

РАЗДЕЛ 2

МОДЕЛИ, СИСТЕМЫ, СЕТИ В ТЕХНИКЕ

УДК 004.032.26

МОДЕЛИРОВАНИЕ ОБЪЕКТОВ С РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ НА НЕЙРОННЫХ СЕТЯХ

В. И. Горбаченко, М. М. Алкезуини

MODELING OBJECTS WITH DISTRIBUTED PARAMETERS ON NEURAL NETWORKS

V. I. Gorbachenko, M. M. Alkezuini

Аннотация. *Предмет и цель работы.* Применение специального вида нейронных сетей – сетей радиальных базисных функций является перспективным направлением решения краевых задач, являющихся моделями объектов с распределенными параметрами. Применение таких нейронных сетей сдерживается отсутствием быстрых и простых алгоритмов обучения. Целью работы является совершенствование алгоритмов обучения сетей радиальных базисных функций при решении краевых задач, позволяющих сократить время решения задачи. *Методы.* Анализ градиентных алгоритмов обучения сетей радиальных базисных функций показал перспективность разработки новых алгоритмов обучения, основанных на методах оптимизации Нестерова и Левенберга – Марквардта. *Результаты.* Разработаны алгоритмы обучения сети на основе методов Нестерова и Левенберга – Марквардта, отличающиеся учетом специфики архитектуры сети и аналитическим вычислением параметров. Алгоритм на основе метода Левенберга – Марквардта достиг на модельной задаче малой погрешности за число итераций, равное числу итераций алгоритма на основе метода доверительных областей, но проще этого алгоритма, так как не требует решения на каждой итерации задачи условной оптимизации. *Выводы.* Разработан быстрый алгоритм обучения сетей радиальных базисных функций, предназначенных для моделирования объектов с распределенными параметрами.

Ключевые слова: краевые задачи, уравнения с частными производными, сети радиальных базисных функций, обучение нейронных сетей, градиентные алгоритмы, метод Нестерова, метод Левенберга – Марквардта.

Abstract. *Subject and goals.* The use of a special type of neural networks – radial basic function networks is a promising direction for solving boundary value problems, which are models of objects with distributed parameters. The use of such neural networks is constrained by the lack of fast and simple learning algorithms. The aim of the work is to improve the learning algorithms for radial basis function networks in solving boundary value problems, which reduce the time to solve the problem. *Methods.* An analysis of gradient learning algorithms for radial basis function networks showed the promise of developing new learning algorithms based on the optimization methods of Nesterov and Levenberg – Marquardt. *Results.* Algorithms for learning the network based on the methods of Nesterov

and Levenberg – Marquardt are developed, which differ by considering the specifics of the network architecture and the analytical calculation of parameters. An algorithm based on the Levenberg – Marquardt method achieved a small error in the model problem for the number of iterations equal to the number of iterations of the algorithm based on trust region method, but simpler than this algorithm, since it does not require a solution to the conditional optimization problem at each iteration. *Conclusions.* A fast algorithm for learning radial basis function networks designed for modeling objects with distributed parameters has been developed.

Keywords: boundary value problems, partial differential equations, radial basis function networks, neural networks learning, gradient algorithms, Nesterov's method, Levenberg – Marquardt method.

Введение

Большое количество объектов реального мира являются объектами с распределенными параметрами [1]. Математическими моделями таких объектов являются краевые задачи для дифференциальных уравнений в частных производных (ДУЧП). Аналитическое решение краевых задач для ДУЧП удается получить только для ограниченного круга задач. Поэтому применяются численные методы. Наиболее популярны методы конечных разностей и конечных элементов [2], требующие построения сетки. Генерация сеток для двух- и трехмерных областей сложной конфигурации является сложной и трудоемкой задачей. Трудоемкость формирования сетки для реальных задач зачастую превосходит трудоемкость решения системы разностных уравнений [3]. Моделирование объектов с распределенными параметрами методами конечных разностей и конечных элементов сводится к решению плохо обусловленных разреженных систем алгебраических уравнений очень большой размерности, что требует больших затрат на их решение. Восстановление решения по его дискретной аппроксимации является отдельной достаточно трудоемкой задачей.

Альтернативой методам конечных разностей и конечных элементов являются бессеточные методы [4] – «численные методы, которые не требуют сетки точек, соединенных между собой для аппроксимации уравнений» [5]. Перспективной является реализация бессеточных методов на нейронных сетях. Решение краевых задач на нейронных сетях представляет собой нейросетевую аппроксимацию неизвестного решения задачи. Известно, что нейронные сети являются хорошими аппроксиматорами функций. В теории нейронных сетей известна универсальная теорема аппроксимации, или теорема Цыбенко (Cybenko G.) [5]: нейронная сеть с одним скрытым слоем, содержащим нейроны с сигмоидальной функцией активации, и линейным выходным слоем может при достаточном количестве нейронов в скрытом слое аппроксимировать с любой степенью точности любую непрерывную функцию. Решение краевых задач для ДУЧП возможно на многослойных сетях прямого распространения (многослойных персептронах) [7], сетях радиальных базисных функций [7], клеточных нейронных сетях [8], сетях Хопфилда [9]. В настоящее время популярно применение глубоких нейронных сетей прямого распространения для решения краевых задач [10, 11]. Для решения краевых задач, описываемых ДУЧП, особенно перспективно применение сетей радиальных базисных функций (РБФ-сетей) [7], так как РБФ-сети содержат всего два слоя, один из которых является линейным, а формирование ре-

шения носит локальный характер, что упрощает обучение таких сетей. В [12] доказано, что РБФ-сеть является универсальным аппроксиматором. Применение РБФ-сетей является развитием проекционных методов на основе радиальных базисных функций (РБФ) [13, 14]. В отличие от использования РБФ, применение РБФ-сетей позволяет в процессе обучения сети находить не только веса, но и параметры базисных функций.

