

УДК 004.032.26

Разработка нейросети для мобильного робота на базе LEGO Mindstorms EV3

Д. В. Черненко✉, А. Р. Ермолаева,
А. Ф. Сукорская, К. П. Швецов
Витебский государственный
технологический университет, Беларусь
E-mail: Chernenko203509@gmail.com

EV3 (Evolution 3) – это программируемый микрокомпьютер, являющийся «мозгом» роботов в системе LEGO Mindstorms Education EV3. Выпущенный в 2013 г. он имеет невысокие технические характеристики (процессор ARM9 300 МГц, 16 МБ флеш-памяти и 64 МБ оперативной памяти), однако позволяет использовать языки высокого уровня, такие как Java, C и Python. Несмотря на эти ограничения, его можно эффективно применять в исследовательских целях для решения задач прототипирования роботизированных систем, алгоритмов компьютерного зрения, навигации, искусственного интеллекта и машинного обучения.

Основная идея проекта разделена на несколько этапов и заключается в следующем:

1) обучение модели. На компьютере создается и обучается нейронная сеть, которая учится предсказывать правильные действия робота на основе данных с датчиков. Это способ был выбран как самый надежный, так как обучение нейросети требует больших вычислительных ресурсов;

2) развертывание модели. Обученная модель преобразуется в формат, понятный EV3, и загружается в память робота;

3) автономная работа. Робот использует модель для принятия решений в реальном времени без участия человека.

Для загрузки программ на языках высокого уровня модулем EV3 понадобится установить операционную систему, основанную на Debian Linux дистрибутиве – Ev3dev и работающую на совместимых с LEGO MINDSTORMS платформах, включая LEGO MINDSTORMS EV3. Скачать дистрибутив можно по ссылке: <https://github.com/ev3dev/ev3dev/releases/download/ev3dev-stretch-2020-04-10/ev3dev-stretch-ev3-generic-2020-04-10.zip> (355 МБ).

Запись образа осуществляется на карту памяти microSD или microSDHC объемом 2 ГБ.

Нужно вставить SD-карту в EV3 при выключенном питании, нажать центральную кнопку и удерживать ее до появления начала загрузки ev3dev.

Следующим шагом является выполнение настройки SSH, компилятора и C-библиотек. Для этого подключимся к ev3dev с помощью SSH. Подключение к SSH позволит выполнять команды на EV3 по сети, чтобы можно было развертывать код, изменять параметры и устанавливать инструменты.

Выполним установку компилятора C/C++ и проведем установку C-библиотек для работы с оборудованием EV3.

ev3dev-c – основная C-библиотека для управления моторами и датчиками LEGO EV3 под Linux. Библиотека ev3dev-c предоставляет методы работы с сенсорами, моторами и всеми ключевыми устройствами EV3. Для каждого типа сенсора данные читаются через устройства /sys/class/lego-sensor или через API-библиотеки.

Главной задачей проекта является разработка нейросети для навигации робота LEGO EV3. Цель проекта – разработать, обучить и внедрить в робот LEGO EV3 нейронную сеть, которая в реальном времени анализирует данные с ультразвуковых датчиков расстояния [2] и принимает решение о направлении движения, чтобы избежать столкновений с препятствием. Конструкция робота представлена на рис. 1.



Рис. 1. Внешний вид робота с двумя ультразвуковыми датчиками

Разрабатываемая модель представляет собой компактную нейросеть, предназначенную для интерпретации сенсорных данных и принятия решений о направлении движения робота. Ее архитектура включает два логических слоя обработки, что позволяет отнести ее к классу односкрыточных нейросетей [1].

Модель принимает на вход два числовых параметра – расстояния до препятствий, полученные с левого и правого сенсоров робота (left, right).

Центральным элементом модели является скрытый слой, состоящий из двух нейронов. Каждый нейрон получает взвешенную комбинацию входных сигналов, преобразованную с помощью гиперболического тангенса (tanh) – активационной функции, обладающей симметрией и способной моделировать как положительные, так и отрицательные зависимости. Весовые коэффициенты (input_weights) на этом этапе играют ключевую роль в формировании внутреннего представления о пространственной конфигурации препятствий.

На финальном этапе скрытые активации поступают в один выходной нейрон, также использующий tanh в качестве активации. Это позволяет получить непрерывный выходной сигнал в диапазоне от -1 до 1, интерпретируемый как направление движения.

В рамках разработки интеллектуальной управляющей системы для мобильного робота была сформирована обучающая выборка в формате CSV, содержащая эмпирические данные, отражающие реакцию робота на препятствия. Данный набор представляет собой основу для обучения компактной нейросети, способной интерпретировать сенсорные сигналы и принимать решения о направлении движения (рис. 2). В датасете представлены разнообразные сценарии, включая симметричные и асимметричные конфигурации препятствий [3], что обеспечивает достаточную выборку для обучения модели.

```
обучение нейросети...
Эпоха 0, Ошибка: 0.352611 | Тест назад: [0.118652, 0.041269]
Эпоха 300, Ошибка: 0.023810 | Тест назад: [-0.591572, -0.577210]
Эпоха 600, Ошибка: 0.023433 | Тест назад: [-0.609288, -0.596101]
Эпоха 900, Ошибка: 0.023290 | Тест назад: [-0.614738, -0.600896]
Эпоха 1200, Ошибка: 0.023137 | Тест назад: [-0.617107, -0.602372]
Эпоха 1500, Ошибка: 0.022985 | Тест назад: [-0.618686, -0.603316]
Эпоха 1800, Ошибка: 0.022836 | Тест назад: [-0.620056, -0.604260]
Эпоха 2100, Ошибка: 0.022688 | Тест назад: [-0.621392, -0.605283]
Эпоха 2400, Ошибка: 0.022540 | Тест назад: [-0.622757, -0.606379]
Эпоха 2700, Ошибка: 0.022389 | Тест назад: [-0.624164, -0.607524]
Достигнуто максимальное количество эпох. Финальная ошибка: 0.022233
```

