networks. Preprint of Biophysics Institute SB AS USSR, Krasnoyarsk, 1986. — No. 59 B. — 20 p. (In Russian)]
14. *Rumelhart, D., Hinton, G., Williams, R.* Learning Internal Representations by Error Propagation / In: Parallel Distributed Processing. — Vol. 1. — P. 318—362. — Cambridge, MA, MIT Press, 1986.
15. *Broomhead, D., Lowe, D.I.* Radial basis functions, multi-variable functional interpolation and adaptive networks / Technical report RSRE. — 1988. — No. 4148.
16. *Cybenko, G.* Approximation by Superpositions of a Sigmoidal Function // Mathematics of Control Signals and Systems. — 1989. — Vol. 2, no. 4. — P. 303—314.
17. *Nielsen, R.* Kolmogorov's Mapping Neural Network Existence Theorem // Proc. of the IEEE First Int. Conf. on Neural Networks (San Diego, CA). — 1987. — Vol. 3. — P. 11—13.
18. *Bridle, J.* Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition / In: Neurocomputing. NATO ASI Series (Series F: Computer and Systems Sciences) / F.F. Soulie, J. Herault (eds). — Vol. 68. — Berlin, Heidelberg: Springer, 1989. — P. 227—236.
19. *Hornik, K.* Approximation Capabilities of Multilayer Feedforward Networks // Neural Networks. — 1991. — Vol. 4, no. 2. — P. 251—257.
20. *Tiwari, S., Singh, A., Shukla, V.* Statistical moments based noise classification using feed forward back propagation neural network // Int. J. of Computer Applications. — 2011. — Vol. 18, no. 2. — P. 36—40.
21. *Portsev, R., Makarenko, A.* Convolutional neural networks for noise signal recognition // IEEE 28th Int. Workshop on MLSP, Aalborg. — 2018. — P. 1—6.
22. *Hinton, G., Salakhutdinov, R.* Reducing the Dimensionality of Data with Neural Networks // Science. — 2006. — Vol. 313, no. 5786. — P. 504—507.
23. *Nair, V., Hinton, G.* Rectified Linear Units Improve Restricted Boltzmann Machines // Proc. of the Int. Conf. on Machine Learning. — 2010. — P. 807—814.
24. *Glorot, X., Bengio, Y.* Understanding the difficulty of training deep feedforward neural networks // Proc. of the Thirteenth Int. Conf. on Artificial Intelligence and Statistics. — 2010. — Vol. 9. — P. 249—256.
25. *Lecun, Y., Boser, B., Denker, J.*, et al. Backpropagation Applied to Handwritten Zip Code Recognition // Neural Computation. — 1989. — Vol. 1, no. 4. — P. 541—551.
26. *Fukushima, K.* Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position // Biological Cybernetics. — 1980. — Vol. 36, no. 4. — P. 193—202.
27. *Mozer, M.* Early Parallel Processing in Reading: A Connectionist Approach / In: Attention and Performance 12: The Psychology of Reading / M. Coltheart (ed.). — 1987. — P. 83—104.
28. *Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.* Gradient-Based Learning Applied to Document Recognition // IEEE Intelligent Signal Processing. — 1998. — P. 306—351.
29. *Ciresan, D., Meier, U. Gambardella, L., Schmidhuber, J.* Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition // ArXiv: 1003.0358.
30. *Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.* Deconvolutional Networks // Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. San Francisco, CA, 2010. — P. 2528—2535.
31. *Krizhevsky, A., Sutskever, I., Hinton, G.* ImageNet Classification with Deep Convolutional Neural Networks // Proc. of the 25th Int. Conf. NIPS. — 2012. — Vol. 1. — P. 1097—1105.
32. *Hinton, G., Srivastava, N., Krizhevsky, A.,* et al. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors // ArXiv: 1207.0580.
33. *Lin, M., Chen, Q., Yan, S.* Network In Network // ArXiv: 1312.4400.
34. *Simonyan, K., Zisserman, A.* Very Deep Convolutional Networks for Large-Scale Image Recognition // ArXiv: 1409.1556.
35. *Szegedy, C., Liu, W., Jia, Y.*, et al. Going Deeper with Convolutions // ArXiv: 1409.4842.
36. *Sifre, L.* Rigid-motion scattering for image classification / Ph.D. Thesis. Ecole Polytechnique, CMAP. Defended October 6th, 2014.
37. *Long, J., Shelhamer, E., Darrell, T.* Fully Convolutional Networks for Semantic Segmentation // ArXiv: 1411.4038.
38. *Dumoulin, V., Visin, F.* A guide to convolution arithmetic for deep learning // ArXiv: 1603.07285. — URL: https://github.com/vdumoulin/conv_arithmetic.
39. *Odena, A., Dumoulin, V., Olah, C.* Deconvolution and Checkerboard Artifacts // Distill, 2016. — DOI: 10.23915/distill.00003.
40. *Ioffe, S., Szegedy, C.* Batch Normalization: Accelerating Deep Net-work Training by Reducing Internal Covariate Shift // ArXiv: 1502.03167.
41. *Ronneberger, O., Fischer, P., Brox, T.* U-Net: Convolutional Networks for Biomedical Image Segmentation // ArXiv: 1505.04597.
42. *Yu, F., Koltun, V.* Multi-Scale Context Aggregation by Dilated Convolutions // ArXiv: 1511.07122.
43. *He, K., Zhang, X., Ren, S., Sun, J.* Deep Residual Learning for Image Recognition // ArXiv: 1512.03385.

