

УДК 004.855

5.2.2. Математические, статистические и инструментальные методы экономики (физико-математические науки, экономические науки)

АНАЛИЗ ВЛИЯНИЯ НЕЙРОННЫХ СЕТЕЙ НА ПРОГРАММИРОВАНИЕ НА PYTHON

Яцкевич Евгений Сергеевич
магистрант
es_yatskevich@internet.ru
ФГБОУ ВО “Кубанский государственный технологический университет”, Краснодар, Россия

Кушнир Надежда Владимировна
старший преподаватель кафедры информационных систем и программирования
РИНЦ SPIN-код=6951-4012
kushnir.06@mail.ru
ФГБОУ ВО “Кубанский государственный технологический университет”, Краснодар, Россия

Тотухов Константин Евгеньевич
доцент кафедры информационных систем и программирования
101KE@mail.ru
ФГБОУ ВО “Кубанский государственный технологический университет”, Краснодар, Россия

Власенко Александра Владимировна
кандидат технических наук, доцент
alex_vlasenko@list.ru
Краснодарский университет МВД России, 350005, Российская Федерация, г. Краснодар, ул. Ярославская 128

Оганян Армен Робертович
магистрант
oganyan21v@mail.ru
ФГБОУ ВО “Кубанский государственный технологический университет”, Краснодар, Россия

Высоцкий Владислав Алексеевич
магистрант
vysotskiy_va@mail.ru
ФГБОУ ВО “Кубанский государственный технологический университет”, Краснодар, Россия

UDC 004.855

5.2.2. Mathematical, statistical and instrumental methods of economics (physical and mathematical sciences, economic sciences)

ANALYSIS OF THE INFLUENCE OF NEURAL NETWORKS ON PYTHON PROGRAMMING

Yatskevich Evgeniy Sergeevich
master student
es_yatskevich@internet.ru
FGBOU VO “Kuban State Technological University”, Krasnodar, Russia

Kushnir Nadezhda Vladimirovna
senior Lecturer in the department of information systems and programming
RSCI SPIN-code=6951-4012
kushnir.06@mail.ru
FGBOU VO “Kuban State Technological University”, Krasnodar, Russia

Totukhov Konstantin Evgenievich
Associate Professor of the Department of Information Systems and Programming
101KE@mail.ru
Kuban State Technological University, Krasnodar, Russia

Vlasenko Alexandra Vladimirovna
Cand.Tech.Sci., associate Professor
alex_vlasenko@list.ru
Krasnodar University of the Ministry of Internal Affairs of Russia, 128 Yaroslavskaya, Krasnodar, 350005, Russia

Oganyan Armen Robertovich
master student
oganyan21v@mail.ru
FGBOU VO “Kuban State Technological University”, Krasnodar, Russia

Vysotskiy Vladislav Alexeevich
master student
vysotskiy_va@mail.ru
FGBOU VO “Kuban State Technological University”, Krasnodar, Russia

Данная научная статья посвящена исследованию влияния нейронных сетей на процесс программирования на языке Python. Авторы исследуют возможности использования нейронных сетей для улучшения процесса разработки и оптимизации программного кода на Python. Статья описывает основные принципы работы нейронных сетей и их потенциальное применение в программировании. Авторы рассматривают различные подходы к использованию нейронных сетей для улучшения производительности и эффективности программирования на Python, включая оптимизацию алгоритмов, автоматическое исправление ошибок и улучшение процесса отладки. Исследование показывает, что использование нейронных сетей может значительно улучшить работу программистов и повысить качество разработки программного обеспечения на Python. Данная статья может быть полезна как профессионалам в области искусственного интеллекта и нейронных сетей, так и разработчикам, интересующимся новыми подходами к программированию.