Решение задачи формируется в процессе обучения РБФ-сети. Поэтому важно сокращение времени обучения сетей. Но в настоящее время для обучения РБФ-сетей при решении краевых задач используются, в основном, простейшие градиентные методы первого порядка на основе градиентного спуска [7]. Быстрые методы второго порядка практически не используются при решении краевых задач на РБФ-сетях. Исключение представляет предложенный в [15] и исследованный в [16, 17] метод доверительных областей. Но метод весьма сложен, так как требует на каждой итерации решения задачи минимизации решать задачу условной минимизации.

Целью данной работы является совершенствование алгоритмов обучения сетей радиальных базисных функций при решении краевых задач, позволяющих сократить время решения задачи.

Материалы и методы

Рассмотрим краевую задачу в операторной форме

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

где L – дифференциальный оператор; u – решение задачи; Ω – область решения; B – оператор граничных условий; $\partial\Omega$ – граница области; f и p – известные функции.

РБФ-сеть включает два слоя [18]. Первый слой состоит из РБФ, производящих нелинейное преобразование входного вектора $\mathbf{x} = [x_1, x_2, \dots, x_d]$ – координат точки, в которой вычисляется приближение к решению (d – размерность пространства). РБФ – это функции расстояния точки пространства от параметра функции, называемого центром функции:

$$\varphi(\|\mathbf{x} - \mathbf{c}\|, \mathbf{p}),$$

где $\mathbf{x}$ – точка пространства; $\mathbf{p}$ – вектор параметров функции; $\mathbf{c}$ – центр радиальной базисной функции; $\|\mathbf{x} - \mathbf{c}\|$ – евклидова норма (расстояние) между точкой и центром. Применяются различные РБФ. В данной работе используется функция Гаусса (гауссиан):

$$\varphi(\|\mathbf{x} - \mathbf{c}\|, a) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2a^2}\right),$$

где $\mathbf{c}$ – положение центра функции; a – параметр формы, часто называемый шириной.

Второй слой РБФ-сети представляет собой линейный взвешенный сумматор

$$u(\mathbf{x}) = \sum_{m=1}^M w_m \varphi_m(\mathbf{x}; \mathbf{p}_m), \quad (2)$$

где M – количество РБФ; w_m – вес РБФ φ_m ; $\mathbf{p}_m$ – вектор параметров.

Процесс решения краевых задач с помощью РБФ-сетей состоит из трех этапов.

1. Внутри области решения Ω выбирается N внутренних пробных точек и на границе $\partial\Omega$ K граничных пробных точек (точек, в которых контролируется невязка решения):

$$\{\mathbf{x}_i \mid i=1,2,\dots,N \subset \Omega\} \cup \{\mathbf{x}_i \mid i=N+1,N,\dots,N+K \subset \partial\Omega\}. \quad (3)$$

Когда нет априорной информации о решении, целесообразно использовать случайное равномерное размещение пробных точек по области и на границе решения. Для задач аппроксимации известно соотношение между количеством РБФ M и количеством пробных точек $N + K$: $M \propto (N + K)^{\frac{1}{3}}$, где $\propto$ означает пропорциональность [19]. Однако при аппроксимации решений краевых задач с помощью РБФ-сетей данная зависимость дает избыточное количество пробных точек, поэтому приходится подбирать количество пробных точек.

2. Определяется структура РБФ-сети: количество РБФ, вид РБФ, задаются начальные значения вектора весов и векторов параметров РБФ. Однозначные рекомендации по выбору вида РБФ отсутствуют. При решении задач, описываемых дифференциальными уравнениями второго порядка, необходимо вычислять вторые производные от выхода сети. Поэтому целесообразно использовать функцию Гаусса, область определения которой сопоставима с областями определения ее производных, чего нельзя сказать о мультиквадрах, для которых наблюдается большой разброс значений. При выборе начальных значений необходимо задать параметры РБФ и вектор весов. Центры РБФ можно расположить в узлах равномерной сетки или случайным образом. Можно увеличить плотность РБФ в областях, где ожидается изменение характера решения.

3. Выполняется обучение сети, т.е. подбор таких значений весов и параметров RBF , чтобы некоторый функционал ошибки в пробных точках принимал минимальное значение. Поскольку решение в пробных точках неизвестно, то возможна только минимизация невязок приближенного решения на множестве пробных точек. Для построения функционала ошибки используется метод наименьших квадратов. Функционал ошибки для поиска весов $\mathbf{w}$ и параметров $\mathbf{p}$ РБФ, минимизирующих невязки в пробных точках, имеет вид

$$J(\mathbf{w}, \mathbf{p}) = \sum_{i=1}^N [Lu_{RBF}(\mathbf{x}_i; \mathbf{w}, \mathbf{p}) - f(\mathbf{x}_i)]^2 + \lambda \sum_{i=N+1}^{N+K} [Bu_{RBF}(\mathbf{x}_i; \mathbf{w}, \mathbf{p}) - p(\mathbf{x}_i)]^2, \quad (4)$$

где $\mathbf{x}_i$ – пробные точки (3); λ – подбираемый штрафной множитель; u_{RBF} – приближенное решение, получаемое на $RBFN$.

Штрафной множитель λ обеспечивает выполнение граничных условий, так как в бессеточных методах условия на границе не фиксируются.

Абсолютное большинство алгоритмов обучения РБФ-сетей основано на градиентных методах оптимизации [20]. Градиентные методы являются методами локальной оптимизации, что в общем случае не гарантирует достижения глобального минимума функционала ошибки. В то же время поиск глобального минимума функционала ошибки не обязателен, достаточно найти локальный минимум с некоторой заданной точностью. Среди градиентных методов выделяют три класса: методы нулевого порядка, использующие при оптимизации только значения оптимизируемой функции, но не значения ее производных, методы первого порядка, применяющие первые производные оптимизируемой функции (градиент функции), и методы второго порядка, применяющие вторые производные (матрицу Гессе, гауссиан).

