Journal of Computer Science

Review

An Overview of the Application of Neural Networks in Orbit Optimization

Pratiksha Devshali Bhavsar and Deepti Pal

Department of Applied Science, Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University), Lavale, Pune, Maharashtra, India

Article history
Received: 20-07-2024
Revised: 03-09-2024
Accepted: 13-09-2024

Corresponding Author:
Pratiksha Devshali Bhavsar
Department of Applied Science, Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University), Lavale, Pune, Maharashtra, India
Email: pratiksha.bhavsar@sitpune.edu.in

Abstract: The idea of neural networks has emerged in recent times and has shown great results in complex and nonlinear systems. Many aerospace engineering areas like autonomous systems, adaptive control, and flight dynamics modeling can be better handled with neural networks. Selecting an orbit that consumes the fewest resources and reduces expenditure can broadly be called orbit optimization. In this study, the application of neural networks in orbit optimization in various space missions and satellite ventures is brought together.

Keywords: Orbit Optimization, Neural Networks, Artificial Neural Networks, Optimization

Introduction

The optimization of orbital trajectories is a major difficulty in the rapidly developing field of aerospace engineering, especially in light of the growing number of satellite installations and complex space missions. With the emergence of Neural Networks (NN), which offer a potent mix of prediction accuracy and computing efficiency, revolutionary techniques for Orbit Optimization (OO) have arisen.

For a particular space maneuver the orbit optimization may be minimizing fuel consumption maximizing the precision of some angle of the trajectory or minimizing/maximizing the time of transfer between the orbits etc. as required by the mission. This is a broad term that encompasses the process of adjusting a spacecraft's orbital trajectory (Addis *et al.*, 2011) and parameters to effectively accomplish the desired aims and objectives of the mission.

As the field of NN is evolving rapidly, there are new modifications getting added to their architecture hence changing the old techniques into better ones. Popular optimization methods like genetic algorithm, and particle swarm optimization have also been modified using NN. This manuscript is a first-of-its-kind attempt to bring together the investigation done in the vast field of orbit determination and optimization combined with NN.

Artificial Neural Networks

Artificial Neural Networks (ANN), colloquially called Neural Networks (NN) or NNets, can be defined as a machine learning model that consists of several computational units that are linked to each other in a well-defined connection (Heaton, 2018; Bishop, 1994; Schmidhuber, 2015). The inspiration for this model came from the human brain, which is incredibly intricate and multitasking, made up of basic unit cells (McCulloch and Pitts, 1988) called neurons. In the case of ANN, they are called artificial neurons or simply neurons or perceptrons.

These are the basic units that facilitate the computation of data by interlinking to one another. Their mechanism is very similar to that of human neuron cells. This biological analogy from the human brain is taken due to its ability to perform parallel processing and be fast and robust.

An artificial neuron (Haykin, 1998; Wu and Feng, 2018) receives input signals, that is, parameters or features from other connected neurons. The strength of these signals is determined by the Synaptic Weights (SW). The product of weights and parameters is summed together and a constant (called bias) is added to it. The bias is an error term that is added to compensate for the simplified assumptions of the model. A transformation is then applied to this summation unit using an activation function that determines the output by limiting the amplification of the summation unit to a finite value. Figures (1-2), show a schematic representation of a neuron with two parameters/features and its model respectively.

This schematic representation shows the flow of input signals in a neuron. The function is the one that classifies the input to give the desired output. Figure (1) is the actual model of a neuron, which is apparently McCulloch-Pitts' model of a neuron. The authors (McCulloch and Pitts,

DOI: 10.3844/jcssp.2025.558.565

1988) were the first ones to describe how a perceptron or neuron would look like, drawing inspiration from a biological neuron.

Mathematically, the neuron can be described from Fig. (1) in the following manner (Fig. 2), where u_k represents a neuron and o its output which is basically given as:

$$o = u_k = f\left((p_1 w_1 + p_2 w_2) + b\right) \tag{1}$$

where, p_1 and p_2 are the parameters, w_1 and w_2 are SWs and b is a bias. For the 'n' number of parameters, the model of a neuron looks like Fig. (3).

