УДК 519.853.4, 004.032.26

Использование нейросетевой аппроксимации целевой функции при решении задач глобальной оптимизации*

Карпенко С. Н., Лебедев И. Г., Надумин Д. В.

ННГУ им. Н.И. Лобачевского

Аннотация: В работе рассматривается задача поиска глобального минимума многоэкстремальной функции. Для решения этой задачи был выбран алгоритм имитации отжига. Для ускорения процесса поиска минимума используется аппроксимация целевой функции нейросетью. Представлены результаты работы комбинированного алгоритма глобальной оптимизации, объединяющего алгоритм имитации отжига, использующий нейронную сеть для аппроксимации функций, и метод Нелдера-Мида для точного решения задачи оптимизации. В данной работе проводились эксперименты с серией двумерных функций из генератора GKLS. Результаты показывают сокращение количество вычислений целевой функции при сохранении высокой точности поиска.

Ключевые слова: методы оптимизации, глобальная оптимизация, нейронные сети, машинное обучение.

1. Введение

Нейронная сеть представляет собой математическую модель, построенную по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы.

С точки зрения математики, обучение нейронных сетей – это многопараметрическая задача нелинейной оптимизации. С точки зрения развития вычислительной техники и программирования, нейронная сеть – способ решения проблемы эффективного параллелизма.

Нейронные сети применяются для решения многих прикладных задач. Наиболее известными сферами применений нейронных сетей являются: распознавание образов и классификация, прогнозирование, аппроксимация. Нейронные сети могут аппроксимировать непрерывные функции: с помощью линейных операций и каскадного соединения можно из произвольного нелинейного элемента получить устройство, вычисляющее любую непрерывную функцию с некоторой наперёд заданной точностью.

История нейронных сетей намного длиннее, чем принято считать. Сама идея «способной к мышлению системы» возникла еще в Древней Греции, и популярность нейронных сетей менялась с течением времени.

В 1943 году Уоррен Маккалок и Уолтер Питтс опубликовали работу «Логическое исчисление идей, относящихся к нервной деятельности» [1]. Целью данного исследования было изучение работы человеческого мозга. В 1958 году, Фрэнк Розенблатт

*Работа выполнена при поддержке Министерства науки и высшего образования РФ (проект № 0729-2020-0055) и научно-образовательного математического центра «Математика технологий будущего» (проект № 075-02-2021-1394).

в своем исследовании «Персептрон: вероятностная модель хранения и организации информации в головном мозге» [2] описал модель персептрона. В 1974 году, первым ученым на территории США, описавшим в своей диссертации [3] использование алгоритма обратного распространения ошибки в нейронных сетях, стал Пол Вербос, хотя развитием этой идеи занимались многие исследователи. В 1989 году, Янн Лекун опубликовал статью [4], в которой было описано практическое использование ограничений обратного распространения ошибки и интеграция в архитектуру нейронной сети для обучения алгоритмов.

2. Постановка задачи глобальной оптимизации

Задача многомерной многоэкстремальной оптимизации может быть определена как проблема поиска наименьшего значения действительной функции $\phi(y)$

$$\phi(y^*) = \min\{\phi(y) : y \in D\},$$
$$D = \{y \in R^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N\}, \tag{1}$$

где $a, b \in R^N$ – заданные векторы.

Численное решение задачи (1) заключается в построении оценки $y_k^* \in D$, отвечающей некоторому понятию близости к точке y^* (например, $||y^* - y_k^*|| \leq \epsilon$, где $\epsilon > 0$ это некоторая заданная точность) на основе конечного числа k вычислений оптимизируемой функции. Относительно класса рассматриваемых задач предполагается выполнение двух важных условий:

1) целевая функция $\phi(y)$ может быть задана алгоритмически, как результат работы некоторой подпрограммы или библиотеки;

2) целевая функция $\phi(y)$ удовлетворяет условию Липшица

$$|\phi(y_1) - \phi(y_2)| \leq L||y_1 - y_2||,\ y_1, y_2 \in D,\ 0 < L < \infty. \tag{2}$$

Второе предположение соответствует ограниченности изменения значений функции при ограниченной вариации аргумента. Это предположение можно интерпретировать (применительно к прикладным задачам) как отражение ограниченности мощностей, порождающих изменения в моделируемой системе.

