УДК 004.852 DOI: 10.31799/2077-5687-2022-4-61-72

ВАРИАНТЫ РЕАЛИЗАЦИИ МАШИННОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ В ПРОГРАММЕ ANYLOGIC

С. А. Андронов, М. С. Прокофьева
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Методы искусственного интеллекта сегодня широко применяются при решении различного рода задач. Существует множество вариаций машинного обучения, среди которых особое место занимает обучение с подкреплением. В данном подходе нет ответов учителя, но есть ответная реакция среды. Такая среда может быть реальной (например, на частных дорогах, в ограниченном воздушном пространстве, либо на учебной конвейерной линии) или виртуальной.

Программные средства имитационного моделирования позволяют создавать реалистичные среды обучения искусственного интеллекта, проводить безопасную тренировку и тестирование обучающихся агентов. В числе таких программ лидирующие позиции принадлежат многоподходной среде моделирования – AnyLogic, которая успешно применяется при решении различных бизнес - задач.

Данная статья посвящена анализу вариантов реализации связки имитационной модели с машинным обучением с подкреплением, применяемых в рассматриваемом программном продукте, выявлению их особенностей, преимуществ и недостатков

Ключевые слова: Anylogic, искусственный интеллект, искусственные нейронные сети, обучение с подкреплением, машинное обучение.

Для цитирования:
Андронов С. А., Прокофьева М. С. Варианты реализации машинного обучения с подкреплением в программе Anylogic // Системный анализ и логистика: журнал.: выпуск №4(34), ISSN 2007-5687. – СПб.: ГУАП., 2022 – с. 61 – 72. РИНЦ. DOI: 10.31799/2077-5687-2022-4-61-72.

VARIANTS FOR IMPLEMENTING MACHINE LEARNING WITH REINFORCEMENT IN THE ANYLOGIC PROGRAM

M. S. Prokofieva, S. A. Andronov
St. Petersburg State University of Aerospace Instrumentation

Artificial intelligence methods are widely applied in the detection of type problems. There are many variations of machine learning, among which reinforcement learning occupies a special place. At present, there is no answer to the teacher's question, but there is a response from the environment. Such environments can be real (for example, on random roads, in confined airspace, or on a training assembly line) or natural.

Simulation software tools allow you to create realistic artificial intelligence environments, safely train and test learning agents. Including such approaches to solving various business problems.

This article is devoted to the analysis of implementation options for a coupling simulation model with machine learning with reinforcement, used in the proposed software product, to identify their features, shortcomings and shortcomings.

Keywords: Anylogic, artificial intelligence, artificial neural networks, reinforcement learning, machine learning.

For citation:
Andronov S. A., Prokofieva M. S. Variants for implementing machine learning with reinforcement in the Anylogic program // System analysis and logistics.: №4(34), ISSN 2007–5687. – Russia, Saint-Petersburg.: SUAI., 2022 – p. 61–72. DOI: 10.31799/2077-5687-2022-4-61-72.

Введение

Anylogic – платформа для имитационного моделирования любых бизнес-систем. Распространенность данного программного продукта также обусловлена, спросом на расширение применения искусственного интеллекта (ИИ) для различных бизнес-задач: управление производством, логистика, цепочка поставок, бизнес-процессы, управление активами и т. д.

Так как, имитационные модели представляют собой мощную реалистичную среду (из-за отсутствия рисков и дешевизны), которая позволяет проводить безопасную тренировку и

тестирование обучающихся агентов, требуется такая платформа, которая будет способна соответствовать постоянно меняющимся требованиям различных реализации ИИ.

Существует множество методов создания ИИ, которыми пользуются для решения различного рода задач (группы этих методов представляют из себя собирательный образ ИИ). На рис.1 показана иерархия подходов создания ИИ.