Hence, in general, the neuron will look like this:

$$o = u_k = f\left(\sum_{k=1}^{n} p_k w_k + b\right) \tag{2}$$

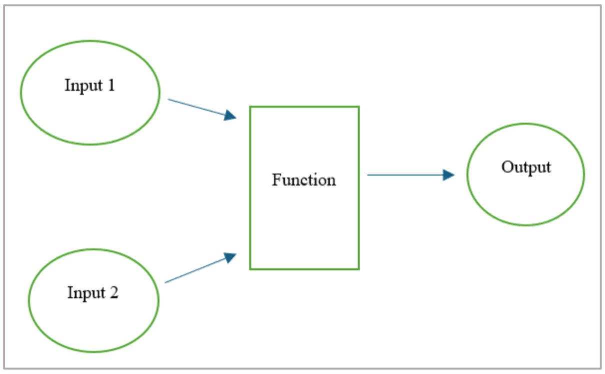

Fig. 1: Schematic diagram of a neuron

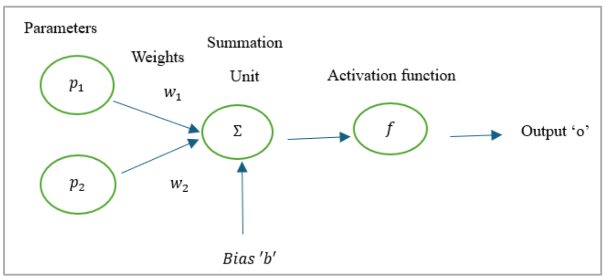

Fig. 2: Model of a neuron when only two parameters are provided

Adding multiple layers between input and output makes the NN deep, hence the name Deep Neural Network (DNN). NNs are a main subset (Fig. 4) of machine learning and artificial intelligence.

There are various types of neural networks, each suitable for different kinds of tasks. The NNs have been broadly classified (Cheng and Titterington, 1994; Eva and Friedman, 1994; Bishop, 1995; Chen and Aihara, 1999; Kenol *et al.*, 2002) as:

- Feed-forward neural networks (information moves in only one direction)
- Convolutional neural networks (specifically for processing structured grid data like images)
- Recurrent neural networks (designed to recognize patterns in sequences of data)
- Generative adversarial networks (consist of two neural networks that compete against each other)
- Radial basis function networks (uses radial basis functions as activation functions)
- Long short-term memory networks (can learn long-term dependencies)
- Gated recurrent units (use reset and update units to control the flow of information)
- Autoencoders (designed for unsupervised learning of efficient codings)
- Multilayer perceptrons (consists of at least three layers of neurons)
- Modular neural networks (consists of many independent networks collaborating to solve a complex problem) etc.

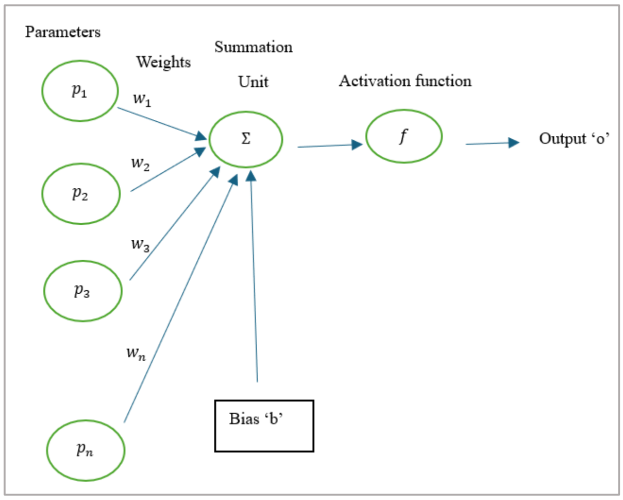

Fig. 3: Model of a neuron with 'n' number of parameters

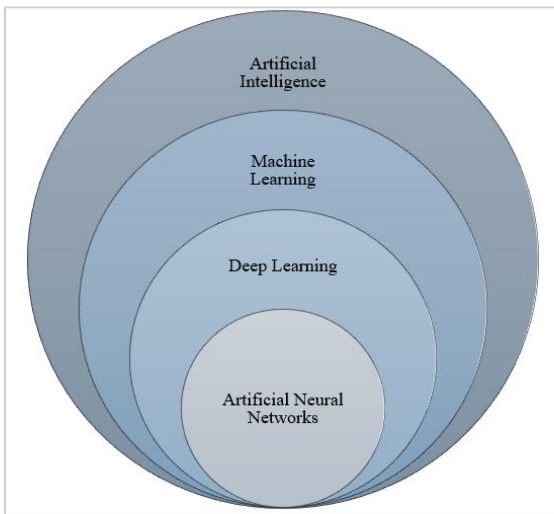

Fig. 4: Hierarchy of artificial intelligence and its subdivisions

The main aim of any type of ANN is to find the optimal solution that classifies every feature of the data set. The difference between the target value and the output value of the model is known as the cost function or the error function. The ANN targets reducing this cost function by modifying SWs and bias. This process of finding the optimal values of weights and biases to minimize the cost function is termed learning of the neural network. There are various cost functions. It is our task to select a cost function that is suitable to our data set. Some of them are Mean Relative Error (MRE) and Mean Square Error (MSE):

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|P_i - A_i|}{A_i}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(A_i - P_i)^2$$

where A_i is the i^{th} actual value and P_i is the i^{th} predicted value.