44. *Khan, H, Yener, B.* Learning Filter Widths of Spectral Decompositions with Wavelets // Proc. of the NIPS Conf. — 2018. — P. 4601—4612.
45. *Fujieda, S., Takayama, K., Hachisuka, T.* Wavelet Convolutional Neural Networks // ArXiv: 1805.08620.
46. *Liu, P., Zhang, H., Lian, W., Zuo, W.* Multi-level Wavelet Convolutional Neural Networks // ArXiv: 1907.03128.
47. *Makarenko, A.* Deep Learning Algorithms for Signal Recognition in Long Perimeter Monitoring Distributed Fiber Optic Sensors // IEEE 26th Int. Workshop on MLSP. — 2016. — Vietri sul Mare, IIASS. — P. 1—6.
48. *Makarenko, A.V.* Deep Learning Algorithms for Estimating Lyapunov Exponents from Observed Time Series in Discrete Dynamic Systems // Proc. of the STAB Conf. — 2018. — P. 1—4.
49. *Elman, J.* Finding Structure in Time // Cognitive Science. — 1990. — Vol. 14, no. 2. — P. 179—211.
50. *Siegelmann, H., Sontag, E.* Turing computability with neural nets // Appl. Math. Lett. — 1991. — Vol. 4, no. 6. — P. 77—80.
51. *Funahashi, K., Nakamura, Y.* Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks // Neural Networks. — 1993. — Vol. 6, no. 6. — P. 801—806.
52. *Jordan, M.* Serial Order: A Parallel Distributed Processing Approach // Advances in Psychology. — 1997. — No. 121. — P. 471—495.
53. *Hochreiter, S., Schmidhuber, J.* Long-Short Term Memory // Neural Computation. — 1997. — Vol. 9, no. 8. — P. 1735—1780.
54. *Chow, T., Li, X.* Modeling of Continuous Time Dynamical Systems with Input by Recurrent Neural Networks // IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications. — 2000. — Vol. 47, no. 4. — P. 575—578.
55. *Morin, F., Bengio, Y.* Hierarchical Probabilistic Neural Network Language Model // Proc. of AISTATS — 2005. — P. 246—252.
56. *Graves, A., Schmidhuber, J.* Framewise Phoneme Classification with Bidirectional LSTM Networks // Int. Joint Conf. on Neural Networks. — 2005. — P. 2047—2052.
57. *Graves, A.* Generating Sequences With Recurrent Neural Networks // ArXiv: 1308.0850.
58. *Shi, X., Chen, Z., Wang, H.,* et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting // ArXiv: 1506.04214.
59. *Kalchbrenner, N., Danihelka, I., Graves, A.* Grid Long Short-Term Memory // ArXiv: 1507.01526.
60. *Laurent, C., Pereyra, G., Brakel, P.,* et al. Batch Normalized Recurrent Neural Networks // ArXiv: 1510.01378.
61. *Amodei, D., Anubhai, R., Battenberg, E.,* et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin // ArXiv: 1512.02595.
62. *Lei Ba, J., Kiros, J.R., Hinton, G.E.* Layer Normalization // ArXiv: 1607.06450.
63. *Vaswani, A., Shazeer, N., Parmar, N.,* et al. Attention is All You Need // ArXiv: 1706.03762.
64. *Dehghani, M., Gouws, S., Vinyals, O.,* et al. Universal Transformers // ArXiv: 1807.03819.
65. *Devlin, J., Chang, M.W., Lee, K., Toutanova, K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // ArXiv: 1810.04805.
66. *Dai, Z., Yang, Z., Yang, Y.,* et al. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context // ArXiv: 1901.02860.
67. *Вапник В.Н., Червоненкис А.Я.* О равномерной сходимости частот появления событий к их вероятностям // Теория вероятностей и ее применение. — 1971. — Т. 16, вып. 2. — С. 264—279. [*Vapnik, V.N., Chervonenkis, A.Ya.* On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities // Theory of Probability & Its Applications. — 1971. — Vol. 16, iss. 2. — P. 264—280. (In Russian)]
68. *Ruder, S.* An Overview of Gradient Descent Optimization Algorithms. — URL: https://ruder.io/optimizing-gradient-descent.

Статья представлена к публикации членом редколлегии чл.-корр. РАН Д.А. Новиковым.

Поступила в редакцию 17.12.2019, после доработки 25.12.2019.
Принята к публикации 25.12.2019.

Макаренко Андрей Викторович — канд. техн. наук,
✉ avm.work@mail.ru,

Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва.

DEEP NEURAL NETWORKS: ORIGINS, DEVELOPMENT, CURRENT STATUS

A.V. Makarenko

V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
✉ avm.work@mail.ru

The article covers the development of neural networks, from their origin in the form of the McCulloch—Pitts neuron to modern deep architectures. Major «neural network crises» are listed together with reasons that led to these crises. The main attention is paid to neural architectures that are trained with supervision learning using labeled datasets. References are given to original publications and mathematical theorems that lay the theoretical foundation for artificial neural networks. An analysis was carried out of the challenges in building effective deep neural architectures, ways to address these challenges are considered, success factors are identified. Main layers are listed for convolutional and recurrent neural networks, as well as their architectural combinations. Examples are given with references to demonstrate that deep neural networks are effective not only in applications with distinct structural patterns in the data (images, voice, music, etc.) but also applications with signals of stochastic/chaotic nature. A major direction of convolutional network development is identified too, which is the implementation of trainable integral transforms into the layers. A basic overview is provided for the modern Transformer architecture, which has become mainstream for sequence processing tasks (including tasks of computational linguistics). A scope of key goals and objectives is defined for the current theory of artificial neural networks.

Keywords: deep learning, convolutional neural networks, recurrent neural networks.