

УДК 37.016:[004.8:004.43]

П. А. Хорошевич

P. A. Khoroshevich

Хорошевич Павел Александрович, преподаватель, БГПУ им. Максима Танка, г. Минск, Беларусь.

Khoroshevich Pavel Alexandrovich, lecturer, BSPU named after Maxim Tank, Minsk, Belarus.

ПРИМЕНЕНИЕ ВИЗУАЛЬНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ В ХОДЕ ПРЕПОДАВАНИЯ ОСНОВ МАШИННОГО ОБУЧЕНИЯ

USING VISUAL PROGRAMMING LANGUAGES IN TEACHING THE BASICS OF MACHINE LEARNING

Аннотация. В данной статье рассматриваются инструменты, приложения и веб-сервисы, предоставляющие необходимый инструментарий для преподавания основ машинного обучения с применением визуальных языков программирования.

Annotation. This article discusses tools, applications and web services that provide teachers with the necessary tools for teaching the basics of machine learning using visual programming languages.

Ключевые слова: машинное обучение, визуальное программирование, Scratch.

Keywords: machine learning, visual programming, Scratch.

Искусственный интеллект и машинное обучение, в частности, всё прочнее входят в повседневную жизнь человека. Углубленное понимание работы алгоритмов машинного обучения требует знания высшей математики, основ статистики и, как минимум, одного высокоуровневого языка программирования, что значительно усложняет процесс изучения данной предметной области [5]. В рамках преподавания основ машинного обучения, преподавателям и учащимся доступны программы [2] и веб-сервисы [4] предоставляющие простой интерфейс для определения классов и подготовки входных данных для обучения моделей на примере изображений и звуков. Обученную благодаря данным сервисам модель, затем можно экспортировать и применить локально на компьютере учащегося. Но, в таком случае, необходимо знать как минимум один высокоуровневый язык программирования, который поддерживает работу с полученной моделью и послужит интерфейсом для взаимодействия с ней. Рассмотрим онлайн-сервис, дающий учащимся возможность создать проект, подготовить примеры для обучения, распределить их по классам, обучить модель и применить её для создания программ в визуальных средах программирования Scratch и AppInventor.

Онлайн-сервис Machine Learning for Kids [3] («Машинное обучение для детей») является платформой для создания и обучения моделей с различным типом входных данных – текстом, изображениями, звуками и числами. Помимо возможности создать и обучить собственные модели, предоставив необходимое количество примеров, сервис содержит ряд уже обученных моделей для распознавания положения черт лица и рук, а также позы человека.

Работу с сервисом можно осуществлять без регистрации, но авторизованный пользователь может создать несколько виртуальных учебных классов и необходимое количество учётных записей для учащихся данного класса. Преподаватель имеет доступ к просмотру проектов учащихся, а также может создать общедоступный проект, который доступен всем членам класса. Работая с таким проектом, учащиеся обучают и применяют в своих программах одну общую модель. Таким образом, количество примеров для обучения можно значительно увеличить и, тем самым, повысить точность классификации тестовых данных.

Отдельная страница сайта содержит библиотеку уроков, сгруппированных по сложности, типу входных данных, применяемому языку и среде разработки. В методическое сопровождение урока входят инструкции для учащихся, рекомендации для учителя и шаблоны проектов с фрагментами программ, необходимых для быстрого старта работы. На данный момент, к сожалению, все инструкции предоставляются лишь на английском языке.

Рассмотрим процесс создания и обучения модели на примере проекта «Умный класс» [1]. Задача модели распознать текстовую команду и, либо включить, либо выключить свет в классе.

В первую очередь, при создании проекта, пользователь выбирает тип входных данных. Затем необходимо создать именованные контейнеры-классы. Каждый такой контейнер будет содержать данные одного класса, на основе которых будет происходить обучение модели и последующая классификация не входивших в обучающую выборку примеров.

Для реализации умного освещения, учащиеся создают два контейнера с именами *lamp_on* («включить свет») и *lamp_off* («выключить свет»), обозначающие состояние освещения, и наполняют их примерами текстовых команд, которые может ввести пользователь (рис. 1). Стоит отметить, что текстовые команды могут поступать в программу, как с клавиатуры, так и с помощью голосового ввода.