NN does not require prior information regarding the relationship between the data set. Rather it is trained by giving experience through a huge number of examples. Each of these examples will have many solutions or search space out of which only few will provide us with the minimum cost function.

An optimization algorithm is needed to train the NN. The choice of cost function and optimization algorithm are closely associated with the process of learning. The type of cost function chosen affects how the optimization algorithm adjusts itself to decrease the cost function. Some of these algorithms are gradient-based viz. Adaptive gradient algorithm, Adams moment estimation, adaptive moment estimation, etc. The Gradient Descent (GD) for cost function $C(w)$, is defined as the negative of the gradient of the cost function i.e., $GD = -\partial C/\partial w_j$.

Here in this equation, the gradient of cost function is being taken with respect to the weight, to update the weights. This gradient can also be taken with respect to bias in order to update it. The updated weight is given by: $w_j = w_j + \eta\Delta w_j$, where, $\Delta w_j = -\partial C/\partial w_j$ and η is the learning rate. The rate of learning is defined as the rate at which the gradient reaches the crest of the search space or the cost function tends to zero.

As there are different cost functions, there are various activation functions (Haykin, 1998) too. The choice of these activation functions depends on what exactly the problem is. Some examples of activation functions are shown in Fig. (5). Apparently, this whole process of choosing an activation function, the right optimization algorithm, and the cost function in order to reduce the cost function, is called backpropagation. In NNs, many neurons come together and are connected to each other through non-linear mapping. Every ANN has an input layer and an output layer. When there are one or more hidden layers present between the input and output layers, they form a multi-layer perceptron (MLP). These neural networks operate in a feedforward manner and are also referred to as Deep Neural Networks (DNNs). It is used when a data set where different classes cannot be separated through just one plane. Hence the presence of a hidden layer aids in working with noisy data.

For an MLP the backpropagation technique is almost the same, however the presence of various valleys and hills in the model poses a problem. This is where momentum comes into the picture. The momentum helps accelerate through the converging optimization process. In simple words, it helps to navigate through different valleys to find the deepest one, which will be considered the best optimal solution. It helps in updating the velocity and weight vector:

$$V(t) = \alpha V(t-1) + (1-\alpha)\frac{\partial C}{\partial w}$$

where, $V(t)$ is the velocity at iteration t and α is the momentum constant that controls the contribution of velocity at the previous iteration.

The weights then can be updated using the learning rate as:

$$w(t) = w(t-1) - \eta V(t)$$

All the research that is happening in *NN* is mainly focused on finding better weights. *NN* is applied in various fields like healthcare and diagnosis, finance, predictive analysis, image recognition, etc. In this study, its application in orbit optimization is studied.

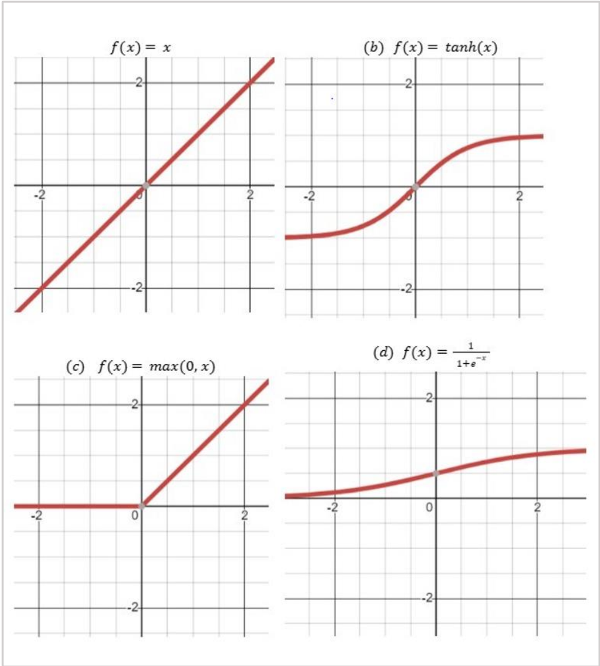

Fig. 5: Activation functions

Orbit Optimization

In order to comprehend how NN aids in orbit optimization, one must first grasp what orbits are and how their study has evolved over time. Trajectory or orbit of planetary bodies has enchanted humans since time immemorial. Many popular physicists and mystics have observed space and realized many intricacies of the universe especially the planetary motion. Then later on with the advancement of science and technology, human-made satellites also added to the purview.