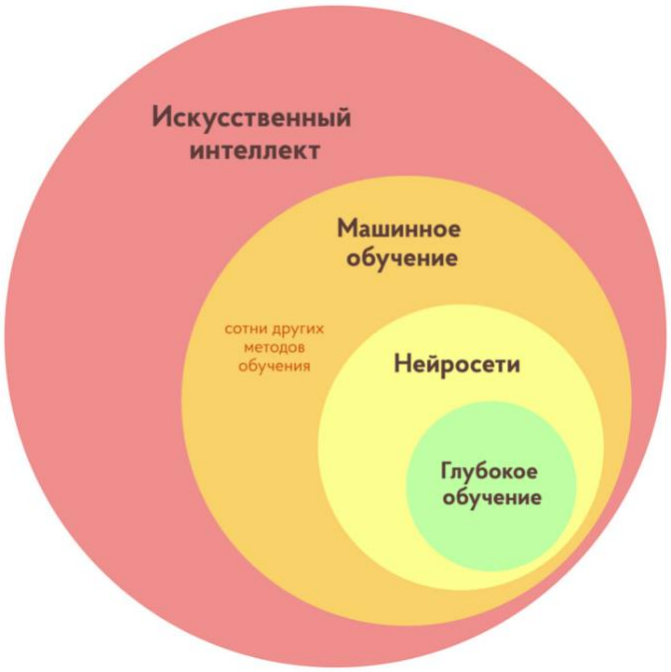

Рис. 1. Иерархия ИИ

В данной статье в основном рассматривается машинное обучение с подкреплением или Reinforcement Learning (RL), совместно с глубоким обучением и искусственными нейронными сетями.

Машинное обучение (МО) - это подход, при котором алгоритм «учится» решать задачу [1].

Термин Глубокое Обучение (Deep Learning, DL) относится к искусственным нейронным сетям, которые в свою очередь представляет собой последовательность слоев, каждый из которых, в свою очередь, состоит из нейронов, и каждый выполняет свою роль (ИНС имитируют работу человеческого мозга). Такие структуры способны учиться без руководства человека на необработанных данных [1].

Таким образом, различия МО и глубокого обучения состоят, главным образом, в том, что во втором случае используются комбинации множества различных по назначению слоев сети и более сложные алгоритмы.

Для организации обучения требуется наличие обучающегося агента (далее агент), который представляет собой интеллектуальную систему, т. е. алгоритм, обучающийся тем или иным способом. Агент учится, взаимодействуя со средой (ИМ реального или виртуального мира): наблюдает состояние (состояние - все, что известно о среде на данный момент, наблюдение – выжимка состояний, которые видит агент), совершает действие (ограниченное правилами), получает обратную связь – награду, которая может быть, как поощрением, так и наказанием.

Результатом такого обучения будет некая политика (стратегия) управления, то есть способ выбора агентом действия, исходя из наблюдения среды. После обучения политика сохраняется.

Так как имитационная модель выступает в качестве среды взаимодействия в программе AnyLogic используют этот метод.

Алгоритмы, которые используют подход обучения с подкреплением, являются лишь одной из подгрупп всего того множества алгоритмов, что принято называть ИИ. В отличии от классических направлений МО: обучение с учителем (в широком смысле задача классификации), обучение без учителя (в широком смысле задача кластеризации), или ансамблевых методов (способ, при котором несколько нестабильных методов машинного обучения объединяют для исправления ошибок друг друга), обучение с подкреплением предполагает путь проб и ошибок (Рис. 2).

Рис. 2. Методы МО

На сегодняшний день существует три варианта использования ИМ в разработке ИИ, на основе алгоритмов обучения с подкреплением:

Рис. 3. Взаимодействие ИМ и ИИ

Так как в статье взяты за основу примеры для 3 варианта, стоит кратко описать возможные области применения для 1 и 2 вариантов:

1. ИМ модель может быть использована как генератор любого количества чистых, детальных, помеченных данных, покрывающих любое нужное множество вариантов развития событий. Разработчики ИИ часто страдают как от недостатка данных, так и от их зашумленности. Поскольку в ИМ мы полностью контролируем детальность и случайность, а также можем измерить и оценить происходящее, модель может быть источником данных в схеме обучения с учителем.
2. ИИ, обученный произвольным способом (как внутри, так и вне модели), может быть включён в имитационную модель просто как компонент по принятию решений – так создается возможность протестировать ИИ до его разворачивания на реальном объекте.

Выше указанные методы имеют по крайней мере один серьезный недостаток, они не масштабируется, так как их обучение существенно зависит от руководства человека. Именно поэтому был выбран вариант на основе обучения с подкреплением.

В AnyLogic метод обучения с подкреплением можно реализовать тремя способами:

1. Импорт инструментов ИИ в среду моделирования (возможно выполнить только через пользовательский (нестандартный) эксперимент) (см. рисунок 4) [2].