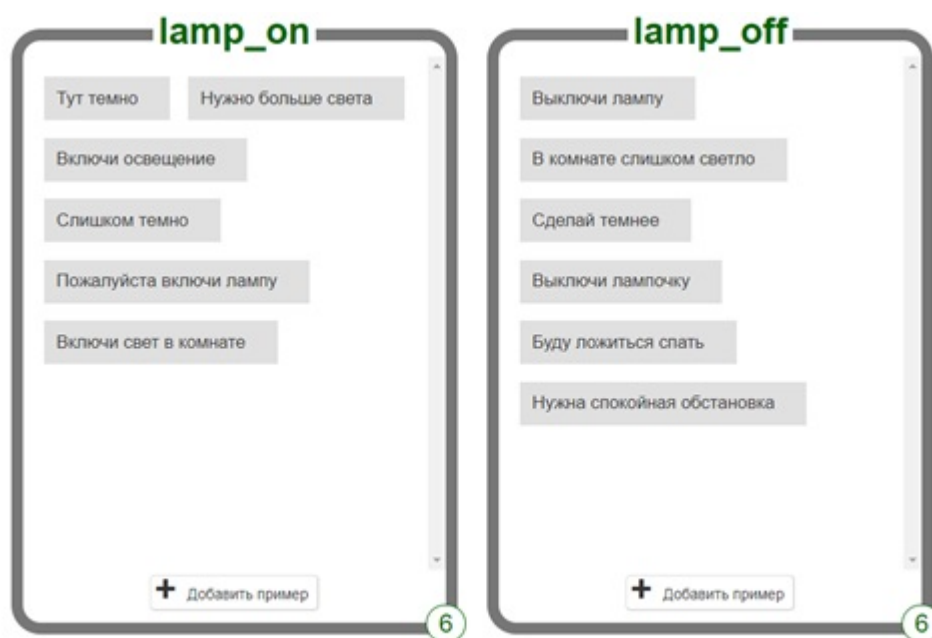


Рисунок 1. Пример контейнеров для модели «Умный класс»

Следующий этап – запуск процесса обучения, по завершению которого можно проверить точность классификации ещё до перехода к среде программирования. После ввода тестовой команды можно получить степень уверенности модели в классификации. Стоит обратить внимание на то, что каждый учащийся сам подбирает примеры для обучения модели и точность классификации, во многом, будет зависеть от количества и качества образцов для обучения.

На третьем этапе обученную модель можно использовать для создания приложения. В случае с визуальной средой программирования Scratch, к обычным категориям, добавляются блоки для передачи данных в модель для последующей классификации. Пример программы, распознающей текст с командой, вводимой с клавиатуры, приведён на рисунке 2.

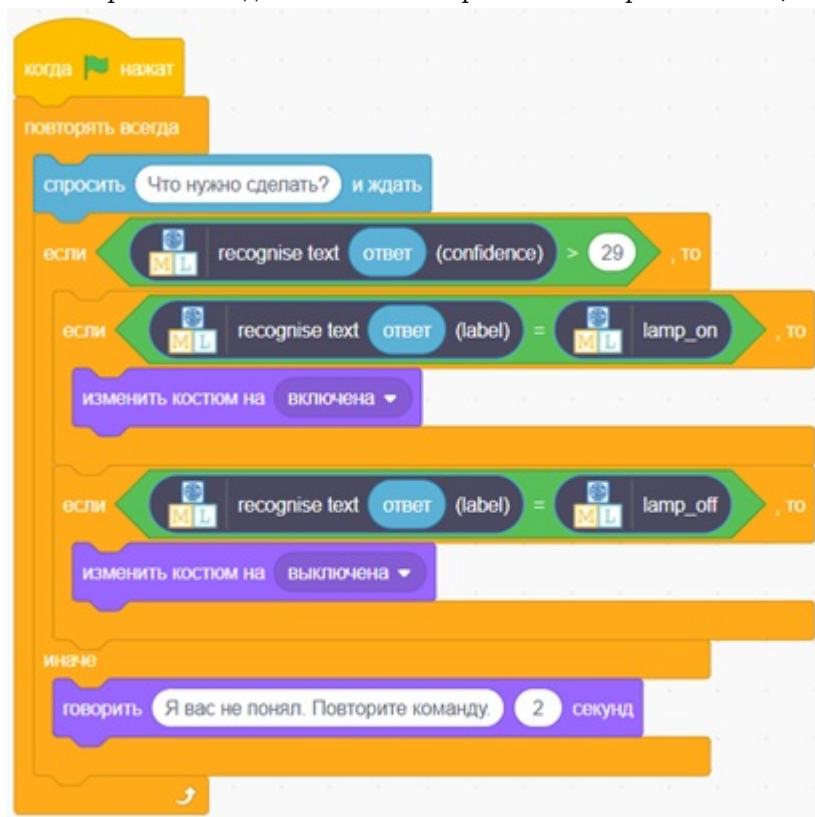


Рисунок 2. Программа для проекта «Умный класс»

Стоит отметить, что помимо Scratch, обученную модель можно применить в визуальной среде разработки мобильных приложений AppInventor, а также в приложениях на языке программирования Python.

В программе выше, команда будет выполнена, если точность классификации текста превышает 30 %. С методической точки зрения, полезно сначала не учитывать точность классификации, чтобы продемонстрировать, как будет вести себя программа при вводе команд, не относящихся к управлению освещением. Подобные команды не должны влиять на состояние умного класса. Для того, чтобы отсеять такие команды и вводится понятие «точность классификации».

Если результаты обучения модели неудовлетворительны, необходимо вернуться к первому этапу, добавить примеры или удалить лишние и повторно обучить модель.

