



УДК. 519.6

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ В СЕКМЕНТЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Сафронов Константин Юрьевич

Полномочное представительство Республики Башкортостан при Президенте Российской Федерации, г. Москва, ул. Доброслободская, 6, стр.2

e-mail: safronovskiy@yandex.ru

тел. 8-927-94-87-830

Аннотация

В данной статье представлена роль языка программирования Python в разработке искусственного интеллекта, описаны примеры применения Python в сегменте ИИ, а также проанализированы преимущества и недостатки данного языка программирования. Поскольку для программирования в ИИ используется большое количество языков программирования, автором проведен их сравнительный анализ с точки зрения удобства использования

Ключевые слова: машинное обучение, искусственный интеллект, язык программирования Python.

COMPARATIVE ANALYSIS OF PROGRAMMING LANGUAGES IN THE ARTIFICIAL INTELLIGENCE SEGMENT

Konstantin Yu. Safronov

Plenipotentiary Representation of the Republic of Bashkortostan under the President of the Russian Federation, Moscow, st. Dobroslobodskaya, 6, building 2.

e-mail: safronovskiy@yandex.ru

Tel. 8-927-94-87-830

ABSTRACT

This article presents the role of the Python programming language in the development of artificial intelligence, describes examples of using Python in the AI segment, and analyzes the advantages and disadvantages of this programming language. Since a large number of programming languages are used for programming in AI, the author carried out a comparative analysis of them in terms of ease of use.

Keywords: machine learning, artificial intelligence, Python programming language.

В конце 1980-х Гвидо Ван Россум разработал язык программирования Python. Сейчас данный язык востребован в различных прикладных областях, таких как разработка программного обеспечения, веб-разработка, разработка графического интерфейса ПК, разработка образовательных и научных приложений. Таким образом, Python охватывает огромный сегмент разработки и завоевал популярность благодаря своей практичности и надежности.

Основатель искусственного интеллекта (далее - ИИ) Джон Маккарти считает, что искусственный интеллект - это наука создания интеллектуальных машин и интеллектуальных компьютерных программ, способных мыслить разумно, подобно тому, как это делает человек [2].

Машину можно обучить только в том случае, если она имитирует процесс, подобный тому, как человеческий мозг учится, думает и действует. Все это можно использовать в качестве платформы для разработки интеллектуальных систем, способных к обучению.

В связи с чем весьма актуален вопрос - какой язык программирования можно использовать для разработки приложений искусственного интеллекта. Существуют различные языки программирования, такие как Python, Lisp, Prolog, C++, Java и т.д., которые можно использовать для разработки ИИ приложений. Среди них набирает популярность в первую очередь язык программирования Python. Так происходит потому, что Python включает очень мало кода и имеет простой синтаксис по сравнению с другими языками программирования, которые могут быть использованы для разработки приложений ИИ. По этой причине тестирование может быть проще и может быть больше сосредоточено на самом программировании. Основное преимущество использования Python для ВВ заключается в том, что он функционирует со встроенными библиотеками. В Python есть библиотеки практически для всех типов проектов ИИ. Например, NumPy, SciPy, Matplotlib, Nltk, SimpleAI являются одними из важнейших встроенных библиотек Python.

Искусственный интеллект как подраздел информатики фокусируется на разработке компьютерных программ и машин, способных выполнять задачи, например, понимание речи и распознавание изображений. Есть много причин актуализирующих изучение ИИ. В частности искусственный интеллект работает с огромным количеством данных, которые человеческий мозг не в силах обработать. Для этого необходима автоматизация и как следствие разработка ИИ, потому что он может учиться на данных и выполнять задачи с заданной точностью.

При этом системе необходимо самообучаться, потому что данные меняются, и знания, полученные из таких данных, должны постоянно обновляться. Система с поддержкой ИИ может обучаться сама. С помощью нейронных сетей ИИ может более глубоко анализировать данные, а также достигать невероятной точности в расчетах. ИИ может думать и реагировать на ситуации, исходя из собственного опыта и ошибок, в режиме реального времени. Для систем, использующих алгоритмы самообучения, данные являются не только объектом обработки, но и источником обучения. В этом контексте в машинном обучении уместно интерпретировать интеллект, как способность системы вычислять, рассуждать, воспринимать отношения и аналогии, учиться на опыте, хранить и извлекать информацию из памяти, решать проблемы, понимать сложные идеи, свободно использовать естественный язык, классифицировать, обобщать и адаптировать новые ситуации.

