HSBI Bielefeld

University of Applied Sciences

Fachbereich Ingenieurwissenschaften und Mathematik

Studiengang Optimierung und Simulation

Lösen von nichtlinearen Gleichungssystemen mit einem Reinforcement-Learning-Agent

Bericht

Vorgelegt von:	Nicolas Schneider
Matrikelnummer:	1208960
Studiengang:	Optimierung und Simulation
Abgabedatum:	07.04.2024
Betreuer:	Prof. Dr. rer. nat. Bernhard Bachmann

Abstract

Nichtlineare Gleichungssysteme (NGS) spielen eine zentrale Rolle in vielen Bereichen der Wissenschaft und Technik, da sie zur Modellierung und Lösung komplexer Probleme in Physik, Chemie, Ingenieurwesen und anderen Disziplinen eingesetzt werden. Trotz ihrer Bedeutung stellt die Lösung von NGS aufgrund ihrer inhärenten Nichtlinearität und des Fehlens geschlossener analytischer Lösungen eine große Herausforderung dar. Traditionelle numerische Verfahren wie das Newton-Raphson-Verfahren oder Optimierungsansätze stoßen oft an ihre Grenzen, insbesondere bei hochdimensionalen Problemen, chaotischem Verhalten oder starker Abhängigkeit von Anfangsbedingungen.

In jüngster Zeit hat der Bereich des Reinforcement Learnings (RL) zunehmend an Bedeutung gewonnen und vielversprechende Ergebnisse bei der Lösung komplexer Probleme geliefert. RL-Agenten lernen durch Interaktion mit einer Umgebung und Belohnungssignale, optimale Strategien zu entwickeln, ohne explizite Programmierung. Dieser Ansatz hat sich in verschiedenen Anwendungsfeldern wie Robotik, Spielen und Optimierungsproblemen als erfolgreich erwiesen.

In dieser Arbeit werden die beiden Ansätze der nichtlinearen Gleichungssysteme kombiniert, wobei ein Fokus auf die Integration von Reinforcement Learning (RL) liegt, um Lösungen zu generieren. Ein RL-Agent wird in einer maßgeschneiderten Umgebung trainiert, die die Struktur des gegebenen nichtlinearen Gleichungssystems widerspiegelt. Durch die Formulierung von Belohnungen für Aktionen, die den Agenten näher an eine Lösung führen, wird dieser befähigt, iterative Strategien zur effizienten Lösungsfindung zu erlernen.

Die vorliegende Arbeit präsentiert einen initiierenden Ansatz und analysiert seine Leistung hinsichtlich Schnelligkeit und Genauigkeit bei der Lösungsfindung von nichtlinearen Gleichungssystemen. Dabei wird eine eingehende Untersuchung durchgeführt, um die Effektivität dieses Ansatzes im Vergleich zu etablierten Methoden wie dem Newton-Raphson-Verfahren zu bewerten. Besonderes Augenmerk wird auf die Identifizierung und Analyse der limitierenden Faktoren dieses Ansatzes gelegt, um potenzielle Schwächen aufzudecken und zu verstehen, inwiefern dieser Ansatz für praktische Anwendungen geeignet ist.

Inhaltsverzeichnis

1 Grundlagen nichtlinearer Gleichungssysteme

Ein nichtlineares Gleichungssystem kann als Nullstellenproblem wie folgt formuliert werden:

Definition 1.1:
Es sei $B \subset \mathbb{R}^n$ und $\mathbf{g} : B \to \mathbb{R}$. Gesucht sind Lösungen von

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}.$$

$\mathbf{f}(\mathbf{x}) = \mathbf{0}$ ist ein System von n nichtlinearen Gleichungen für n Unbekannte $x_1, \ldots, x_n$.

$$\begin{array}{rcl} f_1(x_1, \ldots, x_n) & = & 0 \\ f_2(x_1, \ldots, x_n) & = & 0 \\ \vdots & & \\ f_n(x_1, \ldots, x_n) & = & 0 \end{array}$$

Nichtlineare Gleichungssysteme treten in vielen Bereichen der Wissenschaft und Technik auf, wie beispielsweise in der Physik, Chemie, Biologie, Wirtschaftswissenschaften und Ingenieurwissenschaften. Sie dienen zur Modellierung und Analyse komplexer Systeme, deren Verhalten durch nichtlineare Beziehungen zwischen den Variablen beschrieben wird.
Die Lösung solcher Systeme ist jedoch oft eine große Herausforderung, da nichtlineare Gleichungen im Allgemeinen keine geschlossenen analytischen Lösungen besitzen. Stattdessen müssen numerische Verfahren eingesetzt werden, um approximative Lösungen zu finden. Einige gängige Methoden zur Lösung nichtlinearer Gleichungssysteme sind:

1. **Iterative Verfahren**: Hierzu zählen Methoden wie das Newton-Verfahren, das Quasi-Newton-Verfahren und das Broyden-Verfahren. Diese Verfahren starten mit einer Anfangsschätzung und verbessern diese iterativ, bis eine ausreichend genaue Lösung gefunden ist.

