© 2024 г. Ю.С. ПОПКОВ, д-р техн. наук (popkov@isa.ru)
(Федеральный исследовательский центр
«Информатика и управление» РАН, Москва;
Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ИТЕРАЦИОННЫЕ МЕТОДЫ С САМООБУЧЕНИЕМ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Рассматривается задача решения системы нелинейных уравнений с произвольной, но непрерывной вектор-функцией в левой части, о которой можно иметь только значения ее компонент. Для определения приближенного решения используется какой-нибудь итерационный метод с параметрами, качественные свойства которого оцениваются квадратичным функционалом невязки. Предлагается самообучающаяся процедура (подкрепления), основанная на вспомогательных МК-испытаниях, на функции полезности экспоненциального класса и функции выигрыша, реализующей принцип оптимальности Беллмана. Доказана теорема о строгом монотонном убывании функционала невязки.

Ключевые слова: нелинейные уравнение, итерационные методы, подкрепление, Монте-Карло.

DOI: 10.31857/S0005231024050058, **EDN:** YQBWNO

1. Введение

Подавляющее большинство прикладных задач сводятся к необходимости решать нелинейные уравнения. Параметризованные итерационные методы являются классическим инструментом, позволяющим при определенных условиях получать приближенные решения [1–4] нелинейных уравнений. Этими условиями являются определенные свойства функций (выпуклость, вогнутость, дифференцируемость и др.), входящих в уравнения, и интервальные *достаточные* параметрические условия, при которых обеспечивается сходимость соответствующего итерационного метода.

Возрастающая сложность функций сужает множество классов тех из них, для которых удается проверять эти свойства и использовать результаты проверки в подходящих итерационных методах. Что касается интервальных условий на параметры итерационных методов, то они существенно зависят от свойств функций, которые в общем случае непроверяемы.

Для выхода из этой ситуации предлагается использовать идеи машинного обучения с подкреплением (reinforcement learning), примененных к итерационному вычислительному процессу, а именно к определению значений его параметров с помощью статистического МК-эксперимента и игровой математической модели. Суть этой ветви машинного обучения состоит в том,

чтобы обучать объект (модель, алгоритм, и т.д.) путем взаимодействия не с «учителем», а со «средой», используя метод проб и ошибок, сопровождаемый награждением или наказанием его результатов.

Этот подход применялся к задачам кластеризации и распознавания, по-видимому, потому, что в них удалось вычислять так называемые признаковые характеристики в виде «расстояний» между объектами. Именно на базе матрицы расстояний устраивались некие «поощрения» или «наказания» в настройках параметров алгоритма. В качестве последнего использовались нейронные сети [7] и игровые модели, реализующие принцип конкуренции узлов нейронной сети: преимуществом обладали узлы, для которых расстояние между объектами на каждом шаге алгоритма оказывалось минимальным [8].

В дальнейшем обучение с подкреплением, базирующееся на автоматных моделях взаимодействия объекта (агента) с окружением (средой), имитировалось в игровых терминах (стратегии, функции полезности, выигрыш, проигрыш) и активно развивалось [9]. Появилось множество алгоритмов, которые различались моделями и объемами априорных сведений об окружении (среде), алгоритмами выбора стратегий и процедурами формирования функций полезности [10–13].

Важной компонентой процедур обучения с подкреплением являются МК-испытания, с помощью которых имитируются стратегии агента [14]. Они используются для осреднения фиксированного количества текущих наград с учетом их дисконтирования. Полученная таким образом функция, зависящая от состояния среды и стратегии агента, принималась за функцию полезности (аналог целевой функции в процедурах обучения с учителем), которая в процессе обучения последовательно максимизировалась [15, 16], с использованием принципа оптимальности Беллмана [17] в сочетании с методом стохастической аппроксимации [18].

В данной работе рассматривается задачи численного решения системы нелинейных уравнений с помощью параметризованной итерационной процедуры, сходимость которой зависит от значений этих параметров. Для определения последних развивается обучающая процедура с подкреплением, основанная на игровой модели.

2. Постановка задачи

Рассмотрим нелинейное уравнение в виде

$$\mathbf{f}(\mathbf{x}) = \mathbf{1}, \quad (\mathbf{f}, \mathbf{x}, \mathbf{1}) \in R^n. \tag{2.1}$$

Про функцию $\mathbf{f}$ известно, что доступны только значения ее компонент $f_i(\mathbf{x}^{(k)})$, $i = \overline{1,n}$; $k = 1, \ldots$.

