Neural network and numerical methods performance comparison for 
prey-predator model

DOI: 10.46932/sfjdv5n7-041

Received on: Jun 28th, 2024
Accepted on: Jul 19th, 2024

Fatima Ouaar
PhD in Mathematics
Institution: Laboratory of Applied Mathematics, Mohamed Khider University
Address: BP145 RP, Biskra, Algérie, 07000
E-mail: f.ouaar@univ-biskra.dz

ABSTRACT
In several areas, differential equations are used extensively to simulate a wide range of events. The Prey-
Predator model, sometimes referred to the Lotka-Volterra equations, was used as an example in this work.
On the other hand, occasionally insufficient data is available to build an explicit model for this problem.
Therefore, being able to approximate differential equation solutions is important. This paper's primary
contribution is the performance comparison between the implicit Euler approach and the neural network
method. The outcomes demonstrate that although the neural network approach takes longer to provide an
estimate, it consistently produces better estimates than the implicit Euler technique.

Keywords: Neural Network Methods, Numerical Methods, Prey-Predator Model, Equations.

1 INTRODUCTION

In several scientific domains, differential equations hold significant relevance. When Gotfried
Wilhelm Leibniz and Isaac Newton collaborated to create calculus, they came into play. Differential
equations have now become a common tool in the mathematical toolkit of many fields. These kinds of
equations are frequently used to simulate real-world situations in order to better understand a variety of
phenomena. The Prey-Predator model, which is interpreted by the Lotka-Volterra system of equations and
takes the shape of the well-known Prey-Predator model (Mühlbauer, 2020) is one of the most often
modelled biological phenomena. It is currently widely utilized and has been expanded to numerous fields
of population dynamics and is widely used today in biology, ecology and even chemistry (Volterra, 1931),
(Volterra, 1926), (Volterra, 1927) and (Volterra, 1928).

In this paper we are only interested for positive values of $y_1$ and $y_2$ in this system (1). Lotka
Volterra systems have piqued the interest of several researchers, some of whose works are included below:
S.C. Bhargava described the process of technical substitution and extended the Lotka-Volterra equations
in 1989 (Bhargava, 1989), A hybrid competitive Lotka- Volterra ecosystem was discussed in 2009 by Zhu
and Yin (Zhu et al., 2009). Furthermore, a classical Lotka-Volterra predator-prey model with nonlocal
diffusions and a free boundary was recently researched in September 2022 by Lei Li, Wan-Tong Li, and Mingxin Wang (Li et al., 2022).

Differential equations are evidently used in many scientific domains. Nevertheless, solving them analytically isn't always feasible. Numerical techniques may be applied in these situations to approximate solutions; however, as these techniques frequently entail significant processing costs, it is advantageous to examine how they behave inside neural networks. This study focuses on using neural networks to solve this challenge. Neural networks are becoming more and more popular in scientific computing and approximation theory, with a wide range of applications still to be discovered (Tan et al., 2018). The organic nervous system is the source of the fundamental concept of a neural network. There are neurons in the nervous system. A neuron is a processing unit that, when an electrical input is received and crosses a threshold, transmits messages to other neurons (Zhu, 2009) In 1943, Warren McCulloch and Walter Pitts proposed the first artificial neuron model. A transfer function, inputs, weights, and outputs were all included in this model. McCulloch and Pitts utilized a step function because they chose a model in which a neuron's activity is all or nothing. Because of this, the output is binary, and a neuron is considered to be engaged when its output is one and deactivated when it is zero (McCulloch, 1943). According to (Stanimirovic, 2020) this model may be thought of as a network with weights that reflect the synaptic connections in a biological system. Creating a trial solution is a common step in the neural network solution of ODEs. The purpose of this experimental solution is to meet the specified beginning and boundary criteria (Western, 2023). By modifying weights and biases to minimize approximation error, the neural network approximates the exact/numerical solution to the problem (Lagaris et al., 1998). The mean squared error is frequently computed and used to make the necessary modifications in order to minimize error. Neural networks are a desirable solution for differential equations for a variety of reasons when compared to other approaches.

It cannot be ensured that every differential equation has an exact solution that can be determined analytically, as was previously noted. There could be an answer for certain equations, but it might be rather challenging to locate. Computation time may be greatly reduced by having a neural network assess a differential equation. Consequently, this allows for real-time approximation of complicated differential equations, which opens up a wide range of engineering applications (Lagaris et al., 1998). The created solutions' differentiability is yet another excellent advantage. Discreteizations of the domain are a prerequisite for many other numerical techniques of solving differential equations (Chiaramonte, 2013). As a result, discrete or little differentiable solutions are produced. Conversely, the solutions can be designed to be fully differentiable by the use of an artificial neural network (Lagaris et al., 1998).

This study describes and investigates a neural network approach. Next, this is implemented in the Prey-Predator model. The method's effectiveness is assessed by contrasting it with the Implicit Euler
approach. While a few research have looked at using neural network techniques to solve differential equations, few have looked into systems. The majority of research that examines systems of differential equations doesn’t delve very far into their implementations. Even while the technique is generally generalizable, this does not necessarily translate to an easily modifiable implementation.