2. **Globale Optimierungsverfahren**: Wenn das Gleichungssystem als Optimierungsproblem formuliert werden kann, können globale Optimierungsverfahren wie die Branch-and-Bound-Methode oder evolutionäre Algorithmen zur Lösung eingesetzt werden.

Die Schwierigkeit, nichtlineare Gleichungssysteme zu lösen, besteht oft darin, geeignete Anfangsschätzungen für die Iterationsverfahren zu finden und die Konvergenz der Verfahren sicherzustellen. Viele Systeme weisen mehrere Lösungen auf, von denen einige instabil oder nicht physikalisch sinnvoll sein können. Darüber hinaus können nichtlineare Systeme eine komplexe Struktur mit mehreren lokalen Extrema aufweisen, was die globale Konvergenz erschwert.

2 Grundlagen Reinforcement Learning

Reinforcement Learning (RL) ist ein Teilgebiet des Maschinellen Lernens, bei dem ein Agent lernt, durch Interaktion mit einer Umgebung eine bestimmte Aufgabe oder ein Ziel zu erreichen. Im Gegensatz zu überwachtem Lernen, bei dem ein Modell anhand von Trainingsbeispielen mit bekannten Eingabe-Ausgabe-Paaren gelernt wird, erhält der Agent beim Reinforcement Learning nur eine skalare Bewertung (Reward) für seine Aktionen. Durch Ausprobieren und Lernen aus den erhaltenen Rewards versucht der Agent, eine Strategie (Policy) zu finden, die die kumulative Belohnung über die Zeit maximiert.

Markov-Entscheidungsprozess:

Reinforcement Learning Probleme werden häufig als Markov-Entscheidungsprozesse (Markov Decision Processes, MDPs) formuliert. Ein MDP besteht aus:

- Einem Zustandsraum $\mathcal{S}$, der alle möglichen Zustände der Umgebung enthält.
- Einem Aktionsraum $\mathcal{A}$, der alle möglichen Aktionen des Agenten definiert.
- Einer Übergangswahrscheinlichkeitsfunktion $\mathcal{P}(s, a, s')$, die die Wahrscheinlichkeit angibt, dass der Agent beim Ausführen der Aktion a im Zustand s in den Zustand s' übergeht.
- Einer Belohnungsfunktion $\mathcal{R}(s, a, s')$, die die Belohnung definiert, die der Agent erhält, wenn er aus dem Zustand s durch Ausführen der Aktion a in den Zustand s' übergeht.

Das Ziel des Agenten ist es, eine Policy $\pi : \mathcal{S} \to \mathcal{A}$ zu finden, die die erwartete kumulative Belohnung über die Zeit maximiert.
Abbildung 1 stellt den Lernprozess eines Agenten in einer Umgebung vereinfacht dar.

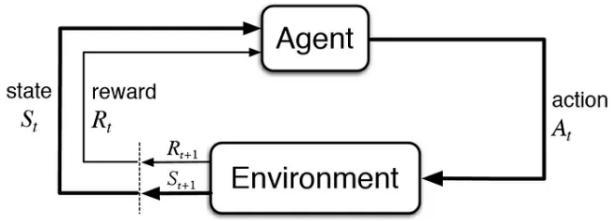

Abbildung 1: Vereinfachter Lernprozess eines Agenten in einer Umgebung[1]

Value Functions und Bellman-Gleichungen:

Eine zentrale Rolle beim Reinforcement Learning spielen die Value Functions, die den erwarteten kumulativen Reward für einen gegebenen Zustand oder eine Zustand-Aktions-Paar angeben. Es gibt zwei wichtige Value Functions:

- Die State-Value Function $V^\pi(s)$ gibt den erwarteten kumulativen Reward an, wenn der Agent im Zustand s startet und der Policy π folgt.
- Die Action-Value Function $Q^\pi(s, a)$ gibt den erwarteten kumulativen Reward an, wenn der Agent im Zustand s startet, die Aktion a ausführt und danach der Policy π folgt.