Определим функционал невязки в следующем виде:

$$J(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{1}\|^2 \geqslant 0. \tag{2.2}$$

Абсолютный минимум этого функционала равен нулю. В общем случае он не единственный, т.е. существует конечное множество точек $\mathbb{X} = \{\mathbf{x}_*^{(1)}, \ldots, \mathbf{x}_*^{(r)}\}$, в которых функционал невязки равен нулю. В этой ситуации будем считать подходящим любое решение из множества $\mathbb{X}$.

Поиск приближенного решение указанного уравнения осуществляется итерационной процедурой марковского типа. В ней приближенное решение $\mathbf{x}^{(p+1)}$ на $(p+1)$-м шаге процедуры приравнивается значению оператора $\mathcal{B}[\mathbf{x}^{(p)}, \mathbf{a}^{(p)}]$ итерационной процедуры на p-м шаге, зависящего от значений компонент функции $\mathbf{f}(\mathbf{x}^{(p)})$ и вектора параметров этой процедуры $\mathbf{a}^{(p)} \in \mathcal{A} \subset R^r$, управляющих качественными свойствами итерационного процесса:

$$\mathbf{x}^{(p+1)} = \mathcal{B}[\mathbf{f}(\mathbf{x}^{(p)}), \mathbf{a}^{(p)}]. \tag{2.3}$$

Под качественными свойствами обычно понимается *сходимость*, *скорость сходимости*, *точность*. Условия их выполнения формулируются в терминах вектора $\mathbf{a}$ и интервальных неравенств, зависящих от свойств оператора $\mathcal{B}$ и от свойств функции $\mathbf{f}(\mathbf{x})$.

Однако поскольку функция $\mathbf{f}(\mathbf{x})$ может иметь произвольную структуру, которая не позволяет постулировать или обнаруживать какие-либо ее свойства, то аналитическая проверка этих неравенств оказывается невозможной.

Для преодоления этой ситуации воспользуемся идеями *подкрепления*, активно применяемыми в различных конкретных реализациях в современных процедурах машинного обучения. В данном случае подкрепление предназначено для определения на каждом шаге итерационной процедуры подходящего вектора параметров $\mathbf{a}$ путем соответствующей самообучающейся процедуры.

Предлагается для определения подходящих параметров $\mathbf{a}$ итерационной процедуры (2.3) игровая модель, функционирующая в интервалах между p-м и $(p+1)$-м шагами, которая имитирует поведение *агента* – стратегию изменения параметров $\mathbf{a}$ в зависимости от качества реакции *среды*. Последнее характеризуется условным *выигрышем* – функцией, зависящий от значений функционала невязки и его декремента.

3. Структура процедуры подкрепления

Рассмотрим исходную задачу (2.1), решение которой будем искать путем минимизации функционала невязки

$$J(\mathbf{x}) \Rightarrow \min, \quad \mathbf{x} \in R^n. \tag{3.1}$$

В некоторых задачах может оказаться полезным преобразование задачи (2.1). Введем новые переменные

$$z_i = \frac{1}{1 + \exp(-b_i x_i)}, \quad x_i = \frac{1}{b_i} \ln \frac{z_i}{1 - z_i}, \quad i = \overline{1, n}.$$

Тогда задача (2.1) приобретает следующий вид:

$$J(\mathbf{z}) = \|\mathbf{\Psi}(\mathbf{z})\| \Rightarrow \min,\ \mathbf{z} \in Z_+^n = [\mathbf{0}, \mathbf{1}], \quad \mathbf{\Psi}(\mathbf{z}) = \mathbf{f}(\mathbf{z}) - \mathbf{1}.$$

Решение задачи (2.1) будем искать, воспользовавшись итерационной процедурой (2.3), полагая, что параметры $\mathbf{a}$ интервального типа: $\mathbf{a} \in [\mathbf{a}^-, \mathbf{a}^+]$.

Для определения значений компонент вектора $\mathbf{a}$ в процедуре (2.3) воспользуемся технологией *подкрепления*, реализуя ее в интервале между p-м и $(p+1)$-м шагами итерационной процедуры (2.3).

Основу этой технологии составляет игровая модель, в которой имитируется игра *агента* со *средой*. Агент генерирует *стратегии* (действия), приводящие к изменениям среды. Ценность этих изменений характеризуется *функцией полезности*. От успешности стратегии агента и полезности ее для среды зависит значение *функции выигрыша*.

В промежутке между шагами итерационной процедуры производится статистический имитационный эксперимент посредством заданного количества Монте-Карло (МК-) испытаний M, которые имитируют стратегии агента, т.е. значения компонент вектора $\mathbf{a}^{(p,k)}$, где $k = \overline{1, M}$.