Задачи многоэкстремальной оптимизации имеют существенно более высокую трудоемкость решения по сравнению с другими типами оптимизационных задач, т. к. глобальный оптимум является интегральной характеристикой решаемой задачи и требует исследования всей области поиска. Как результат, поиск глобального оптимума сводится к построению некоторого покрытия (сетки) в области параметров, и выборе наилучшего значения функции на данной сетке. Вычислительные затраты на решение задачи растут экспоненциально с ростом размерности [5–8].

3. Метод имитации отжига

Алгоритм имитации отжига является общим алгоритмическим методом решения задачи глобальной оптимизации. Он основывается на имитации физического процесса, который происходит при кристаллизации вещества, в том числе при отжиге металлов.

Введём следующие обозначения: Y – множество всех состояний системы, y_i – состояние на i-ом шаге алгоритма; t_i – температура на i-ом шаге.

При помощи моделирования такого процесса ищется такая точка или множество точек, на котором достигается минимум некоторой числовой функции $\phi(y^i), y^i = (y_1, ...y_n) \in Y$. Решение ищется последовательным вычислением точек пространства Y [6], [9].

Определим три функции:

1. Функция энергии (оптимизируемая): $E : Y \to R$.

2. Функция изменения температуры с течением времени: $T : N \to R$.

3. Функция, порождающая новое состояние: $\phi : Y \to Y$.

На входе задаются минимальная t_{min} и начальная t_{max} температуры. Задаём произвольное первое состояние y_1 и $t_{min} = t_{max}$.

Пока $t_i > t_{min}$:

1) $y_c = \phi(y_{i-1})$,

2) $\triangle E = E(y_c) - E(y_{i-1})$,

3) Если $\triangle E \leq 0$, то $y_i = y_c$. Если $\triangle E > 0$, то переход осуществляется с вероятностью $P(\triangle E) = e^{-\triangle E/t_i}$,

4) Понижаем температуру $t_{i+1} = T(i)$.

Возвращаем последнее состояние y.

Особенностью метода имитации отжига является возможность выхода из локального минимума оптимизируемой функции за счет принятия изменений, временно ухудшающих результат, что отражает суть процесса нагрева расплава для предотвращения его быстрого остывания. Еще одно преимущество – даже в условиях нехватки вычислительных ресурсов для нахождения глобального минимума метод отжига выдает один из локальных минимумов.

4. Метод Нелдера-Мида

Метод Нелдера-Мида является методом безусловной оптимизации [10, 11] функции нескольких переменных $\phi(y^1, ..., y^n)$, не использующий градиентов функции. Он относится к группе методов прямого поиска, так как в нем не используются производные. Метод является развитием симплексного метода.

Идея метода заключается в следующем: в пространстве R^N выбираются $(n+1)$ пробных векторов, в каждом из которых можно вычислить значение целевой функции. Вершина A с наихудшим значение целевой функции отражается через центр тяжести других вершин многогранника, в результате чего из оставшихся вершин строится новый многогранник, называемый отраженным, и одна точка, расположенная на линии проекции на соответствующем расстоянии от центра тяжести. При этом предполагается что функция определена во всех точках области определения.

Метод Нелдера-Мида относится к методам локальной оптимизации, т. е. он сходится, вообще говоря, к локальному экстремуму функции.

5. Математический аппарат нейронной сети прямого распространения

Согласно универсальной теореме аппроксимации (теореме Цыбенко), доказанной Джорджем Цыбенко в 1989 году, искусственная нейронная сеть с прямым распространением может аппроксимировать любую непрерывную функцию многих переменных с любой точностью [12] при условии достаточного количества нейронов скрытого слоя и удачного подбора весов и смещений.

На входе нейронная сеть принимает набор значений переменной. Далее эти значения умножаются на некоторые весовые коэффициенты. После чего в качестве взвешенной по весовым коэффициентам суммы подаются на вход каждого из нейронов скрытого слоя, где к этой сумме прибавляется смещение (*bias*). В результате получаем число, которое подаётся на вход активационной функции. Результат активационной функции и есть выходные значение одного нейрона.