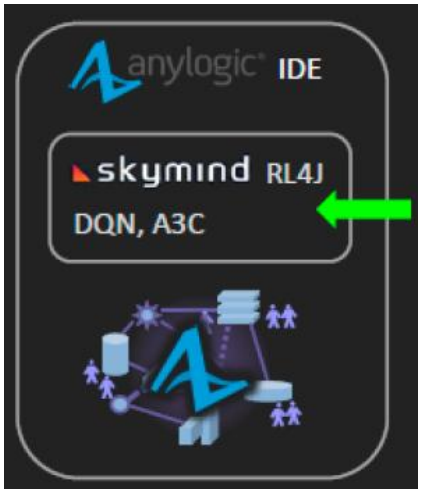

Рис. 4. Импорт инструментов ИИ в среду

Рассмотренное решение предполагает использование и платформы Skymind, которая позволяет автоматизировать интеграцию алгоритмов и имитационной модели и находит оптимальный алгоритм с помощью библиотек для глубокого обучения с подкреплением, и RL4J - это фреймворк для обучения с подкреплением, интегрированный с deeplearning4j и выпущенный под лицензией с открытым исходным кодом Apache 2.0, который эти библиотеки предоставляет [2].

Варианты реализации обучения с подкреплением достаточно разнообразны и предполагают огромное множество алгоритмов, к примеру:

1) Q – обучение - метод, применяемый в искусственном интеллекте при агентном подходе. К нему относятся:
a. Deep Q Learning (DQL);
b. Deep Q Network (DQN);
2) Asynchronous Advantage Actor-Critic – A3C (описывает асинхронные методы для глубокого обучения с подкреплением);
3) ProximalPolicyOptimization (PPO) – решает задачу получения наибольшего возможного улучшения результатов;

4) Importances Weighted Actor-Learner Architectures (IMPALA) - Архитектура основанная на обучении действующего агента со взвешиванием важности.

В текущей версии программы реализованы следующие алгоритмы из подгруппы Model - Free (свободных от модели или безмодельных): Q – leaning (Apex DQN) и алгоритмов с оптимизацией политики: POP, SAC.

Также метод на основе AnyLogic IDE оставляет возможность для интеграции других видов МО за счет реализации через пользовательский эксперимент.

2. Модель экспортируется как Java-приложение из AnyLogic в Project Bonsai, или H2O, см рисунок 5 [2, 3]:

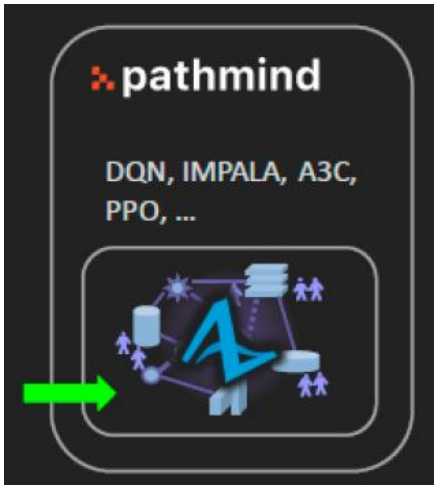

Рис. 5. Работа с автоматизированными платформами

Project Bonsai представляет собой комбинации машинного обучения и глубокого обучения с подкреплением, с помощью которых можно создавать ИИ-модели, способные к оптимизации и автоматизации.

H2O Driverless AI платформа автоматизирующая трудоемкие задачи машинного обучения, включающие в себя: проверку модели, настройку модели, выбор модели и разработку функций.

Ранее была доступна платформа Pathmind, но сейчас она не поддерживается в AnyLogic.

3. Модель выполняется в облаке, ИИ взаимодействует с ней через cloud API, см. рисунок 6 [2]:

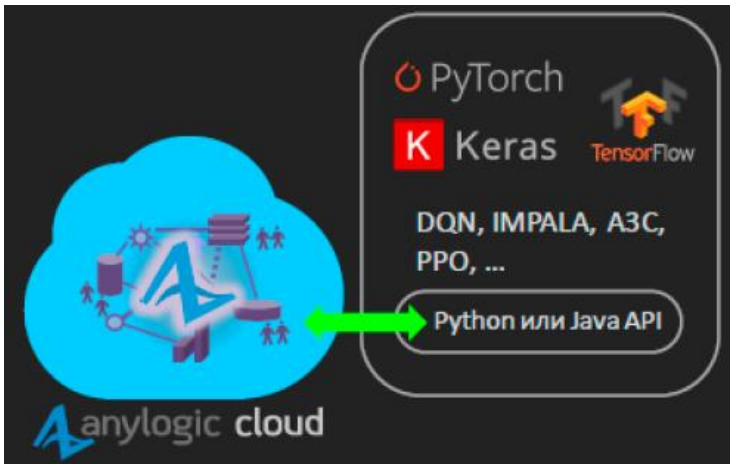

Рис.6. Работа с моделью через Cloud API

Настоящий способ позволяет использовать:

a) PyTorch — фреймворк машинного обучения для языка Python с открытым исходным кодом, созданный на базе Torch;
b) Keras — открытая библиотека, написанная на языке Python и обеспечивающая взаимодействие с искусственными нейронными сетями;
c) TensorFlow — открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия.

