

Автоматизация поиска оптимального алгоритма поведения агента с использованием нейронных сетей

*Национальный аэрокосмический университет им. Н. Е. Жуковского
"Харьковский авиационный институт"*

Описан метод обучения нейронных сетей на основе генетического алгоритма. Рассмотрена структура нейронной сети. Приведен краткий обзор параметров, отличающих нейронные сети (тип нейрона, пороговая функция нейрона, его топология). Рассмотрено, в какой области применения нейронных сетей целесообразно применять обучение без учителя и, в частности, генетического алгоритма. Описаны особенности программной реализации нейронных сетей для их совместимости с генетическим алгоритмом. Приведены архитектура разработанного ПО и результаты тестирования. Рассмотрены результаты обучения с помощью эволюционного алгоритма нейронных сетей с различными топологиями.

Ключевые слова: нейронные сети, генетический алгоритм, применение нейронных сетей, топология нейронных сетей, строение искусственного нейрона, выбор алгоритма обучения НС.

Введение

В современном мире все чаще применяют системы с автоматическим управлением, начиная от контроллеров освещенности и заканчивая беспилотными автомобилями и летательными аппаратами (БПЛА). С течением времени повышается не только степень автоматизации, но и степень автономности, независимости устройств от человека. На этом фоне растет сложность создания алгоритмов поведения, в особенности принятия решений. Например, сейчас перспективными являются автопилотируемые автомобили и БПЛА, а также написание для них алгоритмов поведения. Все рассмотренные примеры представляют собой сложные задачи, поэтому существует множество сообществ с открытыми или коммерческими проектами разработки программного обеспечения (ПО) в данных отраслях.

Нейронная сеть (НС), как методика обработки данных в ПО алгоритмизации поведения сложных объектов, занимает лидирующие позиции по применяемости. Благодаря реализованной возможности самообучения НС сейчас нет необходимости строить математическую модель среды и прописывать в ней все варианты принятия решений, а значит, уменьшается объем работ и проблема решения математически сложных задач.

В данной работе рассмотрена возможность создания и использования НС для управления группой автономных агентов, например, как описано в [1].

В работе применяются следующие термины: агент – объект моделируемой среды, который способен получать информацию из среды и проводить действия в зависимости от полученной информации; алгоритм поведения – однозначный набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата на основе имеющихся данных [2]; оптимальный алгоритм поведения – такой алгоритм, при котором агент будет показывать наилучшие результаты для заданного критерия оценки эффективности; искусственная НС – математическая модель, построенная по принципу организации и функционирования биологических НС [3], а также её программное или аппаратное обеспечение; обучение – это процесс, в котором свободные параметры НС

настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обучения определяется способом подстройки этих параметров.

Объект исследования – процесс поиска применимости и оценки эффективности использования НС для формирования алгоритма поведения агента. Предмет исследования – генетический алгоритм обучения НС. Цель – повысить качество поведения агента с учетом минимизации времени его обучения.

Для достижения поставленной цели в работе формируются и решаются следующие взаимозависимые подзадачи: анализ предметной области; создание модели среды функционирования агента; реализация алгоритма обучения НС в виде ПО; определение критериев эффективности обучения; тестирование разработанного ПО в соответствии с ними.

1. Обоснование использования нейронных сетей

Процесс создания НС не подразумевает фазы анализа алгоритмов поведения агента при построении модели. Это в сущности синтетический, а не аналитический подход к созданию ПО. Синтез модели сводится к параметрической оптимизации шаблона (сконструированного по базису активационных функций нейронов). При этом нет обоснования, какому гипотетическому поведению объекта соответствует активность одного из нейронов (элементарных функций) и какой физический смысл приобретают в этом случае синоптические веса. Задача формулируется для всей НС в целом. Исходя из этого объект исследуется целно в тех режимах, которые интересуют в практическом отношении [4].

Использование подхода создания НС позволит обойти затратную по времени и ресурсам (а иногда и невыполнимую) стадию математического моделирования, которую неизбежно придётся проводить при изменении модели среды и/или введении дополнительных элементов среды или логики, что является важной функцией при создании различных моделей.

