Горбаченко В.И. Решение обратных краевых задач на сетях радиальных базисных функций. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XVIII Междунар. научно-техн. конф. – Пенза: ПДЗ, 2018. – С. 36-42.

УДК 004.032.26

РЕШЕНИЕ ОБРАТНЫХ КРАЕВЫХ ЗАДАЧ НА СЕТЯХ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ

В.И. Горбаченко

SOLVING INVERSE BOUNDARY VALUE PROBLEM ON RADIAL BASIS FUNCTIONS NETWORKS

V.I. Gorbachenko

Аннотация. Предлагается унифицированный алгоритм решения обратных краевых задач на сетях радиальных базисных функций. Решение задач основано на обучении сетей. Рассмотрены особенности решения различных видов обратных задач.

Ключевые слова: обратные краевые задачи, нейронная сеть, сеть радиальных базисных функций, обучение нейронных сетей.

Abstract. A unified algorithm for solving inverse boundary value prob-lems on radial basis functions networks is proposed. Problem solving is based on network training. The features of solving various types of inverse problems are considered.

Keywords: inverse boundary value problems, neural network, radial basis functions network, learning neural networks, Levenberg-Marquardt method.

Среди краевых задач математической физики выделяют прямые и обратные задачи. Прямые задачи характеризуются необходимостью найти решение, которое удовлетворяет заданному уравнению с частными производными и некоторым начальным и граничным условиям. В обратных задачах определяющее уравнение и/или начальные, и/или граничные условия заданы не полностью, но при этом задана некоторая дополнительная информация. При решении таких задач необходимо найти не только решение уравнения, но и восстановить компоненты математической модели [1]. В зависимости от того, какие компоненты модели не заданы, выделяют три вида обратных задач [1]: граничные обратные задачи (неизвестны граничные условия), эволюционные (ретроспективные) обратные задачи (неизвестны начальные условия) и коэффициентные обратные задачи (неизвестны некоторые коэффициенты уравнения).

Известные подходы к решению обратных задач [1] отличаются от подходов к решению прямых задач и различаются для разных классов обратных задач. Предлагается унифицированный подход к решению обратных и прямых задач, основанный на применении нейронных сетей и машинного обучения. Суть предлагаемого подхода [2–4] состоит в следующем. Рассмотрим дифференциальную задачу, заданную в операторной форме

$$Lu(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{1}$$

с граничными условиями

$$Bu(\mathbf{x}) = p(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega \tag{2}$$

и дополнительными условиями

$$Du(\mathbf{z}) = \psi(\mathbf{z}), \quad \mathbf{z} \in Z \subset \Omega \cup \partial\Omega, \tag{3}$$

где u есть искомое решение, L является дифференциальным оператором, B, D – некоторые допустимые операторы, $f(\mathbf{x})$, $p(\mathbf{x})$ – некоторые функции, Ω является областью, в которой ищется решение, $\partial\Omega$ – граница области, $\psi(\mathbf{z})$ – некоторая функция измерений.

В прикладных исследованиях типичной является ситуация, когда дополнительные условия являются результатами измерений и заданы с погрешностью [1], поэтому будем использовать условие

$$Du(\mathbf{z}) \approx \psi^{\delta}(\mathbf{z}),\ \mathbf{z} \in Z., \tag{4}$$

где δ определяет точность задания функции ψ^{δ}.

Погрешность некоторого приближенного решения $\hat{u}(\mathbf{x})$ определяется функционалом ошибки. Зададим функционал ошибки J в дискретной форме

$$J = J_1 + \delta_b J_b + \delta_d J_d, \tag{5}$$

где $J_1 = \sum_{i=1}^{M_1}\left(L\hat{u}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2$, $J_b = \sum_{i=1}^{M_2}\left(B\hat{u}(\mathbf{t}_i) - p(\mathbf{t}_i)\right)^2$, $J_d = \sum_{i=1}^{M_3}\left(D\hat{u}(\mathbf{z}_i) - \psi(\mathbf{z}_i)\right)^2$ являются условиями, удовлетворяющими дифференциальному уравнению (1), граничным условиям (2) и дополнительным данным (3), соответственно; δ_b, δ_d – положительные штрафные коэффициенты; $\hat{\boldsymbol{u}}$ – приближенное решение задачи (1)–(3); $\{\mathbf{x}_i\}_{i=1}^{M_1} \in \Omega$, $\{\mathbf{t}_i\}_{i=1}^{M_2} \in \partial\Omega$, $\{\mathbf{z}_i\}_{i=1}^{M_3} \in Z$ – некоторые множества пробных точек.