For the commercial usage of satellites, it has to return to the same position in the sky, at the same time, every day. In other words, the satellite has to be injected into the geosynchronous orbit. The process involves intermediary transfer orbits, till it settles in its orbital slot. For other non-similar purposes, different types of orbits may be required. The orbits can be classified (Miele *et al*., 2004; Carter and Humi, 2020; Sanjurjo Rivo and Soler, 2021; Ribeiro *et al*., 2023; Serra *et al*., 2021; Sangrà *et al*., 2021; Savitri *et al*., 2017; Song *et al*., 2018) based upon their altitude (Low earth orbit, Medium Earth Orbit, Geostationary Orbit, Geosynchronous Earth Orbit, High Earth Orbit); based upon inclination (Equatorial Orbit, Polar Orbit, Sun Synchronous Orbit); based on shape (Circular Orbit, Elliptic Orbit) etc.

The requirement of having an optimized orbit has become much sought after in recent times to ensure cost-effectiveness, mission success, and longevity of the spacecraft.

The goal of orbit optimization is to allow satellites to function as efficiently as possible by selecting an orbit that consumes the fewest resources and reduces expenditures. It comprises creating a strategy and algorithms that will reduce fuel consumption, reduce the likelihood of a satellite being damaged, and extend the satellite's lifespan while in orbit.

Moreover, OO depends upon the mission analysis. It includes determining the desired orbit, the payload requirement, the vehicle capabilities, mission duration, communication coverage, etc. The OO experience can be enhanced with a better understanding of the orbital mechanics. Kepler's laws of planetary motion serve as the base and other basic principles of orbital mechanics involve the orbital transfers, gravitational assists, and effect of perturbations as in atmospheric drag, solar radiation pressure, etc.

The OO may involve optimization of the parameters that define the shape and orientation of the orbit, fuel usage, mission duration, transfer trajectory, launch time, energy, payload delivery, coverage and revisit time for earth observation satellites, workable life of the spacecraft, cost, etc. It is usually a multi-objective optimization where a trade-off may be required amongst various objectives before deciding upon the final most appropriate orbit in accordance with the mission's priorities.

Many optimization techniques have been directly used so far for OO. Some of them are Hohmann transfer (Miele *et al*., 2004), BiElliptical transfer (Carter and Humi, 2020), Low Thrust Trajectory Optimization (Sanjurjo Rivo and Soler, 2021), Gravity-Assist Maneuvres (Ribeiro *et al*., 2023), Optimal Control Theory (Serra *et al*., 2021), Lambert's Problem (Sangrà *et al*., 2021), Genetic Algorithm (Savitri *et al*., 2017) and Multi Objective Optimization (Song *et al*., 2018).

The mission's integration of technical advancements will determine how OO develops. For instance, using sophisticated propulsion systems can provide precise control over the spaceship, resulting in reduced fuel consumption and shorter journey times. Particularly in the fields of astrodynamics, celestial mechanics, and orbital dynamics, mathematical models can be created that offer a theoretical basis for comprehending how an object behaves in space and, consequently, for optimizing its orbit. An outline of the general flow of an OO process is given in Fig. (6). In the next section the focus is on, NN-based OO, and its general outline is shown in Fig. (7).

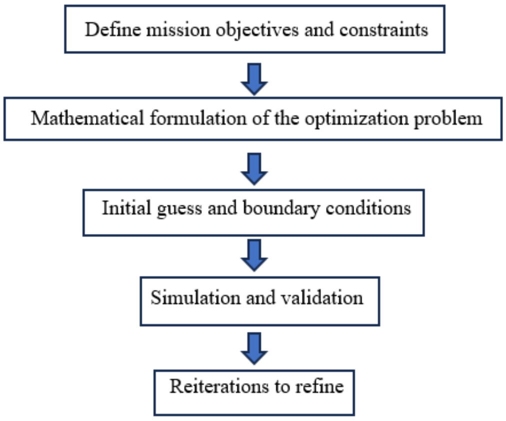

Fig. 6: General flowchart for OO

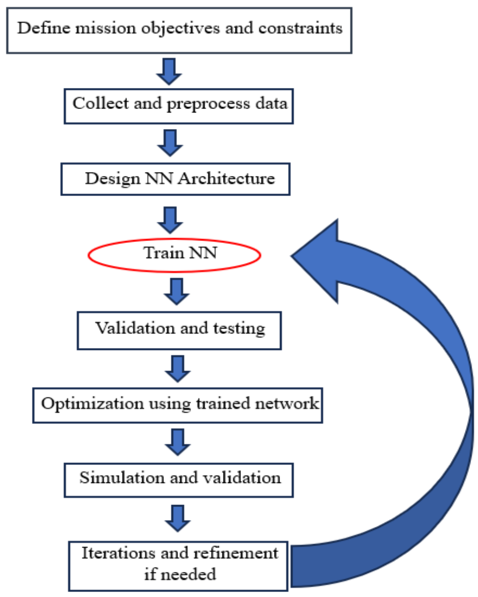

Fig. 7: General flowchart for OO with NN

OO Using NN

A few classic research papers were considered for this study on the OO with NN. Table (1) shows the main parameter that has been optimized as well as the kind of network chosen in the study.