2. Строеение НС

Основная характеристика НС – ее топология (структура) взаимосвязей элементов. Параметрами топологии являются: размерности входного и выходного слоев; количество скрытых слоев; размерность скрытых слоев друг относительно друга; размерность скрытых слоев относительно входного и выходного слоя; наличие обратных связей; наличие специфических скрытых слоев.

Искусственный нейрон — это атомарная часть НС, которая проводит вычисления. Каждый нейрон характеризуется такими параметрами, как весовые коэффициенты для связанных с ним нейронов (обозначаются как w_i); связи с предшествующими нейронами (x_i); нелинейная функция нейрона (обозначается как f); коэффициенты нелинейной функции (обозначаются как θ_j). Основным принципом работы нейрона — это вычисление функции $f(X, \theta)$:

$$OUT = f(X, \theta), \text{ где } X = \sum x_i w_i. \quad (1)$$

Основополагающей характеристикой нейрона является его функция (таблица).

Распространённые функции нейрона

Наименование	Функция
Жесткая ступенька	$OUT = \begin{cases} 0, & NET < \theta \\ 1, & NET \geq \theta \end{cases}$
Логистическая функция (сигмоида, функция Ферми)	$OUT = \frac{1}{1 + e^{-NET}}$
Гиперболический тангенс	$OUT = th(NET) = (e^{NET} - e^{-NET}) / (e^{NET} + e^{-NET})$
Пологая ступенька	$OUT = \begin{cases} 0, & NET < \theta \\ (NET - \theta) / \Delta, & \theta \leq NET < \theta + \Delta \\ 1, & NET \geq \theta + \Delta \end{cases}$
Экспонента	$OUT = e^{-NET}$
Участки синусоиды	$OUT = \sin(NET)$ для $NET = [-\pi/2, \pi/2]$ или $NET = [-\pi, \pi]$
Гауссова кривая	$OUT = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(NET-m)^2}{2\sigma^2}}$
Линейная функция	$OUT = K * NET, K = const$

Еще одной важной характеристикой НС является способ взаимосвязи нейронов (топология НС). Описание различных топологий приведено в [5], а их схемы построения – на рис. 1.

3. Выбор алгоритма обучения НС

Существует два способа обучения НС: с учителем и без него. Для формирования поведения агента первый способ обучения НС не подходит, поскольку, как правило, отсутствует выборка корректного поведения агента для разнообразных условий среды. Методик обучения без учителя существует несколько, например, метод Хебба, также называемый алгоритмом повторения, генетический или эволюционный алгоритм, обучение с подкреплением. Рассмотрим их принципы.

Метод Хебба основывается на биологическом феномене обучения путем повторения и привыкания. Этот феномен еще известен как эффект повторения. При обучении НС сигнальным методом Хебба усиливаются связи между возбужденными нейронами, в данном случае веса изменяются по следующему правилу [6]:

$$w_{ij}(t+1) = w_{ij}(t) + a * y_i[n-1] * y_j[n], \quad (2)$$

где $w_{ij}(t+1)$, $w_{ij}(t)$ – весовой коэффициент синапса;

a – коэффициент скорости обучения;

$y_i[n-1]$ – выходное значение нейрона i слоя $n-1$;

$y_j[n]$ – выходное значение нейрона j слоя n ; соединяющего эти нейроны, на

итерациях t и $t-1$ соответственно.

НС, обученная методом Хебба, может продолжать обучаться в ходе работы без разработки дополнительных функций и с минимальными потерями в скорости работы.

