

## РАЗДЕЛ 2

# МОДЕЛИ, СИСТЕМЫ, МЕХАНИЗМЫ В ТЕХНИКЕ

---

---

УДК 004.032.26

### СОВЕРШЕНСТВОВАНИЕ АЛГОРИТМОВ ОБУЧЕНИЯ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ АППРОКСИМАЦИИ<sup>1</sup>

*М. М. Алкезуни, В. И. Горбаченко*

### IMPROVEMENT OF THE LEARNING ALGORITHMS IN RADIAL BASIS FUNCTIONS NETWORKS FOR SOLVING THE APPROXIMATION TASKS

*M. M. Algezweeni, V. I. Gorbachenko*

*Аннотация.* Актуальность и цели. Перспективным инструментом решения задач аппроксимации, в том числе задач бессеточной аппроксимации, являются сети радиальных базисных функций, которые представляют специальный вид нейронных сетей. Решение задачи формируется в процессе обучения сети. Но в настоящее время отсутствуют быстрые и достаточно простые алгоритмы обучения сетей радиальных базисных функций. Цель данного исследования – разработка и экспериментальное исследование новых быстрых алгоритмов обучения сетей радиальных базисных функций при решении задач аппроксимации. *Материалы и методы.* Реализация поставленных задач достигнута за счет использования для обучения сетей радиальных базисных функций современных ускоренных градиентных методов первого порядка и адаптации метода Левенберга – Марквардта. *Результаты.* Для обучения сетей радиальных базисных функций впервые разработаны алгоритмы на основе методов первого порядка: градиентный спуск с импульсом (импульсный метод), алгоритм ускоренного градиента Нестерова и RMSProp в сочетании с ускоренным градиентом Нестерова. Показаны преимущества последовательной настройки параметров в каждом итерационном цикле обучения сети. Разработана реализация метода Левенберга – Марквардта для обучения сетей радиальных базисных функций. Получены формулы для расчета параметров сетей при реализации алгоритмов. Даны рекомендации по выбору параметров обучения сетей. Показана связь между методом Левенберга – Марквардта и методом доверительных областей. Таким образом, алгоритмом Левенберга – Марквардта можно достичь тех же результатов, что и более сложным алгоритмом метода доверительных областей. Создан комплекс программ в системе MatLab, реализующий разработанные алгоритмы. Проведены экспериментальные ис-

---

<sup>1</sup> Исследование выполнено при финансовой поддержке РФФИ (грант № 16-08-00906).

следования разработанных алгоритмов. *Выводы.* Для решения задач аппроксимации на сетях радиальных базисных функций можно рекомендовать адаптированный в работе алгоритм Левенберга – Марквардта. Если в алгоритме Левенберга – Марквардта возникают проблемы с плохой обусловленностью системы линейных алгебраических уравнений, то можно рекомендовать алгоритм ускоренного градиента Нестерова.

**Ключевые слова:** бессеточная аппроксимация, нейронная сеть, сеть радиальных базисных функций, градиентный алгоритм обучения, импульсный метод, метод ускоренного градиента Нестерова, метод Левенберга – Марквардта.

**Abstract. Background.** A promising tool for solving the approximation tasks, including the tasks of mesh-free approximation, this tool is the radial basis functions networks, which are considered as a special form of the neural networks. The solution of the problem gets formed in the process of the network's learning. But currently, there are no fast and simple algorithms for the learning of the radial basis functions networks. The purpose of this research is the development and experimental study of new fast algorithms for the learning of the radial basis functions networks to solve the approximation tasks. **Materials and methods.** The implementation of the tasks can be achieved by the use of learning networks of the radial basis functions for the modern and accelerated gradient methods of the first order and by the adaptation of Levenberg – Marquardt method. **Results.** For the learning of the radial basis functions networks, for the first time there were such developed algorithms based on the first order methods: gradient descent with an impulse (the impulse method), the accelerated gradient algorithm of Nesterov and RMSProp in combination with the accelerated gradient Nesterov. The showed advantages of the sequential parameter settings in each iterative cycle of the network's learning. The implementation of the Levenberg – Marquardt method for the learning of the radial basis functions networks was developed. The resulted formulas are meant for the calculating of the networks parameters through the algorithms implementation. The given recommendations which are concerning the choosing of parameters for the learning of networks. The showed relationship between the method of Levenberg – Marquardt and the trusted regions method. Thus, now it would be possible with the algorithm of Levenberg – Marquardt to achieve the same results as with the use of the more complex trusted regions method. A software package in the system of MATLAB was created to implement the developed algorithms. Experimental studies of the developed algorithms were carried out. **Conclusions.** For solving the approximation tasks in the radial basis functions networks, it can be recommended to adapt, in the process of work, the algorithm of Levenberg – Marquardt. If the algorithm of Levenberg – Marquardt is having some problems concerning the poor conditionality of the linear algebraic equations system, it would possible to recommend Nesterov's accelerated gradient algorithm.

**Key words:** meshfree approximation, neural network, radial basis functions network, gradient learning algorithm, method of impulse, Nesterov's accelerated gradient method, Levenberg – Marquardt method.

### **Введение**

При моделировании рельефа, реконструкции поверхностей и во многих других случаях возникает необходимость аппроксимации «рассеянных» данных [1], когда узлы интерполяции расположены произвольным образом, а не на некоторой сетке. Методы аппроксимации таких данных являются бессеточными (meshfree) [2, 3]. Они не накладывают ограничений на расположение узлов интерполяции и могут применяться и в случае расположения

узлов интерполяции на сетке. Методы бессеточной аппроксимации являются основой многих бессеточных методов решения уравнений в частных производных.