Решение задачи сводится к минимизации функционала (5). Как правило, для решения таких экстремальных задач применяются градиентные методы минимизации. Известные методы [1] используют функциональную или параметрическую оптимизацию. В случае функциональной оптимизации используется градиент соответствующего интегрального функционала, который находится как решение сопряженной задачи. В случае параметрической оптимизации искомый параметр обратной задачи представляется в виде разложения по некоторому функциональному базису, коэффициенты этого разложения вычисляются в процессе решения. Оба подхода являются сложными и громоздкими и не позволяют найти глобальный минимум функционала (5).

Предлагаетсянейросетевой подход к решению краевых задач [3–8], общий для прямых и обратных задач. Необходимые компоненты ($u(x)$ и другие) задачи (1)–(3) находятся как выходы сети радиальных базисных функций (РБФ-сети)

$$\hat{u} = \sum_{j=1}^{N} c_j \varphi_j\left(\mathbf{x}, \mathbf{a}_j\right), \tag{6}$$

где c_j – веса нейронной сети, φ_j – радиальные базисные функции (РБ-функции), $\mathbf{a}_j$ – вектор параметров базисной функции, N – количество РБ-функций.

РБФ-сеть – сеть, состоящая из двух слоев [5]. Первый слой с помощью РБ-функций осуществляет нелинейное преобразование входного вектора $x = [x_1, x_2, \ldots, x_d]$ в пространство большей размерности. Второй слой производит взвешенное линейное суммирование выходов РБ-функций. Выход сети описывается выражением (6). В качестве РБ-функций используют функции Гаусса, мультиквадрики, обратные мультиквадрики и др. Достоинством применения РБФ-сетей является то, что с их помощью реализуются бессеточные методы [6], исключающие трудоемкий этап построения сетки.

Рассмотрим особенности решения различных видов обратных задач.

В граничных обратных задачах функция p в граничных условиях (2)либо не задана, либо задана на части границы. Для решения такой задачи можно использовать две РБФ-сети: первая аппроксимирует искомое решение задачи, вторая аппроксимирует граничные условия. Обозначим выход первой сети как u_{RBF}, а выход второй сети – как p_{RBF}. Параметры сетей уточняются в процессе минимизации функционала ошибки (5), который принимает вид

$$\begin{aligned} J(\mathbf{w}^u,\mathbf{p}^u,\mathbf{w}^p,\mathbf{p}^p) = \sum_{i=1}^{N}\left[Lu_{RBF}(\mathbf{x}_i) - f(\mathbf{x}_i)\right]^2 + \\ +\delta_b\sum_{i=N+1}^{N+K}\left[Bu_{RBF}(\mathbf{x}_i) - p_{RBF}(\mathbf{x}_i)\right]^2 + \delta_d\sum_{s=1}^{S}\left[Du_{RBF}(\mathbf{z}_s) - \psi^{\delta}(\mathbf{z}_s)\right]^2, \end{aligned} \tag{7}$$

где $\mathbf{w}^u$ и $\mathbf{p}^u$ – веса и параметры РБ-функций сети, аппроксимирующей решение u_{RBF}; $\mathbf{w}^p$ и $\mathbf{p}^p$ – веса и параметры РБ-функций сети, аппроксимирующей граничные условия p_{RBF}; N – количество пробных точек из области решения Ω; K – количество пробных точек из граничной области $\partial\Omega$; S – количество пробных точек из области Z задания дополнительных условий; δ_b, δ_d – положительные штрафные коэффициенты.

Минимизация функционала (7) (обучение РБФ-сети) может быть произведена различными итерационными градиентными методами. Так как дополнительные условия задаются с погрешностью (4), то существует множество решений (7), которые с точностью δ удовлетворяют исходной задаче. Чтобы сузить класс допустимых решений, необходимо использовать итерационный метод регуляризации, в котором в роли регуляризатора выступает число итераций: итерационный процесс минимизации (7) продолжается до тех пор, пока выполняется условие Морозова [1]: $J(\mathbf{w}^u,\mathbf{p}^u,\mathbf{w}^p,\mathbf{p}^p) > \sum_{s=1}^{S}\sigma_s^2$,где $\sigma_s = \left|\psi(\mathbf{z}_s) - \psi^{\delta}(\mathbf{z}_s)\right|$ – погрешность в S -й пробной точке.