Взвешенная сумма представлена ниже:

$$z = \sum_{i=1}^{n} y_i \cdot w_i + bias, \tag{3}$$

где $bias$ – смещение, w_i – весовой коэффициент, y_i – входное значение.

Далее (3) подаётся в качестве входного значения в активационную функцию. Существует несколько видов активационных функций. В данной работе применялась сигмоидальная функции, область значений которой от 0 до 1.

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{4}$$

6. Комбинированный алгоритм глобальной оптимизации

Нами разработан комбинированный алгоритм глобальной оптимизации, использующий нейронную сеть для аппроксимации функций, метод имитации отжига для поиска глобального оптимума и метод Нелдера-Мида для уточнения найденного решения.

В данной работе была выбрана нейронная сеть прямого распространения (feedforward neural network) [1–4, 12] из библиотеки Keras [13].

В ходе исследования параметры нейронной сети были подобраны следующим образом: 5 слоёв по 32 нейрона в каждом слое. В качестве активационной функции выбрана сигмоида. Весовые коэффициенты имеют нормальное распределение со средним равным 0 и стандартным отклонением равным 0.2, что позволяет задавать их значения в диапазоне $[-1, 1]$. Смещение нейрона (*bias*) инициализируется как 0. В выходном слое только один нейрон.

Сам комбинированный алгоритм можно разбить на три пункта:

1. Создание обучающей выборки для нейронной сети.
2. Обучение нейронной сети.
3. Пока не выполнен критерий остановки, либо пока не выполнено ограничение на количество вычислений целевой функции повторяем следующее:

a) Оптимизация нейросетевой аппроксимации с помощью алгоритма имитации отжига.

b) Оптимизация целевой функции с помощью локального метода, где в качестве первоначального значения используется найденная глобальным методом точка.

c) Если разница между найденным значением и глобальным минимумом меньше 0.01, то комбинированный алгоритм сошёлся.

d) Иначе добавляем в обучающую выборку 10 случайных точек и найденную локальным методом точку.

e) После чего, заново обучаем нейронную сеть и уходим на следующую итерацию.

7. Описание программной реализации

Программная реализация выполнена на языке Python с помощью библиотек Keras [13], Numpy [14], Matplotlib [15], SciPy [16].

Работа выполнялась в среде Jupyter Notebook. Программа разбита на 5 блоков:

1. Импортируются все библиотеки, в том числе пользовательские функции (в частности, функция нормализации значений) и генератор задач GKLS.

2. Определена функция задания нейросети, принимающая в качестве аргумента размерность задачи. Функция возвращает объект нейронной сети.

3. Задаётся обучающая выборка, состоящая из 225 значений в области определения функций генератора GKLS.

4. Задаются параметры нейросети и критерии остановки комбинированного алгоритма. Количество эпох обучения равно 6000. Размер выборки для обучения (batch size) на каждой эпохе (каждой итерации) равен одной трети от всего размера обучающей выборки.

5. Выполнение комбинированного алгоритма. В случае повторения обучения нейросети batch size также задаётся равным одной трети от всей выборки с учётом добавленных точек.

8. Эксперименты

Эксперименты проводились на компьютере со следующими характеристиками:

1. Процессор Intel Core i3-9100F, 4100 MHz.

2. ОЗУ 16 Gb (2x8 Gb) DDR4.

3. Видеокарта GeForce GTX 1050 2 Gb.

Задача экспериментов состояла в том, чтобы уменьшить количество вычислений целевой функции.

На рис. 1 представлена поверхность одной из функций, порожденных генератором GKLS, а на рис. 2 – её аппроксимация нейросетью.