Для бессеточной аппроксимации широко используются радиальные базисные функции (РБ-функции) [2, 3]. РБ-функция – это функция, значение которой в некоторой точке зависит от расстояния между точкой и параметром РБ-функции, называемым центром. Обычно параметры РБ-функций подбираются, а веса находятся из условий равенства аппроксимированных значений и известных значений функции в узлах интерполяции. Недостатком применения РБ-функций является трудно формализуемый подбор параметров РБ-функций. Этот недостаток устраняется при использовании специального вида нейронных сетей – сетей радиальных базисных функций (РБФ-сетей) [4–5]. При использовании РБФ-сетей решение формируется в процессе обучения сети с настройкой весов и параметров РБ-функций.

Для обучения РБФ-сетей используются в основном градиентные методы [4–5], среди которых выделяют методы первого порядка, использующие для своей реализации первые производные минимизируемой функции, т.е. градиент функции, и методы второго порядка, использующие вторые производные – матрицу Гессе. Все градиентные алгоритмы позволяют найти только локальный минимум функции. Методы первого порядка просты в реализации, но работают медленно. Методы второго порядка выполняются за гораздо меньшее число итераций, но сложны и ресурсоемки, т.е. одна итерация занимает много времени и требует много памяти. Для обучения РБФ-сетей используют в основном методы первого порядка. Известные методы второго порядка, например метод Левенберга – Марквардта [5], не применяются для обучения РБФ-сетей, а предложенный в [6] быстрый алгоритм метода доверительных областей довольно сложен.

В настоящее время для глубокого обучения сетей, содержащих большое количество слоев и обучающихся на больших объемах данных, усилился интерес к несложным ускоренным градиентным методам первого порядка [7]. Такие методы не применялись для обучения РБФ-сетей.

Целью настоящей работы является разработка и экспериментальное исследование ускоренных градиентных методов первого порядка и метода Левенберга – Марквардта для обучения РБФ-сетей при решении задач аппроксимации.

### *Сеть радиальных базисных функций*

РБФ-сеть – это двухслойная сеть, первый слой которой составляют РБ-функции, а второй – линейный сумматор [4–5]. В случае аппроксимации функции двух переменных сеть имеет два входа – координаты точки – и один выход – значение функции в точке. Выход сети радиальных базисных функций при значении входа  $\mathbf{x} = [x_1, x_2]^T$  (значение функции в точке  $\mathbf{x}$ ) описывается выражением

$$u(\mathbf{x}) = \sum_{k=1}^{n_{RBF}} w_k \varphi_k(\mathbf{x}), \quad (1)$$

где  $\mathbf{x}$  – входной вектор;  $n_{RBF}$  – количество РБ-функций (количество нейронов);  $w_k$  – вес  $k$ -го нейрона;  $\varphi_k(\mathbf{x})$  – значение  $k$ -й РБ-функции в точке  $\mathbf{x}$ .

В качестве РБ-функций используются различные радиальные базисные функции [2]. В данной работе применяется функция Гаусса (гауссиан), в двумерном случае имеющая вид

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

где  $\|\mathbf{x} - \mathbf{c}\| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}$  – евклидова норма (расстояние между точкой  $\mathbf{x}$  и центром  $\mathbf{c}$  РБ-функции);  $\mathbf{c} = [c_1, c_2]^T$  – вектор координат центра РБ-функции;  $a$  – ширина (параметр формы).

При решении задач аппроксимации функций входные векторы при обучении сети представляют собой координаты пробных точек (узлов интерполяции), а целевые значения – значения функции в пробных точках. После обучения сеть при подаче координат произвольной точки области определения функции выдает значение функции в этой точке.

Обучение РБФ-сетей имеет свои особенности. Наличие только двух слоев упрощает обучение сети. Но в процессе обучения РБФ-сети необходимо настраивать две группы параметров сети: веса, которые входят линейно в выходной сигнал сети (1), и параметры РБ-функций (в случае гауссиана (2) – центры и ширина), которые входят нелинейно в выходной сигнал. Открытым является вопрос выбора количества РБ-функций и начальных значений параметров РБ-функций. Известна зависимость между количеством РБ-функций  $n_{RBF}$  для задач аппроксимации и количеством узлов интерполяции  $N$ :  $n_{RBF} \propto N^{1/3}$ , где  $\propto$  означает пропорциональность [8]. Однако эта оценка сильно завышена, и количество РБ-функций требуется подбирать. Если в процессе обучения сети настраиваются параметры РБ-функций, то их начальные значения целесообразно выбирать случайным образом.

Обучение РБФ-сети представляет собой минимизацию некоторого функционала ошибки (функции потерь в терминах машинного обучения). Будем использовать функционал ошибки, представляющий собой сумму квадратов ошибок в пробных точках:

$$I = \frac{1}{2} \sum_{j=1}^n e_j^2 = \frac{1}{2} \sum_{j=1}^n (u(\mathbf{p}_j) - T_j)^2, \quad (3)$$

где  $n$  – количество пробных точек;  $e_j$  – ошибка решения в  $j$ -й пробной точке;  $\mathbf{p}_j$  – координаты  $j$ -й пробной точки (в случае аппроксимации функции двух переменных  $\mathbf{p}_j = [p_{j1}, p_{j2}]^T$ ),  $u(\mathbf{p}_j)$  – решение (1) в  $j$ -й пробной точке;  $T_j$  – целевое значение в  $j$ -й пробной точке; множитель  $1/2$  введен для упрощения вычислений.