В качестве действий агента будем рассматривать вектор $\mathbf{x}^{(p,k+1)}$, который имеет вид

$$\mathbf{x}^{(p,k+1)} = \mathcal{B}[\mathbf{f}(\mathbf{x}^{(p,k)}), \mathbf{a}^{(p,k)}], \quad p = \text{fix},\ k = \overline{1, M}. \tag{3.2}$$

Средой в этой задаче является функционал невязки $J(\mathbf{x} \,|\, \mathbf{a})$. В результате МК-имитируемых действий агента возникает последовательность из M невязок

$$J(\mathbf{x}^{(p,1)} \,|\, \mathbf{a}^{(p,1)}), \ldots, J(\mathbf{x}^{(p,M)} \,|\, \mathbf{a}^{(p,M)}) \tag{3.3}$$

и их декрементов

$$u^{(p,k)}(\mathbf{a}^{(p,k)}) = J(\mathbf{x}^{(p,k+1)} \,|\, \mathbf{a}^{(p,k)}) - J(\mathbf{x}^{(p,k)} \,|\, \mathbf{a}^{(p,k-1)}), \quad k = \overline{1, M}. \tag{3.4}$$

Введем *функцию полезности*, которая характеризует качество реакции среды, измеряемое величиной декремента:

$$\varphi(u^{(p,k)}(\mathbf{a}^{(p,k)})) = \alpha\ \exp[u^{(p,k)}(\mathbf{a}^{(p,k)})]. \tag{3.5}$$

Качество стратегий агента оценивается в терминах *выигрыша*, т.е. соответствующей функции, которая характеризует зависимость величины выигрыша от стратегии агента. Выбор подходящей функции выигрыша представляется творческой задачей [2], связанной с некоторым перебором. Некоторые общие свойства этой функции можно декларировать. Это – непрерывная, ограниченная функция следующего вида:

$$Q(\mathbf{a}^{(p,k)}) = \begin{cases} l(u^{(p,k)}(\mathbf{a}^{(p,k)})), & \varphi(u^{(p,k)}(\mathbf{a}^{(p,k)})) \leqslant 1 \\ 0, & \varphi(u^{(p,k)}(\mathbf{a}^{(p,k)})) > 1. \end{cases} \tag{3.6}$$

Здесь функция

$$l(u^{(p,k)}(\mathbf{a}^{(p,k)})) = \begin{cases} \alpha\,\varphi(u^{(p,k)}(\mathbf{a}^{(p,k)})), & 0 \geqslant u^{(p,k)}(\mathbf{a}^{(p,k)}) \geqslant -U, \\ 0, & u^{(p,k)}(\mathbf{a}^{(p,k)}) \geqslant 0, \end{cases} \tag{3.7}$$

где U – предельное значение модуля декремента.

В результате применения МК-испытаний получаем набор значений функций выигрыша. По концепции подкрепления применительно к итерационной процедуре (2.3) оптимальное значение параметра $\mathbf{a}^{(p+1)}$ определяется по следующему правилу:

$$\mathbf{a}^{(p+1)} = \mathbf{a}^{(p)} + \beta\ \arg \max_{1 \leqslant j \leqslant M} Q(\mathbf{a}^{(p,k_j)}). \tag{3.8}$$

Если агент выбирает стратегию по правилу (3.8), то учитывая (3.6), будем иметь:

$$J(\mathbf{a}^{(p+1)}) < J(\mathbf{a}^{(p)}). \tag{3.9}$$

Таким образом, доказана следующая теорема о свойствах последовательности невязок при использовании итерационной процедуры с подкреплением (3.5)–(3.8)

Теорема 1. Пусть:

а) *для функции* $\mathbf{f}(\mathbf{x})$ *в* (2.1) *доступны только значения ее компонент* $f_i(\mathbf{x}^{(k)})$, $i = \overline{1,n}$;

б) *параметры итерационной процедуры* $\mathbf{a}$ *выбираются по правилу* (3.9), (3.8), (3.6).

Тогда итерационная процедура (3.2) *с подкреплением* (3.5)–(3.8) *генерируют строго монотонно убывающую последовательность функционалов невязки* $J(\mathbf{x})$ (3.1).

Данная теорема не есть теорема сходимости итерационной процедуры в математическом смысле, т.е. сходимости к одному из решений. Однако известно, что этому решению соответствует нулевое значение невязки. Теорема утверждает, что последовательность невязок является строго монотонно убывающей. Поэтому с учетом того, что погрешность вычислений конечна и может быть задана, полученное при ее достижении значение параметров $\mathbf{a}$ может быть принято за решение.