Согласно американскому психологу Говарду Гарднеру, интеллект проявляется во многих формах: лингвистический, музыкальный, логический, математический,

пространственный, телесно-кинестетический, внутриличностный и межличностный интеллект [1]. Система машинного или искусственного интеллекта считается интеллектуальной, если она обладает хотя бы одним из видов интеллекта.

Искусственный интеллект представляет собой обширную область исследований, которая помогает находить решения реальных проблем. Направлениями обучения ИИ являются машинное обучение, искусственные нейронные сети и алгоритм, подсказывающий компьютеру, каким источником данных необходимо воспользоваться.

В этом процессе Python отличает то, что приложение может быть написано на одной платформе, а использоваться на других платформах. Код Python удобочитаем и имеет краткий синтаксис, который позволяет писать приложения с использованием меньшего количества строк кода, чем в других языках программирования. Python поддерживает функциональные, императивные, объектно-ориентированные и процедурные стили кодирования.

Python особенно привлекателен для рабочих нагрузок в области науки о данных, машинного обучения и научных вычислений. К примеру, NumPy - это основная библиотека Python для машинного обучения. Он обеспечивает эффективные операции со структурами данных, часто используемыми в машинном обучении, векторами, матрицами и тензорами. Основная структура данных NumPy - это многомерный массив. Ниже представлена таблица 1 основных библиотек Python.

Таблица 1. Характеристика основных библиотек Python

NumPy	Это библиотека многомерных массивов с базовыми процедурами линейной алгебры, а библиотека SciPy украшает массивы NumPy важными примитивами, числовыми оптимизаторами и обработкой сигналов для статистики и ограниченной линейной алгебры. SciPy используется почти в половине проектов машинного обучения. Библиотека Pandas - это формат представления табличных данных в Python для извлечения, преобразования, загрузки контекста и анализа данных.
TensorFlow	Это программная библиотека или фреймворк, разработанная командой Google для наиболее простого применения концепций машинного обучения и глубокого обучения. Он сочетает в себе методы оптимизации вычислительной алгебры для простого вычисления многих математических выражений. TensorFlow содержит функцию, которая определяет, оптимизирует и вычисляет математические выражения с использованием многомерных массивов, называемых тензорами. Включает программное обеспечение глубокой нейронной сети и методы машинного обучения. Также включает в себя высокомасштабируемую вычислительную функцию с различными наборами данных. TensorFlow использует вычисления на GPU, автоматизируя управление. Он включает уникальную функцию оптимизации памяти и используемых данных. TensorFlow хорошо укомплектован и включает множество библиотек для машинного обучения. TensorFlow является продуктом Google. Он включает в себя различные

	<p>алгоритмы машинного обучения и глубокого обучения. TensorFlow может обучать и запускать глубокие нейронные сети, чтобы вручную классифицировать числа, распознавать изображения, вставлять слова и создавать различные модели последовательностей.</p>
Scikit-learn	<p>Это модуль Python, который объединяет широкий спектр современных алгоритмов машинного обучения для контролируемых и неконтролируемых задач среднего размера. Особое внимание уделяется простоте использования, производительности, укомплектованности и согласованности API. Он распространяется под лицензией BSD, и ориентирован для использования в академической и коммерческой среде.</p>
Theano	<p>Это библиотека Python и оптимизатор компиляции для управления и обработки математических выражений, которые включают многомерные массивы. Возможности Theano заключается в тесной интеграции с NumPy, прозрачном использовании графического процессора, эффективном символьном дифференцировании, оптимизации скорости и стабильности, динамической генерации кода C, а также в уникальном тестировании и самопроверке.</p>
Keras	<p>Это API глубокого обучения, написанный на Python, запущенный на вершине платформы машинного обучения TensorFlow. По сути Keras представляет собой приемлемый, высокопроизводительный интерфейс для решения задач машинного обучения с упором на современное глубокое обучение. Предоставляет базовые абстракции и строительные блоки для разработки и доставки решений машинного обучения с большим числом повторений. Keras может работать на GPU или на больших кластерах графических процессоров, модели Keras могут работать в браузере или на мобильном устройстве.</p>
Natural Language Toolkit (NLTK)	<p>Это платформа для создания программы Python для работы с данными на человеческом языке. Он предоставляет простые в использовании интерфейсы и лексические ресурсы, такие как WordNet, а также пакет библиотек текстовых редакторов для классификации, токенизации, выделения корней, тегов, синтаксического анализа и семантического обоснования. NLTK подходит лингвистам, инженерам, студентам, преподавателям, исследователям и промышленным пользователям.</p>

