

19. NLP | Part of speech tagged – word corpus. Available at: <https://www.geeksforgoeks.org/nlp-part-of-speech-tagged-word-corpus/> (accessed 11 February 2022).
20. Jang B., Kim M., Harerimana G., Kang S-u, Kim J.W. Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism, *Applied Sciences*, 2020; 10(17):5841. Available at: <https://doi.org/10.3390/app10175841>.

Статью рекомендовал к опубликованию д.т.н., профессор А.В. Боженок.

Подоплелова Елизавета Сергеевна – Южный федеральный университет; e-mail: chuzhinova@sfedu.ru; г. Таганрог, Россия; тел.: +79525844188; аспирант.

Podoplelova Elizaveta Sergeevna – Southern Federal University; e-mail: chuzhinova@sfedu.ru; Taganrog, Russia; phone: +79525844188; postgraduate student.

УДК 004.8+004.75

DOI 10.18522/2311-3103-2022-2-189-199

В.В. Курейчик, С.И. Родзин, В.В. Бова

МЕТОДЫ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ОБРАБОТКИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ*

Представлен анализ подходов, основанных на глубоком обучении (DL), к задачам обработки естественного языка (NLP). Исследование охватывает различные задачи NLP, реализованные с помощью искусственных нейронных сетей (ANNs), сверточных нейронных сетей (CNNs) и рекуррентных нейронных сетей (RNNs). Эти архитектуры позволяют решать широкий спектр задач обработки естественного языка, ранее не поддававшихся эффективному решению: моделирование предложений, маркировка семантической роли, распознавание именованных сущностей, ответы на вопросы, категоризация текста, машинный перевод. Наряду с преимуществами использования CNN для решения задач NLP имеются проблемы, связанные с большим числом варьируемых параметров сети и выбором ее архитектуры. Мы предлагаем для оптимизации архитектур сверточных нейронных сетей эволюционный алгоритм. Алгоритм инициализирует случайную популяцию из малого числа агентов (не более 5) и с помощью фитнес функции получает оценки каждого агента в популяции. Затем проводится турнирная селекция между всеми агентами и применяется оператор кроссинговера между wybranными агентами. Алгоритм обладает таким преимуществом как малый размер популяции сетей, он использует несколько типов слоев CNN: сверточный слой, максимальный слой пулинга (субдискретизации), средний слой пулинга и полносвязный слой. Алгоритм тестировался на локальном компьютере с графическим процессором ASUS Cerberus GeForce® GTX 1050 Ti OC Edition 4 GB GDDR5, 8 GB оперативной памяти и процессором Intel(R) Core(TM) i5-4670. Результаты экспериментов показали, что предлагаемый нейроэволюционный подход способен достаточно быстро найти оптимизированную архитектуру CNN для заданного набора данных с приемлемым значением точности. Для завершения выполнения алгоритма потребовалось около 1 часа. Для создания и обучения CNN был использован популярный фреймворк TensorFlow. Для оценки алгоритма использовались общедоступные наборы данных: MNIST и MNIST-RB. Наборы содержали черно-белые изображения рукописных букв и цифр с 50000 обучающими образцами и 10000 тестовыми образцами.

Глубокое обучение; обработка естественного языка; нейронные сети; сверточные нейронные сети; рекурсивные нейронные сети; обучение представлению.

* Исследование выполнено за счет гранта Российского научного фонда № 22-21-00316, <https://rscf.ru/project/22-21-00316/> в Южном федеральном университете.

V.V. Kureichik, S.I. Rodzin, V.V. Bova

DEEP LEARNING METHODS FOR NATURAL LANGUAGE TEXT PROCESSING

The analysis of approaches based on deep learning (DL) to natural language processing (NLP) tasks is presented. The study covers various NLP tasks implemented using artificial neural networks (ANNs), convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These architectures allow solving a wide range of natural language processing tasks that previously could not be effectively solved: sentence modeling, semantic role labeling, named entity recognition, answers to questions, text categorization, machine translation. Along with the advantages of using CNN to solve NLP problems, there are problems associated with a large number of variable network parameters and the choice of its architecture. We propose an evolutionary algorithm for optimizing the architecture of convolutional neural networks. The algorithm initializes a random population of a small number of agents (no more than 5) and uses the fitness function to get estimates of each agent in the population. Then a tournament selection is carried out between all agents and a crossover operator is applied between the selected agents. The algorithm has such an advantage as the small size of the network population, it uses several types of CNN layers: convolutional layer, maximum pooling layer (subdiscretization), medium pooling layer and fully connected layer. The algorithm was tested on a local computer with an ASUS Cerberus GeForce® GTX 1050 Ti OC Edition 4 GB GDDR5, 8 GB of RAM and an Intel(R) Core(TM) i5-4670 processor. The experimental results showed that the proposed neuroevolutionary approach is able to quickly find an optimized CNN architecture for a given data set with an acceptable accuracy value. It took about 1 hour to complete the algorithm execution. The popular TensorFlow framework was used to create and train CNN. To evaluate the algorithm, public datasets were used: MNIST and MNIST-RB. The kits contained black-and-white images of handwritten letters and numbers with 50,000 training samples and 10,000 test samples.