**Разработка ускоренных градиентных алгоритмов  
первого порядка обучения РБФ-сетей**

Рассмотрим алгоритм градиентного спуска обучения РБФ-сети. Если  $I$  – функционал ошибки (3), а  $\theta$  – вектор одного или всех параметров сети (в нашем случае это векторы весов, центров и ширины РБ-функций), то настройка вектора  $\theta$  на  $k$ -й итерации градиентного спуска описывается следующим образом [5]:

$$\theta^{(k+1)} = \theta^{(k)} + \Delta\theta^{(k+1)}, \quad (4)$$

где  $\Delta\theta^{(k+1)} = -\eta \mathbf{g}_0(\theta^{(k)})$  – поправка вектора  $\theta$ , здесь  $\eta$  – подбираемый числовой коэффициент (скорость обучения);  $\mathbf{g}_0(\theta^{(k)})$  – вектор градиента функционала  $I$  (3) по значению параметра  $\theta^{(k)}$  на итерации  $k$ . Процесс вычислений по (4) продолжается до малого значения функционала ошибки (3). Более наглядно применять среднеквадратическую погрешность (Mean Squared Error):

$$I_{MSE} = \frac{1}{n} \sum_{j=1}^n (u(\mathbf{p}_j) - T_j)^2. \quad (5)$$

В алгоритме градиентного спуска с импульсом (в русской литературе принято использовать термин «момент» [5]; принятый в англоязычной литературе термин «momentum» (импульс) по аналогии с импульсом в механике точнее отражает суть метода [7]) поправка к вектору параметров описывается следующим образом:

$$\Delta\theta^{(k+1)} = \alpha\Delta\theta^{(k)} - \eta \mathbf{g}_0(\theta^{(k)}), \quad (6)$$

где  $\alpha$  – коэффициент момента, принимающий значения в интервале  $[0, 1]$ ;  $\eta$  – скорость обучения.

Выражение (6) содержит слагаемые, зависящие от градиента и не зависящие от градиента. Причем чем больше значение коэффициента  $\alpha$ , тем сильнее на настройку весов оказывает влияние слагаемое, не зависящее от градиента. Это влияние существенно возрастает на плоских участках целевой функции и вблизи локальных минимумов. В этих областях слагаемое, не зависящее от градиента, начинает доминировать в (6), что приводит к выходу из этой области. Можно сказать, что изменение весов происходит «по инерции» в направлении вектора поправки предыдущего шага обучения.

В методе ускоренного градиента Нестерова (NAG – Nesterov Accelerated Gradient) [7, 9] поправка вектора параметров описывается следующим образом:

$$\Delta\theta^{(k+1)} = \alpha\Delta\theta^{(k)} - \eta \mathbf{g}_0(\theta^{(k)} + \alpha\Delta\theta^{(k)}).$$

Метод NAG отличается от градиентного спуска с импульсом более точным вычислением вектора поправки.

При обучении глубоких сетей получили распространение алгоритмы с адаптивной скоростью обучения [7]. В этих алгоритмах используется разная скорость обучения для различных компонентов вектора параметров сети, так как разные компоненты могут оказывать различное влияние на обучение сети. В частности, эффективным и практичным методом является RMSProp (Root Mean Square Propagation) и его сочетание с ускоренным градиентом Нестерова [7],  $k$ -я итерация которого включает следующие вычисления:

$$\mathbf{g} = \mathbf{g}_0(\boldsymbol{\theta}^{(k)} + \alpha\Delta\boldsymbol{\theta}^{(k)}), \quad \mathbf{r}^{(k+1)} = \rho\mathbf{r}^{(k)} + (1-\rho)\mathbf{g} \bullet \mathbf{g},$$

$$\Delta\boldsymbol{\theta}^{(k+1)} = \alpha\Delta\boldsymbol{\theta}^{(k)} - \frac{\varepsilon}{\sqrt{\mathbf{r}^{(k+1)}}} \bullet \mathbf{g}, \quad \boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}^{(k+1)},$$

где  $\mathbf{r}^{(0)} = \mathbf{0}$ ;  $\bullet$  – операция поэлементного умножения (произведение Адамара);  $\sqrt{\mathbf{r}^{(k+1)}}$  вычисляется поэлементно;  $\rho, \alpha, \varepsilon$  – подбираемые коэффициенты.

При обучении РБФ-сети можно реализовать одновременную и последовательную стратегию настройки векторов весов, центров и ширины на каждом цикле обучения. При одновременной настройке в каждой итерации обучения при значениях векторов параметров, полученных на предыдущей итерации, находятся векторы градиентов функционала ошибки по каждому параметру, затем вычисляются поправки параметров и вычисляются обновленные векторы параметров. При последовательной настройке сначала вычисляется градиент по весам (настройка весов оказывает наибольшее влияние на обучение сети) и вычисляется обновленный вектор весов. Затем с использованием обновленного вектора весов вычисляется градиент по центрам и вычисляется обновленный вектор центров. Наконец, с использованием обновленных векторов весов и центров вычисляется градиент по ширине и обновляется вектор ширины. Так как при последовательной настройке каждый параметр настраивается при обновленной части других параметров, можно предположить, что последовательная настройка может обеспечить сокращение числа итераций обучения.