В операторном виде эволюционную обратную задачу можно записать следующим образом

$$Au(\mathbf{x},t) = Lu(\mathbf{x},t) + f(\mathbf{x}),\ \mathbf{x}\in\Omega\subset\mathbb{R}^d,\ t\in\left[0,T\right], \tag{8}$$

$$Bu(\mathbf{x},t) = p\left(\mathbf{x},t\right),\ \mathbf{x}\in\partial\Omega,\ t\in\left(0,T\right],$$

$$Gu(\mathbf{x},T) \approx u_T^{\delta}\left(\mathbf{x},T\right),\ \mathbf{x}\in\Omega, \tag{9}$$

где A – дифференциальный оператор по времени (обычно либо $\frac{\partial}{\partial t}$, либо $\frac{\partial^2}{\partial t^2}$, но может быть оператором более высокого порядка); L – дифференциальный оператор по $\mathbf{x}$; B – дифференциальный оператор, задающий граничные условия; G –

дифференциальный оператор, определяющий решение в конечный момент времени $t = T$ или в некоторые промежуточные моменты из интервала $\left(0, T\right]$; f, p – известные функции; T – некоторая заданная временная константа; u_T^δ – известная функция, заданная с погрешностью δ. Задача состоит в определении начальных условий для $u(\mathbf{x},t)$ – $u(\mathbf{x},0)$, $\mathbf{x}\in\Omega$ – если A оператор первого порядка, добавляется $\frac{\partial u}{\partial t}(\mathbf{x},0)$ и другие данные для операторов более общего вида.

Для решения задачи (8)–(9) с помощью нейронной сети можно рассмотреть условие (9) как начальное и решить задачу с помощью нейронной сети как нестационарную задачу с обращением времени. Время можно формально рассматривать как одно из измерений, но это сильно усложняет решение задач оптимизации, если решается задача (8)–(9). Возможно использование метода прямых – явной разностной аппроксимации оператора A и решения задачи на каждом временном слое.

При решении коэффициентных обратных задач предлагается, воспользовавшись методом параметрической оптимизации [1]. Рассмотрим процесс решения обратных краевых задач на примере задачи:

$$L(k(\mathbf{x},u))u(\mathbf{x}) = f\left(\mathbf{x}\right), \mathbf{x}\in\Omega,$$
$$Bu(\mathbf{x}) = p\left(\mathbf{x}\right),\ \mathbf{x}\in\partial\Omega,$$

где u – искомое решение; $k(\mathbf{x},u)$ – неизвестный коэффициент; $L(k(\mathbf{x},u))$ – дифференциальный оператор, зависящий от коэффициента k; оператор B – оператор граничных условий; Ω – область решения; $\partial\Omega$ – граница области; f и p – известные функции; $Du(\mathbf{z}) = \psi\left(\mathbf{z}\right)$, $\mathbf{z}\in Z$ – оператор, задающий дополнительное условие; $Z\subset\Omega\cup\partial\Omega$; φ – известная функция.

Дополнительные условия заданы с погрешностью (4). Неизвестный коэффициент k представляется в параметрическом виде. Необходимо определить параметры этого представления. В качестве такого представления будем использовать РБФ-сеть, выход которой описывается выражением

$$k_{RBF}(\mathbf{x},u) = \sum_{m=1}^{M_k} w_m^k \xi_m(\mathbf{x},u;\mathbf{p}_m^k).$$

Неизвестные параметры $\mathbf{p}_m^k$ и веса w_m^k РБ-функций выбираются из условия минимизации невязки между левой и правой частями дополнительного условия, которое можно записать в виде

$$J(\mathbf{w}^k,\mathbf{p}^k) = \sum_{s=1}^{S}\left(Du'(\mathbf{z}_s) - \psi^\delta\left(\mathbf{z}_s\right)\right)^2 \to \min, \tag{10}$$

где S – количество пробных точек из области Z, u' – решение прямой задачи при $k = k_{RBF}(\mathbf{x},u;\mathbf{w}^k,\mathbf{p}^k)$, $\mathbf{z}_s\in Z$, $\mathbf{w}^k = (w_1^k, w_2^k, ..., w_{M_k}^k)$, $\mathbf{p}^k = (\mathbf{p}_1^k, \mathbf{p}_2^k, ..., \mathbf{p}_{M_k}^k)$.