Deep learning; natural language processing; neural networks; convolutional neural networks; recursive neural networks; representation learning.

Введение. Машинное обучение (*Machine Learning, ML*) – это класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Различают обучение по прецедентам, основанное на выявлении эмпирических закономерностей в данных и дедуктивное обучение, предполагающее формализацию знаний экспертов и их перенос в компьютер в виде базы знаний. Дедуктивное обучение обычно относится к области экспертных систем. Поэтому термины машинное обучение и обучение по прецедентам можно считать синонимами.

Общая постановка задачи обучения по прецедентам заключается в следующем. Имеется множество объектов и множество возможных ответов. Существует некоторая зависимость между ответами и объектами, но она неизвестна. Известна только конечная совокупность прецедентов – пар «объект/ответ», называемая обучающей выборкой. На основе этих данных требуется восстановить неявную зависимость, то есть построить алгоритм, способный для любого возможного входного объекта выдать достаточно точный классифицирующий ответ. Эта зависимость не обязательно выражается аналитически, и здесь искусственные нейронные сети (*Artificial Neural Network, ANN*) реализуют принцип эмпирически формируемого решения. Важной особенностью при этом является способность обучаемой системы к обобщению, то есть к адекватному отклику на данные, выходящие за пределы имеющейся обучающей выборки. Для измерения точности ответов вводится оценочный функционал качества.

Данная постановка является обобщением классических задач аппроксимации функций. В классических задачах аппроксимации объектами являются действительные числа или векторы. В задачах обработки лингвистической информации входные данные являются текстовыми. Эти особенности приводят к большому

разнообразию методов глубокого обучения: с учителем, без учителя, с подкреплением. Методы глубокого обучения основаны на обучении представлениям, а не на специализированных алгоритмах под конкретные задачи. Эти методы были известны ещё в 1980-е годы, но результаты были невпечатляющими, пока продвижения в теории искусственных нейронных сетей и вычислительные мощности не позволили создавать сложные технологические архитектуры нейронных сетей, обладающие достаточной производительностью. Эти архитектуры позволили решать широкий спектр задач, ранее не поддававшихся эффективному решению, например, в компьютерном зрении, машинном переводе, распознавании речи и обработке естественного языка (*Natural Language Processing, NLP*), причём качество решения во многих случаях теперь сопоставимо, а в некоторых случаях превосходит эффективность человека.

Глубокое обучение (*Deep Learning, DL*) включает в себя несколько уровней обработки данных, которые позволяют машине учиться на данных с помощью различных уровней абстракции для конкретной задачи без вмешательства человека, или используя ранее полученные знания. Исследование пригодности подходов *DL* для задач *NLP* привлекло большое внимание со стороны исследовательских сообществ *ML* и *NLP* [1]. Были достигнуты неплохие результаты при решении различных задач *NLP*, таких как моделирование предложений [2], маркировка семантической роли [3], распознавание именованных сущностей [4], ответы на вопросы [5], категоризация текста [6] и машинный перевод [7].

Проанализируем основные подходы *DL* и перспективы их применения для решения задач *NLP*.

Существующие подходы. *Глубокое обучение и искусственные нейронные сети. ANNs* не программируются в привычном смысле этого слова, они обучаются. Возможность обучения – одно из главных преимуществ *ANN* перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения *ANN* способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных. По характеру связей *ANN* делятся на два типа архитектур: сети прямого распространения и рекуррентные нейронные сети (*RNNs*). В архитектуре сети прямого распространения все связи направлены строго от входных нейронов к выходным. Напротив, в *RNN* сигнал с выходных нейронов или нейронов скрытого слоя частично передаётся обратно на входы нейронов входного слоя (обратная связь). Например, рекуррентная сеть Хопфилда «фильтрует» входные данные, возвращаясь к устойчивому состоянию и позволяет решать задачи компрессии данных и построения ассоциативной памяти.

