

УДК: 519.8

Нейросетевой подход к исследованию задач оптимального управления

М. А. Решитько^a, А. Б. Усов^b

Южный федеральный университет,
Россия, 344002, г. Ростов-на-Дону, ул. Б. Садовая, д. 105, стр. 42

E-mail: ^a reshitko@sfedu.ru, ^b abusov@sfedu.ru

Получено 25.09.2021, после доработки — 24.03.2022.

Принято к публикации 27.04.2022.

В статье предлагается метод исследования задач оптимального управления с использованием нейронных сетей. Рассмотрение проводится на примере задачи контроля качества поверхностных вод. При моделировании системы контроля качества поверхностных вод используются теоретико-игровой и иерархический подходы. Исследуется случай динамической двухуровневой системы управления качеством поверхностных вод, включающий ведущего и нескольких ведомых. Рассмотрение ведется с точки зрения ведомых. В этом случае между ними возникает неантагонистическая игра, в которой строится равновесие Нэша. С математической точки зрения при этом решается задача оптимального управления при наличии фазовых ограничений. Для ее аналитического исследования в работе используется принцип максимума Понтрягина, на основе которого формулируются условия оптимальности. Для решения возникающих при этом систем дифференциальных уравнений используется обучаемая нейронная сеть прямого распространения (feedforward). Приводится обзор существующих методов решения подобных задач с помощью нейронных сетей и методов обучения нейронных сетей. Для оценки ошибки решения, получаемого с помощью нейронной сети, предлагается использовать метод анализа дефекта решения, адаптированный для нейронных сетей. Это позволяет получить количественную оценку ошибки численного решения. Приведены примеры использования нейросетевого подхода для решения модельной задачи оптимального управления и задачи контроля качества поверхностных вод. Полученные в этих примерах результаты сравниваются с точным решением и с результатами, полученными методом стрельбы. Во всех случаях величина ошибки оценивается методом анализа дефекта решения. Нейросетевым методом проводится также исследование системы контроля качества поверхностных вод для случаев, когда решение задачи другими методами получить не удалось (большой временной промежуток моделирования и случай нескольких агентов). В статье иллюстрируются возможность использования нейросетевого подхода для решения различных задач оптимального управления и дифференциальных игр, а также возможность количественной оценки точности решения. Полученные результаты численных экспериментов позволяют говорить о необходимости введения регулирующего органа для достижения устойчивого развития системы.

Ключевые слова: оптимальное управление, дифференциальные игры, нейронная сеть, равновесие Нэша, принцип максимума Понтрягина

Работа выполнена при поддержке гранта РФФ, проект № 17-19-01038.

UDC: 519.8

Neural network methods for optimal control problems

M. A. Reshitko^a, A. B. Usov^b

Southern Federal University,
105/42 B. Sadovaya st., Rostov-on-Don, 344002, Russia

E-mail: ^a reshitko@sfedu.ru, ^b abusov@sfedu.ru

Received 25.09.2021, after completion — 24.03.2022.

Accepted for publication 27.04.2022.

In this study we discuss methods to solve optimal control problems based on neural network techniques. We study hierarchical dynamical two-level system for surface water quality control. The system consists of a supervisor (government) and a few agents (enterprises). We consider this problem from the point of agents. In this case we solve optimal control problem with constraints. To solve this problem, we use Pontryagin's maximum principle, with which we obtain optimality conditions. To solve emerging ODEs, we use feedforward neural network. We provide a review of existing techniques to study such problems and a review of neural network's training methods. To estimate the error of numerical solution, we propose to use defect analysis method, adapted for neural networks. This allows one to get quantitative error estimations of numerical solution. We provide examples of our method's usage for solving synthetic problem and a surface water quality control model. We compare the results of this examples with known solution (when provided) and the results of shooting method. In all cases the errors, estimated by our method are of the same order as the errors compared with known solution. Moreover, we study surface water quality control problem when no solutions is provided by other methods. This happens because of relatively large time interval and/or the case of several agents. In the latter case we seek Nash equilibrium between agents. Thus, in this study we show the ability of neural networks to solve various problems including optimal control problems and differential games and we show the ability of quantitative estimation of an error. From the numerical results we conclude that the presence of the supervisor is necessary for achieving the sustainable development.

Keywords: optimal control, differential games, neural network, Nash equilibrium, Pontryagin's maximum principle

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 3, pp. 539–557 (Russian).

This work was supported by Russian Science Foundation, project 17-19-01038.

Введение

Задачи оптимального управления возникают при исследовании систем управления самой разной природы: механических, экономических, эколого-экономических, социальных. Любая задача оптимального управления подразумевает наличие некоторой цели, к достижению которой необходимо прийти оптимальным образом с учетом ограничений на управления и дифференциального уравнения, описывающего состояние системы. Решению таких задач посвящено значительное число работ разных авторов; в частности, обзор методов решения приведен в [Rao, 2009]. Можно выделить два подхода к решению таких задач: косвенный и прямой.

Косвенный подход заключается в исследовании системы аналитическими методами вариационного исчисления, динамического программирования или при помощи принципа максимума Понтрягина [Rao, 2009]. С помощью этих методов аналитически выводятся условия оптимальности. В случае принципа максимума Понтрягина задача сводится к двухточечной краевой задаче и задаче максимизации функции Гамильтона. Двухточечные краевые задачи для систем дифференциальных уравнений редко решаются аналитически и требуют применения численных методов, например метода стрельбы [Rao, 2009]. В целом проблема решения двухточечных краевых задач актуальна [Hua, 1992; Sung, 2001; Holsapple, Venkataraman, 2004], а методы ее решения предлагаются и сегодня [Малкин, 2016], часть их основана на использовании нейронных сетей [Hua, 1992].

Отметим, что метод стрельбы не лишен ряда существенных недостатков; а именно, для метода стрельбы известна проблема сходимости на больших временных отрезках [Rao, 2009], с которой ранее столкнулись и авторы [Решитько, Угольницкий, Усов, 2020].

В прямом подходе [Rao, 2009] функция состояния системы, изменение которой описывается системой дифференциальных уравнений, и неизвестные оптимальные управления аппроксимируются некоторым образом (зачастую различными полиномами). В этом случае осуществляется переход от неизвестных функций к функциям известного вида, но с неизвестными коэффициентами, и далее решается задача скалярной оптимизации. Отметим, что основным недостатком прямых методов является необходимость перехода к задаче численной скалярной оптимизации, где глобальная оптимальность решения не гарантируется.

Задача разработки новых методов исследования задачи оптимального управления остается актуальной и сегодня. В данной работе для решения задач оптимального управления, следуя [Горбаченко, Артюхина, 2007; Коваленко, Черноморец, Петина, 2017; Yize, Yuan, Baisen, 2019; Cybenko, 1989; Zhou et al., 2017; Nazemi, Karami, 2017; Андреева, Цирулева, 2018], предлагается использовать нейронные сети. Суть предлагаемого ниже метода состоит в том, что сначала задача оптимального управления исследуется аналитическими методами, получаются условия оптимальности. Затем неизвестные функции представляются в виде нейронной сети, которая обучается для соответствия условиям оптимальности. Таким образом объединяются косвенный и прямой подходы. Применение именно нейронных сетей оправдано тем, что нейронные сети являются универсальными аппроксиматорами, то есть теоретически могут приблизить любое искомое решение, а кроме того, процесс решения задачи распараллеливается на видеокартах. Подобный подход к исследованию задач оптимального управления использовался в [Nazemi, Karami, 2017; Андреева, Цирулева, 2018]. Отличием данной работы от [Nazemi, Karami, 2017] является использование одной нейронной сети для аппроксимации всех неизвестных функций, кроме того, предлагаемый подход используется не только для задачи оптимального управления, но и для решения некооперативной дифференциальной игры нескольких агентов. Также предложен адаптированный для нейронных сетей метод оценки точности полученного решения на основе анализа дефекта решения. С помощью этого метода удастся количественно оценить точность полученного нейросетевым методом решения. При этом граничные условия удовлетворяются точно, а само решение является непрерывным и дифференцируемым. Кроме того,

в рассмотренных примерах нейросетевой подход позволил получить решения на больших временных промежутках, чем метод стрельбы. Применение нейронных сетей обусловлено тем, что благодаря развитию архитектур сетей, численных методов их обучения [Adam: a method for stochastic optimization; Cyclical learning rates. . . ; Lookahead optimizer. . .] и программных комплексов [PyTorch; autograd; jax; Gym] возможно адаптировать нейронную сеть для решения практически любой задачи [Cybenko, 1989; Zhou et al., 2017].

Статья организована следующим образом: в разделе «Постановка задачи» описывается задача контроля качества поверхностных вод, на примере которой рассматривается предлагаемый метод исследования задач оптимального управления. Далее, в разделе «Методика исследования» приводятся алгоритм решения задачи с помощью нейронных сетей, различные методы обучения сети, а также метод оценки ошибки метода. В разделе «Примеры расчетов» приведены примеры работы метода на модельном примере и задаче контроля качества поверхностных вод.