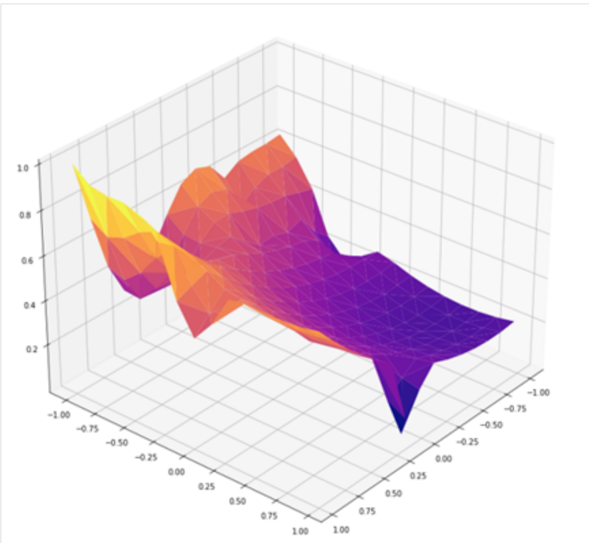

Рис. 1. Поверхность функции.

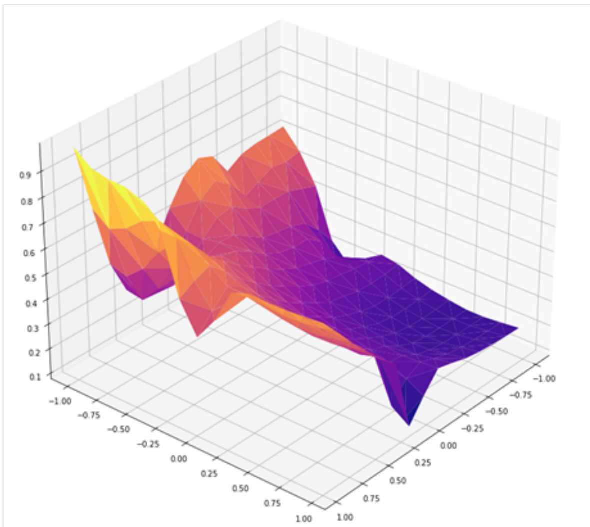

Рис. 2. Аппроксимация.

Эксперименты проводились на 10-ти двумерных задачах из генератора GKLS. В работах [6, 10] описан GKLS-генератор, позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами: количеством локальных минимумов, размерами их областей притяжения, точкой глобального минимума, значением функции в ней и т. п.

В таблице 1 отражено среднее число вычислений целевой функции, которые выполнили метод имитации отжига и разработанный нами комбинированный алгоритм, при решении серии задач. Также приведена средняя точность нахождения решения. Максимальное число итераций алгоритма имитации отжига составляло 10000, остальные параметры соответствуют значениям по умолчанию.

Таблица 1. Результаты работы имитации отжига и комбинированного алгоритма

	Среднее количество вычислений целевой функции	Точность нахождения решений
Имитация отжига	4255.9	$1.8 \cdot 10^{-8}$
Комбинированный алгоритм	293.4	$3.3 \cdot 10^{-5}$

Результаты экспериментов показывают значительное уменьшение количества вычислений целевой функции. При этом сохраняется высокая точность нахождения минимума.

Литература

1. Warren S. McCulloch, Walter Pitts // California State University. Режим доступа: `https://home.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf`
2. F. Rosenblatt // Cornell Aeronautical Laboratory. Режим доступа: `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1type=pdf`
3. Paul Werbos // National Science Foundation. Режим доступа: `https://www.researchgate.net/publication/35657389_Beyond_regression_new_tools_for_prediction_and_analysis_in_the_behavioral_sciences`
4. Y. LeCun, B. Boser // AT & T Bell Laboratories. Режим доступа: `http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf`
5. Гермейер Ю.Б. Введение в теорию исследования операций. М.: «Наука», 1971. 384 с.
6. Сергеев Я.Д., Квасов Д.Е. Диагональные методы глобальной оптимизации. М.: Физматлит, 2008. 352 с.
7. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. М.: «Наука», 1978. 240 с.

8. Стронгин Р.Г., Гергель В.П., Гришагин В.А., Баркалов К.А. Параллельные вычисления в задачах глобальной оптимизации. М.: Издательство Московского университета, 2013. 280 с.

9. Kirkpatrick S., Gelatt Jr.C.D., Vecchi M.P. Optimization by Simulated Annealing // Science. 1983. 220 (4598). P. 671–680.