ANNs, применяемые для решения задач *NLP*, рассматривают синтаксические особенности текста как часть семантического анализа [8]. В [9] были предложены новые модели обучения нейронных сетей, которые применялись к различным задачам обработки текста на естественном языке, таким как маркировка семантических ролей и распознавание именованных сущностей. Преимущество этих подходов состоит в том, что удалось избежать необходимости в предварительном обучении. Применение *ANN* позволило достигнуть высокой производительности при маркировке с использованием небольших вычислительных ресурсов [3]. В [2] предлагается динамическая сверточная нейронная сеть (*Dynamic Convolutional Neural Network, DCNN*) для решения задачи семантического моделирования предложений, которая включает в себя извлечение признаков из предложений. *DCNN*

обучается путем выборочного взвешивания функций между связанными слоями сети. В [10] был предложен подход для контроля качества изображений. Этот подход сочетает в себе *NLP* и представление изображений для текстовой системы ответов на вопросы, касающихся изображений. Однако эта система давала неверные ответы на вопросы относительно размера объектов на изображении, а также на вопросы, связанные с пространственным мышлением (“какой объект самый большой?”). В исследовании [11] был предложен новый подход для сопоставления запросов с возможными ответами (предложениями) в зависимости от их семантического значения. Этот подход использует глубокое обучение без необходимости привлечения лингвистических знаний. Отметим, что современные распределительные модели на основе нейронных сетей достигли высокой производительности при решении таких задач *NLP* как классификация документов [12], устранение неоднозначности и распознавание сущностей [9].

В исследовании [5] используется *RNN* в системе, способная отвечать на вопросы о параграфах. В [13] была предложена нейронная машина для генерации коротких текстовых диалогов. Модели *RNN* демонстрируют неплохую производительность при классификации различных мнений [12, 14] и ответах на вопросы [5]. Модель адаптивной *RNN* (*ADARNN*) для классификации различных мнений используется в рекламных акциях *Twitter* [15].

Другим подходом к решению задач *NLP* является модель *ANNABEL* [7]. Эта модель может обрабатывать вербальный и невербальный естественный язык без предварительных знаний.

Глубокое обучение и сверточные нейронные сети. Сверточные нейронные сети (*Convolutional Neural Network, CNN*) первоначально развивалась в области исследований машинного зрения. Первое применение *CNN* связано с задачами классификации изображений, обучения обнаружению краев изображения для идентификации таких объектов как лицо человека [16]. Идея *CNN* заключается в чередовании сверточных слоев и субдискретизирующих слоев. Структура сети – однонаправленная, принципиально многослойная. Название архитектура сети получила из-за наличия операции свертки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения. Работа *CNN* интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям, и далее к еще более абстрактным деталям вплоть до выделения понятий высокого уровня. При этом сеть самонастраивается и вырабатывает необходимую иерархию абстрактных признаков (последовательности карт признаков), фильтруя маловажные детали и выделяя существенное.

Подобная интерпретация носит скорее метафорический или иллюстративный характер. Фактически «признаки», вырабатываемые сложной *CNN*, малопонятны и трудны для интерпретации настолько, что на практике суть этих признаков даже не пытаются понять, тем более «подправлять», а вместо этого для улучшения результатов распознавания меняют структуру и архитектуру сети. Так, игнорирование системой каких-то существенных явлений может говорить о том, что либо не хватает данных для обучения, либо структура сети обладает недостатками, и система не может выработать эффективных признаков для данных явлений.

Типовая архитектура *CNN* представлена на рис. 1. Сеть состоит из большого количества слоев. После начального слоя (входного изображения) сигнал проходит серию сверточных слоев, в которых чередуется собственно свертка и субдискретизация (пулинг).

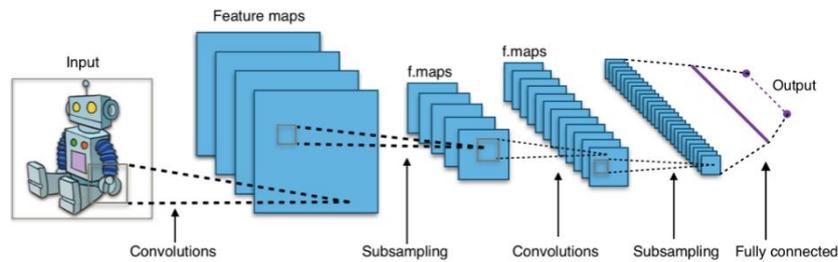


Рис. 1. Типовая архитектура CNN

Чередование слоев позволяет составлять «карты признаков» из карт признаков. На каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов. На практике это означает способность распознавания сложных иерархий признаков. Обычно после прохождения нескольких слоев карта признаков вырождается в вектор или даже скаляр, но таких карт признаков становятся сотни. На выходе *CNN* дополнительно устанавливают несколько слоев полносвязной нейронной сети (перцептрон), на вход которому подаются окончательные карты признаков. В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотненной карты признаков меньшей размерности. За счет данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к масштабу входного изображения.