Постановка задачи

Нейросетевой подход рассматривается на примере задачи контроля качества поверхностных вод, которая уже исследовалась авторами в [Решитько, Угольницкий, Усов, 2020]. Приведем постановку соответствующей задачи.

Рассматривается деятельность предприятий, сбрасывающих в результате своей деятельности загрязняющие вещества (ЗВ) в водоем. Исследуется возможность контроля за сбросом ЗВ предприятием со стороны контролирующего органа посредством установки величины платы за сброс ЗВ. При этом важным является сохранение предприятиями объемов производства. Цели уменьшения сброса ЗВ и сохранения объемов производства являются в некотором смысле противоположными, поскольку с увеличением объемов производства увеличивается и сброс ЗВ, поэтому преследуется цель ограничить концентрацию ЗВ в водоеме в рамках допустимых значений. Задача рассматривается в динамической постановке, то есть сброс ЗВ контролируется на заданном временном отрезке.

Таким образом, рассматриваемая система контроля качества речных вод включает контролирующий орган (супервайзера) и несколько промышленных предприятий (агентов). Промышленные предприятия в результате производства вырабатывают загрязняющие вещества, которые вместе со сточными водами сбрасываются в водоток. Задача рассматривается в игровой постановке и имеет иерархическую структуру. Первым ходит супервайзер, назначая размер платы предприятий за сброс загрязнений. Предприятия выбирают степень очистки сточных вод, когда выбор супервайзера уже известен. В работе исследуется случай безразличного супервайзера. В этом случае плата за сброс загрязнений считается заданной функцией, и модель становится одноуровневой. Агенты стремятся максимизировать свои целевые функционалы, выбирая степень очистки своего стока u_i :

$$\max_{u_i} \int_{t_0}^{t_1} (1 - \eta_i) F_i(t, \phi_i, \rho, u) dt. \quad (1)$$

Ограничения на управления агентов имеют вид

$$u_i \in [u_i^{\min}, u_i^{\max}]. \quad (2)$$

Уравнения динамики системы берутся в виде

$$\dot{\phi} = f(t, \phi_i, \rho, u), \quad \phi(t_0) = \phi_0, \quad (3)$$

$$\dot{\rho} = g(t, \phi_i, \rho, u), \quad \rho(t_0) = \rho_0, \quad (4)$$

$$\phi(t): \mathbb{R} \rightarrow \mathbb{R}^N,$$

$$\rho(t): \mathbb{R} \rightarrow \mathbb{R}.$$

Здесь функция выигрыша i -го агента имеет вид

$$F_i(t, \phi_i, \rho, u) = z_i R_i(\phi_i) - W_i(\phi_i) C_i(u_i) - \frac{\rho v_i W_i(\phi_i)(1 - u_i)}{\rho_{\max}}.$$

Здесь слагаемое $z_i R_i(\phi_i)$ обозначает прибыль агента от реализации произведенной продукции, $W_i(\phi_i) C_i(u_i)$ — затраты агента на очистку сбрасываемого ЗВ, $\frac{\rho v_i W_i(\phi_i)(1 - u_i)}{\rho_{\max}}$ — плата агента за сброс неочищенных ЗВ.

Функции, описывающие динамику системы, взяты в виде

$$\begin{aligned} f_i(t, \phi_i, \rho, u) &= -\beta_i \phi_i + \eta_i F_i(t, \phi_i, \rho, u), \\ g(t, \phi, \rho, u) &= -k\rho + \sum_{i=1}^N W_i(\phi_i)(1 - u_i). \end{aligned}$$

Здесь $\beta_i \phi_i$ характеризует устаревание производственных фондов, а $\eta_i F_i(t, \phi_i, \rho, u)$ — вложение агентом части прибыли в развитие предприятия. $k\rho$ задает самоочищение водоема со временем, а $\sum_{i=1}^N W_i(\phi_i)(1 - u_i)$ — суммарный сброс неочищенных ЗВ агентами.

Приняты следующие обозначения: $u = (u_1, u_2, \dots, u_N)$; u_i — степень очистки стока i -го агента; $[t_0, t_1]$ — временной интервал, на котором ведется рассмотрение; N — число агентов; $\rho(t)$ — концентрация загрязняющих веществ (ЗВ), изменение которой с течением времени описывается уравнением (3); $\phi = (\phi_1, \phi_2, \dots, \phi_N)$ — величина производственных фондов агентов, фонды i -го агента описываются уравнением (4); $R_i(\phi_i)$ — производственная функция i -го агента; $W_i(\phi_i)$ — объем сброса ЗВ i -го агента; ρ_{\max} — максимально допустимая концентрация ЗВ; $v_i(t)$ — размер платы за сброс единицы ЗВ для i -го агента; u_i^{\min}, u_i^{\max} — интервал возможных значений управлений агента u_i ; η_i — доля прибыли, которую агент вкладывает в развитие предприятия; z_i — прибыль агента от реализации единицы продукции; $C_i(u_i)$ — затраты агента на очистку; k — коэффициент распада ЗВ; ρ_0 — концентрация ЗВ в начальный момент времени; $\phi_0 = (\phi_{01}, \phi_{02}, \dots, \phi_{0N})$ — размер производственных фондов агентов в начальный момент времени.

Рассмотрение ведется в игровой постановке, предполагается, что все агенты принимают свои решения независимо друг от друга, стремясь к максимизации своих целевых функционалов. В результате в игре агентов строится равновесие Нэша. При этом каждый агент решает задачу оптимального управления (1)–(4). Для ее решения применяется принцип максимума Понтрягина [Rao, 2009]. Строятся функции Гамильтона H_i :

$$\begin{aligned} H_i(t, \phi_i, \rho, \lambda_{i1}, \lambda_{i2}, u) &= (1 - \eta_i) \left(z_i R_i(\phi_i) - W_i(\phi_i) C_i(u_i) - \frac{\rho v_i W_i(\phi_i)(1 - u_i)}{\rho_{\max}} \right) + \\ &+ \sum_{j=1}^N \lambda_{ij} \left(-\beta_i \phi_i + \eta_i \left(z_i R_i(\phi_i) - W_i(\phi_i) C_i(u_i) - \frac{\rho v_i W_i(\phi_i)(1 - u_i)}{\rho_{\max}} \right) \right) + \\ &+ \lambda_{i0} \left(-k\rho + \sum_{i=1}^N W_i(\phi_i)(1 - u_i) \right) \end{aligned}$$

и выписываются уравнения для присоединенных переменных $\lambda_i(t)$:

$$\begin{aligned} \dot{\lambda}_{ii} &= -\frac{dH_i}{d\phi_i} = -\left(\lambda_{i0} \frac{\partial W_i}{\partial \phi_i} (1 - u_i) \right) - \beta_i \lambda_{ii} + \\ &+ (1 - \eta_i + \eta_i \lambda_{ii}) \left(z_i \frac{\partial R_i}{\partial \phi_i} - \frac{\partial W_i}{\partial \phi_i} C_i(u_i) - \frac{\rho v_i \frac{\partial W_i}{\partial \phi_i} (1 - u_i)}{\rho_{\max}} \right), \quad \lambda_{ii}(t_1) = 0, \\ \dot{\lambda}_{ij} &= -\frac{dH_i}{d\phi_j} = -\left(\lambda_{i0} \frac{\partial W_j}{\partial \phi_j} (1 - u_j) \right) - \beta_i \lambda_{ij} + \\ &+ \eta_j \lambda_{ij} \left(z_j \frac{\partial R_j}{\partial \phi_j} - \frac{\partial W_j}{\partial \phi_j} C_j(u_j) - \frac{\rho v_j \frac{\partial W_j}{\partial \phi_j} (1 - u_j)}{\rho_{\max}} \right), \quad \lambda_{ij}(t_1) = 0, \quad i \neq j, \\ \dot{\lambda}_{i0} &= -\frac{dH_i}{d\rho} = -\left(-\lambda_{i0} l - (1 - \eta_i) \frac{v_i W_i(\phi_i)(1 - u_i)}{\rho_{\max}} - \sum_{j=1}^N \lambda_{ij} \eta_j \frac{v_j W_j(\phi_j)(1 - u_j)}{\rho_{\max}} \right), \quad \lambda_{i0}(t_1) = 0. \end{aligned}$$

В результате получаем две подзадачи ($i = 1, \dots, N$):

1) задача максимизации функций Гамильтона:

$$\max_{u_i} H_i; \quad (5)$$

2) двухточечная краевая задача для переменных состояния и сопряженных переменных:

$$\dot{\phi}_i = -\beta_i \phi_i + \eta_i F_i(t, \phi_i, \rho, u), \quad \phi_i(t_0) = \phi_{0i}, \quad (6)$$

$$\dot{\rho} = -k\rho + \sum_{i=1}^N W_i(\phi_i)(1 - u_i), \quad \rho(t_0) = \rho_0, \quad (7)$$

$$\dot{\lambda}_{ii} = -\frac{dH_i}{d\phi_i}, \quad \lambda_{ii}(t_1) = 0,$$

$$\dot{\lambda}_{ij} = -\frac{dH_i}{d\phi_j}, \quad \lambda_{ij}(t_1) = 0, \quad i \neq j, \quad (8)$$

$$\dot{\lambda}_{i0} = -\frac{dH_i}{d\rho}, \quad \lambda_{i0}(t_1) = 0. \quad (9)$$