10. Gaviano M., Lera D., Kvasov D.E., Sergeyev Y.D. Software for generation of classes of test functions with known local and global minima for global optimization // ACM Transactions on Mathematical Software. 2003. Vol. 29. P. 469-480.

11. Городецкий С.Ю., Гришагин В.А. Нелинейное программирование и многоэкстремальная оптимизация: учебное пособие. Нижний Новгород: Изд-во Нижегородского госун-та, 2007. 489 с.

12. Cybenko G.V. Approximation by Superpositions of a Sigmoidal function // Mathematics of Control Signals and Systems. 1989. Т. 2, № 4. С. 303—314.

13. Официальный сайт библиотеки Keras. Режим доступа: `https://keras.io/`

14. Официальный сайт библиотеки NumPy. Режим доступа: `https://numpy.org/`

15. Официальный сайт библиотеки Matplotlib. Режим доступа: `https://matplotlib.org/`

16. Официальный сайт библиотеки SciPy. Режим доступа: `https://scipy.org/`

MSC 90C26, 82C32

Using neural network approximation of the objective function in global optimization problems

S. N. Karpenko, I. G. Lebedev, D. V. Nadumin

N. I. Lobachevsky State University of Nizhny Novgorod

Abstract: The paper considers the problem of finding the global minimum of a multiextremal function. To solve this problem, the simulation annealing algorithm was used. To speed up the process of finding the minimum, the approximation of the objective function by a neural network is applied. The results of algorithm that combines the simulation annealing algorithm using a neural network to approximate objective function and the Nelder-Meade method to refine the solution of the optimization problem are presented. The paper presentS the results of the experiments carried out with a series of multiextremal test problems. The results show a reduction in the number of calculations of the objective function while maintaining high search accuracy.

Keywords: optimization methods, global optimization, neural networks, machine learning.

References

1. Warren S. McCulloch, Walter Pitts, California State University. Access mode: `https://home.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf`
2. F. Rosenblatt, Cornell Aeronautical Laboratory. Access mode: `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1type=pdf`
3. Paul Werbos, National Science Foundation. Access mode: `https://www.researchgate.net/publication/35657389_Beyond_regression_new_tools_for_prediction_and_analysis_in_the_behavioral_sciences`
4. Y. LeCun, B. Boser, AT & T Bell Laboratories. Access mode: `http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf`
5. Yu.B. Hermeyer, Introduction to the theory of operations research, M., "Science 1971, 384 p.
6. Ya.D. Sergeev, D.E. Kvasov, Diagonal methods of global optimization, M., Fizmatlit, 2008, 352 p.
7. R.G. Strongin, Numerical methods in multi–extreme problems, M., "Science 1978, 240 p.
8. R.G. Strongin, V.P. Gergel, V.A. Grishagin, K.A. Barkalov, Parallel computing in global optimization problems, M., Moscow University Press, 2013, 280 p.
9. S. Kirkpatrick, Jr.C.D. Gelatt, M.P.Vecchi, Optimization by Simulated Annealing, *Science*, 1983, 220 (4598), P. 671–680.

10. M. Gaviano, D. Lera, D.E. Kvasov, Y.D. Sergeyev, Software for generation of classes of test functions with known local and global minima for global optimization, *ACM Transactions on Mathematical Software*, 2003, Vol. 29, P. 469-480.

11. S.Y. Gorodetsky, V. A. Grishagin, Nonlinear programming and multi-extreme optimization: a textbook, Nizhny Novgorod, Publishing House of the Nizhny Novgorod State University, 2007, 489 p.

12. G.V. Cybenko, Approximation by Superpositions of a Sigmoidal function. *Mathematics of Control Signals and Systems*, 1989, v. 2, № 4, P. 303—314.

13. The official website of the Keras Library. Access mode: `https://keras.io/`

14. The official website of the NumPy library. Access mode: `https://numpy.org/`

15. The official website of the Matplotlib library. Access mode: `https://matplotlib.org/`

16. The official website of the SciPy library. Access mode: `https://scipy.org/`