Рис. 2. Отчет об обучении нейронной сети

Для отладки работы нейронной сети использовался режим симуляции робота. После выполнения режима симуляции в консоли программы выводится отчет, а затем и результаты выполнения тестовых ситуаций с пояснениями (рис. 3).

Реализация нейронной сети (2 входа, 2 скрытых нейрона, 1 выход) для управления роботом LEGO EV3 представлена на примере на языке Си (рис. 4).

```

== СИМУЛЯЦИЯ РОБОТА ==

Ситуация: Нет препятствий
Вход: [1.00, 1.00]
Выход: Левый мотор=0.72, Правый мотор=0.77
Действие: Движение вперед

Ситуация: Препятствие слева
Вход: [0.10, 1.00]
Выход: Левый мотор=-0.20, Правый мотор=0.90
Действие: Поворот налево

Ситуация: Препятствие справа
Вход: [1.00, 0.10]
Выход: Левый мотор=0.80, Правый мотор=-0.21
Действие: Поворот направо

Ситуация: Препятствие спереди
Вход: [0.20, 0.20]
Выход: Левый мотор=-0.39, Правый мотор=-0.41
Действие: Движение назад *** ЗАДНИЙ ХОД ***

Ситуация: Очень близко слева
Вход: [0.05, 0.80]
Выход: Левый мотор=-0.37, Правый мотор=0.77
Действие: Поворот налево

Ситуация: Очень близко справа
Вход: [0.80, 0.05]
Выход: Левый мотор=0.67, Правый мотор=-0.32
Действие: Поворот направо

Ситуация: Препятствия с обеих сторон
Вход: [0.10, 0.10]
Выход: Левый мотор=-0.63, Правый мотор=-0.61
Действие: Движение назад *** ЗАДНИЙ ХОД ***

Ситуация: Частичное препятствие спереди
Вход: [0.40, 0.40]
Выход: Левый мотор=0.23, Правый мотор=0.12
Действие: Неопределенное действие

=== ТЕСТ ДВИЖЕНИЯ НАЗАД ===
Вход: [0.10, 0.10] - препятствия с обеих сторон
Выход: Левый мотор=-0.63, Правый мотор=-0.61
Действие: Движение назад
    
```

Рис. 3. Отчет о выполнении симуляции тестовых ситуаций

```

float nn_predict(NeuralNetwork* nn, float* inputs) {
float hidden[2]; // Активации скрытого слоя
// Скрытый слой
for (int i = 0; i < 2; i++) {
hidden[i] = nn->hidden_biases[i]; // Начинаем со смещения
for (int j = 0; j < 2; j++) {
// Взвешенная сумма входов
hidden[i] += inputs[j] * nn->input_weights[j][i];
hidden[i] = activate(hidden[i]); // Применяем нелинейность
}
// Выходной слой
float output = nn->output_bias; // Смещение выхода
for (int i = 0; i < 2; i++) {
output += hidden[i] * nn->output_weights[i]; // Взвешенная сумма
}
return activate(output); // Финальная активация
}
    
```

Рис. 4. Пример вычисления выхода нейросети на языке Си

Как видно из примера на рис. 4, в память робота загружается облегченная версия обученной нейросети в виде отдельного файла с обученными весами, который можно загрузить на робот. Эта версия идеально подходит для роботов с ограниченными ресурсами.

В результате проведенного эксперимента с моделью робота, созданного на базе LEGO Mindstorms Education EV3, можно сделать выводы, что даже слабому процессору ARM9, работающему с частотой 300 МГц и памятью 64 МБ, под силу работа с нейросетью, хотя и предварительно обученной. Таким образом, LEGO EV3 – это идеальный баланс между простотой использования и техническими возможностями. Он открывает путь в робототехнику для начинающих, при этом достаточно мощный для реализации серьезных проектов, включая работы с искусственным интеллектом, как в случае с нейросетями.

Список использованных источников

1. Хайкин, С. Нейронные сети. Полный курс / С. Хайкин. – М. : Вильямс, 2006. – 1104 с.
2. Николенко, С. И. Глубокое обучение. Погружение в мир нейронных сетей / С. И. Николенко, А. А. Кадури, Е. О. Архангельская. – СПб. : Питер, 2018. – 480 с.
3. Джонс, М. Т. Программирование искусственного интеллекта в приложениях / М. Т. Джонс ; пер. с англ. А. И. Осипова. – М. : ДМК Пресс, 2018. – 311 с.