Компоненты градиента функционала по весам, центрам и ширине сложно вычислить аналитически (рассмотрен случай аппроксимации функции двух переменных, номера итераций опущены):

$$\frac{\partial I}{\partial w_i} = \sum_{j=1}^n (u(\mathbf{p}_j) - T_j) \cdot \varphi_i(\mathbf{p}_j), \quad \frac{\partial I}{\partial c_{i1}} = w_i \sum_{j=1}^n (u(\mathbf{p}_j) - T_j) \cdot \varphi_i(\mathbf{p}_j) \cdot \frac{p_{j1} - c_{i1}}{a_i^2};$$

$$\frac{\partial I}{\partial c_{i2}} = w_i \sum_{j=1}^n (u(\mathbf{p}_j) - T_j) \cdot \varphi_i(\mathbf{p}_j) \cdot \frac{p_{j2} - c_{i2}}{a_i^2}, \quad \frac{\partial I}{\partial a_i} = w_i \sum_{j=1}^n (u(\mathbf{p}_j) - T_j) \cdot \varphi_i(\mathbf{p}_j) \cdot \frac{\|\mathbf{p}_j - \mathbf{c}_i\|}{a_i^3},$$

где  $c_{i1}$  и  $c_{i2}$  – координаты центра  $i$ -й РБФ-функции;  $p_{j1}$  и  $p_{j2}$  – координаты пробной точки  $\mathbf{p}_j$ ;  $\|\mathbf{p}_j - \mathbf{c}_i\|$  – евклидова норма.

**Адаптация алгоритма Левенберга – Марквардта  
для обучения РБФ-сетей**

Рассмотрим применение для обучения сети метода второго порядка – метода Левенберга – Марквардта [5, 10], являющегося реализацией известного метода безусловной оптимизации [11]. Метод Левенберга – Марквардта широко применяется для обучения многослойных персептронов [5, 10], но практически не применяется для обучения сетей радиальных базисных функций. Следует отметить работу [12], в которой предложена эффективная с вычислительной точки зрения аппроксимация матрицы Гессе, рассмотрена реализация метода Гаусса – Ньютона для обучения РБФ-сетей, но не рассмотрена реализация метода Левенберга – Марквардта.

Рассмотрим настройку параметров. Введем единый вектор параметров

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

где параметры  $j$ -й РБ-функции ( $j=1, 2, 3, \dots, n_{RBF}$ ):  $w_j$  – вес;  $c_{j1}$  и  $c_{j2}$  – координаты центра (рассматриваем аппроксимацию функций двух переменных);  $a_j$  – ширина.

Настройка вектора параметров  $\boldsymbol{\theta}$  в  $k$ -м цикле (итерации) производится по формуле

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

где вектор поправки  $\Delta\boldsymbol{\theta}^{(k)}$  находится из решения системы линейных алгебраических уравнений

$$\left( \mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E} \right) \Delta\boldsymbol{\theta}^{(k)} = -\mathbf{g}_{k-1}, \quad (7)$$

где  $\mathbf{J}_{k-1}$  – матрица Якоби, вычисленная по значениям параметров сети в  $k-1$  итерации;  $\mu_k$  – параметр регуляризации, изменяющийся на каждом шаге обучения;  $\mathbf{E}$  – единичная матрица;  $\mathbf{g} = \mathbf{J}^T \mathbf{e}$  – вектор градиента функционала (3) по вектору параметров  $\boldsymbol{\theta}$ ;  $\mathbf{e} = [e_1, e_2, \dots, e_n]^T$  – вектор ошибок.

Матрица Якоби имеет вид

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \dots & \frac{\partial e_1}{\partial w_{n_{RBF}}} & \frac{\partial e_1}{\partial c_{11}} & \dots & \frac{\partial e_1}{\partial c_{n_{RBF}1}} & \frac{\partial e_1}{\partial c_{12}} & \dots & \frac{\partial e_1}{\partial c_{n_{RBF}2}} & \frac{\partial e_1}{\partial a_1} & \dots & \frac{\partial e_1}{\partial a_{n_{RBF}}} \\ \frac{\partial e_2}{\partial w_1} & \dots & \frac{\partial e_2}{\partial w_{n_{RBF}}} & \frac{\partial e_2}{\partial c_{11}} & \dots & \frac{\partial e_2}{\partial c_{n_{RBF}1}} & \frac{\partial e_2}{\partial c_{12}} & \dots & \frac{\partial e_2}{\partial c_{n_{RBF}2}} & \frac{\partial e_2}{\partial a_1} & \dots & \frac{\partial e_2}{\partial a_{n_{RBF}}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_n}{\partial w_1} & \dots & \frac{\partial e_n}{\partial w_{n_{RBF}}} & \frac{\partial e_n}{\partial c_{11}} & \dots & \frac{\partial e_n}{\partial c_{n_{RBF}1}} & \frac{\partial e_n}{\partial c_{12}} & \dots & \frac{\partial e_n}{\partial c_{n_{RBF}2}} & \frac{\partial e_n}{\partial a_1} & \dots & \frac{\partial e_n}{\partial a_{n_{RBF}}} \end{bmatrix}. \quad (8)$$

Представим матрицу Якоби (8) в блочном виде:

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

где

$$\mathbf{J}_w = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_{n_{RBF}}} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_{n_{RBF}}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_n}{\partial w_1} & \frac{\partial e_n}{\partial w_2} & \dots & \frac{\partial e_n}{\partial w_{n_{RBF}}} \end{bmatrix}; \quad (9)$$

$$\mathbf{J}_{c_1} = \begin{bmatrix} \frac{\partial e_1}{\partial c_{11}} & \dots & \frac{\partial e_1}{\partial c_{n_{RBF}1}} \\ \frac{\partial e_2}{\partial c_{11}} & \dots & \frac{\partial e_2}{\partial c_{n_{RBF}1}} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial c_{11}} & \dots & \frac{\partial e_n}{\partial c_{n_{RBF}1}} \end{bmatrix}, \quad \mathbf{J}_{c_2} = \begin{bmatrix} \frac{\partial e_1}{\partial c_{12}} & \dots & \frac{\partial e_1}{\partial c_{n_{RBF}2}} \\ \frac{\partial e_2}{\partial c_{12}} & \dots & \frac{\partial e_2}{\partial c_{n_{RBF}2}} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial c_{12}} & \dots & \frac{\partial e_n}{\partial c_{n_{RBF}2}} \end{bmatrix}; \quad (10)$$

$$\mathbf{J}_a = \begin{bmatrix} \frac{\partial e_1}{\partial a_1} & \dots & \frac{\partial e_1}{\partial a_{n_{RBF}}} \\ \frac{\partial e_2}{\partial a_1} & \dots & \frac{\partial e_2}{\partial a_{n_{RBF}}} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial a_1} & \dots & \frac{\partial e_n}{\partial a_{n_{RBF}}} \end{bmatrix}.$$