В дополнение к веб-интерфейсу AnyLogic Cloud предлагает несколько API-интерфейсов, которые можно использовать для программной настройки и запуска симуляций в рамках аналитических рабочих процессов, запроса результатов экспериментов, создания полностью настраиваемых веб-интерфейсов для моделей.

AnyLogic Cloud поддерживает следующие API:

A) RESTful HTTP-API
B) JavaScript API
C) API Python
D) Java-API

Для работы с ИИ чаще всего используют Java-API и API Python. Рассматриваемый нами способ способен решать задачи не только для обучения с подкреплением, но и классические (с учителем и без учителя) и ансамблевые.

Описание методов обучения с подкреплением:

Чтобы лучше понять принцип работы обучения с подкреплением в программе Anylogic рассмотрим более подробно алгоритмы реализации первых двух методов: Импорт инструментов ИИ в среду моделирования и экспорт модели как Java-приложение из AnyLogic в Project Bonsai, или H2O (в качестве примера будет использоваться Project Bonsai). Так как они чаще всего используются из-за большей доступности. В документации есть библиотека с различными решениями и готовыми моделями.

Импорт инструментов ИИ в среду моделирования AnyLogic:

Наиболее наглядно донести суть данного варианта реализации можно при помощи примера на основе обучения с подкреплением, где используется алгоритм Deep Q Learning, реализованный в RL4J (фреймворк для обучения с подкреплением, интегрированный с deeplearning4j) [1, 4].

Другой пример реализации данного типа эксперимента представлен в статье [5], где происходит сравнение обучения с подкреплением и оптимизационным подходом к управлению перекрестком.

Также данный метод возможно реализовать только при помощи пользовательского эксперимента или нестандартного эксперимента (см. пример на Рис. 7).