Слой свертки – это основной блок *CNN*. Слой свертки включает в себя для каждого канала свой фильтр, ядро свертки которого обрабатывает предыдущий слой по фрагментам (суммируя результаты поэлементного произведения для каждого фрагмента). Весовые коэффициенты ядра свертки (небольшой матрицы) неизвестны и устанавливаются в процессе обучения. Особенностью сверточного слоя является сравнительно небольшое количество параметров, устанавливаемое при обучении. Так например, если исходное изображение имеет размерность 100×100 пикселей по трём каналам (это значит 30 000 входных нейронов), а сверточный слой использует фильтры с ядром 3×3 пикселя с выходом на 6 каналов, тогда в процессе обучения определяется только 9 весов ядра, однако по всем сочетаниям каналов, то есть $9 \times 3 \times 6 = 162$, в таком случае данный слой требует нахождения только 162 параметров, что существенно меньше количества искомого параметров полносвязной нейронной сети.

Скалярный результат каждой свертки попадает на функцию активации, которая представляет собой некую нелинейную функцию. Слой активации обычно логически объединяют со слоем. В качестве функции активации используются гиперболический тангенс ($f(x) = \tanh(x)$), сигмоида ($f(x) = (1 + e^{-x})^{-1}$) или *ReLU*, которая позволяет существенно ускорить процесс обучения и одновременно упростить вычисления. По состоянию на 2021 год эта функция, ее модификации *Noisy ReLU*, *Leaky ReLU* и другие являются наиболее часто используемыми функциями активации в *CNN*.

Таким образом ключевым преимуществом использования архитектуры *CNNs* для задач *NLP* является применение нелинейной функции к последовательности слов путем перемещения окна по предложению. Эта функция, которая также называется "фильтром", преобразует входные данные (окно k -слов) в d -мерный вектор, состоящий из значимых характеристик слов в окне. Затем применяется операция объединения векторов, полученных из разных каналов, в один n -мерный вектор.

В настоящее время модели *CNN* используются для решения таких задач классификации *NLP* как семантическая маркировка ролей [3], классификация типов вопросов [17], категоризации текста [6].

Отметим также исследование [18], в котором модель *mQA CNNs* используется для визуализации ответов на вопросы о содержимом изображения. Эта модель состоит из памяти для извлечения представления запроса, памяти для обработки контекста ответов и модуля генерации ответа, который собирает информацию из памяти для предоставления ответов. Ответ может быть в форме слова, фразы или предложения. В [19, 14] *CNNs* применяется для решения проблемы неоднозначности при семантической кластеризации как коротких, так и длинных текстов.

Таким образом, преимущества использования *CNN* для решения задач *NLP* состоят в следующем:

- ◆ по сравнению с полносвязной нейронной сетью требуется гораздо меньшее количество настраиваемых весов, так как одно ядро весов используется целиком для всего изображения, вместо того чтобы искать для каждого пикселя входного изображения свои персональные весовые коэффициенты. Это подталкивает нейросеть при обучении к обобщению демонстрируемой информации, а не попиксельному запоминанию каждой показанной картинке в мириадах весовых коэффициентов, как это делает перцептрон;
- ◆ распараллеливание вычислений, а следовательно, возможность реализации алгоритмов работы и обучения сети на графических процессорах;
- ◆ относительная устойчивость к повороту и сдвигу распознаваемого изображения;
- ◆ обучение при помощи классического метода обратного распространения ошибки.

К недостаткам *CNN* можно отнести большое число варьируемых параметров сети: количество слоев, размерность ядра свертки для каждого из слоев, количество ядер для каждого из слоев, шаг сдвига ядра при обработке слоя, необходимость слоев субдискретизации, степень уменьшения ими размерности, функция по уменьшению размерности (выбор максимума, среднего и т. п.), передаточная функция нейронов, наличие и параметры выходной полносвязной нейросети на выходе сверточной. Все эти параметры существенно влияют на результат, но выбираются исследователями эмпирически. Пока не существует выверенных и работающих рекомендаций, по которым нужно строить оптимальную архитектуру сверточной нейронной сети для новой задачи.