Элементы матрицы  $\mathbf{J}_w$  (9) с учетом (3) и (1) имеют вид

$$\frac{\partial e_i}{\partial w_j} = \frac{\partial}{\partial w_j} [u(\mathbf{p}_i) - T_i] = \frac{\partial u(\mathbf{p}_i)}{\partial w_j} = \varphi_j(\mathbf{p}_i), \quad (11)$$

где  $\varphi_j(\mathbf{p}_i)$  – значение  $j$ -й радиальной базисной функции (2) в пробной точке  $\mathbf{p}_i$ .

Элементы матрицы  $\mathbf{J}_{c_1}$  (10) описываются формулой

$$\begin{aligned} \frac{\partial e_i}{\partial c_{j1}} &= \frac{\partial}{\partial c_{j1}} [u(\mathbf{p}_i) - T_i] = \frac{\partial}{\partial c_{j1}} \left[ \sum_{k=1}^{n_{RBF}} w_k \varphi_k(\mathbf{p}_i) \right] = w_j \frac{\partial}{\partial c_{j1}} \left[ e^{-\frac{(p_{i1}-c_{j1})^2 + (p_{i2}-c_{j2})^2}{2a_j^2}} \right] = \\ &= w_j e^{-\frac{\|\mathbf{p}_i - \mathbf{c}_j\|^2}{2a_j^2}} \cdot \frac{\partial}{\partial c_{j1}} \left[ -\frac{(p_{i1}-c_{j1})^2 + (p_{i2}-c_{j2})^2}{2a_j^2} \right] = w_j \varphi_j(\mathbf{p}_i) \cdot \frac{p_{i1} - c_{j1}}{a_j^2}. \end{aligned}$$



Аналогично для элементов матрицы  $\mathbf{J}_{c_2}$  получаем

$$\frac{\partial e_i}{\partial c_{j2}} = w_j \cdot \varphi_j(\mathbf{p}_i) \cdot \frac{p_{i2} - c_{j2}}{a_j^2}.$$

Элементы матрицы  $\mathbf{J}_a$  вычисляются по формуле

$$\begin{aligned} \frac{\partial e_i}{\partial a_j} &= \frac{\partial}{\partial a_j} [u(\mathbf{p}_i) - T_i] = \frac{\partial}{\partial a_j} \left[ \sum_{k=1}^{n_{RRF}} w_k \varphi_k(\mathbf{p}_i) \right] = \\ &= w_j \frac{\partial}{\partial a_j} \left[ e^{-\frac{\|\mathbf{p}_i - \mathbf{c}_j\|^2}{2a_j^2}} \right] = w_j e^{-\frac{\|\mathbf{p}_i - \mathbf{c}_j\|^2}{2a_j^2}} \cdot \frac{\partial}{\partial a_j} \left[ -\frac{\|\mathbf{p}_i - \mathbf{c}_j\|^2}{2a_j^2} \right] = w_j \varphi_j(\mathbf{p}_i) \cdot \frac{\|\mathbf{p}_i - \mathbf{c}_j\|^2}{a_j^3}. \end{aligned}$$

Матрица  $\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}$  является плотной симметричной и положительно определенной. Поэтому для решения системы (7) целесообразно использовать метод Холецкого [13], реализованный в математических пакетах, например в MatLab, и в математических библиотеках популярных языков программирования. Недостатком метода Холецкого является использование при выполнении разложения матриц длительной операции извлечения квадратного корня. От этого недостатка свободен метод LDL<sup>T</sup>-разложения [13], который представляет матрицу в виде  $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T$ , где  $\mathbf{L}$  – нижняя треугольная матрица с единичной главной диагональю;  $\mathbf{D}$  – диагональная матрица;  $\mathbf{T}$  – операция транспонирования матрицы. При разложении не применяется операция извлечения квадратного корня.

В методе Левенберга – Марквардта важен правильный подбор параметра  $\mu$ . В начале процесса обучения, когда вектор весов далек от оптимального, используется относительно большое значение параметра  $\mu$ . В этом случае гессиан заменяется приближенным значением  $\mathbf{H} \approx \mu \mathbf{E}$ , а вектор поправки определяется методом градиентного спуска с малым шагом

$$\Delta \boldsymbol{\theta}^{(k)} = -\frac{1}{\mu_k} \mathbf{g}_{k-1}.$$

По мере уменьшения погрешности параметр  $\mu$  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. Марквардт рекомендует [14] начинать со значения  $\mu_0$  и коэффициента  $\nu > 1$ . Текущее значение  $\mu$  делится на  $\nu$ , если функционал ошибки уменьшается, или умножается на  $\nu$ , если функционал ошибки увеличивается.

Процесс заканчивается при малом значении функционала ошибки (3) или среднеквадратической погрешности (5).