In Kenol and Paul P (2002), ANN was used in the design and optimization of aircraft/engine propulsion systems to shorten the overall design cycle. Using types of algorithms that produce approximate solutions to unsolvable high-level problems, a universal design tool was developed which can be used for most design optimization processes. Key areas of analysis included aircraft noise and emission, engine cycle analysis, aircraft sizing, aircraft mission analysis system weight prediction, and engine economics analysis. Taguchi techniques and fuzzy logic were used along with NN.

In order to forecast satellite thruster force and regulate osculating orbital elements during maneuvers, an adaptive neural network predictor controller was created by Aly *et al.* (2003). A time-delayed multi-layer feed-forward neural network was employed.

In Ampatzis and Izzo (2009) the authors have shown how trajectory optimization problems can profit from the introduction of approximate models. They used an MLP with two hidden layers. They hybridized the algorithms evolving at times with the original objective function and at times with the approximate objective function due to ANN.

The paper (Meier *et al*., 2012) reports the usage of two NNs for orbit correction in the Australian Synchrotron Storage Ring. The actor neural network learns to take appropriate control actions in order to minimize a long-term cost function modeled by a critical neural network.

In Adrian (2012), a new optimization technique using the direct and inverse kinematics with the Fourier spectrum; for the space trajectory of the tool center point was discussed. They designed a NN, Bipolar Sigmoid Hyperbolic Tangent with Time Delay and Recurrent Links. LabVIEW VI-s was used to verify the results.

In Sánchez-Sánchez and Izzo (2018), the authors chose a feed-forward deep NN to solve the landing problems. They trained separate networks for each of the control variables in all their models. With the help of this interesting aerospace study, real-time optimal control structures for planetary landing might be designed, directly driving the state-action selection with a DNN. The authors of Mereta *et al.* (2017) noted that the machine learning technique outperforms the impulsive Lambert estimate by a significant margin when it comes to multi-revolution transfers between near-Earth objects. Following the best possible low-thrust transfer between two Near Earth Objects, they calculated the final mass of the spacecraft.

Three NN was trained to determine the ideal time, starting costates, and optimal control law of time-optimal low-thrust indirect trajectory optimization by Li *et al.* (2019a). According to their findings, NN was able to predict the initial optimal time and costates with accuracy. In Li *et al.* (2019b), the authors focused on autonomous orbit raising during the orbit transfer. They generated the datasets by solving the low thrust optimization problem using the indirect method with modified orbital state elements and thrust homotopy. Then NN was applied for "autonomous time-optimal many-revolution low-thrust orbit raising problems."

The authors (Han and Wang, 2021) solved the problem of low orbit transfer time while using electric thrusters, using bidirectional stochastic gradient descent method. The authors (Song and Gong, 2019) resolved the challenge of repeated near-earth asteroid rendezvous with a predetermined sequence length. In order to map the relationship between the orbital properties and the transfer time, they established DNN. The goal of the paper (Cheng *et al*., 2020) was to quickly and reliably generate the best landing trajectories for powered descent and landing on an asteroid's surface. The authors created and improved DNN to provide an accurate approximation of the 2-D transfer's costates and an optimum landing trajectory for asteroids. Previously, they used model continuation and state transformation approaches to show that the original asteroid landing challenges are related to 2-D asteroid-free transfer problems.

Table 1: Literature overview on OO with NN

Year	Reference	The optimized parameter	Type of NN
2002	Kenol and Paul P (2002)	Engine propulsion system	Artificial neural network
2002	Aly *et al*. (2003)	Orbit estimation	Neural network
2009	Ampatzis and Izzo (2009)	Trajectory optimization	Artificial neural network
2012	Meier *et al*. (2012)	Orbit correction	Actor-Critic control scheme
2012	Adrian (2012)	Space trajectory	Artificial neural network
2016	Sánchez-Sánchez and Izzo (2018)	Landing	Deep neural network
2017	Mereta *et al*. (2017)	Low thrust transfers and Spacecraft mass	Machine learning
2019	Li *et al*. (2019a)	Time and costates	Deep neural network
2019	Li *et al*. (2019b)	Orbit raising	Artificial neural network
2019	Song and Gong (2019)	Transfer time	Deep neural network
2020	Cheng *et al*. (2020)	Asteroid landing trajectory	Deep neural network
2020	Chai *et al*. (2020)	Trajectory	Deep neural network
2020	Li *et al*. (2020)	Transfer cost	Deep neural network
2021	Zhifei *et al*. (2021)	Trajectory	Radial basis function NN
2022	Schiassi *et al*. (2022)	Orbit transfers	Physics-informed neural networks
2022	Xie and Dempster (2022)	Trajectory	Deep neural network
2023	Tang and Gong (2023)	Trajectory	Artificial neural network
2024	D'Ambrosio and Furfaro (2024)	Propellant consumption	Pontryagin neural networks