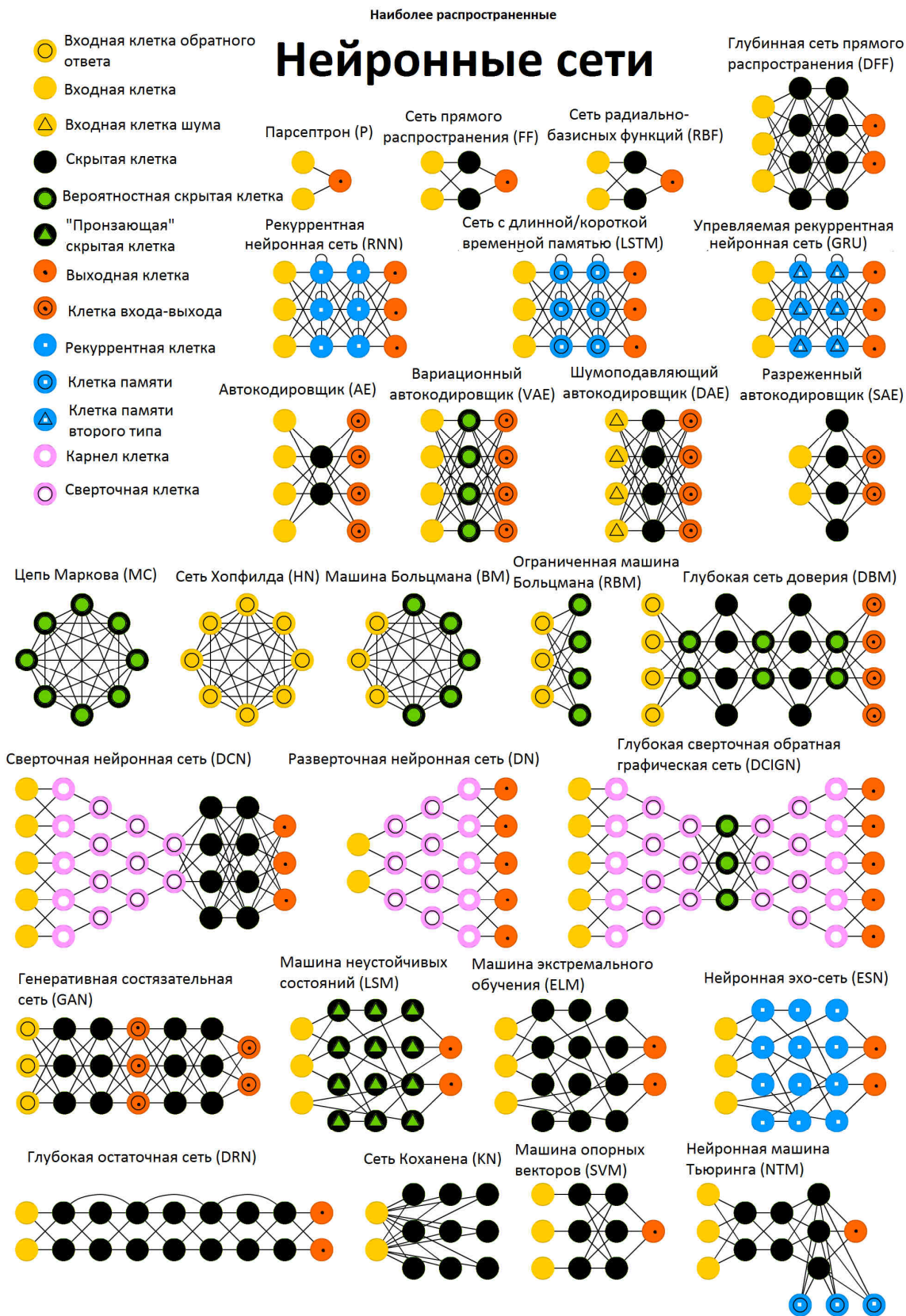


Рис. 1. Большинство используемых видов топологий

Генетические алгоритмы — группа алгоритмов многомерной оптимизации, основанных на моделировании развития биологической популяции [4]. Генетический алгоритм основан на биологических принципах наследственности и изменчивости. Данный алгоритм достаточно медленны и имеет большие требования к объемам оперативной памяти по сравнению с другими алгоритмами обучения НС. Тем не менее для задач формирования поведения выбор этого алгоритма чаще всего бывает единственным вариантом.

Методика обучения с подкреплением требует значительных трудозатрат на создание НС и в данной работе не рассматривается. В качестве основного алгоритма обучения был выбран генетический алгоритм.

4. Описание генетического алгоритма обучения НС

Ключевым понятием в теории генетических алгоритмов является понятие гена и генотипа. Генотип — это вектор значений определенных параметров, которые определяют отличие одной особи в популяции от другой. Ген — значение определенного параметра в генотипе.

