



# ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ И МАШИННОЕ ОБУЧЕНИЕ ДЛЯ ОПТИМИЗАЦИИ ТРЕНИРОВОК СПОРТСМЕНОВ

УДК 004.9

**Сергей Петров,**  
доцент кафедры  
теории физической  
культуры и спортивной  
медицины Гродненского  
государственного  
университета им. Янки  
Купалы (ГрГУ),  
кандидат медицинских  
наук, магистр  
технических наук

**Александр Кадан,**  
заведующий  
кафедрой системного  
программирования  
и компьютерной  
безопасности ГрГУ,  
кандидат технических  
наук, доцент

**Светлана Зайкова,**  
доцент кафедры  
системного  
программирования  
и компьютерной  
безопасности ГрГУ,  
кандидат физико-  
математических наук

**Никита Жёсткин,**  
студент факультета  
математики  
и информатики ГрГУ

Медицинские и спортивные приложения — одни из лучших примеров применения современных технологий интеллектуальной обработки данных, создания новых имитационных моделей, экспертных систем для представления результатов диагностики, в том числе спортивной и медицинской [1, 2].

Большое внимание в актуальных исследованиях по многим направлениям прикладной информатики уделяется машинному обучению как разделу, который занимается разработкой и анализом программных решений, позволяющих вычислительным системам меняться под воздействием внешних факторов, обучаться. Такие алго-

ритмы делают предсказания или принимают решения не на основе строго статических программных команд, а на базе обучающей выборки данных, с помощью которой происходит настройка параметров модели. Для этого процесса применяются математическая статистика и анализ, методы оптимизации и численные, теория вероятностей, линейная алгебра, дискретная математика, теория графов, различные техники работы с цифровой информацией и др. Результатом действия алгоритма обучения становится функция, которая аппроксимирует (восстанавливает) неизвестную зависимость в обрабатываемых данных.

Одна из моделей машинного обучения – искусственные нейронные сети (ИНС). Это иерархические классификаторы, которые способны самостоятельно выделять признаки в исходном сигнале. Общим показателем для ИНС является количество скрытых слоев – некоторые современные сети имеют их сотни тысяч.

Выделяют большое множество архитектур ИНС. Наиболее популярны из них следующие:

- *сети без обратных связей или сети прямого распространения сигнала (перцептроны, байесовские ИНС);*
- *сверточные;*
- *генеративные состязательные;*
- *рекуррентные (рекурсивные, нейронные сети Хэминга, нейронные машины Тьюринга);*
- *автокодировщики (ванильные и сверточные);*
- *глубокие ИНС (сети доверия, глубокие машины Больцмана).*

Указанное разделение условное, так как четких границ между ними нет. Например, сверточная нейронная сеть одновременно является сетью без обратных связей, она же относится к глубоким и может быть автокодировщиком или частью более сложной архитектуры [4]. Также стоит отметить, что в настоящее время постоянно появляются новые архитектуры.

Эволюционный, генетический алгоритм (ГА) – это эвристический поиск, который отражает процесс естественного отбора наиболее подходящего элемента. Из исходной популяции он создает новую, используя этапы отбора, скрещивания и мутации. Алгоритм принимает исходную совокупность в качестве входных данных и выбирает функцию пригодности, которая помогает ему создавать оптимальное или почти оптимальное решение. Он генерирует несколько совокупностей, пока не удовлетворит ограничения оптимизации. Наиболее приемлемым способом настройки весовых

коэффициентов нейронных сетей можно считать именно генетические алгоритмы. Это связано с тем обстоятельством, что на начальной стадии нет абсолютно никакой информации о направлении движения в плане настройки весов матрицы. В условиях неопределенности эволюционные методы, в том числе и ГА, имеют наиболее высокие шансы для достижения требуемых результатов [5].

В рамках данной работы были исследованы способы применения ГА для помощи спортсменам в построении оптимальных режимов тренировок в зависимости от индивидуальных особенностей организма. Практическая значимость работы состоит в реализации программного решения, позволяющего диагностировать физическое состояние спортсмена и рекомендовать индивидуальную нагрузку с возможностью ее дальнейшей корректировки тренером и самим спортсменом.