Ключевые слова: НЕЙРОСЕТЬ, PYTHON, ФРЕЙМВОРК, ПРОГРАММИРОВАНИЕ, NUMPY, БИБЛИОТЕКИ PYTHON, PYTORCH, TENSORFLOW, АПИ-ИНТЕРФЕЙСЫ

This scientific article is devoted to the study of the influence of neural networks on the programming process in Python. The authors explore the possibilities of using neural networks to improve the development process and optimize Python code. The article describes the basic principles of neural networks and their potential application in programming. The authors consider various approaches to using neural networks to improve the performance and efficiency of Python programming, including optimization of algorithms, automatic error correction and improvement of the debugging process. The study shows that the use of neural networks can significantly improve the work of programmers and improve the quality of software development in Python. This article can be useful for both professionals in the field of artificial intelligence and neural networks, as well as developers interested in new approaches to programming.

Keywords: NEURAL NETWORK, PYTHON, FRAMEWORK, PROGRAMMING, NUMPY, PYTHON LIBRARIES, PYTORCH, TENSORFLOW, APIS

<http://dx.doi.org/10.21515/1990-4665-198-042>

Введение. Эволюция и рост использования нейронных сетей в программировании на Python сыграли важную роль в формировании ландшафта искусственного интеллекта и машинного обучения.

1. Появление Python как языка нейронных сетей:

- Рост Python как универсального и удобного для пользователя языка программирования, создающего благоприятную среду для научных вычислений и анализа данных.

- Разработка библиотек и фреймворков, таких как NumPy, SciPy и Pandas, которые заложили основу для использования Python в сфере машинного обучения и нейронных сетей.

2. Новаторские библиотеки и фреймворки нейронных сетей:

- Внедрение базовых библиотек нейронных сетей в Python, включая Theano и Caffe, которые обеспечивали раннюю поддержку моделей

<http://ej.kubagro.ru/2024/04/pdf/42.pdf>

глубокого обучения.

- Ключевая роль TensorFlow, среды машинного обучения с открытым исходным кодом, разработанной Google, в демократизации реализации нейронных сетей на Python [3].

3. Демократизация развития нейронных сетей:

- Доступность и простота использования Python снизили входные барьеры для начинающих специалистов по данным и специалистов по машинному обучению, способствуя формированию активного сообщества разработчиков.

- Влияние фреймворков нейронных сетей высокого уровня, таких как Keras, построенных на основе TensorFlow, на упрощение создания и обучения моделей нейронных сетей.

4. Интеграция нейронных сетей с экосистемой Python:

- Бесшовная интеграция нейронных сетей с другими библиотеками и инструментами Python, такими как Matplotlib для визуализации данных и блокноты Jupyter для интерактивного исследования данных и разработки моделей.

- Рост интернет-сообществ, форумов и образовательных ресурсов, посвященных разработке нейронных сетей на Python, способствующих распространению знаний и повышению квалификации.

5. Улучшения в масштабируемости и производительности:

- Эволюция Python в устранении узких мест в производительности за счет внедрения оптимизированных библиотек числовых вычислений и возможностей параллельной обработки, повышающих масштабируемость реализаций нейронных сетей.

- Эволюция решений аппаратного ускорения, таких как интеграция поддержки графических процессоров в структуры нейронных сетей, приводит к значительному увеличению производительности в приложениях глубокого обучения на основе Python.

6. Принятие в промышленности и реальное применение:

- Распространение приложений нейронных сетей на основе Python в различных отраслях, включая здравоохранение, финансы, автономные транспортные средства и обработку естественного языка, демонстрируя практическое влияние разработки нейронных сетей на Python.

Изучая эволюцию и рост использования нейронных сетей в программировании на Python, эта статья призвана обеспечить всестороннее понимание исторического контекста, который сформировал влияние нейронных сетей на Python. Эта историческая перспектива создает основу для дальнейшего анализа влияния и последствий нейронных сетей на экосистему программирования Python.

Вехи и ключевые разработки в области интеграции нейронных сетей с Python существенно сформировали ландшафт искусственного интеллекта и машинного обучения. В этом разделе научной статьи будут освещены основные достижения и важные вехи, которые способствовали широкому распространению и влиянию нейронных сетей в программировании на Python [5].