Предлагается биоинспирированный алгоритм для оптимизации архитектуры *CNN*.

Оптимизация архитектуры сверточных нейронных сетей с помощью эволюционного алгоритма. Рассмотрим нейроэволюционный подход для решения существующих *CNN* проблем.

Нейроэволюция – это метод машинного обучения, который применяет эволюционные алгоритмы для построения и обучения нейронных сетей [20]. В нейроэволюции моделируются базовые положения теории биологической эволюции – процессы отбора, мутации и воспроизводства. Поведение агентов определяется окружающей средой. Множество агентов принято называть популяцией. Такая популяция эволюционирует в соответствии с правилами отбора и целевой функцией, задаваемой окружающей средой. Таким образом, каждому агенту популяции назначается значение его пригодности в окружающей среде. Размножаются только наиболее пригодные виды. Рекомбинация и мутация позволяют изменяться агентам и приспособляться к среде.

В нейроэволюции генотип – это представление нейросети. В схеме с прямым кодированием генотип эквивалентен фенотипу, нейроны и связи непосредственно указаны в генотипе. Напротив, в схеме с непрямым кодированием в генотипе указаны правила и структуры для создания нейросети. Процесс нейроэволюции сначала инициализирует набор этих геномов, а затем применяет их к проблемной среде. Каждому геному присваивается фитнес оценка, основанная на том, насколько хорошо нейронная сеть решает прикладная задача. Например, этот показатель пригодности может быть функцией точности, достигнутой в задаче обработки лингвистической информации.

Эволюционные сверточные нейронные сети (*Evolutionary CNN, ECNN*) в дополнение к обучению используют эволюцию для оптимизации архитектуры *CNN*. Нейроэволюция – это агностический процесс, который не предусматривает какой-либо конкретной топологии или гиперпараметров для генерируемых нейронных сетей. Нейросеть эволюционирует на основе обратной связи о том, насколько эффективно она работает в проблемной среде.

Предлагаемый эволюционный алгоритм работает следующим образом. В начале он инициализирует случайную популяцию из малого числа агентов (не более 5) и с помощью фитнес функции получает оценки каждого агента в популяции. Затем проводится турнирная селекция между всеми агентами и с вероятностью 1 применяется оператор кроссинговера между выбранными агентами. В следующем поколении сохраняется лучший агент из прошлого поколения (стратегия элитизма). Таким образом, есть гарантия того, что информация о хорошей *CNN* не будет потеряна.

В предлагаемом алгоритме используются несколько типов слоев *CNN*: сверточный слой, максимальный слой пулинга (субдискретизации), средний слой пулинга и полносвязный слой. Позиции в списке слоев в хромосоме содержат информацию о типе слоя, количестве выходных карт, количестве нейронов, а также о размере ядра. Первым шагом предлагаемого алгоритма является создание популяции из малого числа *CNN*, удовлетворяющих следующим ограничениям:

- ◆ в архитектуре *CNN* должно быть не менее трех слоев;
- ◆ первый слой должен быть сверточным;
- ◆ выходной слой должен быть полносвязным слоем с количеством нейронов, равным количеству классов;
- ◆ в сверточном слое количество фильтров (выходных карт объектов) не должно превышать заранее установленного максимального количества;
- ◆ количество слоев пулинга не должно превышать заданную границу.

Следующим шагом алгоритма является вычисление фитнес функции каждого агента в популяции, турнирная селекция и применение оператора кроссинговера между *CNN*. Худший агент заменяется на лучший и проверяется условие останова алгоритма.

Предложенный алгоритм был протестирован на локальном компьютере с графическим процессором *ASUS Cerberus GeForce® GTX 1050 Ti OC Edition* 4 Гб *GDDR5*, 8 Гб оперативной памяти и процессором *Intel(R) Core(TM) i5-4670*. Для завершения выполнения алгоритма потребовалось около 1 часа. Для создания и обучения *CNN* был использован популярный фреймворк *TensorFlow* [21]. Для оценки алгоритма использовались общедоступные наборы данных: *MNIST* [22] и *MNIST-RB* [23]. *MNIST* содержит черно-белые изображения рукописных букв и цифр с 50 000 обучающими образцами и 10 000 тестовыми образцами. *MNIST-RB* является модифицированной версией *MNIST*, потому что проверяет, насколько хорошо *CNN* может обобщать данные, содержащие не относящуюся к проблеме информацию. Все они содержат 12 000 обучающих образцов и 50 000 тестовых образцов.