Примеры для обучения можно добавлять в модель непосредственно из программы в процессе её работы. Эта функция особенно полезна в проектах-играх [1]. Учащийся, играя, может собирать данные для обучения модели. Чем лучше играет человек, тем лучше, впоследствии, будет играть искусственный интеллект, обученный на примерах успешных стратегий. Учащимся необходимо описать состояние игры и игрока рядом параметров и создать контейнеры-классы, которые являются отражением действий игрока или ожидаемым результатом этих действий. В результате обучения такой модели строится дерево решений, на основе которого искусственный интеллект будет определять, как действовать в той или иной ситуации. Фрагмент такого дерева решений показан на рисунке 3.

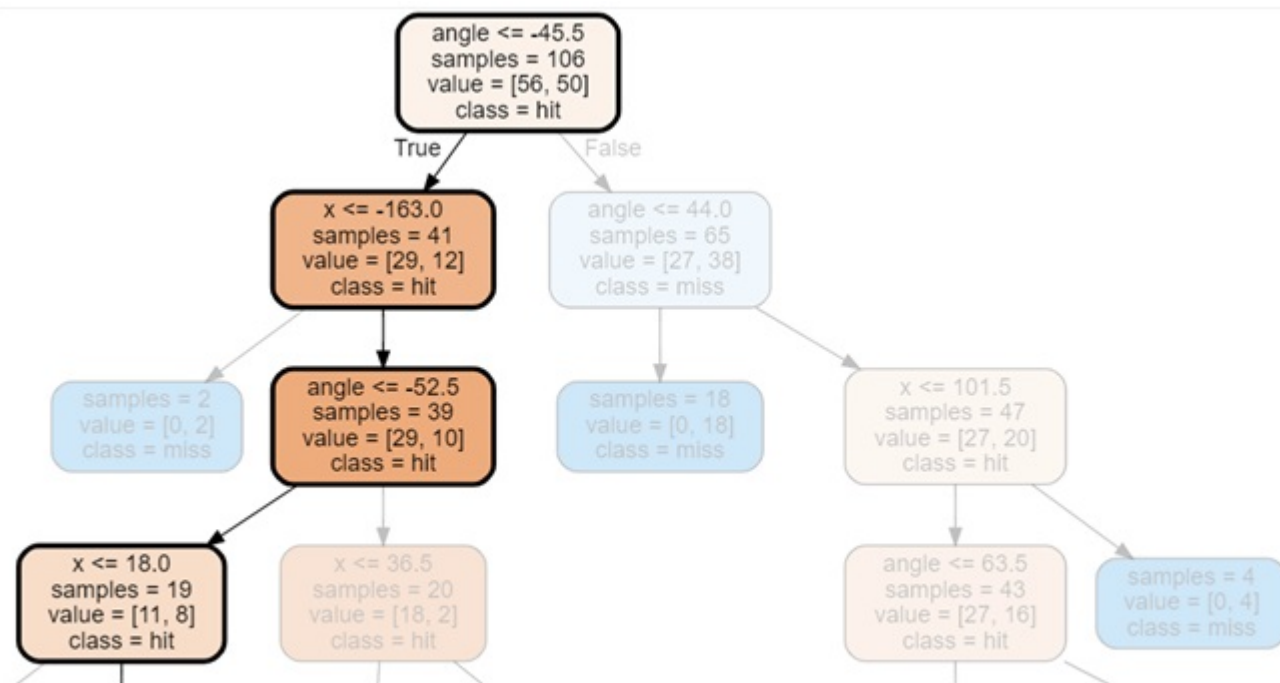


Рисунок 3. Пример дерева решений

В этом примере состояние игры описывалось двумя параметрами – *angle* и *x* (направление броска мяча и координата *x* цели соответственно). Данные для обучения были распределены по двум классам – *hit* (попадание) и *miss* (промах). Программа выбирала угол броска мяча таким образом, чтобы набор параметров, описывающих игру, попадал в класс *hit*. На этапе проверки модели, учащийся может ввести значения параметров и проследить за тем, как будет происходить выбор тех или иных вершин дерева (на рисунке подсвечены).

Проекты, реализуемые благодаря сервису «Машинное обучение для детей» позволяют преподавателю в игровой форме познакомить учащихся с рядом понятий машинного обучения, а также на практике продемонстрировать те трудности, с которыми сталкиваются разработчики в процессе применения моделей машинного обучения.

Список литературы

1. Lane D. Machine Learning for Kids: A Project-Based Introduction to Artificial Intelligence [Текст]. / D. Lane. – SF.: No Starch Press, 2020. – 250 р.
2. Lobe: сайт [Электронный ресурс]. // сайт. – 2021. – URL : <https://www.lobe.ai> (дата обращения : 10.05.2022).
3. Machine Learning for Kids [Электронный ресурс]. // сайт. – 2017. – URL : <https://machinelearningforkids.co.uk> (дата обращения : 10.05.2022).
4. Teachable Machine [Электронный ресурс]. // сайт. – 2017. – URL : <https://teachablemachine.withgoogle.com> (дата обращения : 10.05.2022).
5. Шолле, Ф. Глубокое обучение на Python [Текст]. / Ф. Шолле. – СПб. : Питер, 2018. – 400 с.