1. Появление Python как языка нейронных сетей:

- Появление Python как популярного языка программирования для научных вычислений, анализа данных и машинного обучения, обусловленное его простотой, выразительностью и богатой экосистемой библиотек.

- Фундаментальная роль библиотек Python, включая NumPy, SciPy и Matplotlib, в предоставлении необходимых инструментов для численных вычислений, манипулирования данными и визуализации, закладывающих основу для разработки нейронных сетей.

2. Введение новаторских структур нейронных сетей:

- Выпуск Theano, библиотеки числовых вычислений с открытым исходным кодом, которая облегчила разработку моделей глубокого

обучения и символьных математических операций, ознаменовав раннюю веху в поддержке нейронных сетей для Python.

- Появление Caffe, системы глубокого обучения, разработанной Центром видения и обучения Беркли, предлагающей удобный интерфейс и предварительно обученные модели для классификации изображений и других задач компьютерного зрения.

3. TensorFlow: революция в разработке нейронных сетей на Python:

- Запуск TensorFlow, платформы машинного обучения с открытым исходным кодом, разработанной Google, ознаменовал важный поворотный момент в демократизации разработки нейронных сетей на Python, предоставив масштабируемую и гибкую платформу для построения и обучения моделей глубокого обучения.

- Полный набор инструментов TensorFlow, включая TensorFlow Extended (TFX) для сквозных конвейеров машинного обучения и TensorFlow Lite для мобильных и встраиваемых устройств, укрепил свое влияние на интеграцию нейронных сетей с Python.

4. Расцвет Кераса и абстракций высокого уровня:

- Создание Keras, библиотеки нейронных сетей с открытым исходным кодом, разработанной для простоты использования и модульности, которая позже стала частью TensorFlow, упрощая разработку и развертывание моделей нейронных сетей на Python.

- Внедрение абстракций высокого уровня для разработки нейронных сетей, таких как последовательные и функциональные API-интерфейсы Keras, что дает разработчикам возможность быстро создавать прототипы и выполнять итерации архитектур глубокого обучения [10].

5. Общественные и образовательные инициативы:

- Распространение онлайн-сообществ, форумов и образовательных ресурсов, посвященных разработке нейронных сетей на Python, способствующих обмену знаниями, лучшими практиками и совместному

обучению.

- Создание образовательных платформ, таких как Coursera, Udacity и edX, предлагающих курсы и специализации по глубокому обучению с использованием Python, способствующих распространению опыта работы в области нейронных сетей среди глобальной аудитории.

6. Принятие в промышленности и реальное применение:

- Внедрение приложений нейронных сетей на основе Python в различных отраслях, включая здравоохранение, финансы, электронную коммерцию и автономные системы, демонстрируя практическое влияние нейронных сетей, интегрированных с Python, в реальных сценариях.

- Заметные достижения в области обработки естественного языка (НЛП) и компьютерного зрения, обусловленные решениями нейронных сетей на основе Python, демонстрируют преобразующее влияние этих технологий на отрасли промышленности [1-2].

Освещая эти вехи и ключевые события в интеграции нейронных сетей с Python, эта статья призвана проиллюстрировать исторический прогресс, который привел к глубокому влиянию нейронных сетей на программирование на Python. Эта историческая перспектива служит основой для более глубокого анализа влияния и последствий нейронных сетей в экосистеме Python.

Влияние библиотек и фреймворков нейронных сетей на программирование на Python было глубоким, произведя революцию в области искусственного интеллекта и машинного обучения. В этом разделе будет рассмотрено преобразующее влияние этих инструментов, подчеркнута их значимость в формировании роли Python как доминирующей платформы для разработки нейронных сетей.

1. Демократизация глубокого обучения:

- Библиотеки и фреймворки нейронных сетей, такие как TensorFlow, PyTorch и Keras, демократизировали глубокое обучение, предоставляя

высокоуровневые API и абстракции, которые упрощают реализацию сложных архитектур нейронных сетей на Python.

- Эти платформы дали разработчикам, исследователям и специалистам по данным возможность экспериментировать с передовыми моделями глубокого обучения, способствуя инновациям и ускоряя внедрение нейронных сетей в сообществе Python.