## Выбор инструментов

Для разработки требуемого решения в нашем случае было целесообразно обратить внимание на языки программирования, которые поставляются с готовыми библиотеками и имеют расширенную поддержку науки о данных и моделях данных, – Python, C++, Java, JavaScript, R. Исходя из поставленной задачи построения модели с целью диагностики состояния спортсмена, выбор был сделан в пользу последнего [3]. Это динамический, основанный на массивах, объектно-ориентированный, императивный, функциональный, процедурный и рефлексивный язык программирования.

В последние несколько лет R стал популярным среди специалистов по данным и разработчиков машинного обучения благодаря своим функциональным и статистическим алгоритмам. Он имеет открытый исходный код и доступен на r-project.org и Github. Этот язык управляется и разрабатывается в рамках R Foundation и R Core Team. Кроме того, интерфейс R для TensorFlow позволяет продуктивно работать с высокоуровневыми API-интерфейсами Keras и Estimator. В случае необходимости большего контроля он предоставляет полный доступ к основному API-интерфейсу TensorFlow.

## Разработка и тестирование модели физической нагрузки спортсмена

Система энергообеспечения физической нагрузки описана как трехкомпонентная резервуарная модель с коэффициентами перетоков между резервуарами,

соответствующими сложившимся теоретическим представлениям о связи между частями этой системы. Резервуары отождествляются с источниками энергии и имеют неотрицательное значение. Их вклады в обеспечение текущей физической нагрузки определяет появляющееся неравенство в системе дифференциальных уравнений.

Для наблюдения соответствия модели системе энергообеспечения конкретного спортсмена введен показатель, соответствующий частоте сердечных сокращений, равный отношению текущего уровня нагрузки к максимально доступному, с учетом имеющегося источника всех трех источников энергии суммарно. На предварительном этапе было необходимо установить интерпретатор R и интегрированную среду разработки (IDE) RStudio. Для решения поставленной задачи диагностики параметров спортсмена и динамической смены времени между графиками физического состояния были созданы 20 «временных» скрипт-файлов, названных Vremena1.R, Vremena2.R и т.д.

Их содержание различалось вторым параметром, увеличиваемым в последующих временных файлах на 5 (он может корректироваться). Затем формируется скрипт в RStudio, в котором содержатся данные для модели машинного обучения. Специальный

файл FileSp1.R сначала подключает пакет, библиотеку deSolve, которая устанавливается в IDE через `install.packages(deSolve)`. Далее вводятся «временной» файл и данные для модели машинного обучения, затем переменная, которой передаются состояния.

Их можно отобразить на графиках и интерполировать, записать в переменную используемой дифференциальной функции. В результате будут созданы рекурсивная функция `recc` и функция `Sports`, на основе вычисления контролируемых физических параметров спортсмена. В переменную `out` записывается функция `ode`, которая и служит для решения дифференциальных уравнений (рис. 1).

На заключительном этапе обработки данных с помощью модели выводятся временные графики (рис. 2) и финальный результат изменения контролируемых параметров спортсмена, поведение частоты сердечных сокращений (рис. 3). UK и AK – это параметры, отражающие содержание кислорода в крови спортсмена, причем: UK ( $-kK*UK + wnK*recc()$ ["wK"]), AK (`recc()`["wK"]). Аналогично UL и AL – параметры, связанные с молочной кислотой (лактатом), где UL ( $-kL*UL + wnL*recc()$ ["wL"]) и AL (`recc()`["wL"]). UO и AO – параметры, контролирующие фосфат, здесь UO ( $-kO*UO + wnO*recc()$ ["wO"]) +  $wnK*kK*UK + wnL*kL*UL$  и AO (`recc()`["wO"]).

Контролируемыми параметрами являются:

- объемы трех резервуаров энергии: фосфагенный, гликолитический и окислительный;
- коэффициенты перетока между резервуарами;
- уровень текущей физической нагрузки.

Визуализация данных позволяет отметить следующее. Поведение пульсовой кривой отвечает поведению пульса при реальной нагрузке. Значительно легче оценить сценарии работы системы энергообеспечения физической нагрузки у спортсмена по результатам численного решения системы уравнений, чем вывести следствия логическим путем из теоретических правил взаимодействия источников энергообеспечения (рис. 3).