Задача (5)–(9) решается в два этапа. Первый этап состоит в решении задачи (5), которая при $N = 1$ представляет собой задачу максимизации функции Гамильтона, а при $N > 1$ — задачу нахождения равновесия Нэша между всеми агентами. Таким образом, определяется вид оптимальных управлений агентов u_i относительно пространственных и присоединенных переменных $u_i = u_i(t, \phi_i, \rho, \lambda_{i1}, \lambda_{i2})$. Для частного вида входных функций задача (5) решается аналитически, а в общем случае из (5) получаются условия оптимальности вида

$$\frac{\partial H_i}{\partial u_i} = -\lambda_{i0} W_i(\phi_i) + (1 - \eta_i + \eta_i \lambda_{ii}) \left(-W_i(\phi_i) \frac{\partial C_i}{\partial u_i} + \frac{\rho v_i W_i(\phi_i)}{\rho_{\max}} \right) = 0,$$

$$\frac{\partial^2 H_i}{\partial u_i^2} = -(1 - \eta_i + \eta_i \lambda_{ii}) W_i(\phi_i) \frac{\partial^2 C_i}{\partial u_i^2} < 0,$$

$$u_i \in [u_i^{\min}, u_i^{\max}].$$

На втором этапе решается двухточечная краевая задача (6)–(9) с учетом найденных функций $u_i(t, \phi_i, \rho, \lambda)$ или условий оптимальности для (5).

Методика исследования

Для исследования задачи контроля качества поверхностных вод (5)–(9) предлагается использовать подход на основе нейронной сети, который дальше будет называться нейросетевым. Суть подхода заключается в том, что искомое решение представляется в виде нейронной сети $NN: \mathbb{R} \rightarrow \mathbb{R}^M$, где M — число неизвестных функций. Вход сети представляет собой временную координату t , а выход сети длиной M включает в себя значения переменных состояния ϕ , ρ , присоединенных переменных λ_i и, возможно, управлений u . Последнее зависит от того, возможно ли аналитическое решение задачи (5). Такая структура нейронной сети соответствует искомому решению задачи, поскольку в конечном итоге ищутся функции, зависящие только от времени. Применению подхода предшествует аналитическое исследование системы (5)–(9), результатом которого может стать, например, решение задачи (5), либо получение условий оптимальности для (5), что упростит применение метода.

Алгоритм решения задачи (5)–(9) при помощи нейросетевого подхода состоит в следующем:

- 1) определение конфигурации нейронной сети $NN(t, w)$ для конкретного набора входных данных с учетом того, что на входе сети — временная координата, а на выходе — значения переменных состояния ϕ , ρ , присоединенных переменных λ и других функций, не определенных при аналитическом исследовании;
- 2) вводится функция потерь, описывающая отклонение значений сети от условий оптимальности (5)–(9);
- 3) выбирается метод обучения сети;
- 4) генерируется обучающая выборка;
- 5) проводится оптимизация функции потерь выбранным методом; обучение прекращается при достижении достаточно малого значения функции потерь;
- 6) оценка полученных результатов, при необходимости переход к пункту 1.

Поясним указанные выше пункты алгоритма.

Вводимая на первом этапе нейронная сеть имеет фиксированную для задачи структуру и описывается своими весами $w \in \mathbb{R}$. Значение решения в момент t определяется парой (t, w) . Задачей, следовательно, является нахождение весов w , при которых нейронная сеть удовлетворяет условиям задачи (5)–(9) с некоторой требуемой точностью. Такой подход аналогичен методу коллокаций [Рао, 2009], где решение ищется в виде полиномов различных степеней с неизвестными коэффициентами. Для решения задачи возьмем множество точек $T \in [t_0, t_1]$ и для нейронной сети NN потребуем выполнения в них условий (5)–(9) ($i = 1 \dots, N, t \in T$).

Условие максимума функций Гамильтона (5) принимает вид

$$\max_{u_i} H_i(t, NN(t, w), u^*). \quad (10)$$

Обозначим через $NN(t, w)[x]$ часть выхода нейронной сети (либо ее производной $\frac{\partial NN}{\partial t}(t, w)[x]$), соответствующую переменной x . Двухточечная краевая задача (6)–(9) принимает вид

$$\frac{\partial NN}{\partial t}[\phi_i] = f_i(t, NN(t, w), u^*), \quad NN(t, w)[\phi_i](t_0, w) = \phi_{0i}, \quad (11)$$

$$\frac{\partial NN}{\partial t}[\rho] = g(t, NN(t, w), u^*), \quad NN(t, w)[\rho](t_0, w) = \rho_0 \quad (12)$$

$$\frac{\partial NN}{\partial t}[\lambda_{ii}] = -\frac{dH_i}{d\phi_i}(t, NN(t, w), u^*), \quad NN(t, w)[\lambda_{ii}](t_1, w) = 0,$$

$$\frac{\partial NN}{\partial t}[\lambda_{ij}] = -\frac{dH_i}{d\phi_j}(t, NN(t, w), u^*), \quad NN(t, w)[\lambda_{ij}](t_1, w) = 0, \quad i \neq j, \quad (13)$$

$$\frac{\partial NN}{\partial t}[\lambda_{i0}] = -\frac{dH_i}{d\rho}(t, NN(t, w), u^*), \quad NN(t, w)[\lambda_{i0}](t_1, w) = 0. \quad (14)$$

Здесь u^* обозначает либо соответствующий выход сети $NN(t, w)[u]$, либо найденные аналитически оптимальные управления агентов $u^*(t, NN(t, w))$. Заметим, что удовлетворить краевые условия для уравнений (11)–(14) можно, изменив конфигурацию нейронной сети. Достаточно искать решение в виде [Yadav, Yadav, Kumar, 2015] $\overline{NN}(t, w) = A(t) + F(t, NN(t, w))$, где первое слагаемое удовлетворяет краевому условию, а второе обращается в ноль в соответствующий момент времени. Теперь функция $\overline{NN}(t, w)$ заменяет нейронную сеть и автоматически удовлетворяет краевым условиям. Для упрощения выкладок будем далее обозначать $\overline{NN}(t, w)$ как $NN(t, w)$.

В работе используются нейронные сети прямого распространения, которые последовательно пропускают входной сигнал через последовательность слоев. В примерах статьи используются сети с тремя внутренними слоями. Каждый слой осуществляет линейное преобразование входного сигнала и применяет к нему функцию активации. Таким образом i -й слой имеет вид

$$l_i(x_{i-1}) = s_i(x_{i-1}w_i + b_i) \rightarrow x_i,$$

где w_i, b_i — веса i -го слоя, s_i — функция активации, x_{i-1} — сигнал предыдущего слоя либо входной сигнал, w_i — матрица $N_i \times M_i$, b_i — вектор длины M_i , N_i — длина входного вектора, M_i — длина выхода слоя. Параметры M_i, N_i выбираются исследователем с учетом размерностей соседних слоев, размера входа и требуемого размера выхода. Нейронная сеть из k слоев выглядит следующим образом:

$$s_k(s_{k-1}(\dots s_2(s_1(x_0w_1 + b_1)w_2 + b_2)\dots w_{k-1} + b_{k-1})w_k + b_k) \rightarrow x_k.$$

На втором этапе определяется функция потерь L , описывающая отклонение вывода нейронной сети от условий (10)–(14):

$$L = \frac{1}{|T|} \sum_{t \in T} \sum_{i=1}^N \left(\left(\frac{\partial NN}{\partial t}[\phi_i] - f_i(\cdot) \right)^2 + \left(\frac{\partial NN}{\partial t}[\rho] - g(\cdot) \right)^2 + \left(\frac{\partial NN}{\partial t}[\lambda_{ii}] + \frac{dH_i}{d\phi_i}(\cdot) \right)^2 + \right. \\ \left. + \left(\frac{\partial NN}{\partial t}[\lambda_{ij}] + \frac{dH_i}{d\phi_j}(\cdot) \right)^2 + \left(\frac{\partial NN}{\partial t}[\lambda_{i0}] + \frac{dH_i}{d\rho}(\cdot) \right)^2 + U^2 \right). \quad (15)$$

Здесь U обозначает условие оптимальности управлений, которое добавляется, если оптимальное управление не найдено аналитически. Условие зависит от конкретной задачи, примером может служить условие максимума первого порядка $U = \frac{\partial H}{\partial u}(t, NN(t, w))$.

На третьем этапе для обучения сети возможно выбрать различные методы оптимизации. Обычно для нейронных сетей используются модификации стохастического градиентного спуска, такие как momentum, Nesterov's momentum [Sutskever et al., 2013], Adam [Adam: a method for stochastic optimization], Lookahead [Lookahead optimizer. . .], которые направлены на ускорение сходимости. Также важным является выбор скорости обучения (lr — learning rate) для этих алгоритмов. Одним из методов ее выбора является эвристика cyclicLr [Cyclical learning rates. . .], которая позволяет избежать застревания в локальных экстремумах. Ее идея состоит в циклическом изменении lr в некоторых пределах с постепенным уменьшением и сдвигом пределов к нулю. В примерах ниже используются модификация градиентного спуска adam и стратегия выбора скорости обучения cyclicLr.