A bi-level control method with improved trajectory optimization and DNNs was created by Chai *et al*. (2020) to guide Hypersonic Vehicle (HV) reentry flight. By taking into account the rotational effects, during the trajectory optimization phase, the HV model with three degrees of freedom was expanded to six degrees of freedom. DNNs were designed to simulate optimal state-control actions and provide real-time feedback.

The authors of Li *et al*. (2020) employed feed-forward DNNs to estimate optimal transfer costs for three different kinds of optimization problems: Minimum velocity change in J_2 perturbed multi-impulse transfers, fuel-optimal and time-optimal low-thrust transfers.

The authors of Zhifei *et al*. (2021) integrated trajectory features with temporal continuity to create a phase space reconstruction-based target maneuver trajectory prediction model known as radial basis function NN (RBFNN).

A recent Physics-Informed NN-based approach was presented by the authors of Schiassi *et al*. (2022) to solve optimal planar orbit transfer issues through the indirect technique. Using the state-costate pair that is the solution to the Two-Point Boundary Value Problem resulting from the application of the Pontryagin Minimum Principle, this method seeks to model an NN representation of the optimal control.

In Xie and Dempster (2022), the authors gave the construction, training, and optimization of a DNN-classifier that can identify feasible low thrust transfer, prior to the optimization process.

The article (Tang and Gong, 2023) analyzed the power recovery of the rocket's first stage. It takes into account the entire trajectory optimization process, beginning with the separation of the first and second phases and ending with the landing. The authors employed an NN to turn the problem into a parameter optimization problem, which was then solved using a genetic algorithm.

Pontryagin NNs (PoNNs) are used in the paper (D'Ambrosio and Furfaro, 2024) to obtain control techniques for reaching fuel-optimal trajectories. The new Extreme Theory of Functional Connections was the main tool that the authors used in the Physics Informed NN architecture. The suggested method was used for both a landing trajectory on Mars and a fixed-time fuel-optimal trajectory with low thrust propulsion from Earth to Mars, where the number of switches is unknown beforehand. PoNNs can identify discontinuities in the Hamiltonian that lead to discontinuous control in linear control issues. The outcomes showed a high degree of accuracy.

Discussion and Conclusion

The major concerns related to any aircraft maneuver vary from shortening the design cycle to the low-thrust adjustments to the minimal fuel consumption etc. All these requirements can be brought under one term called 'orbit optimization.' Various numerical methods like the shooting method, collocation method, etc. have been used to solve the boundary value problems arising in finding the most efficient trajectory that meets the mission requirements. Moreover, Pontryagin's maximum principle and Bellman's dynamic programming are the popular approaches under optimal control theory, for optimization.

The introduction of NN in the initial phase converts the problem into a parameter optimization problem which can then be solved by direct and indirect methods, particle swarm optimization, genetic algorithm, etc. using the NN-trained data. The Deep NN has been proven to be highly accurate in predicting low thrust transfers in comparison to the Ensemble learning model, K-Nearest Neighbors model, Support Vector Machine, etc. The software used for NN has evolved from Merlin to MATLAB to Python and so on. The NN architecture has also been extended in the last twenty years from NN → DNN → RBFNN → PINN → PoNN especially for OO. Although the NN version was used for a specific problem its validity and accuracy have been established well in the literature.

In conclusion, it is anticipated that the discovery of Physics informed NN (Schiassi *et al*., 2022; D'Ambrosio and Furfaro, 2024) brings in greater possibilities for OO with NN, providing new avenues for pattern identification, prediction, and optimization. The inclusion of a split domain can be used to effectively address complicated multi-revolution situations. Improvised optimization methods (Zuo *et al*., 2024; Reyad *et al*., 2023) like Nesterov accelerated gradient, Adam, Nadam, AdaMax, AMSGrad, etc. can be incorporated with NN for OO problems.

Acknowledgment

The authors are grateful to all the anonymous reviewers for their valuable remarks and suggestions.

Funding Information

No funding was received for this study.

Author's Contributions

Both authors have equal contribution.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

Conflict of Interest

The authors declare that there is no conflict of interest.