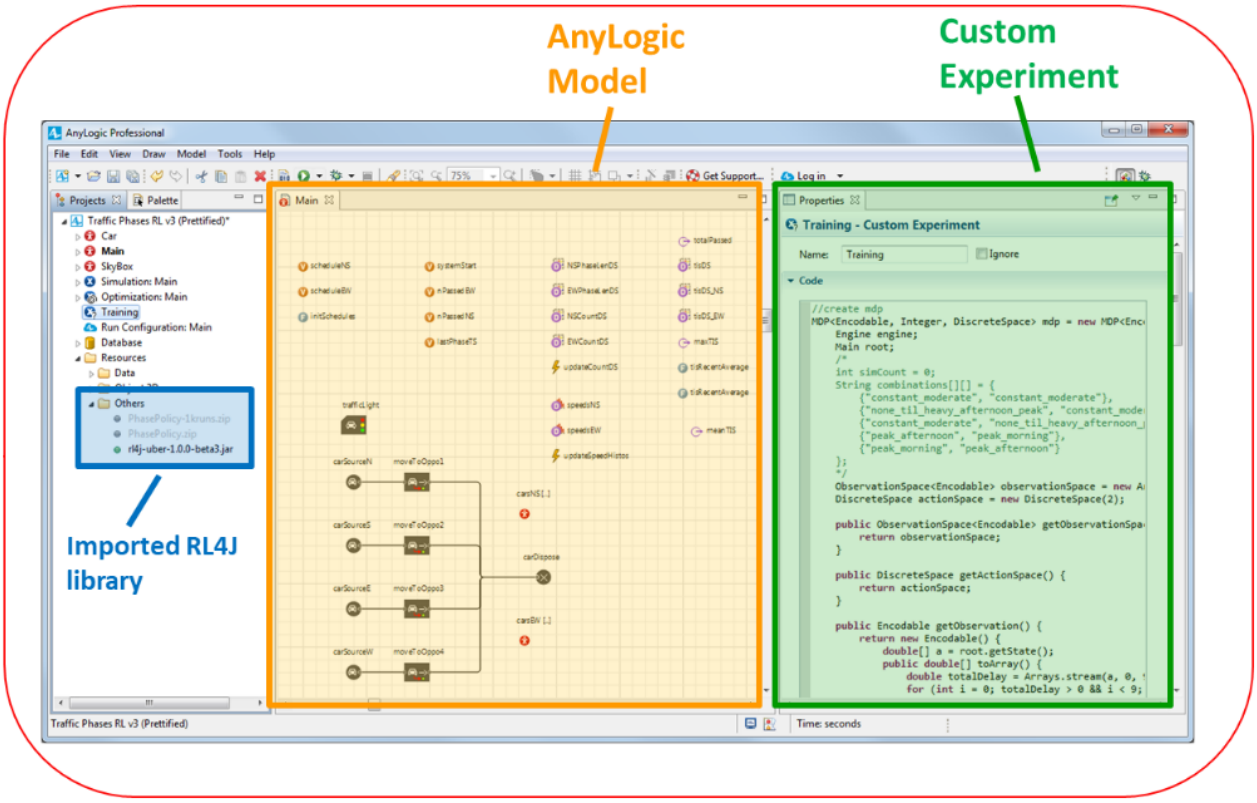

Рис. 7. Интерфейс нестандартного эксперимента (custom Experiment)

Упрощенный алгоритм процесса обучения можно описать следующим образом:

1. на каждом временном шаге обучающийся агент наблюдает за текущим состоянием среды, т.е ИМ;
2. на основе этой информации (которая может не включать полностью всю информацию, содержащуюся в состоянии), он выбирает действие, которое нужно предпринять.
3. после этого происходит временной шаг, и последствия этого действия применяются к системе.
4. затем агент получает числовое вознаграждение и по прошествии времени снова начинает процесс в новом состоянии.

Соответственно, математическая постановка задачи будет выглядеть следующим образом (см. рисунок 8):

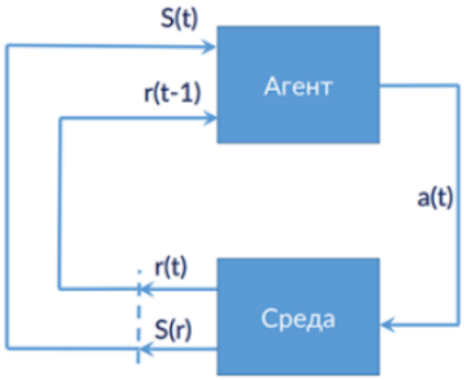

Рис. 8. Схема взаимодействия агента со средой

Среда описывается марковским процессом принятия решений (МППР), который включает S (states) − множество состояний среды; A (actions) – множество возможных действий; условное распределение P(s'|s,a) следующего состояния при условии текущего и действия; R(rewards) − вознаграждения при переходе в новое состояние R(s,s'); коэффициента дисконтирования 0<γ<1, который определяет предпочтение текущего вознаграждения по сравнению с будущим [5].

В произвольный момент времени t агент характеризуется состоянием $s_t \in S$ и множеством возможных действий $A(s_t)$. Выбирая действие $a \in A(s_t)$, он переходит в состояние s_{t+1} и получает выигрыш r_t. Основываясь на таком взаимодействии с окружающей средой, агент, обучающийся с подкреплением, должен выработать оптимальную политику, т.е. стратегию $\pi: S \to A$, которая максимизирует функцию ценности действия математическое ожидание совокупного вознаграждения:

$$Q = E\sum_t \gamma^t r_t \quad (1)$$

Поскольку r(t) всегда лежит в диапазоне между -1 и 1, то значение функции Q определяется в процессе поиска точки в интервале между двумя экстремумами [-1,1].

Чтобы настроить этот процесс, необходимо выполнить три шага [2]:

1. Библиотека RL4J импортируется как зависимость AnyLogic.
2. Создается имитационная модель с добавлением необходимых функций, обеспечивающих связь между моделью и структурой обучения с подкреплением. Чаще всего в модель AnyLogic добавляются две функции: одна для получения наблюдения, а другая для выполнения действия.
3. ИНС настраивается с использованием фреймворка RL4J и подключается к ИМ. Вся эта настройка реализована внутри пользовательского эксперимента.

Работа в Project Bonsai:

Алгоритм процесса обучение с подкрепление в данном методе можно проиллюстрировать следующим образом:

Рис. 9. Схематический поэтапный алгоритм процесса обучения с подкреплением

Алгоритм реализации обучения при помощи Project Bonsai в Anylogic поэтапно описан в документации. Суть данного алгоритма заключается в создании эксперимента RLExperiment, где описываются базовые концепции (см. Рисунок 10) [2]:

1. Наблюдение – ключевые значения, которые передаются агенту ИИ для анализа во время обучения;
2. Действие – значения, которые агент ИИ определяет на каждом этапе, а затем присваивает переменным модели (или функциям)
3. Конфигурация — это набор значений, определяющих начальное состояние модели перед запуском моделирования. Этот код запускается при настройке модели и инициализирует каждый прогон симуляции обучения RL.