The remainder of this essay is structured as follows: Section 2 presents the description of the Prey-Predator paradigm, while Section 3 delves into the case used for this research. The example in part 4 is implemented using the Implicit Euler technique, and in section 5, a neural network approach is used to study the same example. In Section 6, the two methodologies under investigation are compared, and the performance outcomes of each are discussed. In Section 7, the task is concluded and evaluated, and the advantages of each approach are further discussed.

2 THE PREY-PREDATOR MODEL

Following the First World War, there was a noticeable decline in fishing during which the percentage of sharks and other predators (unfit for human eating) was obviously greater than it was before to the conflict. Italian mathematician Vito Volterra created a model to explain the phenomenon that "the predator fish's proportion increases when the fishing decreases" after the Italian fishing authorities in Trieste became aware of it. On the other hand, American mathematician Alfred James Lotka independently published a model that was similar. The Lotka-Volterra model, commonly referred to as the Prey-Predator system, was developed to simulate how the populations of two interacting species, one of which is thought of as a predator and the other as prey, evolve over time. It is a system of two first order nonlinear ordinary differential equations (Ginoux, 2017). The system can be expressed as:

\[
\begin{align*}
\frac{dy_1}{dt} &= \alpha y_1 - \beta y_2 \\
\frac{dy_2}{dt} &= \delta y_1 y_2 - \gamma y_2
\end{align*}
\]

(1)

The model is based on two populations whose numbers at time t the first is \(y_1\) represents the population of the prey, and the second \(y_2\) the population of the predators, the second (predators) feeding on the first (prey). \(\alpha\) the growth rate of the prey, \(\beta\) the loss rate of prey after each encounter with predator, \(\delta\) the growth rate of predators when they meet with the prey and \(\gamma\) the loss rate of the predators. The parameters \(\alpha, \beta, \delta\) and \(\gamma\) are all real positive numbers. The following hypothesis are made:

1. the prey \(y_1\) have unlimited food, only the predators \(y_2\) oppose their increasing and in the absence of predators the prey population has an exponential increasing (Malthus law);
2. the number of predators is limited by the amount of prey available for feeding and in the absence of prey, the predator population has an exponential decreasing (Malthus law);
3. the number of encounters between prey and predators is both proportional to $y_1$ and $y_2$ and therefore proportional to the product $y_1y_2$;
4. the rate of prey's decreasing and the rate of predator's increasing due to these encounters are both proportional to the number of encounters between the two populations.

For more details about different analytical proprieties of Prey-Predator model Eq. (2.1) like the local existence, the existence and positivity of solution, periodicity and equilibrium points see (Li et al., 2022), (Tribut, 2013), (Volterra, 1931), (Volterra, 1926), (Volterra, 1927) and (Volterra, 1928).

3 THE PREY-PREDATOR EXAMPLE

Given that the Prey-Predator model Eq (1) is unsolvable Analytically, the Deer (the prey) vs. Leopard (the predator) shown in Figure (1) is a true case that exists in the wild. To compare performance, a numerical technique was applied to the same example as the neural network. The primary contribution of this study is this comparison, which was made in order to enable a proper discussion of the outcomes of the systems that lacked precise answers. Discussing the benefits and drawbacks of each approach might be done by using both approaches on the same case. The Implicit Euler technique was selected as the numerical approach for this thesis because it can yield better approximations than the explicit method. To demonstrate and explain the computational efficacy of the studied methods, we always employed the same step size, $k$ for each method problem. To obtain an approximation of the solution, the parameters were set in Table (1). The Matlab environment version R2013a compiler was used to do all calculations on an Intel Duo Core 2.20 GHz PC running Microsoft Windows 2007 Professional. The iteration is continued until the end timespan is reached.

Figure 1. Prey-Predator example: deer vs leopard.

Source: Internet

Table 1. Parameters Values of the Prey-Predator model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Prey population growth parameter</td>
<td>1.5</td>
</tr>
</tbody>
</table>
Table (2) displays the outcomes of the numerical approximation of the solutions. The equations' solutions over time are displayed in Figure (2-a), and the phase-space diagram of the solutions is displayed in Figure (2-b). We can see that the approach diverges (in a counter-clockwise manner) for $k = 0.05$ because of an enormous global error; nevertheless, we may take smaller steps to get a useful result. Additionally, Figure (2) demonstrates how the rise in prey fuels the rise in predators. But as the number of predators rises, prey declines and eventually hits a minimum, which causes the predator population to drop as well. Prey grows again as the number of predators declines, and so on. Thus, a never-ending cycle of development and decrease is caused by this dynamic. We note that when the predator's population decreases the prey's increases. So we can say that they have a negative relationship (For more details about the method see (Henrici, 1964)).
approach. Nonetheless, the time taken is still rather short, indicating that the technique may be applied to get a trustworthy estimate in a reasonable amount of time.

Table 2. Performance results of Neural network method

| Elapsed Time (s) | 0.02670035807000724 |

Source: Prepared by the authors themselves.