Чтобы сузить класс допустимых решений, используем итерационный метод регуляризации: итерационный процесс минимизации (10) продолжается до тех пор, пока выполняется условие Морозова [1]

$$J(\mathbf{w}^k,\mathbf{p}^k) > \sum_{s=1}^{S}\sigma_s^2,$$

где $\sigma_s = \left|\psi\left(\mathbf{z}_s\right) - \psi^{\delta}\left(\mathbf{z}_s\right)\right|$ – погрешность в S -й пробной точке.

Аппроксимацию функции u' найдем, решив с помощью РБФ-сети, описываемой выражением $u_{RBF}(\mathbf{x}) = \sum_{m=1}^{M_u} w_m^u \varphi_m(\mathbf{x};\mathbf{p}_m^u)$ прямую задачу, в которой $k = k_{RBF}$. Неизвестные параметры $\mathbf{w}^u = (w_1^u, w_2^u, ..., w_{M_u}^u)$, $\mathbf{p}^u = (\mathbf{p}_1^u, \mathbf{p}_2^u, ..., \mathbf{p}_{M_u}^u)$ можно найти, минимизировав функционал

$$I(\mathbf{w}^u, \mathbf{p}^u) = \sum_{i=1}^{N} \left(L(k_{RBF}(\mathbf{x}_i, u_{RBF}(\mathbf{x}_i; \mathbf{w}^u, \mathbf{p}^u); \mathbf{w}^k, \mathbf{p}^k)) u_{RBF}(\mathbf{x}_i; \mathbf{w}^u, \mathbf{p}^u) - f\left(\mathbf{x}_i\right) \right)^2 +$$

$$+\delta_b \sum_{i=N+1}^{N+K} \left(B u_{RBF}(\mathbf{x}_i; \mathbf{w}^u, \mathbf{p}^u) - p\left(\mathbf{x}_i\right) \right)^2,$$

где δ_b – подбираемый штрафной множитель за нарушение граничных условий.

Решение задачи находится в процессе минимизации функционала $R(\mathbf{w}^u, \mathbf{p}^u, \mathbf{w}^k, \mathbf{p}^k) = I(\mathbf{w}^u, \mathbf{p}^u) + \delta_d J(\mathbf{w}^k, \mathbf{p}^k)$, где δ_d – подбираемый штрафной множитель за нарушение дополнительных условий, т.е. настройки параметров сетей u_{RBF} и k_{RBF}, минимизация продолжается до тех пор, пока выполняется условие Морозова $J(\mathbf{w}^k, \mathbf{p}^k) > \sum_{s=1}^{S} \sigma_s^2$, где $\sigma_s = \left|\psi\left(\mathbf{z}_s\right) - \psi^{\delta}\left(\mathbf{z}_s\right)\right|$ – погрешность в S -й пробной точке.

Таким образом, возможен унифицированный подход к решению прямых и обратных краевых задач математической физики, описывающих сложные реальные объекты.

Библиографический список

1. Самарский А.А., Вабищевич П.Н. Численные методы решения обратных задач математической физики. М.: Изд-во ЛКИ, 2009. 480 с.

2. Жуков М.В. Решение коэффициентных обратных задач математической физики с помощью сетей радиальных базисных функций // Нейрокомпьютеры: разработка и применение. 2014. №2. С. 32–39.

3. Горбаченко В.И., Жуков М.В. Решение краевых задач математической физики с помощью сетей радиальных базисных функций // Журнал вычислительной математики и математической физики. 2017, том 57. № 1. С. 133–143.

4. Neural Network Technique in Some Inverse Problems of Mathematical Physics / V. I. Gorhochenko, T. V. Lazovskaya, D. A. Tarkhov, A. N. Vasiljev, M. V. Zhukov // Advances in Neural Networks - ISNN 2016: 13th International Symposium on Neural Networks, ISNN 2016, St. Petersburg, Russia, July 6-8, 2016, Proceedings (Lecture Notes in Computer Science). Springer, 2016. P. 310–316.

5. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook. – Springer, 2018. 497 p.

6. Liu G. R. Mesh free methods: moving beyond the finite element method. CRC Press, 2003. 712 p.

Горбаченко Владимир Иванович
Пензенский государственный
университет, г. Пенза, Россия
E-mail: gorvi@mail.ru

Gorbachenko V.I.
Penza State University,
Penza, Russia