На четвертом этапе создается обучающая выборка. Поскольку решение задачи сводится к минимизации введенной выше функции потерь (15), то и обучающая выборка состоит из использованного выше множества точек $T \in [t_0, t_1]$, в которых требуется выполнение условий (11)–(14). Такая структура выборки возможна, поскольку параметры задачи (в том числе пространственные и присоединенные переменные) уже отражены в функции потерь. Исследователь выбирает количество точек и их значения. В примерах ниже обучающая выборка берется равномерно из рассматриваемого отрезка $[t_0, t_1]$ и дополнительно нормируется к отрезку $[-0,5, 0,5]$. Для проверки решения используется в 2 раза больше точек, чем в обучающей выборке.

На пятом этапе проводится обучение нейронной сети выбранным методом, решается задача $\min_w L$. Для данной задачи требуется находить производную нейронной сети, как по входу t , так и по весам w_i, b_i . Учитывая, что каждый слой осуществляет линейное преобразование входа, а производная функции активации известна, то, используя формулу производной сложной функции, сравнительно легко вычислить производные $\frac{\partial NN}{\partial t}$ и $\frac{\partial NN}{\partial w}$. Однако большинство библиотек для машинного обучения (например, PyTorch, Tensorflow) совершают эту операцию автоматически. При вычислениях на видеокarte целесообразно для ускорения вычислений обрабатывать сразу несколько примеров из выборки. Современные библиотеки для машинного обучения позволяют выписать функцию потерь в векторном виде, обрабатывающую любое число примеров из выборки. В этом случае на каждом шаге минимизируется значение L по выбранному подмножеству примеров. Добавление случайного шума к выборке во время обучения позволит увеличить размер выборки и улучшить сходимость выбранного метода.

На последнем этапе производится оценка результата. Традиционной для задач машинного обучения является оценка значения функции потерь, поскольку ее малость свидетельствует (в данном случае) о соответствии обученной сети условиям (10)–(14) с некоторой точностью. Однако такой метод не выражает это соответствие в количественном виде. Поэтому в данной работе предлагается использовать значение функции потерь на этапе обучения, а затем оценивать полученное решение более точно. Предлагается использовать методы на основе анализа дефекта (defect) решения [Auzinger, 2011; Stetter, 1978; Zadunaisky, 1976]. Приведем подход к оценке ошибки для случая, когда оптимальные управления известны, то есть решается система ДУ (6)–(9). Сначала находится решение исходной задачи описанным методом, то есть такие веса w_1 , которые обеспечивают малое значение L . Далее вводится величина

$$D = \frac{\partial NN}{\partial t}(t, w_1) - f(t, NN(t, w_1)),$$

называемая дефектом (defect), где f обозначает правые части уравнений (11)–(14), и рассматривается вспомогательная задача:

$$\dot{z} = f(t, z) + D(t), \quad z(t^*) = NN(t^*, w_1). \quad (16)$$

Здесь t^* соответствует одному из концов отрезка $[t_0, t_1]$ в соответствии с граничными условиями в (11)–(14). Заметим, что $NN(t, w_1)$ является точным решением вспомогательной задачи (16). Теперь найдем решение задачи (16) описанным методом, получим веса w_2 . Далее ошибка исходного решения оценивается как $e(t) = NN(t, w_2) - NN(t, w_1)$, $t \in T$. Если получены неудовлетворительные значения ошибки, исследователь может поменять структуру сети (увеличить число и размер слоев, поменять функции активации), метод обучения сети, размер и вид обучающей выборки, затем повторить процесс обучения сети. В частности, итеративное увеличение размера и числа слоев нейронной сети возможно благодаря предложенному методу оценки ошибки.

Примеры расчетов

Исходный код всех примеров доступен на github [NN-for-Optimal-Control]. Наиболее простой способ запустить примеры — использовать google colab. Примеры 1 и 2 являются модельными и демонстрируют работу нейросетевого метода и оценку его точности по сравнению с другими методами.

Пример 1

В этом примере рассмотрена задача оптимального управления, решаемая аналитически. Для этой задачи применяется предложенный метод решения, что позволило сравнить ошибку метода относительно точного решения. Рассмотрим задачу оптимального управления вида [Kamien, Schwartz, 1991]

$$\max_{u(t)} \int_1^5 (ux - u^2 - x^2) dt,$$

$$\dot{x} = x + u, \quad x(1) = 2.$$

Здесь x — пространственная переменная, а u — неизвестное управление. На управление не накладывается ограничений. Найдем оптимальное управление, используя принцип максимума Понтрягина:

$$H = ux - u^2 - x^2 + \lambda(x + u),$$

$$\lambda = -H_x = -u + 2x - \lambda, \quad x(1) = 2, \quad \lambda(5) = 0,$$

$$H_u = x - 2u + \lambda,$$

$$H_{uu} = -2,$$

$$x - 2u + \lambda = 0 \rightarrow u^* = \frac{x + \lambda}{2}.$$

Точное решение двухточечной краевой задачи имеет вид

$$x(t) = C_1 e^{\sqrt{3}t} \left(1 + \frac{2\sqrt{3}}{3} \right) + C_2 e^{-\sqrt{3}t} \left(1 - \frac{2\sqrt{3}}{3} \right),$$

$$\lambda(t) = C_1 e^{\sqrt{3}t} + C_2 e^{-\sqrt{3}t},$$

$$C_1 = \frac{6e^{-\sqrt{3}}}{2\sqrt{3} + 3 - 3e^{8\sqrt{3}} + 2\sqrt{3}e^{8\sqrt{3}}},$$

$$C_2 = \frac{6e^9 \sqrt{3}}{2\sqrt{3} + 3 - 3e^{8\sqrt{3}} + 2\sqrt{3}e^{8\sqrt{3}}}.$$

Найдем решение нейросетевым методом согласно сформулированному алгоритму.

Введем нейронную сеть из трех внутренних слоев с функциями активации сигмоид и линейной активацией на последнем слое. Обозначим ее как $nn(t)$, а ее выходы — соответственно как $nn_x(t)$, $nn_l(t)$. Вход сети соответствует времени, которое будем нормировать как $t = \frac{t-1}{4} - 0,5$; таким образом, t лежит в отрезке $[-0,5; 0,5]$. Число нейронов в слоях — соответственно 4, 8 и 32, считая со входа сети. Для удовлетворения краевых условий будем искать решение в следующем виде: $\overline{nn}(t) = A(t) + F(t, nn(t))$, где $A(t) = [1, 0]$, $F(t) = [(t + 0,5)nn_x(t), (t - 0,5)nn_l(t)]$. Введем функцию потерь, описывающую отклонение нейронной сети от условия задачи:

$$L = \left(\frac{\partial \overline{nn}}{\partial x} [x] - (\overline{nn}_x + u^*) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \lambda} [\lambda] - (-u + 2\overline{nn}_x - \overline{nn}_\lambda) \right)^2.$$

В качестве обучающей выборки введем множество временных моментов T , равномерно распределенных на отрезке $[1, 5]$. В примере берется 51 точка, включая концы отрезка. Таким образом, решается задача минимизации среднего значения L :

$$\min_w \overline{L(t)}, \quad t \in T.$$

Для ее решения используется модификация метода градиентного спуска adam со стратегией выбора скорости обучения cyclicLr. В результате обучения среднее значение L уменьшилось до 0,001. Ниже на графиках приводится сравнение полученных траекторий с точным решением. Также оценена ошибка относительно точного решения, $\Delta = \max_{t \in T} (|nn_1(t) - [x, \lambda]|)$, и ошибка на основе значения дефекта решения $\overline{\Delta} = \max_{t \in T} (|nn_2(t) - nn_1(t)|)$. Здесь $nn_1(t)$ — решение задачи нейросетевым методом, $nn_2(t)$ — решение вспомогательной задачи, для которой точным решением является $nn_1(t)$, полученное нейросетевым методом. Получены значения $\Delta = 0,016; 0,033$ и $\overline{\Delta} = 0,017; 0,02$ для $x(t)$, $\lambda(t)$ соответственно.