В известных работах, посвященных решению краевых задач на РБФ-сетях [7], используется простейший метод первого порядка – метод градиентного спуска. Рассмотрим реализацию метода скорейшего спуска на примере двумерной задачи (1). Рассмотрим единый вектор параметров РБФ:

$$\boldsymbol{\theta} = [w_1, w_2, \dots, w_{n_{RBF}}, c_{11}, c_{21}, \dots, c_{n_{RBF}1}, c_{12}, c_{22}, \dots, c_{n_{RBF}2}, a_1, a_2, \dots, a_{n_{RBF}}]^T, \quad (5)$$

где w_j – веса РБФ; $j = 1, 2, 3, \dots, n_{RBF}$; n_{RBF} – количество РБФ; c_{j1} и c_{j2} – координаты центров; a_j – ширина.

Коррекция вектора (5) на итерации k в методе градиентного спуска производится по формуле

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}^{(k+1)}, \quad (6)$$

где $\Delta\boldsymbol{\theta}^{(k+1)} = -\eta \nabla J(\boldsymbol{\theta}^{(k)})$ – вектор поправки параметров; η – скорость обучения (подбираемый гиперпараметр алгоритма); $\nabla J(\boldsymbol{\theta}^{(k)})$ – вектор градиента функционала ошибки (4) по компонентам вектора $\boldsymbol{\theta}^{(k)}$ (5) на итерации k .

Вычисления по (6) заканчиваются при малом значении функционала (4). Метод градиентного спуска обладает низкой скоростью сходимости, что не позволяет решать задачи с большой точностью.

Методы второго порядка основаны на квадратичной аппроксимации функционала ошибки. В окрестности вектора параметров $\boldsymbol{\theta}^{(k)}$ сети функционал ошибки (4) аппроксимируется формулой Тейлора:

$$J(\boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}^{(k+1)}) \approx J(\boldsymbol{\theta}^{(k)}) + [\nabla J(\boldsymbol{\theta}^{(k)})]^T \Delta\boldsymbol{\theta}^{(k+1)} + \frac{1}{2} [\Delta\boldsymbol{\theta}^{(k+1)}]^T \mathbf{H}(J(\boldsymbol{\theta}^{(k)})) \Delta\boldsymbol{\theta}^{(k+1)}, \quad (7)$$

где $\nabla J(\boldsymbol{\theta}^{(k)})$ – градиент функционала; $\mathbf{H}(J(\boldsymbol{\theta}^{(k)}))$ – матрица Гессе (матрица вторых производных функционала), вычисленная при $\boldsymbol{\theta}^{(k)}$.

Из условия минимума функционала (7) может быть получен вектор $\Delta\boldsymbol{\theta}^{(k+1)}$ поправки параметров сети, обеспечивающий уменьшение функционала ошибки. Из-за сложности вычисления матрицы Гессе для многослойных персептронов используются различные приближения матрицы Гессе. Например, в методе сопряженных градиентов используются формулы Флетчера –

Ривса (Fletcher R., Reeves C. M.) и Полака – Рибьера (Polak E., Ribiere G.). В квазиньютоновских методах матрица приближения к гессиану рассчитывается на каждом шаге обучения, например, по формуле Бройдена – Флетчера – Гольдфарба – Шанно (Broyden – Fletcher – Goldfarb – Shanno – BFGS). В методе Левенберга – Марквардта матрица Гессе аппроксимируется с помощью произведения матриц Якоби вектора ошибок сети.

Методы второго порядка не получили распространения при обучении РБФ-сетей. Хотя наличие только одного слоя с нелинейными функциями и дифференцируемость большинства РБФ обеспечивают возможность применения методов оптимизации второго порядка при обучении РБФ-сетей. Предложенный в [15] быстрый алгоритм обучения РБФ-сетей основан на эффективном методе оптимизации – методе доверительных областей (МДО) [21]. Метод позволяет одновременно оптимизировать большое количество параметров, обладает высокой скоростью сходимости даже для плохо обусловленных задачах, позволяет преодолевать локальные минимумы. Алгоритм МДО достаточно сложен, так как минимум функционала ошибки отыскивается в ограниченных областях, что требует на каждом шаге процесса оптимизации решения условной задачи оптимизации. Поэтому целесообразно исследовать возможность адаптации для обучения РБФ-сетей современных быстрых методов первого порядка и метода Левенберга – Марквардта. Особый интерес представляет метод Левенберга – Марквардта, который проще в реализации, чем МДО, и, как показано в [22], эквивалентен МДО.

Результаты

При обучении сетей глубокой архитектуры [23] успешно используется быстрый градиентный алгоритм первого порядка – алгоритм ускоренного градиента Нестерова (NAG – Nesterov Accelerated Gradient) [24]. Применение этого алгоритма для обучения РБФ-сетей при решении задач аппроксимации функций впервые предложено в [25].

В [24] доказано, что метод Нестерова можно представить в виде метода с импульсом (моментом) с модифицированным вычислением градиента:

$$\begin{aligned}\theta^{(k+1)} &= \theta^{(k)} + \Delta\theta^{(k+1)}, \\ \Delta\theta^{(k+1)} &= \alpha\Delta\theta^{(k)} - \eta\mathbf{g}_0\left(\theta^{(k)} + \alpha\Delta\theta^{(k)}\right),\end{aligned}\quad (8)$$

где α и η – подбираемые коэффициенты; $\mathbf{g}_0\left(\theta^{(k)} + \alpha\Delta\theta^{(k)}\right) = \nabla J\left(\theta^{(k)} + \alpha\Delta\theta^{(k)}\right)$ – вектор градиента функционала ошибки в промежуточной точке.