2. Масштабируемость и производительность:

- Интеграция библиотек нейронных сетей с Python значительно повысила масштабируемость и производительность приложений машинного обучения. Такие платформы, как TensorFlow и PyTorch, используют оптимизированные вычислительные серверные части, включая ускорение графического процессора, для ускорения процессов обучения и вывода, что приводит к значительному увеличению производительности.

3. Исследования и разработки:

- Библиотеки и фреймворки нейронных сетей стали катализатором прогресса в исследованиях и разработках в области глубокого обучения. Полная интеграция Python с этими инструментами облегчила реализацию передовых алгоритмов и моделей, создав динамичную экосистему для академических и промышленных исследований.

4. Обогащение экосистемы:

- Доступность надежных библиотек и фреймворков нейронных сетей обогатила экосистему Python, привлекая процветающее сообщество участников, исследователей и энтузиастов. Это привело к созданию разнообразного набора инструментов, расширений и утилит, которые дополняют рабочий процесс разработки нейронных сетей на Python.

5. Внедрение и стандартизация в отрасли:

- Библиотеки и фреймворки нейронных сетей стали стандартными инструментами для разработки решений машинного обучения в различных отраслях. Совместимость Python с этими платформами повлияла на их

широкое распространение, обеспечивая поддержку самых разных приложений: от обработки естественного языка до компьютерного зрения и автономных систем.

6. Образовательное воздействие:

- Доступность и простота использования библиотек нейронных сетей и фреймворков Python сыграли ключевую роль в образовательной среде. Эти инструменты стали полезными в преподавании и изучении концепций глубокого обучения, способствуя увеличению количества образовательных ресурсов, учебных пособий и курсов, посвященных нейронным сетям на Python.

7. Инновации и междисциплинарные приложения:

- Влияние библиотек и фреймворков нейронных сетей на программирование на Python выходит за рамки традиционных областей машинного обучения. Эти инструменты способствовали инновациям в междисциплинарных приложениях, включая здравоохранение, финансы, робототехнику и творческое искусство, демонстрируя свою универсальность и преобразующий потенциал [4].

На рисунке 1 показана простейшая модель нейронной сети.

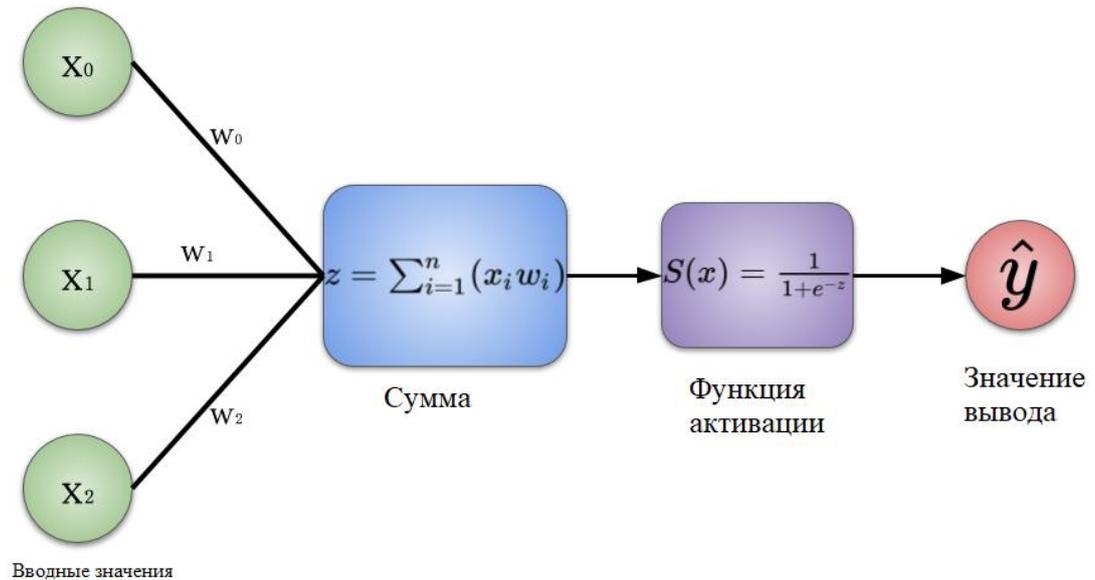


Рисунок 1 – Модель реализации нейронной сети

Технический анализ реализации нейронных сетей в Python.