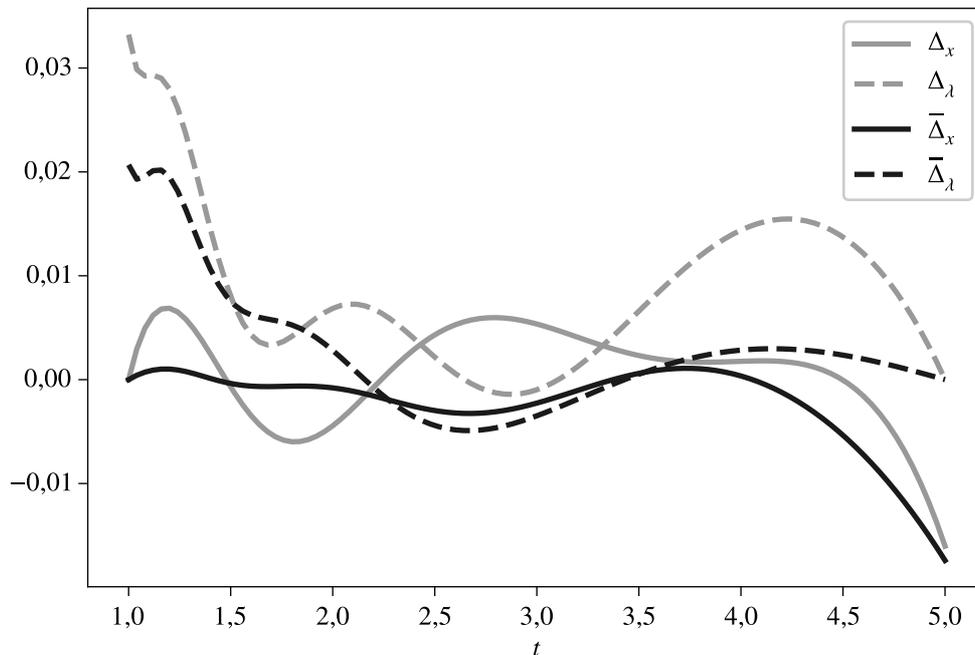


Рис. 1. Сравнение ошибки решения относительно точного решения Δ_x, Δ_λ и ошибки решения, вычисленной предложенным методом, $\overline{\Delta}_x, \overline{\Delta}_\lambda$ для модельного примера 1

Поскольку для задачи известно точное решение, проводится сравнение нейросетевого метода с точным решением, а также оценивается ошибка решения. На рис. 1 показано, что значения Δ и $\bar{\Delta}$ имеют один порядок. Таким образом, возможности предложенного метода демонстрируются на модельной задаче.

Пример 2

В этом примере рассмотрена сформулированная ранее задача контроля качества поверхностных вод (1)–(4) для случая одного агента. В ней точное аналитическое решение не найдено. Проводится сравнение результатов применения предложенного нейросетевого метода с результатами, полученными использованием метода стрельбы, реализованного методами библиотеки для Python SciPy [SciPy]. В этом случае задача (1)–(4) принимает вид

$$\begin{aligned} & \max_{u \in [u_{\min}, u_{\max}]} \int_{t_0}^{t_1} F(t, \phi, \rho, u) dt, \\ & \dot{\phi} = -\beta\phi + \eta F(t, \phi, \rho, u), \\ & \dot{\rho} = -k\rho + W(\phi)(1 - u), \\ & F(t, \phi, \rho, u) = (1 - \eta) \left(zR(\phi) - W(\phi)C(u) - \frac{\rho v W(\phi)(1 - u)}{\rho_{\max}} \right). \end{aligned}$$

Расчеты проводились в случае $N = 1$, $t_0 = 0$, $t_1 = 1$ мес., $\beta = 0,5$, $\eta = 0,5$, $k = 0,5$, $z = 0,1$, $u_{\max} = 0,9$, $u_{\min} = 0$, $\rho_{\max} = 1$ мг/л, $\phi_0 = 1$ у. е., $\rho_0 = 0$ мг/л, $v(t) = 1$, $R(\phi) = \phi$, $W(\phi) = \phi$, $C(u) = u^2$. Функция Гамильтона имеет вид

$$\begin{aligned} H(\cdot) &= \lambda_0(-k\rho + w(\phi)(1 - u)) - \beta\phi\lambda_1 + (1 - \eta + \eta\lambda_1) \left(zR(\phi) - W(\phi)C(u) - \frac{\rho v W(\phi)(1 - u)}{\rho_{\max}} \right), \\ \lambda_1 &= -\frac{\partial H}{\partial \phi}, \quad \lambda_1(t_1) = 0, \\ \lambda_0 &= -\frac{\partial H}{\partial \rho}, \quad \lambda_0(t_1) = 0. \end{aligned}$$

Найдем максимум функции H по u . Для этого найдем ноль ее производной и, с учетом ограничений на u , получим

$$u^{opt} = \begin{cases} \operatorname{argmax}(H(u_{\min}), H(u_{\max}), H(u^*)), & u^* \in [u_{\min}, u_{\max}], \\ \operatorname{argmax}(H(u_{\min}), H(u_{\max})), & u^* \notin [u_{\min}, u_{\max}], \end{cases}$$

где

$$u^* = \frac{-\lambda_0 \rho_{\max} + \rho v (1 - \eta + \eta \lambda_1)}{2 \rho_{\max} (1 - \eta + \eta \lambda_1)}$$

— ноль производной H по u .

С учетом найденного управления решим задачу нейросетевым методом.

Аналогично примеру 1 введем сеть из трех слоев с функциями активации сигмоид и линейной активацией на последнем слое. Обозначим ее как $nn(t)$, а ее выходы — соответственно как $nn_{\phi}(t)$, $nn_{\rho}(t)$, $nn_{\lambda_1}(t)$, $nn_{\lambda_0}(t)$. Вход сети соответствует времени, которое будем нормировать как $t = t - 0,5$; таким образом, t лежит в отрезке $[-0,5; 0,5]$. Число нейронов в слоях — соответственно 4, 16 и 64, считая со входа сети. Для удовлетворения краевых условий будем искать решение в следующем виде: $\bar{nn}(t) = A(t) + F(t, nn(t))$, где $A(t) = [\phi_0, \rho_0, 0, 0]$, $F(t) = [(t + 0,5)nn_{\phi}(t), (t + 0,5)nn_{\rho}(t), (t - 0,5)nn_{\lambda_1}(t), (t - 0,5)nn_{\lambda_2}(t)]$. Введем функцию потерь, описывающую

отклонение производной нейронной сети от условия задачи:

$$L = \left(\frac{\partial \bar{nn}}{\partial \phi} [\phi] - \dot{\phi}(\bar{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \bar{nn}(t)}{\partial \rho} [\rho] - \dot{\rho}(\bar{nn}, u^{opt}) \right)^2 + \left(\frac{\partial \bar{nn}(t)}{\partial \lambda_1} [\lambda_1] - \dot{\lambda}_1(\bar{nn}, u^{opt}) \right)^2 + \left(\frac{\partial \bar{nn}(t)}{\partial \lambda_0} [\lambda_0] - \dot{\lambda}_0(\bar{nn}, u^{opt}) \right)^2.$$

Здесь $\dot{\phi}$, $\dot{\rho}$, $\dot{\lambda}_1$, $\dot{\lambda}_0$ — правые части соответствующих уравнений. В качестве обучающей выборки введем множество временных моментов T , выбранных из равномерного разбиения отрезка $[0, 1]$. В примере берется 1000 точек, включая концы отрезка. Таким образом, решается задача минимизации среднего значения L :

$$\min_w \overline{L(t)}, \quad t \in T.$$

Для ее решения используется модификация метода градиентного спуска adam со стратегией выбора скорости обучения cyclicLr. В результате обучения среднее значение L уменьшилось до 10^{-6} . Получены значения ошибки $\bar{\Delta} = 10^{-5}$; 0,0001; 0,0001; 0,0001 для $\phi(t)$, $\rho(t)$, $\lambda_1(t)$, $\lambda_0(t)$ соответственно. Значения Δ относительно метода стрельбы составили 0,0002; 0,0003; 0,0002; 0,0002 для $\phi(t)$, $\rho(t)$, $\lambda_1(t)$, $\lambda_0(t)$ соответственно. На рис. 2, 3 приведено сравнение с решением, полученным методом стрельбы. При этом решения, полученные двумя методами, идентичны.

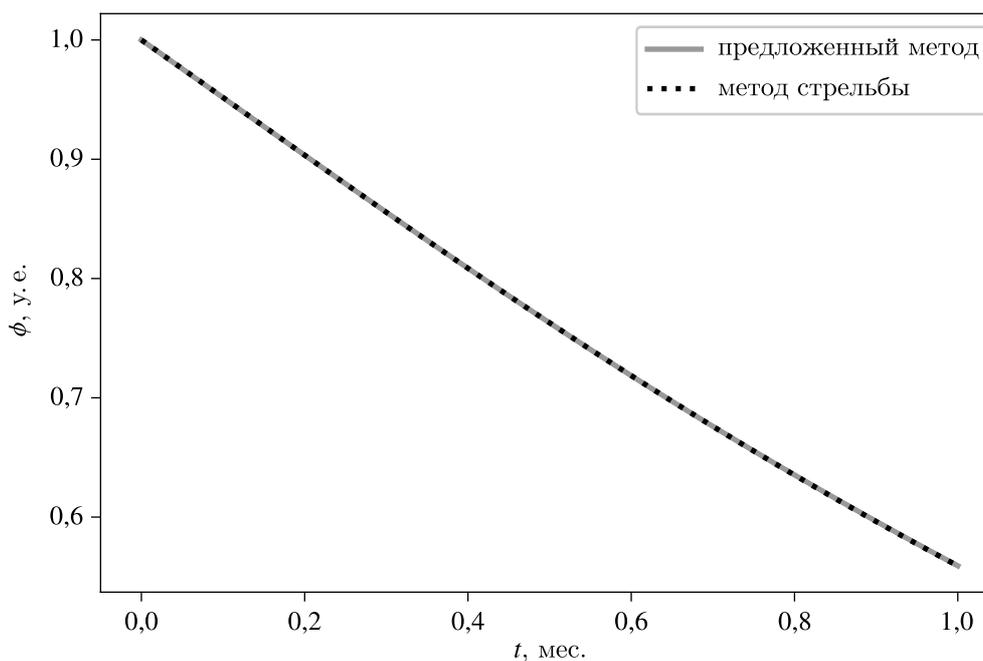


Рис. 2. Сравнение динамики фондов агента для задачи контроля качества поверхностных вод с одним агентом (пример 2), рассчитанной нейросетевым методом и методом стрельбы

Данный пример иллюстрирует способность нейросетевого метода решать задачу контроля качества поверхностных вод с одним агентом. Полученное решение соответствует решению методом стрельбы.