На качественном уровне эффект ускорения алгоритма за счет вычисления градиента в промежуточной точке объясняется тем, что, как правило, вектор градиента указывает правильное направление минимума. Градиент, вычисленный чуть дальше в правильном направлении, будет несколько точнее градиента, вычисленного в исходной точке.

В реализации методов первого порядка для обучения РБФ-сетей предлагается компоненты вектора градиента функционала ошибки по весам, центрам и ширине вычислять аналитически. В качестве примера рассмотрим модельную задачу, описываемую уравнением Лапласа с граничными условием Дирихле:

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = f(x_1, x_2), \quad (x_1, x_2) \in \Omega, \quad u = p(x_1, x_2), \quad (x_1, x_2) \in \partial\Omega, \quad (9)$$

где $\partial\Omega$ – граница области; f и p – известные функции (x, y) .

Тогда функционал ошибки (4) запишется в виде

$$I = \left(\sum_{i=1}^N r_i^2 + \lambda \sum_{j=1}^K r_j^2 \right) = \left[\sum_{i=1}^N (\Delta u_i - f_i)^2 + \lambda \sum_{j=1}^K (u_j - p_j)^2 \right], \quad (10)$$

где r_i и r_j – невязки приближенного решения во внутренних и граничных пробных точках; Δu_i – лапласиан в точке i .

Компоненты градиента по весам *RBFN* вычисляются по формуле

$$\frac{\partial I}{\partial w_p} = \sum_{i=1}^N (\Delta v_i - f_i) e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2}{2a_p^2}} \frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2 - 2a_p^2}{a_p^4} + \lambda \sum_{j=1}^K (v_j - p_j) w_p e^{-\frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{2a_p^2}}.$$

Компонент градиента по координате c_{p1} центра *RBFN* имеет вид

$$\begin{aligned} \frac{\partial I}{\partial c_{p1}} = & w_p \sum_{i=1}^N (\Delta v_i - f_i) e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2}{2a_p^2}} (x_{i1} - c_{p1}) \frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2 - 4a_p^2}{a_p^6} + \\ & + \lambda w_p \sum_{j=1}^K (v_j - p_j) e^{-\frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{2a_p^2}} \frac{(x_{j1} - c_{p1})}{a_p^2}. \end{aligned}$$

Аналогичный вид имеет компонент градиента по координате c_{p2} .

Компонент градиента по ширине равен

$$\begin{aligned} \frac{\partial I}{\partial a_p} = & w_p \sum_{i=1}^N (\Delta v_i - f_i) \frac{\|\mathbf{x}_i - \mathbf{c}_p\|^4 - 6a_p \|\mathbf{x}_i - \mathbf{c}_p\|^2 + 4a_p^4}{a_p^7} e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2}{2a_p^2}} + \\ & + \lambda w_p \sum_{j=1}^K (v_j - p_j) \frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{a_p^3} e^{-\frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{2a_p^2}}. \end{aligned}$$

Процесс обучения сети заканчивается при малом значении функционала ошибки (10) или средней квадратической погрешности.

Реализацию метода Левенберга – Марквардта обучения РБФ-сети для решения краевой задачи рассмотрим на примере модельной задачи (9). В методе Левенберга – Марквардта поправка $\Delta \theta^{(k)}$ вектора параметров θ (5) находится из решения системы линейных алгебраических уравнений:

$$(\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}) \Delta \theta^{(k)} = -\mathbf{g}_{k-1}, \quad (11)$$

где матрица $\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}$ – аппроксимация матрицы Гессе; $\mathbf{J}_{k-1}$ – матрица Якоби, вычисленная в $k-1$ итерации; μ_k – параметр регуляризации, изменяющийся на каждом шаге обучения; $\mathbf{E}$ – единичная матрица; $\mathbf{g} = \mathbf{J}^T \mathbf{r}$ – век-

тор градиента функционала (4) по вектору параметров θ , здесь $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_n]^T$ – вектор невязок во внутренних и граничных пробных точках.

Представим матрицу Якоби в блочном виде:

$$\mathbf{J} = [\mathbf{J}_w \mid \mathbf{J}_{c_1} \mid \mathbf{J}_{c_2} \mid \mathbf{J}_a],$$

где

$$\mathbf{J}_w = \begin{bmatrix} \frac{\partial r_1}{\partial w_1} & \dots & \frac{\partial r_1}{\partial w_{n_{RBF}}} \\ \frac{\partial r_2}{\partial w_1} & \dots & \frac{\partial r_2}{\partial w_{n_{RBF}}} \\ \dots & \dots & \dots \\ \frac{\partial r_n}{\partial w_1} & \dots & \frac{\partial r_n}{\partial w_{n_{RBF}}} \end{bmatrix}, \quad \mathbf{J}_{c_1} = \begin{bmatrix} \frac{\partial r_1}{\partial c_{11}} & \dots & \frac{\partial r_1}{\partial c_{n_{RBF}1}} \\ \frac{\partial r_2}{\partial c_{11}} & \dots & \frac{\partial r_2}{\partial c_{n_{RBF}1}} \\ \dots & \dots & \dots \\ \frac{\partial r_n}{\partial c_{11}} & \dots & \frac{\partial r_n}{\partial c_{n_{RBF}1}} \end{bmatrix},$$

$$\mathbf{J}_{c_2} = \begin{bmatrix} \frac{\partial r_1}{\partial c_{12}} & \dots & \frac{\partial r_1}{\partial c_{n_{RBF}2}} \\ \frac{\partial r_2}{\partial c_{12}} & \dots & \frac{\partial r_2}{\partial c_{n_{RBF}2}} \\ \dots & \dots & \dots \\ \frac{\partial r_n}{\partial c_{12}} & \dots & \frac{\partial r_n}{\partial c_{n_{RBF}2}} \end{bmatrix}, \quad \mathbf{J}_a = \begin{bmatrix} \frac{\partial e_1}{\partial a_1} & \dots & \frac{\partial e_1}{\partial a_{n_{RBF}}} \\ \frac{\partial e_2}{\partial a_1} & \dots & \frac{\partial e_2}{\partial a_{n_{RBF}}} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial a_1} & \dots & \frac{\partial e_n}{\partial a_{n_{RBF}}} \end{bmatrix},$$

где $n = N + K$ – суммарное количество пробных точек.