Можно показать, что метод Левенберга – Марквардта является вариантом метода доверительных областей (trust-region methods) [15]. Особенно

стью метода доверительных областей является поиск минимума функции в ограниченной доверительной области. В [6] разработана реализация метода доверительных областей для обучения сетей радиальных базисных функций. Но метод доверительных областей является достаточно сложным, так как из-за условия поиска минимума в доверительной области требует решения задачи условной оптимизации, в отличие от других градиентных методов обучения нейронных сетей, в которых решается задача безусловной оптимизации. Марквардт доказал [14], что этот метод эквивалентен методу доверительных областей, при этом радиус доверительной области регулируется параметром  $\mu$ . Таким образом, метод Левенберга – Марквардта, сохраняя положительные качества метода доверительных областей, является более простым в реализации.

### *Экспериментальное исследование алгоритмов обучения РБФ-сетей*

Экспериментальное исследование рассмотренных методов проводилось на примере аппроксимации функции  $z = x^2 + y^2$  в области  $(x = -3 \dots +3, y = -3 \dots +3)$ . Количество узлов интерполяции равно 100. Узлы интерполяции располагались случайным образом в области (рис. 1). Количество РБ-функций (нейронов) равно 16. В начальном состоянии центры РБ-функций располагались на сетке (рис. 2).

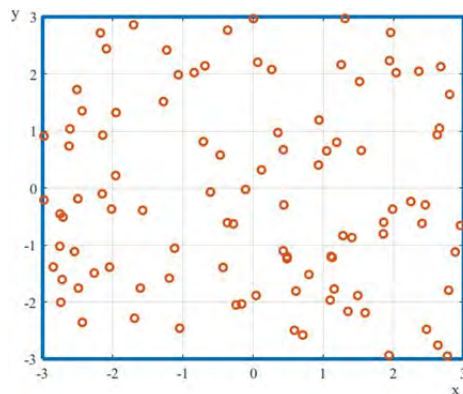


Рис. 1. Пример расположения узлов интерполяции

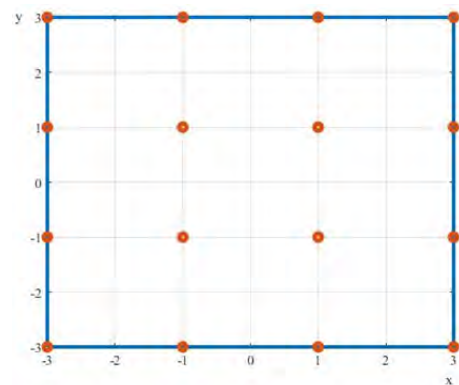


Рис. 2. Начальное расположение центров РБ-функций

Веса иницировались случайными числами, равномерно распределенными от 0 до 0,001. Начальная ширина всех РБ-функций была постоянной, равной 3,0 для методов спуска и NAG и 1,0 – для метода Левенберга – Марквардта. Итерационный процесс обучения продолжался до достижения значения среднеквадратической ошибки (5), равной 0,01.

Для проведения экспериментов разработан комплекс программ в системе MatLab. Эксперименты проводились на компьютере со следующими характеристиками: процессор Intel Core i5-2500K, 3,30 GHz, ОЗУ 8,0 GB. Результаты экспериментов представлены в табл. 1. Так как число итераций и время решения зависят от случайных начальных значений весов, то для каждого метода проводилось 10 экспериментов, и в таблице представлены полу-

ченные диапазоны числа итераций и времени решения. Индексы у коэффициентов означают: 1 – коэффициент для весов, 2 – для центров, 3 – для ширины. Значения коэффициентов подбирались экспериментально.

Таблица 1

Результаты экспериментов

Метод	Стратегия обучения	Параметры	Число итераций	Время решения, с
Градиентный спуск	Одновременная настройка	$\eta_1 = 0,00150$ , $\eta_2 = 0,00100$ , $\eta_3 = 0,00050$	45000–70000	2500–5000
Градиентный спуск	Последовательная настройка	$\eta_1 = 0,05000$ , $\eta_2 = 0,00100$ , $\eta_3 = 0,00050$	35000–50000	2000–3500
Градиентный спуск с импульсом	Одновременная настройка	$\eta_1 = 0,00700$ , $\alpha_1 = 0,9$ , $\eta_2 = 0,00002$ , $\alpha_2 = 0,9$ , $\eta_3 = 0,00020$ , $\alpha_3 = 0,9$	1500–2100	50–70
Градиентный спуск с импульсом	Последовательная настройка	$\eta_1 = 0,00700$ , $\alpha_1 = 0,9$ , $\eta_2 = 0,00002$ , $\alpha_2 = 0,9$ , $\eta_3 = 0,00020$ , $\alpha_3 = 0,9$	200–9000	9–360
NAG	Последовательная настройка	$\eta_1 = 0,00500$ , $\alpha_1 = 0,9$ , $\eta_2 = 0,00200$ , $\alpha_2 = 0,5$ , $\eta_3 = 0,00100$ , $\alpha_3 = 0,3$	270–470	14–24
RMSProp+NAG	Последовательная настройка	$\eta_1 = 0,00100$ , $\eta_2 = 0,00200$ , $\eta_3 = 0,00100$ , $\alpha_1 = 0,90000$ , $\rho_1 = 0,90000$ , $\alpha_2 = 0,50000$ , $\rho_2 = 0,90000$ , $\alpha_3 = 0,10000$ , $\rho_3 = 0,90000$	5700–13400	245–600
Метод Левенберга – Марквардта	Одновременная настройка	$\mu_0 = 0,1$ , $\nu = 10$	6–11	1,77–1,96

Результаты экспериментов показали явное преимущество ускоренных методов первого порядка по сравнению с классическим методом градиентного спуска. Последовательная настройка параметров также дает существенный эффект. Градиентный спуск с импульсом очень чувствителен к случайным начальным значениям весов и параметрам обучения. Процесс изменения среднеквадратической погрешности носит колебательный характер и неустойчив. Наиболее быстрым и устойчивым является метод NAG, он наименее чувствителен к начальным значениям весов и параметрам обучения. Процесс изменения среднеквадратической погрешности носит гладкий характер и устойчив (рис. 3). Экспериментально обнаружена необходимость использования различных значений коэффициентов  $\alpha$  для различных параметров сети. Метод RMSProp+NAG устойчив к изменению начальных значений весов и параметров обучения. Процесс изменения среднеквадратической погрешности носит гладкий характер и устойчив. Но метод проигрывает NAG по числу итераций.