Пример 3

В данном примере рассмотрена задача из примера 2, но на большем временном промежутке ($t_1 = 7$ мес.) и $z = 1,0$. Если моделируемый интервал времени значителен, то решение

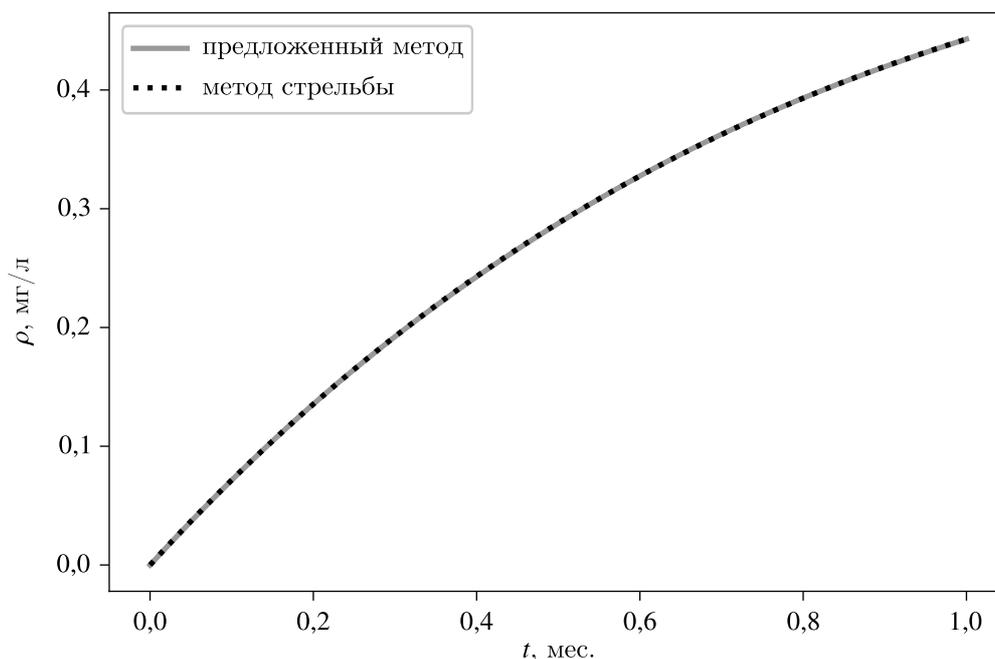


Рис. 3. Сравнение динамики концентрации ЗВ для задачи контроля качества поверхностных вод с одним агентом (пример 2), рассчитанной нейросетевым методом и методом стрельбы

методом стрельбы получить не удастся, метод расходится. Для больших промежутков времени метод стрельбы сходится медленно или расходится. Для метода стрельбы данная проблема описана в литературе [Rao, 2009]. Поэтому в данном примере решение методом стрельбы не находится. Найдем решение нейросетевым методом.

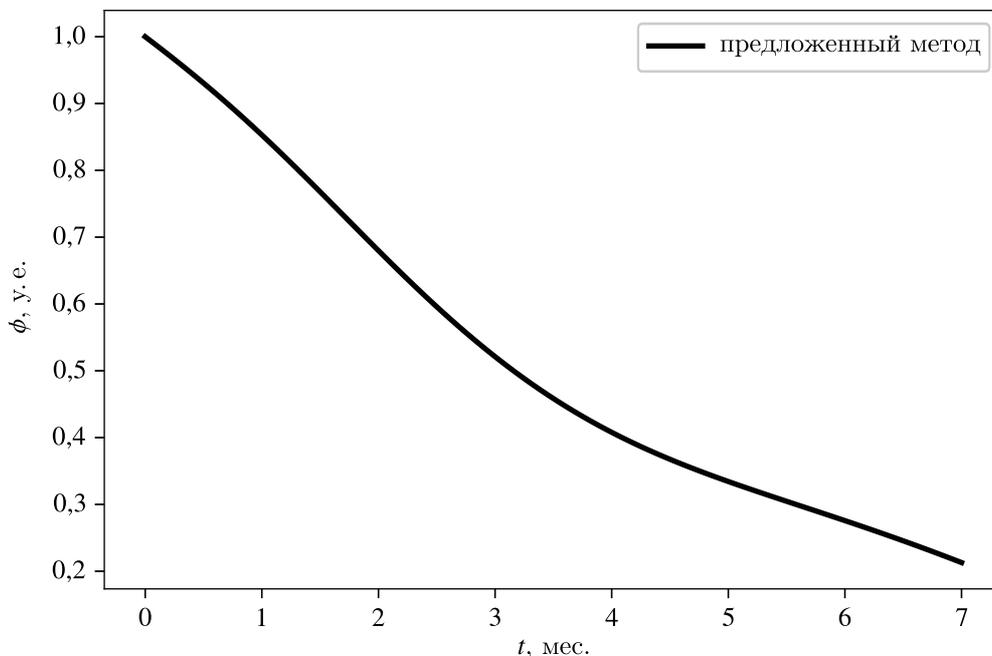


Рис. 4. Изменение величины фондов агента с течением времени для задачи контроля качества поверхностных вод с одним агентом и увеличенным временем (пример 3). Расчеты проведены нейросетевым методом, другими методами решение получить не удалось

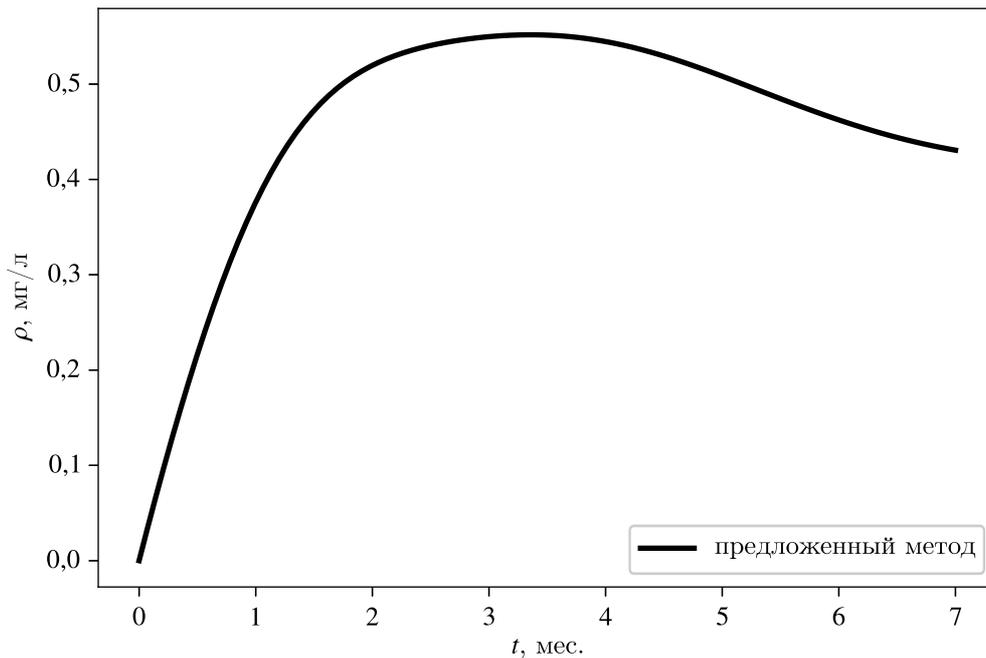


Рис. 5. Изменение концентрации ЗВ с течением времени для задачи контроля качества поверхностных вод с одним агентом и увеличенным временем (пример 3). Расчеты проведены нейросетевым методом, другими методами решение получить не удалось

Введем аналогичную предыдущим примерам нейронную сеть с числом нейронов 4, 32, 128. Вход сети соответствует времени, которое будем нормировать как $t = \frac{t}{7} - 0,5$; таким образом, t лежит в отрезке $[-0,5; 0,5]$. Остальное соответствует примеру 2. Для обучения используется модификация метода градиентного спуска adam+Lookahead со стратегией выбора скорости обучения cyclicLr. В результате обучения среднее значение L уменьшилось до 0,001. Получены значения ошибки $\bar{\Delta} = 0,002; 0,002; 0,003; 0,002$ для $\phi(t), \rho(t), \lambda_1(t), \lambda_0(t)$ соответственно. На рис. 4 и 5 приведены результаты расчетов. Данный пример демонстрирует возможность исследования задачи контроля качества поверхностных вод предложенным методом на больших временных промежутках.