Элементы матрицы Якоби предлагается вычислять аналитически. Элементы матрицы $\mathbf{J}_w$ для внутренних пробных точек вычисляются по формуле

$$\frac{\partial r_i}{\partial w_j} = \frac{\partial (\Delta v_i - f_i)}{\partial w_j} = e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2a_j^2}} \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 2a_j^2}{a_j^4} = \varphi_j(\mathbf{x}_i) \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 2a_j^2}{a_j^4},$$

где $\varphi_j(\mathbf{x}_i)$ – значение РБФ φ_j в пробной точке $\mathbf{x}_i$.

Для граничных пробных точек вычисления производятся по формуле

$$\frac{\partial r_i}{\partial w_j} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2a_j^2}\right) = \varphi_j(\mathbf{x}_i). \quad \text{Элементы матрицы } \mathbf{J}_{c_1} \text{ для внутренних}$$

пробных точек имеют вид

$$\begin{aligned} \frac{\partial r_i}{\partial c_{j1}} &= \frac{w_j}{a_j^4} e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2a_j^2}} (x_1 - c_{j1}) \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 4a_j^2}{a_j^2} = \frac{w_j}{a_j^4} \varphi_j(\mathbf{x}_i) (x_1 - c_{j1}) \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 4a_j^2}{a_j^2} = \\ &= \frac{w_j}{a_j^6} \varphi_j(\mathbf{x}_i) (x_1 - c_{j1}) (\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 4a_j^2). \end{aligned}$$

Для граничных точек элементы матрицы записываются в виде

$$\frac{\partial r_i}{\partial c_{j1}} = w_j e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2a_j^2}} \frac{(x_{i1} - c_{j1})}{a_j^2} = w_j \varphi_j(\mathbf{x}_i) \frac{(x_{i1} - c_{j1})}{a_j^2}.$$

Аналогично вычисляются элементы матрицы $\mathbf{J}_{c_2}$. Элементы матрицы $\mathbf{J}_a$ для внутренних пробных точек имеют вид

$$\begin{aligned} \frac{\partial e_i}{\partial a_j} &= \frac{w_j}{a_j^5} e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2a_j^2}} \left[\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{a_j^2} (\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 2a_j^2) - 4(\|\mathbf{x}_i - \mathbf{c}_j\|^2 - a_j^2) \right] = \\ &= \frac{w_j}{a_j^5} \varphi_j(\mathbf{x}_i) \left[\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{a_j^2} (\|\mathbf{x}_i - \mathbf{c}_j\|^2 - 2a_j^2) - 4(\|\mathbf{x}_i - \mathbf{c}_j\|^2 - a_j^2) \right]. \end{aligned}$$

Для граничных точек элементы матрицы записываются в виде

$$\frac{\partial r_i}{\partial a_j} = w_j e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2a_j^2}} \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{a_j^3} = w_j \varphi_j(\mathbf{x}_i) \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{a_j^3}.$$

Условием завершения процесса обучения методом Левенберга – Марквардта является малое значение функционала ошибки (10) или средней квадратической ошибки.

В методе Левенберга – Марквардта параметр регуляризации μ должен изменяться в процессе обучения сети. Процесс обучения начинается при относительно большом значении параметра μ . Это означает, что в начале процесса обучения гессиан в (11) близок к приближенному значению $\mathbf{H} \approx \mu \mathbf{E}$, а вектор поправки определяется методом градиентного спуска с малым шагом $\Delta \boldsymbol{\theta}^{(k)} = -\mathbf{g}_{k-1} / \mu_k$. По мере уменьшения функционала ошибки параметр μ уменьшается и метод приближается к методу Ньютона с аппроксимацией гессиана $\mathbf{H} \approx \mathbf{J}^T \mathbf{J}$. Это обеспечивает высокую скорость сходимости, так как метод Ньютона вблизи минимума функционала ошибки имеет хорошую сходимость. В [22] рекомендуется начинать с некоторого значения μ_0 и коэффициента $\nu > 1$. Текущее значение μ делится на ν , если функционал ошибки уменьшается, или умножается на ν , если функционал ошибки увеличивается.

В [22] показано, что метод Левенберга – Марквардта эквивалентен методу доверительных областей, а радиус доверительной области регулируется параметром μ . Но в отличие от известных реализаций метода доверительных областей метод Левенберга – Марквардта не требует решения на каждой итерации обучения достаточно сложной задачи условной оптимизации. То есть метод Левенберга – Марквардта, сохраняя положительные свойства метода доверительных областей, является более простым.

Недостатком метода Левенберга – Марквардта является плохая обусловленность системы (11), зависящая от ширины РБФ и увеличивающаяся с

ростом точности вычислений. Известно [26], что матрица, элементами которой являются РБФ, является плохо обусловленной и обусловленность матрицы зависит от ширины РБФ. С ростом ширины РБФ элементы матрицы $\mathbf{J}_w$ стремятся к единице, а элементы матриц $\mathbf{J}_c$ и $\mathbf{J}_a$ стремятся к нулю. Число обусловленности матрицы $\mathbf{J}^T \mathbf{J}$ растет. Параметр регуляризации μ улучшает обусловленность системы (11), но уменьшение параметра μ по мере уменьшения погрешности приводит к ухудшению обусловленности.