References

Addis, B., Cassioli, A., Locatelli, M., & Schoen, F. (2011). A Global Optimization Method for the Design of Space Trajectories. *Computational Optimization and Applications*, *48*(3), 635–652. https://doi.org/10.1007/s10589-009-9261-6

Adrian, O. (2012). Optimization of The Robot Space Trajectory by Using Kinematics, Dynamics, Intelligent Damper and Proper Neural Network. *Proceedings in Manufacturing Systems*.

Aly, A. F., Nguib Aly, M., Elshishtawy, M. E., & Zayan, M. A. (2003). Optimization Techniques for Orbit Estimation and Determination to Control the Satellite Motion. *Proceedings, IEEE Aerospace Conference*, 5–5. https://doi.org/10.1109/aero.2002.1035389

Ampatzis, C., & Izzo, D. (2009). Machine Learning Techniques for Approximation of Objective Functions in Trajectory Optimisation. *Proceedings of the Ijcai-09 Workshop on Artificial Intelligence in Space*, 1–6.

Bishop, C. M. (1994). Neural Networks and Their Applications. In *Review of Scientific Instruments* (Vol. 65, Issue 6, pp. 1803–1832). https://doi.org/10.1063/1.1144830

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*.

Carter, T., & Humi, M. (2020). Two-Impulse, Bi-parabolic and Bi-Elliptical Transfers Between Coplanar Elliptical Orbits. *Acta Astronautica*, *177*, 158–171. https://doi.org/10.1016/j.actaastro.2020.06.013

Chai, R., Tsourdos, A., Savvaris, A., Chai, S., Xia, Y., & Chen, C. L. P. (2020). Six-DOF Spacecraft Optimal Trajectory Planning and Real-Time Attitude Control: A Deep Neural Network-Based Approach. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(11), 5005–5013. https://doi.org/10.1109/tnnls.2019.2955400

Chen, L., & Aihara, K. (1999). Global Searching Ability of Chaotic Neural Networks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *46*(8), 974–993. https://doi.org/10.1109/81.780378

Cheng, B., & Titterington, D. M. (1994). Neural Networks: A Review from a Statistical Perspective. In *Statistical Science* (Vol. 9, Issue 1, pp. 2–30). https://doi.org/10.1214/ss/1177010638

Cheng, L., Wang, Z., Jiang, F., & Li, J. (2020). Fast Generation of Optimal Asteroid Landing Trajectories Using Deep Neural Networks. *IEEE Transactions on Aerospace and Electronic Systems*, *56*(4), 2642–2655. https://doi.org/10.1109/taes.2019.2952700

D'Ambrosio, A., & Furfaro, R. (2024). Learning Fuel-Optimal Trajectories for Space Applications via Pontryagin Neural Networks. *Aerospace*, *11*(3), 228–245. https://doi.org/10.3390/aerospace11030228

Eva, B., & Friedman, A. (1994). *Neural Networks and Orbit Control in Accelerators*. 4. European particle accelerator conference, London United Kingdom.

Han, M., & Wang, Y. (2021). Engineering Optimization Method of Orbit Transfer Strategy for All-electric Propulsion Satellites. *Journal of Physics: Conference Series*, *2029*(1), 012011. https://doi.org/10.1088/1742-6596/2029/1/012011

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*.

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio and Aaron Courville: Deep Learning. In *Genetic Programming and Evolvable Machines* (Vol. 19, Issue 1, pp. 305–307). https://doi.org/10.1007/s10710-017-9314-z

Kenol, J., & Paul P, Lin. (2002). Artificial Neural Networks Applications: from Aircraft Design Optimization to Orbiting Spacecraft on Board Environment Monitoring. *2001 Advanced Study Institute on Neural Networks for Instrumentation, Measurement and Related Industrial Applications*, 2002.

Li, H., Baoyin, H., & Topputo, F. (2019a). Neural Networks in Time-Optimal Low-Thrust Interplanetary Transfers. *IEEE Access*, *7*, 156413–156419. https://doi.org/10.1109/access.2019.2946657

Li, H., Chen, S., Izzo, D., & Baoyin, H. (2020). Deep Networks as Approximators of Optimal Low-Thrust and Multi-Impulse Cost in Multitarget Missions. *Acta Astronautica*, *166*(4), 469–481. https://doi.org/10.1016/j.actaastro.2019.09.023

Li, H., Topputo, F., & Baoyin, H. (2019b). Autonomous Time-Optimal Many-Revolution Orbit Raising for Electric Propulsion Geo Satellites Via Neural Networks. *Optimization and Control*. https://doi.org/https://doi.org/10.48550/arXiv.1909.08768

Meier, E., Tan, Y. E., & LeBlanc, G. (2012). Orbit Correction Studies Using Neural Networks. *Proc. 3rd Int. Particle Accelerator Conf*, 2837–2839.