Обзор популярных библиотек и фреймворков нейронных сетей на Python (например, TensorFlow, Keras, PyTorch)

На ландшафт разработки нейронных сетей на Python значительное влияние оказывают несколько популярных библиотек и фреймворков, которые сыграли ключевую роль в развитии области глубокого обучения. Среди них следует выделить TensorFlow, Keras и PyTorch, каждый из которых предлагает уникальные возможности и способствует широкому внедрению нейронных сетей в программировании на Python.

TensorFlow, разработанный Google, представляет собой выдающуюся среду машинного обучения с открытым исходным кодом, которая обеспечивает комплексную поддержку для создания и обучения моделей нейронных сетей. Его универсальная архитектура обеспечивает эффективную реализацию широкого спектра алгоритмов машинного обучения, включая глубокое обучение, а также предлагает надежные инструменты для масштабируемого развертывания и создания приложений

машинного обучения.

Keras, известный своим удобным интерфейсом и модульностью, приобрел известность как библиотека нейронных сетей высокого уровня, построенная на основе TensorFlow. Который облегчает быстрое прототипирование и экспериментирование с архитектурами нейронных сетей, позволяя разработчикам создавать и обучать модели с минимальной сложностью, тем самым служа ценным инструментом как для новичков, так и для опытных практиков в области глубокого обучения.

PyTorch, разработанный исследовательской лабораторией искусственного интеллекта Facebook, превратился в мощную и динамичную среду глубокого обучения, которая привлекла значительное внимание сообщества Python. Который особенно известен своей гибкостью и простотой использования, предлагая императивный стиль программирования, который хорошо находит отклик у исследователей и практиков, ищущих более интуитивный и выразительный подход к разработке нейронных сетей.

Эти библиотеки и платформы в совокупности изменили ландшафт разработки нейронных сетей на Python, предоставив специалистам разнообразные возможности для создания, обучения и развертывания передовых моделей машинного обучения. Их влияние выходит за рамки традиционных областей машинного обучения, оказывая влияние на междисциплинарные приложения и стимулируя инновации в таких областях, как обработка естественного языка, компьютерное зрение и обучение с подкреплением [8].

Тематические исследования и приложения. Реальные применения нейронных сетей в программировании на Python охватывают широкий спектр областей, отражая широкое влияние технологий глубокого обучения на многочисленные отрасли и области. Эти приложения демонстрируют практическую полезность и преобразующий потенциал

нейронных сетей, интегрированных с Python, в решении сложных реальных задач. Яркие примеры включают:

1. Здравоохранение и медицинская визуализация:

Нейронные сети в Python широко используются для анализа медицинских изображений, помогая обнаруживать аномалии, диагностировать заболевания и интерпретировать медицинские изображения. Приложения включают использование сверточных нейронных сетей (CNN) для идентификации опухолей на радиологических изображениях и анализа медицинских изображений для обнаружения патологий.

2. Обработка естественного языка (НЛП) и анализ текста:

Нейронные сети на основе Python играют важную роль в задачах обработки естественного языка, таких как анализ настроений, языковой перевод и разработка чат-ботов. Рекуррентные нейронные сети (RNN) и модели на основе преобразователей применяются к текстовым данным для таких задач, как генерация языка, обобщение и извлечение информации.

3. Финансовое прогнозирование и торговля:

Нейронные сети в Python используются для прогнозной аналитики и финансового моделирования, помогая прогнозировать фондовый рынок, оценивать риски и разрабатывать алгоритмические торговые стратегии. Модели глубокого обучения используются для анализа рыночных тенденций, выявления закономерностей и принятия обоснованных инвестиционных решений [7].