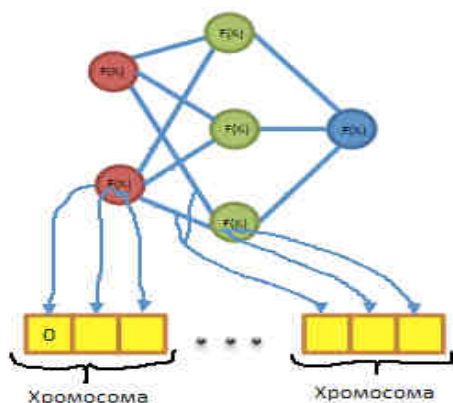


Рис. 2. Кодирование нейронной сети в виде генов

Рассмотрим в качестве генов веса связей в нейронах, типы пороговых функций и значения их параметров (рис. 2). Следует отметить, что сама топология НС остается неизменной.

Важной частью генетического алгоритма является принятие решения о том, какие из представителей данной популяции наиболее пригодны для скрещивания. Для решения этой задачи всегда необходимы два компонента — функция эффективности агента и принцип выбора представителей агентов среди ранжированных по эффективности. В качестве функции эффективности агента будет принято количество собранной агентом еды при условии, что он выжил. Для скрещивания будут выбраны два наиболее эффективных агента, от которых для скрещивания будут взяты их нейронные сети.

условии, что он выжил. Для скрещивания будут выбраны два наиболее эффективных агента, от которых для скрещивания будут взяты их нейронные сети.

5. Программная реализация

Для реализации представленных выше алгоритмов были спроектированы в виде UML диаграмм и реализованы в программном коде три пакета классов: «Experiment_Package», «Environment_Package» и «Network_Package» (рис. 3).

Их назначение и вертикаль управления таковы:

- «Experiment_Package» — основной управляющий пакет, содержащий main функцию и классы, отвечающие за проведение эксперимента и получение управляющих команд от пользователя;

- «Environment_Package» — пакет, отвечающий за логику работы и отображения «среды» и всех ее составляющих, включая агентов;

- «Network_Package» — пакет, отвечающий за работу и эволюционирование НС, которая является решателем (центром принятия решений) агента.