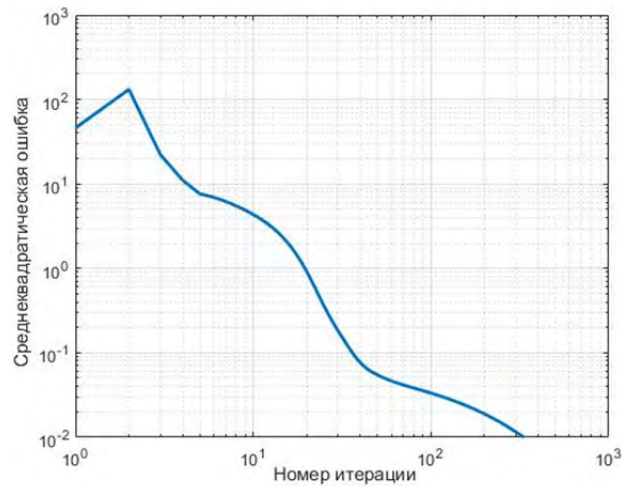


Рис. 3. Зависимость среднеквадратической ошибки от номера итерации в методе NAG

Эксперименты подтвердили важность настройки не только весов, но и параметров РБ-функций. Конечное положение центров РБ-функций, полученное в результате обучения сети (рис. 4), радикально отличается от начального положения (см. рис. 2), причем после обучения сети центры вышли за пределы области решения.

Метод Левенберга – Марквардта обеспечивает качественное преимущество по числу циклов обучения сети по сравнению с методами первого порядка. С помощью метода Левенберга – Марквардта удалось достичь значения среднеквадратической погрешности, равного  $10^{-6}$ , что практически невозможно сделать методами первого порядка. Решение при этом было достигнуто за число циклов обучения, изменявшееся от 23 до 98 (время решения менялось от 2,23 с до 4,65 с).

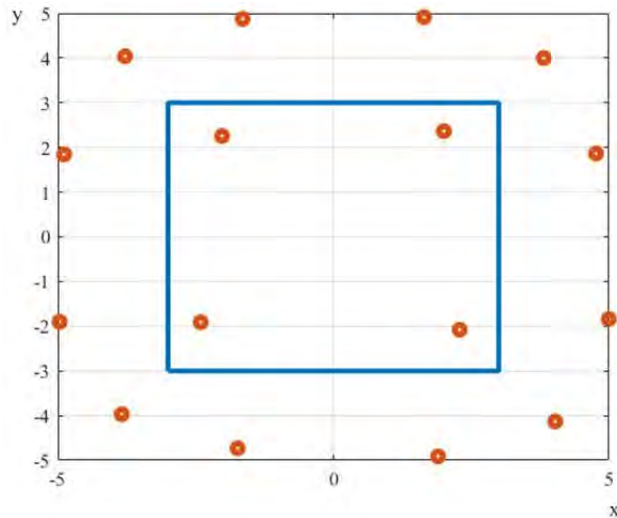


Рис. 4. Пример конечного положения центров РБ-функций

Недостаток метода Левенберга – Марквардта – плохая обусловленность системы (7), зависящая от начальных значений ширины РБ-функций и увеличивающаяся с ростом точности вычислений. Известно, что матрица, элементами которой являются функции Гаусса, является плохо обусловленной и обусловленность матрицы зависит от ширины РБ-функций [16]. С ростом ширины значения РБ-функций (2), являющиеся элементами матрицы  $\mathbf{J}_w$  (11), 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. В пределе матрица  $\mathbf{J}^T \mathbf{J}$  будет содержать  $3n_{RBF}$  нулевых строк и столбцов и становится особенной. Это следует из известного свойства, что определитель матрицы равен нулю, если матрица имеет хотя бы два одинаковых столбца (или две одинаковые строки). Параметр регуляризации  $\mu$  улучшает обусловленность системы (7), но уменьшение параметра  $\mu$  по мере уменьшения погрешности приводит к ухудшению обусловленности. В отличие от обучения многослойного персептрона методом Левенберга – Марквардта, обучение РБФ-сети требует значительно бóльших значений параметра регуляризации  $\mu$ . Так, для обучения многослойного персептрона рекомендуется  $\mu = 0,001$  [17], а обучение РБФ-сети работает даже при  $\mu > 1$ , но при этом процесс изменения среднеквадратической погрешности носит сильно колебательный характер.