Пример 4

В данном примере исследуется задача (1)–(4) для случая двух агентов. Рассуждая аналогично предыдущим примерам, получим выражения для оптимальных управлений:

$$u_i^{opt} = \begin{cases} \operatorname{argmax}(H_i(u_{\min}), H_i(u_{\max}), H_i(u_i^*)), & u_i^* \in [u_{\min}, u_{\max}], \\ \operatorname{argmax}(H_i(u_{\min}), H_i(u_{\max})), & u_i^* \notin [u_{\min}, u_{\max}], \end{cases}$$

где

$$u_i^* = \frac{-\lambda_{i0}\rho_{\max} + \rho v_i(1 - \eta_i + \eta_i\lambda_{ii})}{2\rho_{\max}(1 - \eta_i + \eta_i\lambda_{ii})}$$

— ноль производной H_i по u_i . Существенным отличием в данном случае является то, что величины H_i в общем случае зависят не только от управлений i -го агента, но и от управлений других агентов; таким образом, получаем задачу нахождения равновесия Нэша для агентов. Определим функцию оптимального ответа агентов $P(u): \mathbb{R}^N \rightarrow \mathbb{R}^N, P_i(u) = u_i^{opt}$. Тогда решение уравнения $P(u) = u$ даст искомое равновесие Нэша. Кроме того, поскольку агентов двое, динамика

системы описывается тремя пространственными переменными ρ , ϕ_1 , ϕ_2 и шестью присоединенными переменными λ_{10} , λ_{11} , λ_{12} , λ_{20} , λ_{21} , λ_{22} . Используем полученные условия оптимальности для решения задачи нейросетевым методом.

Введем сеть из трех слоев с функциями активации сигмоид и линейной активацией на последнем слое. Обозначим ее как $nn(t)$, а ее выходы — соответственно как $nn_{\phi_1}(t)$, $nn_{\phi_2}(t)$, $nn_{\rho}(t)$, $nn_{\lambda_{10}}(t)$, $nn_{\lambda_{11}}(t)$, $nn_{\lambda_{12}}(t)$, $nn_{\lambda_{20}}(t)$, $nn_{\lambda_{21}}(t)$, $nn_{\lambda_{22}}(t)$, $nn_{u_1}(t)$, $nn_{u_2}(t)$. Вход сети соответствует времени, которое будем нормировать как $t = t - 0,5$; таким образом, t лежит в отрезке $[-0,5; 0,5]$. Число нейронов в слоях — соответственно 4, 32 и 128, считая со входа сети. Для удовлетворения краевых условий будем искать решение в следующем виде: $\overline{nn}(t) = A(t) + F(t, nn(t))$, где $A(t) = [\phi_{10}, \phi_{20}, \rho, 0, 0, 0, 0, 0, 0, 0, 0]$, $F(t) = \left[(t + 0,5)nn_{\phi_1}, (t + 0,5)nn_{\phi_2}, (t + 0,5)nn_{\rho}, (t - 0,5)nn_{\lambda_{10}}, (t - 0,5)nn_{\lambda_{11}}, (t - 0,5)nn_{\lambda_{12}}, (t - 0,5)nn_{\lambda_{20}}, (t - 0,5)nn_{\lambda_{21}}, (t - 0,5)nn_{\lambda_{22}}, \frac{u_{\max}}{(1+e^{-nn_{u_1}})}, \frac{u_{\max}}{(1+e^{-nn_{u_2}})} \right]$.

Введем функцию потерь, описывающую отклонение производной нейронной сети от условия задачи:

$$L = \left(\frac{\partial \overline{nn}}{\partial \phi_1} [\phi_1] - \dot{\phi}_1(\overline{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \phi_2} [\phi_2] - \dot{\phi}_2(\overline{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \rho} [\rho] - \dot{\rho}(\overline{nn}(t), u^{opt}) \right)^2 + \\ + \left(\frac{\partial \overline{nn}}{\partial \lambda_{10}} [\lambda_{10}] - \dot{\lambda}_{10}(\overline{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \lambda_{11}} [\lambda_{11}] - \dot{\lambda}_{11}(\overline{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \lambda_{12}} [\lambda_{12}] - \dot{\lambda}_{12}(\overline{nn}(t), u^{opt}) \right)^2 + \\ + \left(\frac{\partial \overline{nn}}{\partial \lambda_{20}} [\lambda_{20}] - \dot{\lambda}_{20}(\overline{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \lambda_{21}} [\lambda_{21}] - \dot{\lambda}_{21}(\overline{nn}(t), u^{opt}) \right)^2 + \left(\frac{\partial \overline{nn}}{\partial \lambda_{22}} [\lambda_{22}] - \dot{\lambda}_{22}(\overline{nn}(t), u^{opt}) \right)^2 + \\ + (P(nn(t)[u_1]) - nn(t)[u_1])^2 + (P(nn(t)[u_2]) - nn(t)[u_2])^2.$$

В качестве обучающей выборки введем множество временных моментов T , выбранных из равномерного разбиения отрезка $[0, 1]$. В примере берется 5000 точек, включая концы отрезка. Таким образом, решается задача минимизации среднего значения L :

$$\min_w \overline{L}(t), \quad t \in T.$$

Для ее решения используется модификация метода градиентного спуска adam со стратегией выбора скорости обучения cyclicLr. Проводились расчеты со следующими входными данными: $N = 2$, $t_0 = 0$, $t_1 = 1$ мес., $k = 0,5$, $z = [1,0; 1,0]$, $u_{\max} = 0,9$, $u_{\min} = 0$, $\rho_{\max} = 1$, $\phi_0 = [1,0; 1,5]$ у.е., $\rho_0 = 0$ мг/л, $R(\phi) = \phi$, $W(\phi) = \phi$, $C(u) = u^2$, значения v , η , β варьировались. В таблице 1 приведены результаты счета и значения v , η , β . Таким образом, увеличение платы за сброс ЗВ может привести к существенному уменьшению фондов агентом и их выигрышей. При этом уменьшение платы за сброс ЗВ приводит к увеличению концентрации ЗВ в водоеме. Следовательно, для достижения устойчивого развития системы «агенты – водоем» требуется введение регулирующего органа для задания платы за сброс ЗВ. Эти выводы соответствуют выводам другой работы [Решитько, Угольницкий, Усов, 2020], где расчеты проводились методом стрельбы. Кроме того, на временном промежутке в 1 месяц не наблюдается превышения предельно допустимой концентрации ЗВ.

Таким образом примеры 1 и 2 иллюстрируют применение метода для задач с известным решением, при этом решение нейросетевым методом соответствует точному решению в случае примера 1 и решению методом стрельбы в случае примера 2. Кроме того, ошибка метода, вычисленная относительно точного решения, и ошибка, вычисленная предложенным методом, имеют один порядок, что демонстрирует состоятельность предложенного метода оценки ошибки. В примерах 2, 3 и 4 проведены расчеты системы контроля качества поверхностных вод для одного и двух агентов различного вида.

Таблица 1. Результаты расчетов, проведенных нейросетевым методом, для задачи контроля качества поверхностных вод с двумя агентами (пример 4)

J_1 , млн руб.	J_2 , млн руб.	β_1	β_2	η_1	η_2	ν_1	ν_2	$\phi_1(T)$, у. е.	$\phi_2(T)$, у. е.	$\rho(T)$, мг/л	$\max \bar{\Delta}$
0,3160	0,1053	0,5	0,5	0,5	0,5	1	1	0,847	0,9902	0,6154	0,0063
0,3281	0,1072	0,5	0,5	0,5	0,5	0,9	0,9	0,8569	0,9915	0,6639	0,005
0,3560	0,1114	0,5	0,5	0,5	0,5	0,7	0,7	0,8797	0,9953	0,6756	0,0054
0,3882	0,1143	0,5	0,5	0,5	0,5	0,5	0,5	0,9063	0,9975	0,725	0,008
0,4263	0,1235	0,5	0,5	0,5	0,5	0,3	0,3	0,938	1,005	0,7865	0,006
0,4730	0,1246	0,5	0,5	0,5	0,5	0,1	0,1	0,9773	1,006	0,8522	0,0046
0,0366	0,0113	0,5	0,5	0,95	0,95	0,9	0,9	1,1416	1,0739	0,7080	0,0128
0,1554	0,0441	0,5	0,5	0,8	0,8	0,7	0,7	1,0867	1,0448	0,7369	0,0194
0,3210	0,0933	0,5	0,5	0,6	0,6	0,5	0,5	0,9793	1,0174	0,7518	0,0131
0,5525	0,1663	0,5	0,5	0,3	0,3	0,3	0,3	0,7898	0,9647	0,7254	0,0118
0,7119	0,2137	0,5	0,5	0,1	0,1	0,1	0,1	0,6678	0,9283	0,7132	0,0115
0,0419	0,0146	0,0	0,0	0,95	0,95	0,9	0,9	1,7926	1,7779	0,8480	0,0161
0,1681	0,0546	0,25	0,25	0,8	0,8	0,7	0,7	1,3705	1,3597	0,8331	0,0066
0,3154	0,0919	0,6	0,6	0,6	0,6	0,5	0,5	0,9194	0,9223	0,7556	0,0141
0,5070	0,1590	0,8	0,8	0,3	0,3	0,3	0,3	0,6042	0,7123	0,6919	0,0146
0,6223	0,1802	1,0	1,0	0,1	0,1	0,1	0,1	0,4486	0,5473	0,5924	0,0085