Таким образом, Python можно назвать одним из самых популярных языков программирования. Он включает очень мало кода и имеет простой синтаксис, в отличие от других языков программирования, которые можно использовать для разработки приложений искусственного интеллекта. В целом можно выделить несколько приоритетных характеристик Python:

1. Поддержка функциональных и структурных методов программирования, а также ООП.
2. Его можно использовать как язык сценариев или скомпилировать в байт-код для создания больших приложений.

3. Предоставляет высокоуровневые динамические типы данных и поддерживает проверку динамических типов.

Его можно легко интегрировать с C, C++, COM, ActiveX, CORBA и Java. Как и любой язык программирования, у Python есть свои достоинства и недостатки, которые рассматриваются ниже [3].

Python - это язык программирования с англоязычным синтаксисом, что упрощает написание, чтение и понимание кода. Python является интерпретируемым языком и выполняется напрямую, код за строкой. В случае любой ошибки выполнение останавливается, и сообщается о возникшей ошибке. Python показывает только одну ошибку, даже если ошибок несколько. Это упрощает устранение неполадок. Python не знает тип переменной до тех пор, пока код не будет запущен, а при выполнении автоматически присваивает тип данных. Программисту не нужно беспокоиться об объявлении переменных и их типов данных. Python распространяется под лицензией с открытым исходным кодом, одобренной OSI, что делает его бесплатным для использования и распространения. Стандартная библиотека Python огромна, и есть почти все функции для решения различных задач. Менеджер пакетов Python (pip) облегчает импорт других пакетов из индекса пакетов Python (PyPi). Код Python может работать на разных платформах.

В то же время выполнение кода часто приводит к замедлению процесса вычисления, потому что он должен выполнять дополнительную работу во время выполнения кода. Язык программирования Python использует большой объем памяти, что является недостатком при создании приложений, когда необходимо выполнить оптимизацию памяти. Python обычно используется в серверном программировании и не используется в клиентских или мобильных приложениях, потому что он неэффективен с точки зрения памяти и имеет низкую вычислительную мощность по сравнению с другими языками. Связь с базой данных - не сильная сторона Python. Уровень доступа к базе данных Python примитивен и недостаточно развит по сравнению с такими технологиями, как JDBC и ODBC. Python - это язык с динамической типизацией, поэтому тип данных переменной можно изменить в любое время. Переменная, содержащая целое число, может в будущем содержать строку, что может привести к ошибкам выполнения. В связи с чем, необходимо проводить тщательное тестирование приложений.

Существуют разные языки для написания ИИ, но не существует идеального языка программирования, который выделялся бы как лучший из них, используемый в искусственном интеллекте. Процесс разработки зависит от желаемой функциональности разрабатываемого приложения ИИ [4].

Java, Python, Lisp, Prolog и C++ - основные языки программирования ИИ, используемые для искусственного интеллекта и отвечающие различным потребностям при разработке и проектировании различного программного обеспечения. Ниже представлена таблица 2, позволяющая сравнить языки программирования и сопоставить их функционал с функционалом Python.