Обсуждение

Эксперименты проводились в системе MATLAB R2019b. Для решения системы (11) использовался решатель системы MATLAB.

Эксперименты проводились на примере задачи (9) при $f(x_1, x_2) = \sin(\pi x_1) \cdot \sin(\pi x_2)$, $p(x_1, x_2) = 0$. Задача имеет аналитическое решение $u = -\frac{1}{2\pi^2} \sin \pi x \cdot \sin \pi y$ в единичном квадрате. Количество внутренних

и граничных пробных точек равно $N = 100$, $K = 40$. Штрафной коэффициент равен $\lambda = 100$. Центры RBF располагались регулярно на квадратной сетке с количеством центров по каждой координате, равным 8. Пробные точки располагались случайным образом в области решения и на границе области. Веса инициализировались нулевыми значениями. Начальная ширина всех РБФ была постоянной, равной 0,2.

На рис. 1 показано расположение центров, условное обозначение ширины (в виде окружностей с радиусами, равными ширине) РБФ и значения весов с использованием цветовой палитры MATLAB перед обучением сети (рис. 1,а) и после обучения (рис. 1,б). Рис. 1 показывает важность настройки параметров РБФ.

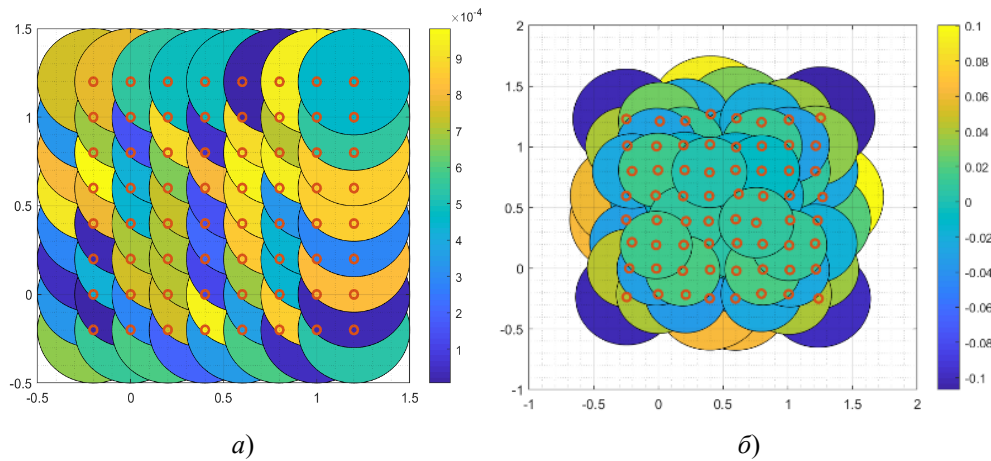


Рис. 1. Центры и ширина РБФ при решении первой задачи:
а – перед обучением сети; б – после обучения сети
методом Левенберга – Марквардта

По сравнению с аналитическим решением достигнута среднеквадратическая относительная погрешность $8,519 \cdot 10^{-4}$.

Зависимость среднеквадратической невязки различных алгоритмов от номера итерации при обучении РБФ-сети различными алгоритмами показана на рис. 2.

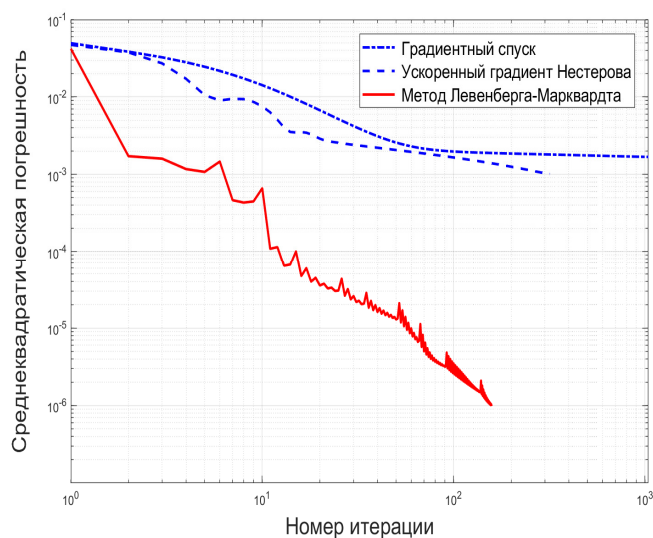


Рис. 2. Зависимости среднеквадратической невязки различных алгоритмов от номера итерации

Результаты экспериментов по решению краевой задачи на РБФ-сетях, обучаемых различными алгоритмами, представлены в табл. 1.

Таблица 1

Результаты экспериментов при решении краевой задачи 1

Алгоритм	Среднеквадратическая погрешность	Число итераций	Время решения, с
Градиентный спуск	$3 \cdot 10^{-3}$	1000	1100
Метод Нестерова	$7 \cdot 10^{-4}$	1000	1120
Метод Левенберга – Марквардта	10^{-6}	30	8,5

Метод градиентного спуска позволил решить модельную задачу с небольшой точностью. Для решения с большой точностью метод практически неприменим. Несколько бóльшую точность обеспечивает метод Нестерова. Только метод Левенберга – Марквардта позволил решить задачу с высокой точностью за приемлемое время. Метод Левенберга – Марквардта показал практически одинаковые результаты по сравнению с методом доверительных областей [15], но реализация метода Левенберга – Марквардта проще. Недостатками метода Левенберга – Марквардта являются плохая обусловленность системы, формирующей коррекцию параметров, и негладкий характер сходимости.