McCulloch, W. S., & Pitts, W. (1988). (1943) Warren S. McCulloch and Walter Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," Bulletin of Mathematical Biophysics 5: 115-133. In *The bulletin of mathematical biophysics* (Vol. 5, pp. 115–133). Springer. https://doi.org/https://doi.org/10.7551/mitpress/4943.001.0001

Mereta, A., Izzo, D., & Wittig, A. (2017). *Machine Learning of Optimal Low-Thrust Transfers Between Near-Earth Objects*. *10334*. https://doi.org/10.1007/978-3-319-59650-1_46

Miele, A., Ciarci, M., & Mathwig, J. (2004). Reflections on the Hohmann Transfer. *Journal of Optimization Theory and Applications*, *123*(2), 233–253. https://doi.org/10.1007/s10957-004-5147-z

Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A Modified Adam Algorithm for Deep Neural Network Optimization. *Neural Computing and Applications*, *35*(23), 17095–17112. https://doi.org/10.1007/s00521-023-08568-z

Ribeiro, R. S., Meireles, L. G., Prado, A. F. B. A., & de Melo, C. F. (2023). Alternative Approach to The DART Mission by The use of Gravity Assist Maneuvers with the Moon and Solar Sails. *Scientific Reports*, *13*(1), 6937. https://doi.org/10.1038/s41598-023-33680-4

Sánchez-Sánchez, C., & Izzo, D. (2018). Real-Time Optimal Control via Deep Neural Networks: Study on Landing Problems. *Journal of Guidance, Control and Dynamics*, *41*(5), 1122–1135. https://doi.org/10.2514/1.g002357

Sangrà, T., la, D. de, & Elena, F. (2021). Review of Lambert's problem. *ISSFD 2015: 25th International Symposium on Space Flight Dynamics*, 1–15.

Sanjurjo Rivo, M., & Soler, M. (2021). A Survey on Low-Thrust Trajectory Optimization Approaches. *Aerospace*, *8*(3), 88–89. https://doi.org/10.3390/aerospace8030088

Savitri, T., Kim, Y., Jo, S., & Bang, H. (2017). Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach. *International Journal of Aerospace Engineering*, *2017*(1), 1–17. https://doi.org/10.1155/2017/1235692

Schiassi, E., D'Ambrosio, A., Drozd, K., Curti, F., & Furfaro, R. (2022). Physics-Informed Neural Networks for Optimal Planar Orbit Transfers. *Journal of Spacecraft and Rockets*, *59*(3), 834–849. https://doi.org/10.2514/1.a35138

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Serra, R., Yanez, C., & Frueh, C. (2021). Tracklet-to-Orbit Association for Maneuvering Space Objects Using Optimal Control Theory. *Acta Astronautica*, *181*, 271–281. https://doi.org/10.1016/j.actaastro.2021.01.026

Song, Y., & Gong, S. (2019). Solar-Sail Trajectory Design for Multiple Near-Earth Asteroid Exploration Based on Deep Neural Networks. *Aerospace Science and Technology*, *91*, 28–40. https://doi.org/10.1016/j.ast.2019.04.056

Song, Z., Chen, X., Luo, X., Wang, M., & Dai, G. (2018). Multi-Objective Optimization of Agile Satellite Orbit Design. *Advances in Space Research*, *62*(11), 3053–3064. https://doi.org/10.1016/j.asr.2018.08.037

Tang, D., & Gong, S. (2023). Trajectory Optimization of Rocket Recovery Based on Neural Network and Genetic Algorithm. *Advances in Space Research*, *72*(8), 3344–3356. https://doi.org/10.1016/j.asr.2023.07.028

Wu, Y., & Feng, J. (2018). Development and Application of Artificial Neural Network. *Wireless Personal Communications*, *102*(2), 1645–1656. https://doi.org/10.1007/s11277-017-5224-x

Xie, R., & Dempster, A. G. (2022). Feasible Low-Thrust Trajectory Identification Via a Deep Neural Network Classifier. *Optimization and Control*, arXiv:2202.04962.

Zhifei, X., An, X., Yingxin, K., Zhanwu, L., & Aiwu, Y. (2021). Target Maneuver Trajectory Prediction Based on RBF Neural Network Optimized by Hybrid Algorithm. *Journal of Systems Engineering and Electronics*, *32*(2), 498–516. https://doi.org/10.23919/jsee.2021.000042

Zuo, X., Li, H.-Y., Gao, S., Zhang, P., & Du, W.-R. (2024). NALA: A Nesterov Accelerated Look-Ahead Optimizer for Deep Learning. *PeerJ Computer Science*, *10*, 2167. https://doi.org/10.7717/peerj-cs.2167