4. Заключение

Рассмотрена задача поиска решения системы нелинейных уравнений с непрерывными функциями в левых частях. Доступной априорной информацией об этих функциях являются только их значения. Для поиска решений при таких условиях используется итерационная процедура с параметрами, с

помощью управления значениями которых можно обеспечить ее сходимость в каком-либо смысле.

Предлагается использовать идеи подкрепления, достаточно активно развиваемые в теории и практике машинного обучения. Разработана самообучающаяся процедура, в которой на каждом шаге итераций производится заданное количество МК-испытаний, имитирующих стратегии агента, которыми в данном случае являются значения параметров итерационной процедуры. Функции среды в данной процедуре выполняет функционал невязки (3.2), а его реакцией на действия агента является декремент (3.3) этого функционала. Для приемлемого течения итерационного процесса необходимо, чтобы декремент уменьшался. Величина декремента характеризуется функцией полезности экспоненциального типа, в терминах которой меньшим (с учетом знака) значениям декремента соответствуют большие значения функции полезности. Действия агента, т.е. реализованные параметры итерационной процедуры, оцениваются функцией выигрыша, морфология которой учитывает как состояние среды, так и степень успешности действий агента.

Доказано, что в результате применения указанной процедуры самообучения итерационный алгоритм с подкреплением генерирует строго монотонна убывающую последовательность функционалов невязки.

СПИСОК ЛИТЕРАТУРЫ

1. *Красносельский М.А., Вайникко Г.М., Забрейко П.П. и др.* Приближенные решения операторных уравнений. М.: Наука, 1969.
2. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.* Численные методы. М.: Бином, 2003.
3. *Поляк Б.Т.* Введение в оптимизацию. М.: Наука, 1983.
4. *Стрекаловский А.С.* Элементы невыпуклой оптимизации. Новосибирск, Наука, 2003.
5. *Lyle C., Rowland M., Dabney W., Kwiatkowska M., Gal Y.* Learning dynamics and generalization in deep reinforcement learning // Int. Conf. on Machin. Learning. PMLR. 2022. P. 14560–14581.
6. *Che Wang, Shushan Yaun, Keit W. Ross.* On the Convergence of the Monte Carlo Exploring Starts Algorithm for Reinforcement Learning. ICLR. 2022.
7. *Уоссерман Ф.* Нейрокомпьютерная техника. Теория и практика. М.: Мир, 1992.
8. *Kohonen T.* Self-organizing Maps. Springer Berlin, Heidelberg, 1995.
9. *Mnih V., Kavukcuoglu K., Silver D., Rusu A.A., Veness J., Bellemare M.G., Graves A., Riedmiller M., FIdjeland A.* Human-level control through deep reinforcement learning // Nature. 2015. Vol. 518. No. 7540. P. 529–533.
10. *Sutton R.S., Barto A.G.* Introduction to reinforcement Learning. Cambridge, MIT press, 1998.
11. *Russel S.J., Norvig P.* Artificial Intelligemce: A Modern Approach (Third Ed.) Prentice Hall, Upper Saddle River, 2010.

12. *van Hasselt H.* Reinforcement Learning in Continuous State and Action Spaces. In: Wiering M., van Otterio M.(eds.) Reinforcement Learning: State-of-the-Art, 2012. Springer Sciences & Business Media, P. 207–257.

13. *Ivanov S.* Reinforcement Learning Textbook // ArXiv, 2022. https://doi.org/10.48550/arXiv.2201.09746

14. *Bozinovski S.* Crossbar Adaptive Array: The first connectionist network that solved the delayed reinforcement learning problem. In: Dobnikar A., Steele N.C., Pearson D.W., Albrecht R.F. (eds.) Artificial Neural Nets and Genetic Algorithms // Proc. Int. Conf. Portoroz, Slovenia, Springer Science & Business Media, 1999, P. 320–325.

15. *Watkins C., Dayan P.* Q-learning // Machine Learning. 1992. Vol. 8. No. 3–4. P. 279–292.

16. *van Hasselt H., Guez A., Silver D.* Deep reinforcement learning with double Q-learning // Proc. AAAI Conf. Artificial Intelligence. 2016. Vol. 30. No. 1. P. 2094–2100.

17. *Bellman R.* Dynamic Programming. Princeton University Press, 1957.

18. *Robbins H., Monro S.* A stochastic approximation method // The Annals of Mathematical Statistics. 1951. P. 400–407.

Статья представлена к публикации членом редколлегии Д.В. Виноградовым.

Поступила в редакцию 10.01.2024

После доработки 21.03.2024

Принята к публикации 30.03.2024