Таким образом, алгоритм метода Левенберга – Марквардта показал явное преимущество перед алгоритмами первого порядка и обеспечил точность

на уровне известных реализаций алгоритма доверительных областей, но проще этих алгоритмов.

Выводы

Сети радиальных базисных функций являются перспективным средством решения краевых задач, описываемых дифференциальными уравнениями в частных производных. Но известные методы обучения сетей радиальных базисных функций не обеспечивают быстрого обучения сетей радиальных базисных функций. В качестве пути устранения этого недостатка предложено совершенствовать алгоритмы обучения сетей.

Для обучения сетей радиальных базисных функций, предназначенных для решения ДУЧП, разработаны алгоритмы обучения на основе методов Нестерова и Левенберга – Марквардта, отличающиеся учетом специфики архитектуры сети и аналитическим вычислением параметров. Алгоритм на основе метода Левенберга – Марквардта позволил на модельной задаче достичь среднеквадратической невязки, не достижимой известными алгоритмами первого порядка. Предложенный алгоритм достигает малой погрешности за число итераций, равное числу итераций алгоритма на основе метода доверительных областей, но проще этого алгоритма, так как не требует решения на каждой итерации задачи условной оптимизации.

Библиографический список

1. Самарский, А. А. Математическое моделирование: Идеи. Методы. Примеры / А. А. Самарский, А. П. Михайлов. – Москва : Физматлит, 2005. – 320 с.
2. Mazumder, S. Numerical methods for partial differential equations: finite difference and finite volume methods / S. Mazumder. – Academic Press, 2015. – 461 p.
3. Толстых, А. И. Бессеточный метод на основе радиальных базисных функций / А. И. Толстых, Д. А. Ширококов // Журнал вычислительной математики и математической физики. – 2005. – Т. 45, № 8. – С. 1498–1505.
4. Meshfree Methods for Partial Differential Equations / ed. by M. Griebel, M. A. Schweitzer. – Springer, 2008. – 412 p.
5. ГОСТ Р 57188–2016. Численное моделирование физических процессов. Термины и определения. – Москва : Стандартинформ, 2016. – 12 с.
6. Cybenko, G. Approximation by Superposition of a Sigmoidal Function / G. Cybenko // Mathematics of Control, Signals and Systems. – 1989. – Vol. 2. – P. 303–314.
7. Yadav, N. An Introduction to Neural Network Methods for Differential Equations / N. Yadav, A. Yadav, M. Kumar. – Springer, 2015. – 115 p.
8. Горбаченко, В. И. Нейрокомпьютеры в решении краевых задач теории поля / В. И. Горбаченко. – Москва : Радиотехника, 2003. – 336 с.
9. Бойков, И. В. Приближенное решение эллиптических уравнений на нейронных сетях Хопфилда / И. В. Бойков, О. А. Баулина. // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2014. – № 1 (29). – С. 39–53.
10. Berg, O. A unified deep artificial neural network approach to partial differential equations in complex geometries / O. Berg, K. Nystrom. // Neurocomputing. – 2018. – Vol. 317, № 23. – P. 28–41.
11. Liu, Z. Neural network as a function approximator and its application in solving differential equations / Z. Liu, Y. Yantao, Q. Cai // Applied Mathematics and Mechanics. – 2019. – Vol. 40, № 2. – P. 237–248.
12. Park, J. Universal Approximation Using Radial-Basis-Function Networks / J. Park, I. W. Sandberg // Neural Computation. – 1991. – Vol. 3, № 2. – P. 246–257.
13. Chen, W. Recent Advances in Radial Basis Function Collocation Methods / W. Chen, Z.-J. Fu. – Springer, 2014. – 90 p.

14. Kansa, E. J. Motivation for using Radial Basis Function to solve PDEs. – URL <http://www.cityu.edu.hk/rbf-pde/files/overview-html.html>
15. Горбаченко, В. И. Решение краевых задач математической физики с помощью сетей радиальных базисных функций / В. И. Горбаченко, М. В. Жуков. // Журнал вычислительной математики и математической физики. – 2017. – Т. 57, № 1. – С. 115–126.
16. Елисов, Л. Н. Обучение методом доверительных областей сетей радиальных базисных функций при решении краевых задач / Л. Н. Елисов, В. И. Горбаченко, М. В. Жуков // Автоматика и телемеханика. – 2018. – № 9. – С. 95–105.
17. Alqezweeni, M. M. Efficient Solving of Boundary Value Problems Using Radial Basis Function Networks Learned by Trust Region Method / M. M. Alqezweeni, V. I. Gorbachenko, M. V. Zhukov, M. S. Jaafar // International Journal of Mathematics and Mathematical Sciences. – Vol. 2018, Article ID 9457578, 4 pages, 2018.
18. Aggarwal, C. C. Neural Networks and Deep Learning: A Textbook / C. C. Aggarwal. – Springer, 2018. – 497 p.
19. Niyogi, P. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions / P. Niyogi, F. Girosi. // Neural Computation. – 1996. – Vol. 8, № 4. – P. 819–842.
20. Kachenderfer, M. J. Algorithms for Optimization / M. J. Kachenderfer, T. A. Wheeler. – MIT Press, 2019. – 520 p.
21. Conn, A. R. Trust-region methods / A. R. Conn, N. I. M. Gould, P. L. Toint. – SIAM, 1987. – 972 p.
22. Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters / D. W. Marquardt // Journal of the Society for Industrial and Applied Mathematics. – 1963. – Vol. 11, № 2. – P. 431–441.
23. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвиль. – Москва : ДМК Пресс, 2018. – 652 с.
24. Sutskever, I. On the importance of initialization and momentum in deep learning / I. Sutskever, J. Martens, G. Dahl, G. Hinton // ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning. – 2013. – Vol. 28. – P. III-1139-III-1147.
25. Алкезуини, М. М. Совершенствование алгоритмов обучения сетей радиальных базисных функций для решения задач аппроксимации / М. М. Алкезуини, В. И. Горбаченко // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 3 (23). – С. 123–138.
26. Boyd, J. P. Numerical experiments on the condition number of the interpolation matrices for radial basis functions / J. P. Boyd, K. W. Gildersleeve // Applied Numerical Mathematics. – 2011. – № 61 (4). – P. 443–459.