Заключение

В работе представлен подход к решению задач оптимального управления и дифференциальных игр с использованием нейронных сетей. Представлено описание нейросетевого подхода, включая методы оценки ошибки и методы обучения нейронной сети. Метод основан на существующих методах решения систем дифференциальных уравнений и задач оптимального управления. При этом в работе предлагается способ количественной оценки ошибки решения, а также одна нейросеть используется для приближения всех неизвестных функций. Развитие нейросетевого метода решения задач оптимального управления позволяет расширить круг решаемых задач, поскольку метод использует теорию оптимального управления в объединении с численными методами нейронных сетей.

Выше приводятся примеры использования предложенного метода. В примерах 1 и 2 решение, найденное предлагаемым методом, сравнивается с точным решением и решением, полученным методом стрельбы. Вычисляются абсолютная погрешность Δ и ошибка с использованием дефекта решения $\bar{\Delta}$, которые имеют один порядок, что говорит о возможности применения $\bar{\Delta}$ для оценки ошибки нейросетевого подхода. Для примера 3 неизвестно точное решение, а решение методом стрельбы получить не удалось, однако решение получено с помощью нейросетевого метода с приемлемым значением ошибки $\bar{\Delta}$. В примере 4 нейросетевым методом решается дифференциальная игра двух агентов, при этом оптимальные управления находятся не аналитически, а при обучении нейронной сети. Проведен ряд численных экспериментов, результаты которых согласуются с полученными ранее. Таким образом, можно сделать вывод о применимости метода к решению различных постановок задачи контроля качества поверхностных вод и выделить следующие преимущества:

- 1) в отличие от различных модификаций метода стрельбы граничные условия удовлетворяются точно, причем непосредственно архитектурой нейронной сети до ее обучения;
- 2) получаемое решение является непрерывным и дифференцируемым;

- 3) метод позволяет проводить расчеты на больших временных интервалах по сравнению с методом стрельбы;
- 4) возможно ускорение расчетов на видеокартах;
- 5) возможно использовать одну нейронную сеть для аппроксимации всего решения, что ускоряет и упрощает расчеты;
- 6) возможно количественно оценить ошибку полученного решения, используя дефект решения.

Список литературы (References)

- Андреева Е. А., Цирулева В. М.* Математическое моделирование оптимального управления динамическими системами с помощью искусственных нейронных сетей // Моделирование, оптимизация и информационные технологии. — 2018. — Т. 6, № 2. — С. 119–131.
Andreeva E. A., Tsiruleva V. M. Matematicheskoe modelirovanie optimal'nogo upravleniya dinamicheskimi sistemami s pomoshch'yu iskusstvennykh neuronnykh setei [Mathematical modeling of optimal control of dynamic systems with neural networks] // Modelirovanie, optimizatsiya i informatsionnye tekhnologii. — 2018. — Vol. 6, No. 2. — P. 119–131 (in Russian).
- Горбаченко В. И., Артюхина Е. В.* Два подхода к обучению радиально-базисных нейронных сетей при решении дифференциальных уравнений в частных производных // Известия вузов, Поволжский регион. Технические науки. — 2007. — № 2. — С. 56–66.
Gorbachenko V. I., Artyukhina E. V. Dva podkhoda k obucheniyu radial'no-bazisnykh neuronnykh setei pri reshenii differentsial'nykh uravnenii v chastnykh proizvodnykh [Two approaches for solving PDE with radial basis neural networks] // Izvestiya vuzov, Povolzhskii region. Tekhnicheskie nauki. — 2007. — No. 2. — P. 56–66 (in Russian).
- Коваленко А. Н., Черноморец А. А., Петина М. А.* О применении нейронных сетей для решения дифференциальных уравнений в частных производных // Научные ведомости Белгородского государственного университета. Сер. Экономика. Информатика. — 2017. — № 9. — С. 103–110.
Kovalenko A. N., Chernomorets A. A., Petina M. A. O primeneniі neuronnykh setei dlya resheniya differentsial'nykh uravnenii v chastnykh proizvodnykh [Using neural networks to solve PDEs] // Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. Ser. Ekonomika. Informatika. — 2017. — No. 9. — P. 103–110 (in Russian).
- Малкин В. А.* Решение двухточечной краевой задачи методом неградиентного случайного поиска // Системный анализ и прикладная информатика. — 2016. — С. 29–34.
Malkin V. A. Reshenie dvukhtochechnoi kraevoi zadachi metodom negradientnogo sluchainogo poiska [Solving two-point BVP using non-gradient random search] // Sistemnyi analiz i prikladnaya informatika. — 2016. — P. 29–34 (in Russian).
- Решитько М. А., Угольницкий Г. А., Усов А. Б.* Численный метод нахождения равновесий Нэша и Штакельберга в моделях контроля качества поверхностных вод // Компьютерные исследования и моделирование. — 2020. — Т. 12, № 3. — С. 653–667.
Reshitko M. A., Ugolnitsky G. A., Usov A. B. Numerical method for finding Nash and Shtakelberg equilibria in river water quality control models // Computer Research and Modeling. — 2020. — Vol. 12, No. 3. — P. 653–667 (in Russian).
- Auzinger W.* Error estimation via defect computation and reconstruction: some practical techniques // JNAIAM. — 2011. — Vol. 6, No. 1-2. — P. 15–27.
- Adam: a method for stochastic optimization. — URL: arxiv.org/abs/1412.6980 (accessed: 10.01.2021).
- Cybenko G.* Approximation by superpositions of a sigmoidal function // Mathematics of control, signals and systems. — 1989. — No. 2. — P. 303–314.
- Cyclical learning rates for training neural networks. — URL: arxiv.org/abs/1506.01186 (accessed: 10.01.2021).
- Google/jax. — URL: github.com/google/jax (accessed: 10.01.2021).
- Gym. — URL: gym.openai.com (accessed: 10.01.2021).
- HIPS/autograd. — URL: github.com/HIPS/autograd (accessed: 10.01.2021).
- Holsapple R., Venkataraman R.* New, fast numerical method for solving two-point boundary value problems // Journal of guidance control and dynamics. — 2004. — No. 27. — P. 301–304.

- Hua L.* Solving a nonlinear two-point boundary value problem // IEEE International Conference on Systems, Man, and Cybernetics. — 1992. — P. 1038–1042.
- Kamien M. I., Schwartz N. L.* Dynamic Optimization. The calculus of variations and optimal control in economics and management. — Elsevier, 1991.
- Lookahead optimizer: k steps forward, 1 step back. — URL: arxiv.org/abs/1907.08610 (accessed: 10.01.2021).
- Nazemi A., Karami R.* A neural network approach for solving optimal control problems with inequality constraints and some applications // Neural processing letters. — 2017. — No. 45. — P. 995–1023.
- NN-for-Optimal-Control. — URL: github.com/ReshitkoM/NN-for-Optimal-Control (accessed: 10.01.2021).
- PyTorch. — URL: pytorch.org (accessed: 10.01.2021).
- Rao A. V.* A survey of numerical methods for optimal control // Advances in the Astronautical sciences. — 2009. — No. 135. — P. 497–528.
- SciPy. — URL: www.scipy.org (accessed: 10.01.2021).
- Stetter H. J.* The defect correction principle and discretization methods // Numerische Mathematik. — 1978. — No. 29. — P. 425–443.
- Sung N. H.* A nonlinear shooting method for two-point boundary value problem // Computers and mathematics with applications. — 2001. — No. 42. — P. 1411–1420.
- Sutskever I., Martens J., Dahl J., Hinton G.* On the importance of initialization and momentum in deep learning // Proceedings of the 30th international conference on machine learning. — 2013. — P. 1139–1147.
- Yadav N., Yadav A., Kumar M.* An introduction to neural network methods for differential equations. — Springer Briefs in Applied Sciences and Technology, 2015.
- Yize C., Yuanyuan S., Baisen Z.* Optimal Control Via Neural Networks: A Convex Approach // ICLR Conference. — 2019.
- Zadunaisky P. E.* On the estimation of errors propagated in the numerical integration of ordinary differential equations // Numerische Mathematik. — 1976. — No. 27. — P. 21–39.
- Zhou L., Hongming P., Feicheng W., Zhiqiang H., Liwei W.* The expressive power of neural networks: a view from the width // NIPS. — 2017.