Предлагаемый для оптимизации архитектур сверточных нейронных сетей эволюционный алгоритм обладает явным преимуществом – малый размер популяции, которую использует алгоритм для оптимизации. Это важно, поскольку из-за сложности *CNN* для обучения одной сети требуется значительное время. Результаты экспериментов показывают, что предлагаемый нейроэволюционный подход способен достаточно быстро найти оптимизированную архитектуру *CNN* для любого заданного набора данных. Алгоритм позволяет находить архитектуру *CNN*, которая имеет приемлемые значения точности.

Дискуссия и будущие исследования. Создание системы машинного обучения требует специальных знаний в предметной области для разработки модели классификатора, преобразования необработанных данных во внутренние входные данные представления или векторы. Это называется обучением представлению (*Representation Learning, RL*), при котором модель автоматически подает необработанные данные для определения необходимого представления.

В частности, способность точно представлять слова, фразы, предложения (утверждение или вопрос) на естественном языке, а также их классификацию, имеет важное значение для понимания языка. Подходы к глубокому обучению аналогичны методам *RL*, использующим несколько уровней репрезентативной обработки [1].

Эти подходы могут быть объединены с другими моделями *DL* для повышения производительности задач *NLP*. Нейронные сети прямой связи могут обеспечить конкурентоспособную производительность в задачах *NLP* с последовательным вводом. Однако, если проблема связана с разреженностью входных данных, то перспективнее ее решать с помощью *CNNs*. Реализация *CNN* чаще всего используется в таких задачах *NLP*, как классификация предложений, категоризация текста, семантическая кластеризация и ответы на вопросы. Однако есть проблема с выбором оптимальной архитектуры сверточной нейронной сети для новой задачи. Эту проблему можно решить, используя предлагаемый выше эволюционный алгоритм, отличительной чертой которого является малый размер популяции *CNNs*, которую использует алгоритм для оптимизации.

Методы *DL* обеспечивают повышенную производительность для задач *NLP* не только благодаря современным процессорам. Методы *DL* используют многослойную систему нелинейных фильтров для извлечения признаков с преобразованием. Каждый последующий слой получает на входе выходные данные предыдущего слоя. Система глубокого обучения обладает несколькими слоями выявления признаков или параметров представления данных. При этом признаки организованы иерархически, признаки более высокого уровня являются производными от признаков более низкого уровня. В процессе обучения слои формируются на нескольких уровнях представлений, которые соответствуют различным уровням абстракции; слои образуют иерархию понятий.

Состав и количество конкретных нелинейных слоев зависит от решаемой проблемы. Й. Шмидхубер выделяет «очень глубокое обучение», когда число слоев больше десяти.

Мы ожидаем, что модели *RNN*, имеющие дело с последовательностями входных данных, окажут большое влияние на понимание естественного языка в течение ближайших нескольких лет, особенно при решении таких задач, как текстовый разговор, ответы на вопросы, анализ высказанных мнений.

Обзор задач *NLP* с использованием подходов *DL* для семантического анализа текста с помощью *ANN*, *CNN* и *RNN* представлен в табл. 1.

Таблица 1

Обзор задач *NLP* с использованием подходов *DL* для семантического анализа текста

Задачи понимания текста	ANNs	CNNs	RNNs
Классификация предложений		√	
Распознавание именованных сущностей	√		
Категоризация текста		√	
Маркировка семантической роли		√	
Семантическая кластеризация		√	
Короткий текстовый разговор	√		√
Ответ на вопрос	√	√	√
Анализ мнений	√		√
Обнаружение перефразирования	√		
Классификация документов	√		
Категоризация тем		√	
Прогнозирование выражения мнения			√

Заключение. В статье анализируются подходы, основанные на глубоком обучении, к задачам обработки естественного языка. Представлены исследования, охватывающие различные задачи *NLP*, реализованные с помощью *ANNs*, *CNNs* и *RNNs*. Эти архитектуры позволили решать широкий спектр задач, ранее не поддававшихся эффективному решению, например, в компьютерном зрении, машинном переводе, распознавании речи и обработке естественного языка. Были достигнуты неплохие результаты при решении различных задачах *NLP*, таких как моделирование предложений, маркировка семантической роли, распознавание именованных сущностей, ответы на вопросы, категоризация текста и машинный перевод.