Система уравнений легко запускается как задача оптимизации с полезной целевой функцией,

```
Fizkult <- function(t, state, parameters) {
  with(as.list(c(state, parameters)),{
    flagO <- (w0O - UO <= 0.)

    wPROP <- getrec(myw(t),
      c(wK = if(w0K - UK <= 0) 0 else w0K - UK,
        wL = if(w0L - UL <= 0) 0 else w0L - UL,
        wO = if(flagO) 0 else w0O - UO))

    ## rate of change
    dUK <- if(flagO) 0. else {-kK*UK} + wnK*wPROP["wK"]
    dAK <- wPROP["wK"]
    dUL <- if(flagO) 0. else {-kL*UL} + wnL*wPROP["wL"]
    dAL <- wPROP["wL"]
    dUO <- -kO*UO + wnO*(wPROP["wO"] + if(flagO) 0. else {kK*UK/wnK + kL*UL/wnL})
    dAO <- wPROP["wO"]
    PP <- ((170-60)/(w0O - UO))*(wPROP["wO"] + if(flagO) 0. else {kK*UK/wnK + kL*UL/wnL})+60
    realativWK <- if(wPROP["wK"] > 0) 1/((w0K - UK)/wPROP["wK"]) else 0
    realativWL <- if(wPROP["wL"] > 0) 1/((w0L - UL)/wPROP["wL"]) else 0
    realativWO <- if(wPROP["wO"] > 0) 1/((w0O - UO)/wPROP["wO"]) else 0
    ## return the rate of change
    list(c(dUK, dAK, dUL, dAL, dUO, dAO),
      "PP" = PP,
      "realativWK" = realativWK,
      "realativWL" = realativWL,
      "realativWO" = realativWO)
  }) # end with(as.list ...)
}
```

Рис. 1. Описание резервуарной модели энергообеспечения физической нагрузки с вычислением частоты сердечных сокращений