4. Автономные транспортные средства и робототехника:

Нейронные сети на основе Python играют решающую роль в навигации беспилотных транспортных средств, обнаружении объектов и планировании пути. Сверточные нейронные сети используются для распознавания объектов в реальном времени, позволяя транспортным средствам и роботизированным системам воспринимать окружающую

среду и взаимодействовать с ней.

5. Промышленная автоматизация и контроль качества:

Нейронные сети, интегрированные с Python, используются для контроля качества и обнаружения дефектов в производственных процессах. Модели нейронных сетей на основе машинного зрения используются для автоматизированного контроля, выявления дефектов продукции и обеспечения стандартов качества производства.

6. Экологический мониторинг и анализ:

Нейронные сети на основе Python применяются для анализа данных об окружающей среде, включая моделирование климата, прогнозирование погоды и экологический мониторинг. Модели глубокого обучения используются для обработки крупномасштабных наборов данных об окружающей среде и составления прогнозов, связанных с климатическими явлениями и закономерностями окружающей среды.

Производительность и эффективность. Оценка показателей производительности и эффективности реализаций нейронных сетей в Python является важнейшим аспектом оценки вычислительных возможностей и практической полезности моделей глубокого обучения. Эта оценка включает в себя различные факторы, влияющие на эффективность и результативность реализации нейронных сетей, включая скорость вычислений, масштабируемость, использование ресурсов, сравнительный анализ, аппаратное ускорение, а также сложность модели и компромиссы с эффективностью.

Тесты производительности проводятся для сравнения скорости вычислений различных архитектур нейронных сетей, оптимизации проектирования моделей и процессов обучения для повышения эффективности. Тесты масштабируемости включают анализ способности модели обрабатывать большие объемы данных и эффективно использовать вычислительные ресурсы, такие как многоядерные процессоры или

ускорение графического процессора, для задач обучения и вывода.

Оценка эффективного использования аппаратных ресурсов, таких как ЦП, память и графический процессор, во время реализации нейронных сетей на Python имеет решающее значение для оптимизации использования вычислительных ресурсов и выявления потенциальных узких мест, которые могут повлиять на производительность. Сравнительный анализ реализаций нейронных сетей на Python включает в себя сравнение различных платформ, моделей и методов оптимизации для оценки их производительности, и эффективности, что позволяет специалистам-практикам выбирать наиболее подходящие инструменты и методы для их конкретных приложений [9].

Тесты производительности также оценивают ускорение, достигнутое за счет использования аппаратных ускорителей для задач обучения и вывода, оптимизации использования специализированного оборудования для рабочих нагрузок глубокого обучения. Кроме того, оценка компромисса между сложностью модели и эффективностью вычислений важна для оптимизации общей эффективности и производительности моделей глубокого обучения.

Проводя всестороннюю оценку показателей производительности и эффективности реализаций нейронных сетей на Python, специалисты-практики могут получить представление о вычислительных возможностях, использовании ресурсов и масштабируемости моделей глубокого обучения. Эта оценка облегчает оптимизацию реализации нейронных сетей для повышения эффективности и результативности в широком спектре приложений и вариантов использования [6]. Сравнение языков программирования по важным критериям для нейросетевого программирования приведено в таблице 1. 1

Таблица 1 – Сравнение Python с другими языками программирования

| Аспект | Python | Другой язык программирования (например, C++, Java) |
|------------------------|---|---|
| Скорость | Обычно медленнее из-за интерпретируемого характера и глобальной блокировки интерпретатора (GIL). | Часто быстрее из-за скомпилированной природы и отсутствия GIL. |
| Масштабируемость | Ограниченная масштабируемость для многопоточных задач из-за GIL. | Улучшенная масштабируемость для многопоточных и параллельных задач. |
| Простота использования | Очень популярен и удобен для пользователя благодаря широкому спектру библиотек и фреймворков для нейронных сетей. | Требуется более сложное управление памятью и более крутая кривая обучения для разработки нейронных сетей. |

Заключение. В ходе исследования было выявлено, что использование нейронных сетей в программировании на Python позволяет значительно улучшить производительность и эффективность разработки. Нейронные сети способны автоматизировать многие процессы, что позволяет программистам сосредоточиться на более творческих задачах. Благодаря анализу влияния нейронных сетей на программирование на Python можно сделать вывод о их огромном потенциале и перспективности для будущего развития информационных технологий. Дальнейшие исследования в этом направлении помогут расширить возможности использования нейронных сетей в программировании и создании инновационных решений.