5 NEURAL NETWORKS FOR SOLVING PREY-PREDATOR MODEL

To solve the system of equations (1) by neural network method (Figure 3), one trial solution is constructed for each equation. By letting $y_{trial1}$ be the trial solution for $y_1$ and $y_{trial2}$ be the trial solution for $y_2$, the trial solutions could look as follows:

$$y_{trial1} = y_1(0) + tN_1(t, \bar{p}_1)$$  \hspace{1cm} (2)  

$$y_{trial2} = y_2(0) + tN_2(t, \bar{p}_2)$$  \hspace{1cm} (3)

where:

$y_1(0)$ and $y_2(0)$ are the given initial conditions and

$N(t, \bar{p})$ represents the output of the neural network with the input $t$ and $\bar{p}$.

To train the system and adjust the weights, a modified version of the error function is needed; it can be written as (For more details about the method see (Western, 2023)):

$$E[\bar{p}] = \frac{1}{n} \sum_{k=1}^{2} \sum_{i=1}^{n} \frac{dy_{trial1}}{dt} - f_k(t_i, y_{trial1}, y_{trial2})^2, n \in N.$$  \hspace{1cm} (3)
We employ a fully connected neural network to execute the solution, which means that all nodes are connected to all nodes in the layers that come before and after, but the activation function has been altered. Activation was previously accomplished using the sigmoid function. However, it was soon discovered that the sine activation function worked better because of the Prey-Predator model’s rhythmic character. Figure (4-a) shows the neural network approximation of the solutions to the Prey-Predator equations shown over time and Figure (4-b) shows the same solutions but plotted against each other in a phase-space diagram. The graphs demonstrate the oscillatory behavior of the system, where a reduction in the number of predators occurs subsequent to a decline in the population of prey.

Table 3. Performance results of Neural network method

<table>
<thead>
<tr>
<th>Elapsed Time (s)</th>
<th>37.6856715410000010</th>
</tr>
</thead>
</table>

Source: Prepared by the authors themselves

The time interval between starting the solution and getting the program’s approximations is also displayed in Table (3). The technique took 37.6856715410000010 seconds to complete, although changes to the network settings can have a significant impact on this number.

It may be inferred from Figure (5) that the model fits the situation rather well. Train loss likewise behaves in accordance with the validation loss’s decreasing trend before becoming more steady. Although the training loss shows some oscillations suggesting that the model would benefit from a few less epochs, but the loss still indicates a good fit.
6 COMPARAISON AND RESULTS

The approximated answers to the case given in the preceding parts are shown together in this section so that the performances of the implicit Euler approach and the neural network method may be compared. Plotting the numerical and neural network approximations of the answers against one another is shown in Figure (6-a). As an alternative, Figure (6-b) displays the phase space diagram for both the numerical and neural network solutions. The similarity between the two approximations is demonstrated in Figure (6-a), but the differences between them are more clearly shown in Figure (6-b). It is clear that both estimates begin similarly and then diverge more with time. Since there isn’t a precise solution to compare this situation with, it is likely that the neural network technique yields a more accurate estimate than the implicit Euler method. However, this is difficult to determine for sure.

It is easier to compare the computational time needed for each method, as seen in Table (4):
Table 4. Performance results of Neural network method

<table>
<thead>
<tr>
<th>Method</th>
<th>Elapsed Time (s)</th>
<th>Neural Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical method</td>
<td>0.02670035807000724</td>
<td>37.685671541000010</td>
</tr>
</tbody>
</table>

Source: Prepared by the authors themselves

7 CONCLUDING REMARKS

This study examined the application of neural networks in solving the model of Prey- Predator which has a wide range of practical uses, but in these situations it is sometimes exceedingly challenging, if not impossible, to identify the precise solution to the issue. If a solution is found, computing the result may take a very lengthy time using the techniques now in use. For the same cases as the neural network, a numerical technique was used to compare performance. The primary contribution of this paper is this comparison, which was carried out to make sure that the outcomes of the systems missing precise solutions. It was evident from comparing the results of the two approaches that the neural net- work constantly outperformed the other in terms of accuracy; yet, there was a noticeable variation in the techniques' execution times. Despite being an implicit method, the numerical method executed very quickly. Of course, this depends on the situation at hand and the step size that is employed. Reducing the step size might have increased the method's accuracy, but doing so would have increased computing time.

Which approach is best will depend on the application. In environments where it is most important to obtain accurate results regardless of computation time, the neural network method presented would be a good fit. In environments where processing time and computational cost are of the essence, however, the Implicit Euler method might be a better fit. Based on the exploration done in this thesis, it does seem reasonable to draw the conclusion that the neural network generally does provide better results but with longer computation time.

The use of hybrid techniques is another option also. These hybrid approaches exploit the neural network's universal approximation capabilities, but they may also employ a numerical technique to solve an equation between the layers. In addition, there is a chance to use a different approach inspired by nature instead of using metaheuristic algorithms.
REFERENCES


Volterra, V. (1928). Variation and euctuations of the number of individuals in animal species living together. Translated by Miss MARY EVELYN WELLS. Journal du Conseil international l'exploration de la mer, Copenhagen, III(I), 3 51.