Таблица 2. Сравнение языков программирования и сопоставление их функционала с функционалом Python

Python	Один из востребованных языков программирования при разработке ИИ среди разработчиков из-за его синтаксической простоты и универсальности. Python используется для машинного обучения разработчиками, поскольку он менее сложен по
--------	---

	<p>сравнению с C ++ и Java. Python - переносимый язык, он может использоваться на платформах Linux, Windows, Mac OS и UNIX. Он привлекателен благодаря своим функциям, таким как интерактивность, интерпретируемость, модульность, динамичность, переносимость и высокоуровневость, что делает его более уникальным, чем Java. Python - это многопарадигмальная программа, поддерживающая объектно-ориентированные, процедурные и функциональные стили программирования. Python поддерживает нейронные сети и разработку решений NLP благодаря своей простой библиотеке функций и более удобной структуре. Поддерживает тестирование алгоритмов без необходимости их применения. Python развивается быстрее, чем Java и C ++.</p> <p>Однако, разработчики, использующие Python, испытывают трудности с адаптацией к совершенно другому синтаксису, когда им нужно использовать другой язык для программирования ИИ. В отличие от C ++ и Java, Python работает с помощью интерпретатора, который замедляет компиляцию и выполнение при разработке ИИ. Программы на Python работают медленнее, чем программы на Java, но при этом на их разработку уходит гораздо меньше времени. Программы на Python обычно в 3-5 раз короче эквивалентных программ на Java. Подобное различие объясняется встроенными в Python типами данных высокого уровня и его динамической типизацией. Например, разработчик Python не тратит время на объявление типов аргументов или переменных. Мощный полиморфный список Python и тип словаря, для которых богатая синтаксическая поддержка встроена непосредственно в язык, находят применение почти в каждой программе Python. Из-за типизации во время выполнения, среда разработки Python нагружается больше, чем Java. Например, при оценке выражения $a + b$ оно должно исследовать объекты a и b, чтобы найти их тип, который не известен во время компиляции. Затем вызывается соответствующая операция сложения, которую можно перегрузить с помощью пользовательского метода. Java может выполнять сложение целых чисел, он требует объявления переменных для a и b и не допускает перегрузки для оператора $+$ для объектов, определяемых пользователем классов. По этим причинам Python гораздо лучше подходит в качестве «связующего» языка, в то время как Java можно охарактеризовать как язык реализации низкого уровня.</p>
C ++	<p>Самый быстрый компьютерный язык. Его скорость важна для разработчиков, которые больше всего ценят время. Он обеспечивает быстрое выполнение и меньшее время загрузки, поэтому применяется в поисковых системах и при разработке компьютерных игр. C ++ позволяет широко использовать алгоритмы и эффективно использовать статистические методы искусственного интеллекта. C ++ поддерживает повторное использование программ в разработке за счет наследования и</p>

	<p>сокрытия данных, что эффективно с точки зрения экономии времени и затрат. С ++ подходит для машинного обучения и нейронных сетей. Данный язык программирования хорошо применим для поиска решений сложных проблем ИИ. Он богат библиотечными функциями и набором программных инструментов. С ++ - это многопарадигмальная программа, поддерживающая объектно-ориентированные принципы, полезные для получения организованных данных.</p> <p>В то же время С ++ плохо справляется с многозадачностью. Данный язык программирования подходит только для применения ядра или базы данных определенных систем или алгоритмов. Подход снизу вверх чрезвычайно сложен и затрудняет для начинающих разработчиков написание программ ИИ.</p> <p>Код Python зачастую в 5-10 раз короче эквивалентного кода С ++. К примеру, разработчик Python может выполнить за два месяца то, что два разработчика С ++ не смогут выполнить за год.</p>
Java	<p>Язык программирования с несколькими парадигмами, который следует объектно-ориентированным принципам и принципу Once Written Read / Run Anywhere (WORA). Это язык программирования искусственного интеллекта, который может работать на любой поддерживающей его платформе без необходимости перекомпиляции. Большая часть синтаксиса заимствована из С и С ++. Java подходит не только для NLP и поисковых алгоритмов, но и для нейронных сетей. В отличие от С ++, Java прост в использовании. Имеет автоматический менеджер памяти, облегчающий работу разработчиков. Java медленнее С ++, имеет более низкую скорость выполнения и большее время отклика. Java портативен, но старые платформы потребуют изменений программного и аппаратного обеспечения.</p>
Lisp	<p>Язык, используемый для разработки ИИ. Это второй по возрасту язык программирования после Fortran. Lisp превратился в мощный и динамичный язык программирования. Некоторые считают Lisp лучшим языком программирования искусственного интеллекта из-за свободы, которую он предлагает разработчикам. Lisp используется в ИИ из-за его гибкости для быстрого прототипирования и экспериментов. Он имеет уникальную макросистему, которая облегчает исследование и применение искусственного интеллекта на разных уровнях. Lisp подходит для проектов индуктивной логики и машинного обучения, поддерживает компилятор вместо интерпретатора и, следовательно, является быстрым и эффективным в процессе кодирования. Для Lisp был изобретен автоматический диспетчер памяти, который «собирает мусор». Lisp предлагает особый контроль над системами, что приводит к его максимальному использованию. Будучи старинным языком программирования Lisp требует настройки нового программного и аппаратного обеспечения, чтобы приспособиться к его использованию. Lisp близок к Python по своей динамической семантике, но несколько</p>