К преимуществам использования *CNN* для решения задач *NLP* относятся гораздо меньшее количество настраиваемых весов по сравнению с полносвязной нейронной сетью, возможность реализации алгоритмов работы и обучения сети на графических процессорах, относительная устойчивость к повороту и сдвигу распознаваемых объектов, возможность обучения при помощи метода обратного распространения ошибки. К недостаткам *CNN* относится большое число варьируемых параметров сети, выбираемых исследователями эмпирически. Мы предлагаем для оптимизации архитектур сверточных нейронных сетей эволюционный алгоритм, обладающий таким преимуществом как малый размер популяции сетей. Это оказалось важным, поскольку из-за сложности *CNN* для обучения одной сети требуется значительное время. Результаты экспериментов показали, что предлагаемый нейроэволюционный подход способен достаточно быстро найти оптимизированную архитектуру *CNN* для заданного набора данных с приемлемым значением точности.

Мы не обсуждали вопросы сравнения производительности нейронных сетей, связанные с методами глубокого обучения, используемыми для задач обработки естественного языка. Это следует всесторонне изучить в рамках будущих исследований.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ba L. & Caurana R. Do Deep Nets Really Need to be Deep?, *arXiv preprint arXiv:1312.6184*, 2013, 521(7553), pp. 1-6.
2. Kalchbrenner N., Grefenstette E., & Blunsom P. A Convolutional Neural Network for Modeling Sentences, *In Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore, MD, USA*, 2014, Vol. 1, pp. 655-665.

3. Collobert R., Weston J., Bottou L., Karlen M., Kavukcuglu K., Kuksa P. Natural Language Processing from Scratch, *Journal of Machine Learning Research*, 2011, 12:2493-2537.
4. Santos C.N., Dos & Guimarães V. Boosting Named Entity Recognition with Neural Character Embeddings, *ACL'2015*, pp. 25-33.
5. Iyyer M., Boyd-Graber J., Claudino L., Socher R. A neural network for factoid question answering over paragraphs, *In EMNLP*, 2014.
6. Johnson R. & Zhang T. Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding, 2015. pp. 1-12.
7. Golosio B., Cangelosi A., Gamotina O. & Masala G.L. A Cognitive Neural Architecture Able to Learn and Communicate through Natural Language, *Plos-One*, 2015, 10 (11).
8. Weston J., America N.E. C.L. & Way I. A Unified Architecture for Natural Language, *Processing: Deep Neural Networks with Multitask Learning*, 2008, pp. 160-167.
9. Sun Y., Lin L., Tang D., Yang N., Ji Z. & Wang X. Modelling Mention, Con-text and Entity with Neural Networks for Entity Disambiguation, 2015. pp. 1333-1339.
10. Malinowski M., Rohrbach M. & Fritz M. 2015. Ask Your Neurons: A Neural based Approach to Answering Questions about Images, *IEEE Int. Conf. on Computer Vision*, 2015, pp. 1-9.
11. Yu L., Hermann K.M., Blunsom P. & Pulman S. Deep Learning for Answer Sentence Selection, *NIPS Deep Learning Workshop*, 2014, 9.
12. Karl Moritz Hermann and Phil Blunsom. Multilingual Models for Compositional Distributional Semantics, *In Proceedings of ACL*, 2014.
13. Shang L., Lu Z. & Li H. Neural Responding Machine for Short-Text Conversation, *ACL'2015*, 2015. pp. 1577-1586.
14. Wang P., Xu J., Xu B., Liu C., Zhang H., Wang F. & Hao H. Semantic Cluster-ing and Convolutional Neural Network for Short Text Categorization. *Proceedings ACL'2015*, 2015, pp. 352-357.
15. Dong L., Wei F., Tan C., Tang D., Zhou M., & Xu K. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification, *ACL'2014*, 2014, pp. 49-54.
16. LeCun Y. and Bengio, YConvolutional networks for images, speech, and time-series. MIT Press, 1995, pp. 255-258.
17. Kim Y. Convolutional Neural Networks for Sentence Classification, *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'2014)*, 2014, pp. 1746-1751.
18. Gao H., Mao J., Zhou J., Huang Z., Wang L. & Xu W. Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering, *Arxiv*, 2015, pp. 1-10.
19. Shen Y., He X., Gao J., Deng L. & Mesnil G. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval, *Proc. of the 23rd ACM Int. Conf. on CIKM'2014*, 2014), pp. 101-110.
20. Rodzin S., Rodzina O., Rodzina L. Neuroevolution: Problems, algorithms, and experiments, *Proc. of the 10th IEEE Int. Conf. on Application of Information and Communication Technologies (AICT'2016)*, 2016, 7991745.
21. Alshahrani S., and Kapetanios E. Are Deep Learning Approaches Suitable for Natural Language Processing?, *Proc. of the 21st Int. Conf. on Applications of Natural Language to Information Systems (NLDB'2016)*, 2016. DOI: 10.1007/978-3-319-41754-7_33.
22. LeCun Y., et al. Back propagation applied to handwritten zip code recognition, *Neural Comput*, 1989, 1 (4), pp. 541-551.
23. Larochelle H., Erhan D., Courville A., Bergstra J., Bengio Y. An empirical evaluation of deep architectures on problems with many factors of variation, *in ACM International Conf. Proc. Series*, 2007, Vol. 227, No. 2006, pp. 473-480.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