Во внутренней структуре эксперимента RL в AnyLogic все эти значения представлены в виде классов Java. Эксперимент с RL позволяет настроить эти значения в модели и управлять ими до начала фактического процесса обучения RL в Project Bonsai. После чего происходит экспорт готовой к RL модели в Microsoft Bonsai, где устанавливается функция вознаграждения и проходится обучение.

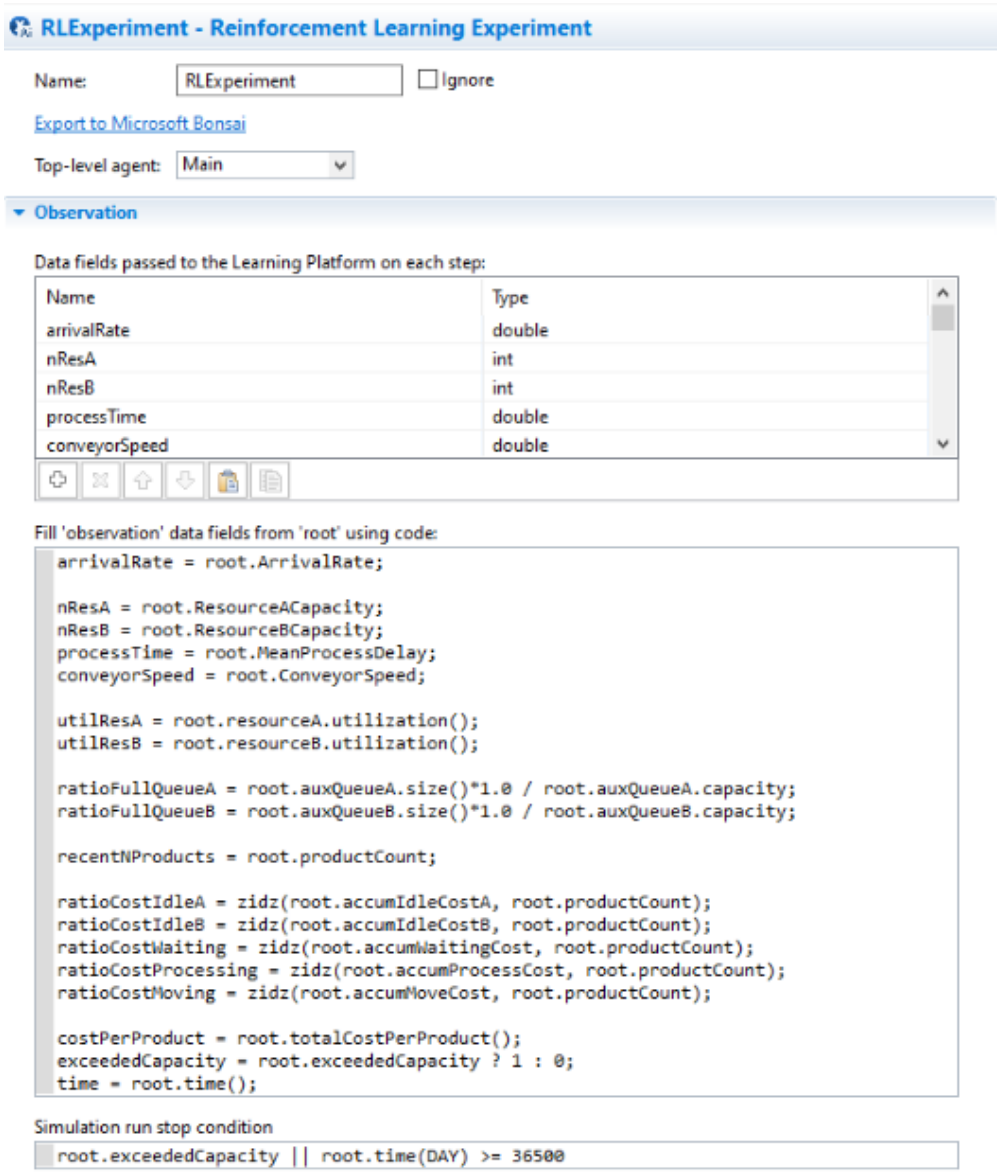

Рис. 10. Пример реализации

Однако существует более интересный способ, который также представлен на официальном сайте AnyLogic, осуществления процесса обучения, через пользовательский эксперимент (см. Рисунок 11) [3]:

1. Добавить заранее установленный элемент Bonsai Connector (позволяет поддерживать разработку ИИ путем интеграции программного обеспечения моделирования с корпорацией Microsoft) в палитру AnyLogic.
2. Создать функцию наблюдение, которая будет передавать агенту ИИ значения для анализа во время обучения.
3. Реализовываются три класса, для наблюдения, действия и конфигурации. Как говорилось выше, они необходимо для взаимосвязи обучения и ИМ.