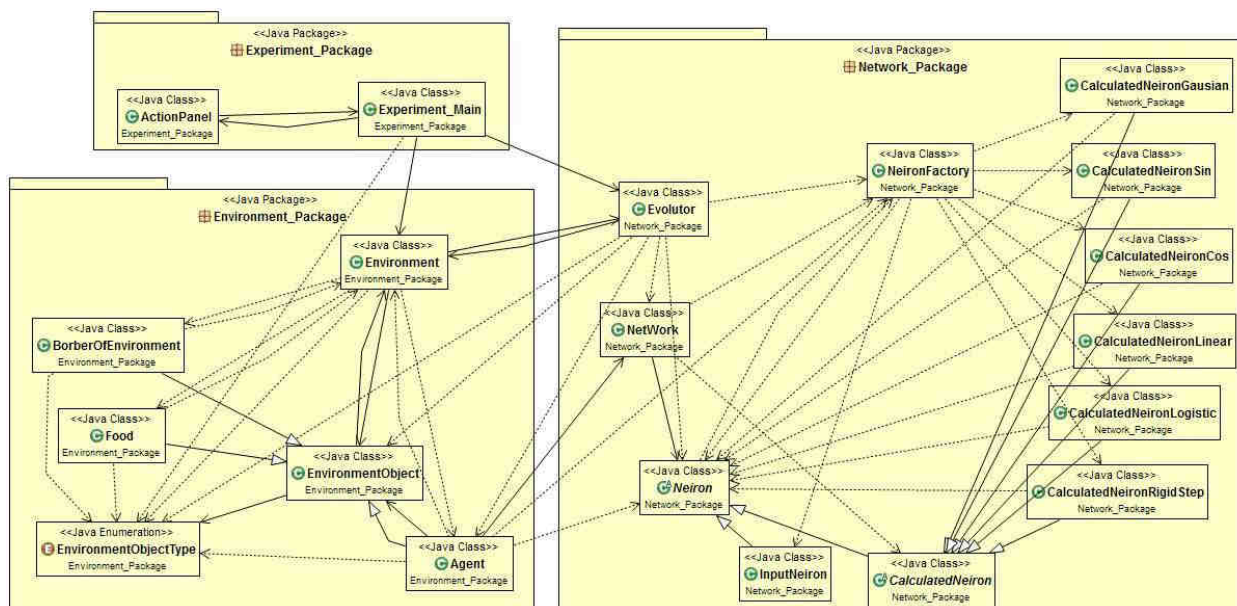


Рис. 3. UML диаграмма пакетов

Управляющий модуль автоматического обучения является скриптом, который заведует начальной инициализацией проведения экспериментов и подает команду на выполнение действий генетическим алгоритмом между экспериментами. Параметрами данного скрипта должны выступать: размеры среды; количество агентов (особей) в популяции; стартовая топология НС агента; количество целей в среде; параметр автоматического проведения эволюции; имя файла для сохранения результатов эволюции.

Было проведено тестирование работы приложения на ноутбуке Lenovo Z500 под управлением ЦП модели Intel CORE i5-3230M (2.6ГГц). В ходе тестирования были выявлены следующие особенности:

- реализованная модель среды работает исправно на больших промежутках времени, что делает ее пригодной для дальнейшего использования;
- генетический алгоритм работает исправно и в соответствии с заданными критериями, однако было выявлено, что скорость прогресса в поведении агента незначительна.

В ходе тестирования было выявлено, что метод обучения с использованием генетического алгоритма зависит от начальных параметров НС (исходного генотипа). Большинство тестов, показавших заметное увеличение эффективности работы агентов на первых 10 – 20 поколениях обучения, в дальнейшем его усиливали, остальные – имели незначительный прогресс или не имели его вовсе. Следует отметить, что примерно 2% тестов показали регресс вначале, но после введения порога минимальной эффективности регресс автоматически пресекался смертью всей популяции. Ввиду всего этого необходимо обратить внимание на выбор начального состояния НС или на то, что не следует сразу начинать долговременное обучение, а сделать ряд стартов по 50 – 100 поколений для поиска хороших кандидатов на обучение.

Отметим, что при максимальной скорости смена одного поколения проходит за 20 секунд. А первые результаты обучения при такой скорости смены поколений будут проявляться через 10 – 15 минут. Наиболее длительные тесты заняли до 32 часов. Лучший результат, которого удалось добиться, – это 100 – 120 собранных

единиц еды одним агентом против 2 – 3 при случайном движении. Но выбранное этими агентами спиралевидное движение к цели не было оптимальным маршрутом. Кроме того, следует отметить, что ввиду технических сложностей наиболее долговременным тестом был тест, развивший популяцию только до 4700-го поколения, что для НС является средним показателем.

Тестами было подтверждено и то, что с увеличением количества нейронов в сети ее обучаемость растет, однако время на ее обучение также значительно возрастает.

В ходе данной работы было протестировано несколько видов топологий НС, а именно: НС прямого распространения (FF), автокодировщик (AE), вариационные автокодировщики (VAE), сеть типа «deep belief» (DBN), глубинные остаточные сети (DRN), свёрточные НС (CNN). НС из разряда кодировщиков и автокодировщиков показали себя плохо. У них не наблюдалось значительного прогресса в обучении, чаще всего именно у них было регрессивное развитие. Нейронные сети прямого распространения показали себя лучше: на малых дистанциях обучения они имели максимальные показатели качества работ агентов, хотя наилучшие показатели у них не были зафиксированы. Кроме того стоит отметить, что правильным решением было применять FF сети на этапе отладки и тестирования программы:

- во-первых, они имеют значительно меньше нейронов и связей (по сравнению с другими сетями), что дает возможность вручную проверить корректность топологии и отследить изменения параметров функций и весов нейронов;

- во-вторых, они дают первые заметные положительные результаты значительно раньше, что позволяет сократить время тестирования.

Лучше всего себя показали глубинные НС. На длинных дистанциях обучения (больше 1000 поколений) они имели наилучшие результаты, однако на небольших дистанциях они были хуже, чем сети прямого распространения.

НС свёрточного типа давали неоднозначные результаты. Так как на больших дистанциях был получен результат, сравнимый с результатом глубинных НС, но на малых дистанциях их результат был заметно хуже. Минусом является то, что у них регрессия или отсутствие прогресса встречалось чаще, чем в глубинных НС.