Курейчик Владимир Викторович – Южный федеральный университет; e-mail: vkur@sfedu.ru; г. Таганрог, Россия; тел.: 88634371651; кафедра систем автоматизированного проектирования; зав. кафедрой САПР; д.т.н.; профессор.

Родзин Сергей Иванович – e-mail: srodzin@sfedu.ru; тел.: 88634371673; кафедра математического обеспечения и применения ЭВМ; к.т.н.; профессор.

Бова Виктория Викторовна – e-mail: bova@sfedu.ru; кафедра систем автоматизированного проектирования; доцент.

Kureichik Vladimir Victorovich – Southern Federal University; e-mail: vkur@sfnu.ru; Taganrog, Russia; phone: +78634371651; the department of computer aided design; head of department; dr. of eng. sc.; professor.

Rodzina Sergey Ivanovich – e-mail: srodzin@sfnu.ru; phone: +78634371673; the department of software engineering; cand. of eng. sc.; professor.

Bova Viktoriya Viktorovna – e-mail: bova@sfnu.ru; the department of computer aided design; associate professor.

УДК 004.42+004.3; 57.08+615.47+615.8 DOI 10.18522/2311-3103-2022-2-199-212

А.В. Проскуряков

**СИНТЕЗ ПРОГРАММНОГО И ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ
РЕАЛИЗАЦИИ МЕТОДОВ ВЕРИФИКАЦИИ СОСТОЯНИЯ
МЕДИЦИНСКИХ БИОЛОГИЧЕСКИХ ОБЪЕКТОВ ДЛЯ МЕДИЦИНСКОЙ
АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Описаны информационное и программное обеспечение реализации различных методов верификации состояния фрагментов биологических объектов по компьютерно-томографическим изображениям подсистемой поддержки принятия решения для диагностики заболеваний. Указано на современное состояние развития медицинской диагностической техники, оснащённость которой медицинских учреждений страны и не оперативная её доступность для населения способствовало и привело к появлению и активному развитию новых направлений в области лучевой диагностики, к которым относятся: цифровая и пленочная рентгенография, компьютерная томография, магнитно-резонансная томография. В статье акцент сделан на анализ рентгенографических изображений, принятие решений на основании анализа этих изображений, постановка диагноза на основании принятых решений. Сделан анализ преимуществ и недостатков рентгенографии, как современного способа диагностики, относительно своих аналогов. Важной задачей при анализе рентгенографических изображений медицинских биологических объектов и их фрагментов является решение задачи улучшения качества изображения. С целью улучшения качества рентгеновских снимков и повышения их информативности разработан алгоритм и реализовано программное обеспечение подсистемы программного обеспечения медицинской автоматизированной информационной системы для их коррекции и анализа. В статье рассматривается реализация решения задач диагностики заболеваний, таких как: анализ рентгенографических изображений, принятие решений на основании анализа этих изображений, постановка диагноза на основании принятых решений путём разработки и применения программного и информационного обеспечения реализации методов верификации состояния фрагментов биологических объектов, как эффективных методов диагностики состояния параназальных пазух по их рентгенографическим и компьютерно-томографическим изображениям. Описаны основные методы, лежащие в основе верификации по рентгеновским и компьютерно-томографическим изображениям. Приведен детальный анализ реализации математических моделей методов диагностики в виде алгоритмов, реализуемых программным обеспечением для функционирования подсистемы поддержки принятия решения медицинской автоматизированной информационной системы. Показаны примеры практической реализации программного и информационного обеспечения методов верификации медицинских объектов в виде экранных форм для работы с фрагментами исследуемого объекта и результатами анализа рентгенографических изображений. Это позволяет повысить оперативность, точность верификации состояния медицинских биологических объектов, достоверность процесса диагностики заболеваний. Показана научная новизна, результаты апробации материала, представленного в статье на международных, всероссийских конференциях, научных журналах.

Биологический; верификация; безэталонный метод верификации; диагностика; информация; медицинский; метод; метод эталонный; статистика; рентгенография; томография; энтропия; эталон.