References

1. Samarskiy A. A., Mikhaylov A. P. *Matematicheskoe modelirovanie: Idei. Metody. Primery* [Mathematical modeling: Ideas. Methods. Examples]. Moscow: Fizmatlit, 2005, 320 p. [In Russian]
2. Mazumder S. *Numerical methods for partial differential equations: finite difference and finite volume methods*. Academic Press, 2015, 461 p.
3. Tolstykh A. I., Shirobokov D. A. *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki* [Journal of computational mathematics and mathematical physics]. 2005, vol. 45, no. 8, pp. 1498–1505. [In Russian]
4. *Meshfree Methods for Partial Differential Equations*. Ed. by M. Griebel, M. A. Schweitzer. Springer, 2008, 412 p.
5. GOST R 57188–2016. *Chislennoe modelirovanie fizicheskikh protsessov. Terminy i opredeleniya* [GOST R 57188-2016. Numerical modeling of physical processes. Terms and definitions]. Moscow: Standartinform, 2016, 12 p. [In Russian]

6. Cybenko G. *Mathematics of Control, Signals and Systems*. 1989, vol. 2, pp. 303–314.
7. Yadav N., Yadav A., Kumar M. *An Introduction to Neural Network Methods for Differential Equations*. Springer, 2015, 115 p.
8. Gorbachenko V. I. *Neyrokomp'yutery v reshenii kraevykh zadach teorii polya* [Neurocomputers in solving boundary value problems of field theory]. Moscow: Radio-tekhnika, 2003, 336 p. [In Russian]
9. Boykov I. V., Baulina O. A. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki* [University proceedings. Volga region. Physical and mathematical sciences]. 2014, no. 1 (29), pp. 39–53. [In Russian]
10. Berg O., Nystrom K. *Neurocomputing*. 2018, vol. 317, no. 23, pp. 28–41.
11. Liu Z., Yantao Y., Cai Q. *Applied Mathematics and Mechanics*. 2019, vol. 40, no. 2, pp. 237–248.
12. Park J., Sandberg I. W. *Neural Computation*. 1991, vol. 3, no. 2, pp. 246–257.
13. Chen W., Fu Z. J. *Recent Advances in Radial Basis Function Collocation Methods*. Springer, 2014, 90 p.
14. Kansa E. J. *Motivation for using Radial Basis Function to solve PDEs*. Available at <http://www.cityu.edu.hk/rbf-pde/files/overview-html.html>
15. Gorbachenko V. I., Zhukov M. V. *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki* [Journal of computational mathematics and mathematical physics]. 2017, vol. 57, no. 1, pp. 115–126. [In Russian]
16. Elisov L. N., Gorbachenko V. I., Zhukov M. V. *Avtomatika i telemekhanika* [Automation and remote control]. 2018, no. 9, pp. 95–105. [In Russian]
17. Alqezweeni M. M., Gorbachenko V. I., Zhukov M. V., Jaafar M. S. *International Journal of Mathematics and Mathematical Sciences*. Vol. 2018, Article ID 9457578, 4 pages, 2018.
18. Aggarwal C. C. *Neural Networks and Deep Learning: A Textbook*. Springer, 2018, 497 p.
19. Niyogi P., Girosi F. *Neural Computation*. 1996, vol. 8, no. 4, pp. 819–842.
20. Kachenderfer M. J., Wheeler T. A. *Algorithms for Optimization*. MIT Press, 2019, 520 p.
21. Conn A. R., Gould N. I. M., Toint P. L. *Trust-region methods*. SIAM, 1987, 972 p.
22. Marquardt D. W. *Journal of the Society for Industrial and Applied Mathematics*. 1963, vol. 11, no. 2, pp. 431–441.
23. Gudfellou Ya., Bendzhio I., Kurvil' A. *Glubokoe obuchenie* [Deep learning]. Moscow: DMK Press, 2018, 652 p. [In Russian]
24. Sutskever I., Martens J., Dahl G., Hinton G. *ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning*. 2013, vol. 28, pp. III-1139-III-1147.
25. Alkezuini M. M., Gorbachenko V. I. *Modeli, sistemy, seti v ekonomike, tekhnike, prirode i obshchestve* [Models, systems, networks in economy, technology, nature and society]. 2017, no. 3 (23), pp. 123–138. [In Russian]
26. Boyd J. P., Gildersleeve K. W. *Applied Numerical Mathematics*. 2011, no. 61 (4), pp. 443–459.

Горбаченко Владимир Иванович
 доктор технических наук, профессор,
 заведующий кафедрой компьютерных
 технологий,
 Пензенский государственный
 университет
 (Россия, г. Пенза, ул. Красная, 40)
 E-mail: gorvi@mail.ru

Gorbachenko Vladimir Ivanovich
 doctor of technical sciences, professor,
 head of sub-department
 of computer technologies,
 Penza State University
 (40 Krasnaya street, Penza, Russia)

Алкезуини Мухи Муртада Мухи
аспирант,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: mohieit@mail.ru

Alkezuini Mukhi Murtada Mukhi
postgraduate student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Образец цитирования:

Горбаченко, В. И. Моделирование объектов с распределенными параметрами на нейронных сетях / В. И. Горбаченко, М. М. Алкезуини // Модели, системы, сети в экономике, технике, природе и обществе. – 2019. – № 4 (32). – С. 50–64.