Die Value Functions erfüllen die Bellman-Gleichungen, die eine rekursive Beziehung zwischen den Value Functions benachbarter Zustände herstellen. Für die State-Value Function lautet die Bellman-Gleichung:

$$V^\pi(s) = \mathbb{E}_\pi \left[R(s, a, s') + \gamma V^\pi(s') \right] \tag{1}$$

Und für die Action-Value Function:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[R(s, a, s') + \gamma \sum_{s'} \mathcal{P}(s, a, s') V^\pi(s') \right] \tag{2}$$

Hier ist $\gamma \in [0, 1]$ der Diskontierungsfaktor, der bestimmt, wie viel Gewicht zukünftigen Rewards beigemessen wird. Ein Wert von γ nahe 0 bedeutet, dass der Agent nur die unmittelbaren Rewards optimiert, während

[1] `https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292`

ein Wert nahe 1 längerfristige Belohnungen stärker gewichtet.

Proximal Policy Algorithmus (PPO):

Proximal Policy Optimization (PPO) ist ein moderner Algorithmus im Bereich des Reinforcement Learning, der zur Klasse der Policy Gradient Methoden gehört. PPO wurde von Schulman et al. [**schulman2017proximal**] entwickelt und hat sich aufgrund seiner guten Performanz und Stabilität in verschiedenen Anwendungsbereichen etabliert.

Das Ziel von PPO ist es, eine optimale Policy zu finden, die die erwartete kumulative Belohnung maximiert. Im Gegensatz zu klassischen Policy Gradient Methoden, die oft große Schritte im Parameterraum machen und dadurch instabil werden können, verwendet PPO eine Clipping-Technik, um die Größe der Policy-Updates zu begrenzen. Dadurch wird eine stabilere Konvergenz erreicht und die Wahrscheinlichkeit von Divergenz oder schlechten Performanz reduziert.

Der Kerngedanke von PPO besteht darin, die Policy-Updates so zu gestalten, dass sie innerhalb einer vertrauenswürdigen Region (trust region) bleiben. Dies wird erreicht, indem die Differenz zwischen der alten und der neuen Policy durch eine Clipping-Funktion begrenzt wird. Die Clipping-Funktion schneidet die Wahrscheinlichkeitsverhältnisse (probability ratios) zwischen der alten und der neuen Policy bei vordefinierten Schwellenwerten ab.

Formal lässt sich die Zielfunktion von PPO wie folgt darstellen:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right] \tag{3}$$

Hierbei ist θ der Parametervektor der Policy, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ das Wahrscheinlichkeitsverhältnis zwischen der neuen und der alten Policy, $\hat{A}_t$ der geschätzte Vorteil und ϵ ein Hyperparameter, der die Größe der vertrauenswürdigen Region steuert.

Der Term $\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$ sorgt dafür, dass die Policy-Updates innerhalb der vertrauenswürdigen Region bleiben. Wenn das Wahrscheinlichkeitsverhältnis $r_t(\theta)$ außerhalb des Intervalls $[1-\epsilon, 1+\epsilon]$ liegt, wird es auf die Grenzen des Intervalls gesetzt. Dadurch werden zu große Policy-Updates vermieden und die Stabilität des Lernprozesses verbessert.

PPO verwendet häufig auch eine Value Function $V_\phi(s)$, um den Wert eines Zustands zu schätzen. Die Value Function wird verwendet, um den Vorteil $\hat{A}_t$ zu berechnen, der die Qualität einer Aktion im Vergleich zum erwarteten Wert des Zustands angibt. Der Vorteil kann beispielsweise durch die Generalized Advantage Estimation (GAE) [**schulman2015high**] geschätzt werden.

Insgesamt bietet PPO eine effektive und stabile Methode zum Lernen von Policies in Reinforcement Learning Problemen. Durch die Verwendung der Clipping-Technik und der vertrauenswürdigen Region werden große Policy-Updates vermieden und eine stabile Konvergenz erreicht. PPO hat sich in verschiedenen Benchmark-Problemen und realen Anwendungen bewährt und ist ein weit verbreiteter Algorithmus im Bereich des Reinforcement Learning.

Neben dem Proximal-Policy-Algorithmus gibt es noch eine Vielzahl weiterer Algorithmen, die sich über die Zeit entwickelt haben, wie bspw. dem *Deep Deterministic Policy Gradient* (DDPG)-, *Twin Delayed DDPG* (TD3)-, oder dem *Soft Actor Critic*-Algorithmus.

3 Reinforcement Learning für die Lösung nichtlinearer Gleichungssysteme

Bevor ein RL-Agent nichtlineare Gleichungssysteme lösen kann, sollten zunächst einige wichtige Fragen beantwortet werden, die für die Qualität der Lösung von Relevanz sind:

1. **Umgebung:** Was ist die Umgebung, in dem der Agent agiert?
2. **Aktionen:** Was sind zulässige und sinnvolle Aktionen, die der Agent wählen kann?
3. **Belohnung:** Wie sieht eine zielführende Belohnungsfunktion aus?

4 Umsetzung

5 Ergebnisse

6 Fazit