При некоторых значениях начальных параметров не удалось решить систему (7) в MatLab из-за практической вырожденности. В частности, систему не всегда удавалось решить при постоянном начальном значении ширины, бóльшем единицы. Например, при начальном расположении центров на сетке, постоянном начальном значении ширины, равном 3,0, и случайных малых начальных значениях весов ранг матрицы  $\mathbf{J}_0^T \mathbf{J}_0$  при начальных значениях параметров сети равен 22, когда размер этой матрицы  $64 \times 64$ , т.е. значительная часть столбцов матрицы линейно зависима. Определитель матрицы

$\mathbf{J}_0^T \mathbf{J}_0 + \mu_k \mathbf{E}$  равен  $2,487968848564625 \cdot 10^{-51}$ , т.е. практически равен нулю. Поэтому зачастую не удавалось решить систему с использованием стандартного решателя MATLAB. Уменьшение начального значения ширины РБ-функций до 1,0 позволило преодолеть этот недостаток. Число обусловленности матрицы  $\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}$  увеличивалось по мере повышения точности решения. Например, число обусловленности матрицы  $\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}$  менялось в экспериментах от  $1,9349 \cdot 10^{+2}$  при начальных значениях сети до  $2,1619 \cdot 10^{+6}$  и при значении среднеквадратической погрешности, равной 0,01 и  $3,2367 \cdot 10^{+8}$ , при среднеквадратической погрешности  $10^{-6}$ . Процесс изменения среднеквадратической погрешности в методе Левенберга – Марквардта носит колебательный характер (рис. 5). Более гладкий характер изменения среднеквадратической погрешности можно обеспечить, уменьшая коэффициент  $\nu$ , но при этом растет число циклов обучения.

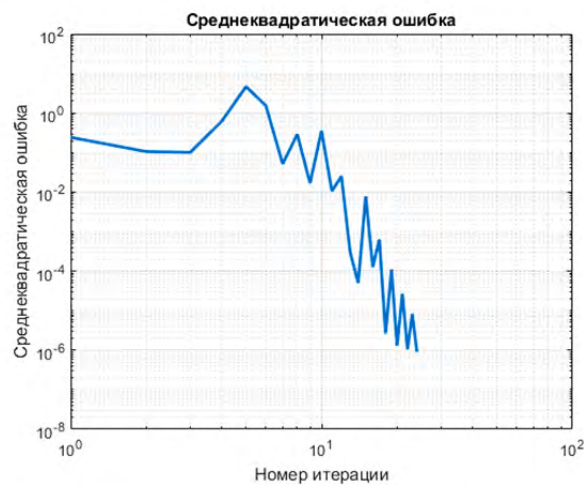


Рис. 5. Зависимость среднеквадратической ошибки от номера итерации в методе Левенберга – Марквардта

### Заключение

Впервые разработаны и исследованы ускоренные алгоритмы первого порядка и Левенберга – Марквардта обучения сетей радиальных базисных функций для решения задач аппроксимации функций. Экспериментальное исследование алгоритмов показало преимущество алгоритма Левенберга – Марквардта и проблемы использования этого алгоритма, связанные с плохой обусловленностью решаемой системы линейных алгебраических уравнений.

Для решения задач аппроксимации на сетях радиальных базисных функций можно рекомендовать адаптированный в работе алгоритм Левенберга – Марквардта, но при этом необходимо оценивать обусловленность решаемой системы. При возникновении проблем с плохой обусловленностью системы можно рекомендовать использовать алгоритм ускоренного градиента Нестерова.

### *Библиографический список*

1. Wendland, H. Scattered Data Approximation / H. Wendland. – Cambridge : Cambridge University Press, 2010. – 348 p.
2. Buhmann, M. D. Radial Basis Functions: Theory and Implementations / M. D. Buhmann. – Cambridge : Cambridge University Press, 2009. – 272 p.
3. Fasshauer, G. F. Meshfree Approximation Methods with MATLAB / G. F. Fasshauer. – New Jersey : World Scientific Publishing Company, 2007. – 520 p.
4. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М. : Вильямс, 2006. – 1104 с.
5. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Горячая линия - Телеком, 2016. – 448 с.
6. 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. 133–143.
7. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – М. : ДМК Пресс, 2017. – 652 с.
8. 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, Issue 4. – P. 819–842.
9. Sutskever, I. On the importance of initialization and momentum in deep learning / I. Sutskever, J. Martens, G. Dahl, G. Hinton // Proceedings of the 30th International Conference on International Conference on Machine Learning. – Vol. 28. – Atlanta, Georgia, 2013. – P. 1139–1147.
10. Bishop, C. M. Neural Networks for Pattern Recognition / C. M. Bishop. – Oxford : Oxford University Press, 1996. – 504 p.
11. Гилл, Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. – М. : Мир, 1985. – 509 с.
12. Fast and Efficient Second-Order Method for Training Radial Basis Function Networks / T. Xie, H. Yu, J. Hewlett, P. Rozycki, B. Wilamowski // IEEE transactions on neural networks and learning systems. – 2012. – Vol. 23, № 4. – P. 609–619.
13. Горбаченко, В. И. Вычислительная линейная алгебра с примерами на MATLAB / В. И. Горбаченко. – СПб. : БХВ-Петербург, 2011. – 320 с.
14. 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.
15. Conn, A. R. Trust regions methods / A. R. Conn, N. I. M. Gould, P. L. Toint. – Philadelphia : Society for Industrial and Applied Mathematics, 2000. – 959 p.
16. 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. – Vol. 61, Issue 4. – P. 443–459.
17. Beale, M. H. Neural Network Toolbox. User's Guide / M. H. Beale, M. T. Hagan, H. B. Demuth. – Natick : MathWorks, Inc., 2017. – 446 p.

---

*Алгезуни Мухи Муртада Мухи*

аспирант,

Пензенский государственный университет

E-mail: mohieit@mail.ru

*Algezweeny Mohie Mortadha Mohie*

postgraduate student,

Penza State University

**Горбаченко Владимир Иванович**  
доктор технических наук, профессор,  
заведующий кафедрой компьютерных  
технологий,  
Пензенский государственный университет  
E-mail: gorvi@mail.ru

**Gorbachenko Vladimir Ivanovich**  
doctor of technical sciences, professor,  
head of sub-department  
of computer technologies,  
Penza State University

---

УДК 004.032.26

**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.**