4. В Bonsai Connector осуществляются построение взаимосвязей между средой, где будет проходить обучение и программой, где создавалась ИМ (ссылки на классы, функции и т.д.).
5. В Bonsai Connector описываются действия (эпизоды в терминах Bonsai) в Connector они называются episodeStart и episodeStep.
6. Создается Пользовательский эксперимент, где прописываются необходимые параметры для осуществления дальнейшего обучения на платформе (перекликается с предыдущим вариантом реализации обучения Импорт инструментов ИИ в среду моделирования AnyLogic).
7. После чего осуществляется экспорт в Microsoft Bonsai, где устанавливается функция вознаграждения и проходится обучение.

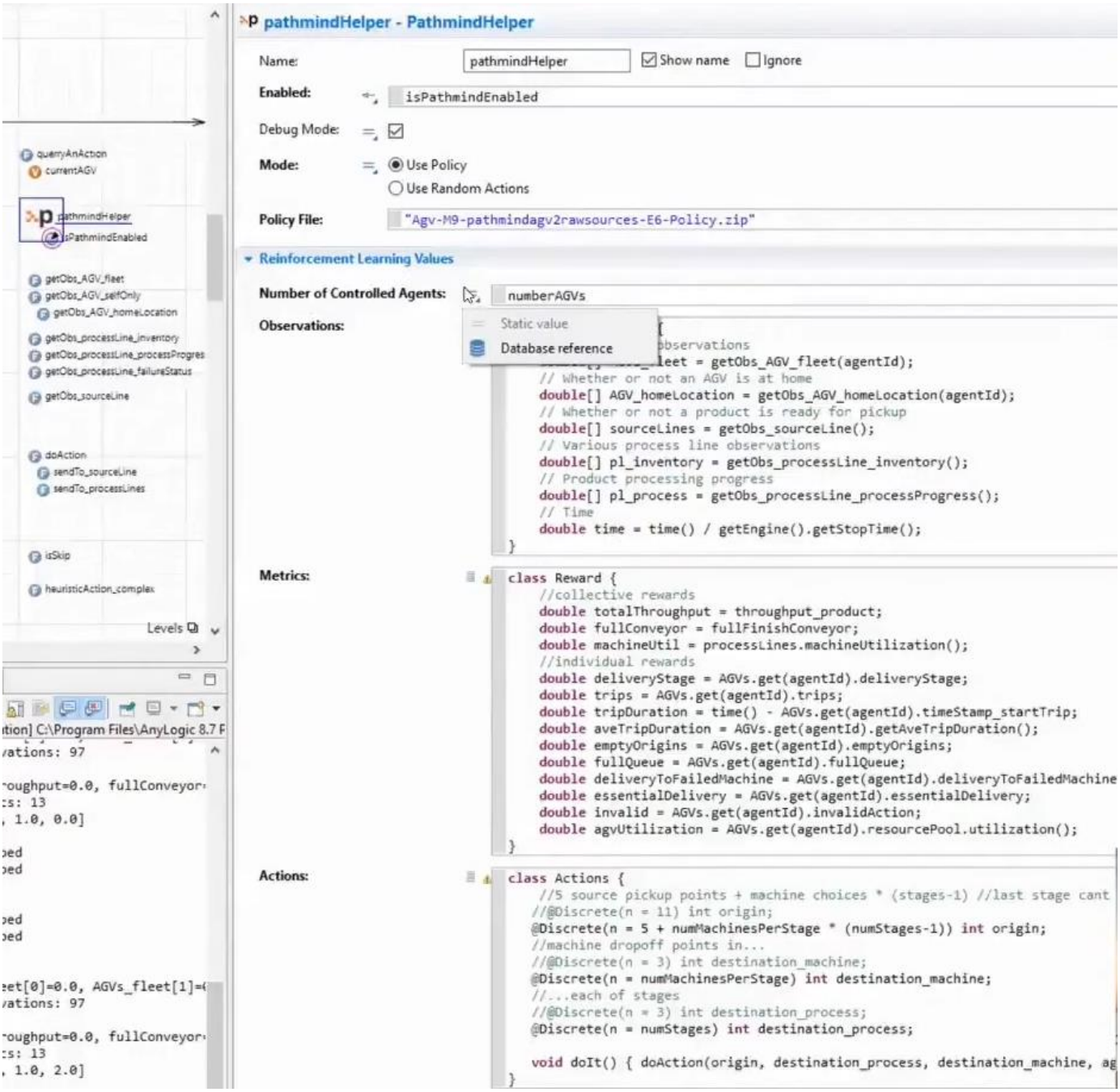

Рис. 11. Пример реализации с использованием Bonsai Connector и пользовательского эксперимента

Результаты

Результаты данной обзорной статьи показали, что все вышеперечисленные методы реализации ИИ в программе более чем удовлетворяют общемировым тенденциям. Также стоит обратить внимание на хорошую связку ИИ и ИМ, имитация позволяет более наглядным образом показывать результаты обучения и позволяет демонстрировать итоги проделанной работы.

Однако для наглядности стоит продемонстрировать главные преимущества и недостатки методов осуществления ИИ в рассматриваемой программе:

Таблица 1 – Преимущества и недостатки

	Преимущества	Недостатки
Импорт инструментов ИИ в среду моделирования AnyLogic	1. Отсутствие сторонних платформ и приложений. Благодаря интейфейсу RL4J можно выполнить поставленные задачи средствами AnyLogic. 2. Быстродействие. За счёт использования пользовательского эксперимента. 3. Возможна интеграция с методом на Project Bonsai или h20.	Доступен только в профессиональной версии
Работа в Project Bonsai или H2O.ai	1. Упрощенное понимание. Из-за минимального использования программирования. 2. Доступ во всех версиях AnyLogic. 3. Большая база разнообразных примеров. 4. Возможна интеграция с первым методом.	Из-за вынужденного использования разных платформ и приложений возможно возникновение ошибок не по вине разработчика