Таким образом, результаты данного исследования подтверждают перспективность разработки и использования систем автоматизированного полива растений, что имеет значение как для повседневного ухода за

растениями в домашних условиях, так и для применения в аграрной сфере.

ЛИТЕРАТУРА

1. Баррель, Д. Искусственный интеллект: сборник статей / Д. Баррель - М.: Кудиц-образ, 2017, 315 с.
2. Васильев, А. Введение в машинное обучение: обзор современных методов и алгоритмов / А. Васильев. - М.: ДМК Пресс, 2018, 224 с.
3. Гельфанд, И.М. Машинное обучение: в современной практике / И.М. Гельфанд, С.Г. Селиверстов. - М.: ДМК Пресс, 2016, 562 с.
4. Даниэль, Ш. Глубокое обучение на Python / Ш. Даниэль. - СПб.: Питер, 2018, 168 с.
5. Кеплер, Л. Программирование на Python для начинающих: учебник / Л. Кеплер. - М.: Эксмо, 2019, 421 с.
6. Лиддл, М. Машинное обучение в действии: Практическое руководство / М. Лиддл. - М.: ДМК Пресс, 2019, 312 с.
7. Митичкин, А. Нейронные сети и искусственный интеллект: сборник научных трудов / А. Митичкин, И. Данилов. - М.: ФИСК, 2016, 123 с.
8. Никифоров, И. Алгоритмы обучения на примерах / И. Никифоров. - М.: Вильямс, 2017, 612 с.
9. Петров, В. Современные методы машинного обучения: методическое пособие / В. Петров, А. Сидоров. - М.: Горячая Линия, 2016, 333 с.
10. Пыскин, И. Машинное обучение в практике программиста: Практические примеры на Python / И. Пыскин. - М.: ДМК Пресс, 2019, 411 с.

REFERENCES

1. Barrel, D. Iskusstvennyj intellekt: sbornik statej [Artificial intelligence: a collection of articles], 2017, 315 p.
2. Vasiliev, A. Vvedenie v mashinnoe obuchenie: obzor sovremennyh metodov i algoritmov [Introduction to machine learning: a review of modern methods and algorithms], 2018, 224 p.
3. Gelfand I.M., Seliverstov S.G. Mashinnoe obuchenie: v sovremennoj praktike [Machine learning: in modern practice], 2016, 562 p.
4. Daniel, S. Glubokoe obuchenie na Python [Deep learning in Python], 2018, 168 p.
5. Kepler, L. Programirovanie na Python dlja nachinajushhih: uchebnik [Python programming for beginners: textbook], 2019, 421 p.
6. Liddle, M. Mashinnoe obuchenie v dejstvii: Prakticheskoe rukovodstvo [Machine learning in action: A practical guide], 2019, 312 p.
7. Mitichkin A., Danilov I. Nejronnye seti i iskusstvennyj intellekt: sbornik nauchnyh trudov [Neural networks and artificial intelligence: a collection of scientific papers], 2016, 123 p.
8. Nikiforov, I. Algoritmy obuchenija na primerah [Algorithms of learning by examples], 2017, 612 p.
9. Petrov V., Sidorov A. Sovremennye metody mashinnogo obuchenija: metodicheskoe posobie [Modern methods of machine learning: a methodological guide], 2016, 333 p.
10. Pyskin, I. Mashinnoe obuchenie v praktike programmista: Prakticheskie primery na Python [Machine learning in programmer's practice: Practical examples in Python], 2019, 411 p.