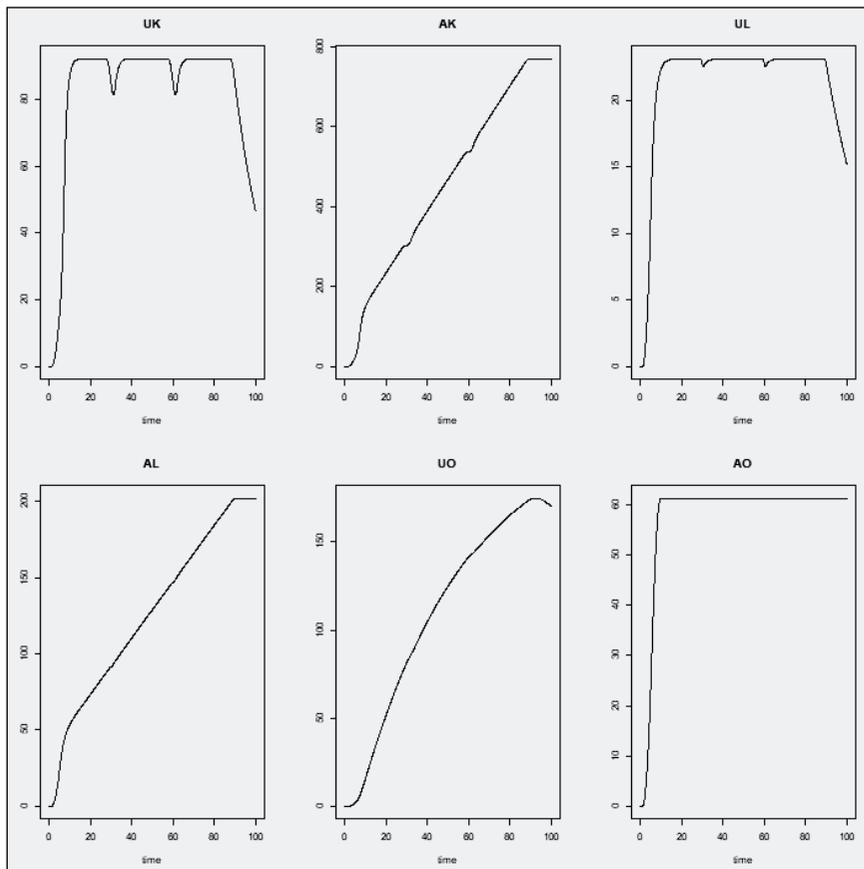


Рис. 2. Представление изменения параметров спортсмена

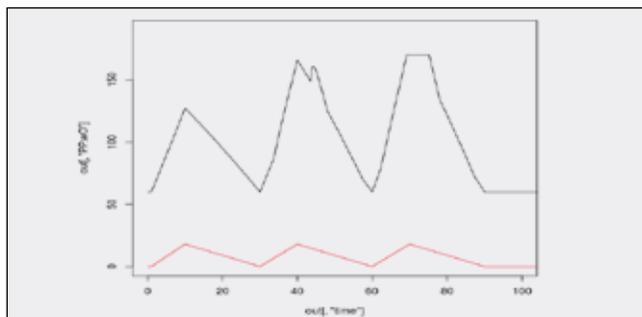


Рис. 3. Поведение частоты сердечных сокращений при заданной повторяющейся нагрузке. Черная линия – поведение частоты сердечных сокращений, красная – заданная и повторяющаяся нагрузка на спортсмена

отражающей важные параметры для планирования объемов и динамики нагрузки в ходе тренировки и соревновательного упражнения. Данные спортсмена без проблем подвергаются индивидуализации путем автоматического подбора параметров источников энергии в соответствии с откликом частоты сердечных сокращений на кривую нагрузки.

Таким образом, язык программирования R, как и среда его выполнения, – хороший выбор для такого рода задач с машинным обучением, включая иден-

тификацию модели выполнения физической нагрузки спортсменом. Используемый подход позволил найти решение быстро и качественно, визуализировать результаты с возможностью быстрого анализа, классификации и сохранения в популярных форматах.

На следующем этапе проводимого исследования с интеллектуальной обработкой данных предполагается применение модели для оценки состояния легкоатлетов на базе факультета физической культуры Гродненского государственного университета им. Я. Купалы. Заключительный этап разработки приложения даст возможность провести финальную корректировку используемого алгоритма с машинным обучением и совершенствовать диагностический функционал предлагаемого программного решения. 

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. А.И. Урбанович, А.М. Кадан, С.А. Зайкова. Кроссплатформенное приложение для диагностики и профилактики тиннитуса // Наука и инновации. 2022. №6. С. 80–83.
2. Петров С.В. О вкладе частоты сердечных сокращений и систолического выброса крови в формирование минутного объема кровообращения / С.В. Петров // Вопросы экспериментальной и клинической физиологии: сборник научных трудов, посвященный 100-летию со дня рождения Аринчина Н.И. – Гродно, 2014. С. 246–251.
3. С.В. Петров, Е.М. Балдин, В.Е. Лявшук. R: GUI на примере. Оснастите свои статистические приложения графическим интерфейсом // Linux Format. 2010. №3(129). С. 62–65.
4. Введение в машинное обучение и искусственные нейронные сети // <https://foobar167.github.io/page/vvedeniye-v-mashinnoye-obucheniye-i-iskusstvennyye-neyronnyye-seti.html>.
5. Мищенко В.А. Использование генетических алгоритмов в обучении нейронных сетей / Мищенко В.А, Коробкин А.А. // <https://science-education.ru/ru/article/view?id=5138>.
6. Абахин А.В. Технология планирования микроциклов подготовки спортсменов в циклических видах спорта с использованием имитационного моделирования / А.В. Абахин [и др.] // <http://prosportlab.com/works/pedagogy/work-5>.
7. Soetaert K. Solving ODEs, DAEs, DDEs and PDEs in R / K. Soetaert, T. Petzoldt // Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM). 2011. №6(1–2). P. 51–65.
8. Scrucca L. GA: A Package for Genetic Algorithms in R // Journal of Statistical Software. 2013. №53(4). P. 21–37.
9. Scrucca L. On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution // The R Journal. 2017. №9(1). P. 187–206.