Метод на основе cloud API	1. Подстраивается под ситуацию в режиме онлайн. 2. Большой арсенал различных архитектурных решений и способов работы с ИИ. 3. Интеграция с другими платформами, приложениями и базами данных, которые минимизируют сторонние взаимодействие и могут адаптироваться под разработчика	AnyLogic Cloud API доступен для платных пользователей и пользователей Private Cloud

Таким образом, можно сделать вывод, что рассмотренные выше методы внедрения искусственного интеллекта обучения с подкреплением в имитационную модель не имеют существенных недостатков, но некоторые из них ограничены средой AnyLogic.

СПИСОК ЛИТЕРАТУРЫ

1. Андрей Борщев. Имитационные модели как виртуальная среда для обучения и тестирования искусственного интеллекта для бизнес-приложений / Андрей Борщев, Arash Mahdavi, Анатолий Жеребцов. – Документация AnyLogic, 2019. – 28 с.
2. Справка Anylogic [Электронный ресурс]. – URL: https://anylogic.help/anylogic/index.html (дата обращения: 30.10.2022).

3. Документация Project Bonsai [Электронный ресурс]. – URL: https://learn.microsoft.com/ru-ru/autonomous-systems/bonsai-connectors/ (дата обращения: 30.10.2022).
4. Patterson, J. Deep learning: a practitioner's approach / Patterson, J., Gibson, A. – O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2017 – 532 p.
5. Андронов С. А., Прокофьева М. С. Сравнительная оценка эффективности алгоритмов машинного обучения с подкреплением и оптимизации при управлении транспортными процессами в среде Anylogic // Системный анализ и логистика: журнал.: выпуск №2, ISSN 2077-5687. – СПб.: ГУАП., 2022. РИНЦ. DOI: 10.31799/2077-5687-2021-2-22-41

ИНФОРМАЦИЯ ОБ АВТОРАХ

Прокофьева Марина Сергеевна –
аспирант
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: m4riprokofjeva@yandex.ru

Андронов Сергей Александрович –
кандидат технических наук, доцент
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: andronov_00@mail.ru

INFORMATION ABOUT THE AUTHORS

Prokofeva Marina Sergeevna –
graduate student
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaia str., Saint-Petersburg, 190000, Russia
E-mail: m4riprokofjeva@yandex.ru

Andronov Sergey Alexandrovich –
Candidate of Technical Sciences, Associate Professor
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaia str., Saint-Petersburg, 190000, Russia
E-mail: andronov_00@mail.ru