	отличается по подходу к синтаксису. У Python есть интроспективные возможности, такие как и у Lisp. Программы Python могут создавать и выполнять фрагменты программ "на лету".
Prolog	Является одним из старейших языков программирования и поэтому подходит для разработки ИИ. В нем есть механизмы, которые упрощают создание гибких фреймворков, с которыми разработчикам нравится работать. Prolog поддерживает базовые механизмы, такие как сопоставление с образцом, древовидное структурирование данных и автоматический поиск с возвратом, необходимые для программирования ИИ. В Prolog есть встроенная обработка списков, необходимая для представления древовидных структур данных. Он эффективен для быстрого создания прототипов часто публикуемых программ ИИ. Это позволяет создавать базу данных одновременно с запуском программы.

Исходя из вышеизложенного можно заключить, что ИИ играет важную роль в развитии технологий, а также в упрощении выполнения различных работ. ИИ стремится создавать машины, которые смогут заменять людей на различных должностях, достигая точность и эффективность в выполнении необходимых работ. В области ИИ достигнуты огромные результаты, но тенденция к его усовершенствованию все еще сохраняется. Этому способствуют различные языки программирования. В частности, Python используется во всевозможных областях искусственного интеллекта, и является одним из востребованных языков программирования среди разработчиков ИИ из-за его синтаксической простоты и универсальности.

Однако для программирования в ИИ используется множество языков программирования. Разработчики постоянно задаются вопросом, какой язык программирования лучше всего подходит для программирования в ИИ, и не находят ответа на этот вопрос. Потому, что у каждого языка программирования есть свои преимущества и недостатки. В представленной статье проведено сравнение языков программирования, наиболее популярных среди разработчиков в сегменте ИИ: Python, C++, Java, Lisp и Prolog. В зависимости от профиля и функциональности приложения ИИ используется тот или иной язык программирования. Выбрав соответствующий язык, достигается соответствующий результат.

Список литературы

1. Гарднер Г. Структура разума: теория множественного интеллекта / Г. Гарднер. – М.: ООО «И.Д. Вильямс», 2007. – 512 с.
2. Информатика [Электронный ресурс]: курс лекций / Л.И. Шехтман [и др.]; ГОУ ВПО УГАТУ – Уфа: УГАТУ, 2011.
3. Кауфман В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс]: [пособие для студентов вузов]. В.Ш. Кауфман – Москва: ДМК ПРЕСС, 2020 – 464 с.
4. Филиппова А.С., Валиахметов Ю.И. Теория оптимального использования ресурсов Л. В. Канторовича в работах уфимской научной школы / А.С. Филиппова, Ю.И. Валиахметов // Омский научный вестник. - №1 (17). - 2012. – С. 39-42.

References

1. Gardner G. The structure of mind: the theory of multiple intelligences / G. Gardner. - М .: LLC "I.D. Williams ", 2007. - 512 p.
2. Informatics [Electronic resource]: a course of lectures / L.I. Shekhtman [and others]; GOU VPO USATU - Ufa: USATU, 2011.
3. Kaufman V.Sh. Programming languages. Concepts and principles [Electronic resource]: [manual for university students]. V.Sh. Kaufman - Moscow: DMK PRESS, 2020 - 464 p.
4. Filippova A.S., Valiakhmetov Yu.I. L. V. Kantorovich's theory of optimal use of resources in the works of the Ufa scientific school / A.S. Filippova, Yu.I. Valiakhmetov // Omsk Scientific Bulletin. - No. 1 (17), - 2012. - P. 39-42.