Проведены тесты при большем количестве входных данных для оценки свёрточно-глубинных НС. Получить удовлетворяющие результаты не удалось из-за технических ограничений аппаратного обеспечения.

Выводы

В данной работе были рассмотрены принципы построения и обучения НС. Приведены различные варианты топологий НС и выбраны наиболее подходящие для решения поставленной задачи. Подробно описаны особенности использования генетического алгоритма для обучения НС. Было проведено теоретическое обоснование выбора использованного набора методик для решения поставленной задачи. По выбранным методикам и алгоритмам создана UML диаграмма архитектуры программного приложения. На основании спроектированной архитектуры описаны алгоритмы, которые были реализованы в виде программного приложения, предназначенного для автоматизированного проведения экспериментов по обучению НС.

Было выполнено тестирование работы приложения, в ходе которого были выявлены недостатки выбранного алгоритма:

- зависимость от случайных величин и от начальных данных;
- большие затраты по времени (на указанном оборудовании).

К плюсам выбранных алгоритмов следует отнести: имеющийся потенциал, а также то, что обученные с их помощью сети легко приспособить для действий при изменившихся параметрах среды.

Список литературы

1. Погудина, О. К. Разработка имитационной модели взаимодействия беспилотных летательных аппаратов для исследования возможности совместного полета [Текст] / О. К. Погудина // Системы обработки информации. – 2012. – Вып. 7 (105). – С. 140-145. – Режим доступа: file:///I:/vidnew/soi_2012_7_30.pdf
2. Кольчугина, Е. А. Результаты эксперимента по созданию эволюционирующего программного обеспечения [Текст] / Е. А. Кольчугина // Известия вузов. Поволжский регион. Технические науки. 2007. №1. URL: <http://cyberleninka.ru/article/n/rezultaty-eksperimenta-po-sozdaniyu-evolyutsioniruyuschego-programmnogo-obespecheniya> (дата обращения: 27.04.2017).
3. Заенцев, И. В. Нейронные сети. Основные модели. [Текст] / И. В. Заенцев, – Воронеж, 1999.
4. Вороновский, Г. К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности, [Текст] / Г. К. Вороновский. – Харьков: Основа. 1997.
5. Фёдор ван Вин. Зоопарк нейронных сетей. [Электронный ресурс]. – <http://www.asimovinstitute.org/neural-network-zoo/>
6. Петунин, Ю. И. Приложение теории случайных процессов в биологии и медицине, [Текст] / Ю. И. Петунин. – Киев: Наук. думка. – 1981.

Поступила в редакцию 07.11.2017

Автоматизация поиска оптимального алгоритма поведения агента с использованием нейронных сетей

Описано метод обучения нейронных сетей на основе генетического алгоритма. Рассмотрена структура нейронной сети. Приведено краткое описание параметров, которые отличают нейронные сети (тип нейрона, пороговая функция нейрона, его топология). Рассмотрено, в какой области применения нейронных сетей целесообразно использовать обучение без учителя и, в частности, генетический алгоритм. Описаны особенности программной реализации нейронных сетей для их совместности с генетическим алгоритмом. Приведена архитектура разработанного ПО и результаты тестирования. Рассмотрены результаты обучения с помощью эволюционного алгоритма нейронных сетей с разными топологиями.

Ключевые слова: нейронные сети, генетический алгоритм, применение нейронных сетей, топология нейронных сетей, структура искусственного нейрона, выбор алгоритма обучения МС.

Automation of Searching for the Optimal Algorithm for Agent Behavior Using Neural Networks

Described the method of training neural networks based on a genetic algorithm. The structure of a neural network is considered. A brief review of the parameters distinguishing neural networks (neuron type, threshold function of the neuron, its topology) is given. It considered is in which field of application of neural networks it is expedient to apply training without a teacher and, in particular, a genetic algorithm. Described features of the software implementation of neural networks for their compatibility with the genetic algorithm. Presented the architecture of the developed software and the test results. Considered the results of training using the evolutionary algorithm of neural networks with various typologies.

Keywords: neural networks, genetic algorithm, application of neural networks, topology of neural networks, artificial neuron structure, choice of learning algorithm.

Сведения об авторах:

Великий Ярослав Александрович – студент 5 курса кафедры 105, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Украина, Email: VelikiyYaroslavMail@Gamil.com.