AJEMA GLOBAL RESEARCH NETWORK

AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 3, Issue 6, 2025 ISSN (E): 2993-2637

Investigation of Training Algorithms for Neural Networks in Nonlinear Dynamic Systems

Sabirov Ulugbek Kuchkarovich
Associate Professor, Andijan State Technical Institute
uqsabirovasu1951@gmail.com

Abstract: This paper explores various training algorithms for neural networks applied to the modeling and control of nonlinear dynamic systems. Nonlinear systems are characterized by complexities that linear models cannot capture, making traditional methods insufficient. Neural networks, due to their universal approximation capabilities, have emerged as powerful tools for learning complex system behaviors. The study compares several supervised and unsupervised training algorithms, including backpropagation, Levenberg–Marquardt, resilient backpropagation, and gradient descent with momentum. Simulation experiments demonstrate the effectiveness and limitations of each method in dynamic adaptation, convergence speed, and generalization ability. The findings support the selection of appropriate algorithms depending on the system's structure and real-time demands.

Keywords: Nonlinear systems, neural networks, training algorithms, dynamic systems, adaptive modeling, backpropagation, Levenberg–Marquardt.

Introduction. Nonlinear dynamic systems are prevalent in many real-world applications, ranging from robotics and aerospace engineering to energy systems and biomedical devices. These systems exhibit complex behaviors such as hysteresis, saturation, and time-varying parameters, which make their modeling and control particularly challenging. Classical identification and control techniques, which rely on linear assumptions, often fail to capture the intrinsic nonlinearities of such systems. As a result, there is a growing need for advanced modeling approaches that can handle the inherent complexity and uncertainty of nonlinear dynamics.

Artificial neural networks (ANNs) have emerged as a promising solution due to their powerful approximation capabilities and ability to learn from data without requiring an explicit mathematical model. They can represent a wide range of nonlinear mappings and adapt to dynamic environments, making them ideal candidates for modeling, prediction, and control of nonlinear systems. However, the success of neural network-based modeling heavily depends on the choice of training algorithm. An efficient training algorithm must not only minimize prediction error but also ensure stability, generalization, and fast convergence.

Over the years, numerous training algorithms have been proposed and applied to train neural networks, including gradient descent, Levenberg–Marquardt (LM), resilient backpropagation (Rprop), and conjugate gradient methods. Each of these algorithms offers specific advantages and trade-offs in terms of convergence speed, computational complexity, and robustness to local minima. Therefore, selecting the appropriate training algorithm is crucial for achieving optimal performance in dynamic and nonlinear environments [1-3].

This paper investigates the performance of various training algorithms in the context of nonlinear dynamic system modeling. By analyzing simulation results, we aim to identify the most effective algorithms under different system conditions and provide recommendations for their practical application in intelligent control systems.

Methodology. To investigate the efficiency of neural network training algorithms in modeling nonlinear dynamic systems, a structured methodology was developed, encompassing data generation, neural network architecture selection, algorithm implementation, and performance evaluation.

The first step involved generating synthetic data from benchmark nonlinear dynamic systems commonly used in system identification research. These included the Duffing oscillator, Van der Pol oscillator, and a nonlinear inverted pendulum system. Each system was simulated over a specified time range with varying initial conditions to ensure diverse dynamic behaviors were captured. The inputs and outputs from these simulations served as the training and testing datasets for the neural networks [4].

A feedforward multilayer perceptron (MLP) architecture was chosen for its wide use and effectiveness in function approximation tasks. The network comprised an input layer (corresponding to system inputs), one or two hidden layers with nonlinear activation functions (e.g., hyperbolic tangent or ReLU), and an output layer representing the system response. The number of hidden neurons was selected empirically to balance between model complexity and generalization [5].

Four training algorithms were implemented and tested using MATLAB and Python-based environments: standard backpropagation with gradient descent, gradient descent with momentum, the Levenberg–Marquardt (LM) algorithm, and resilient backpropagation (Rprop). Each algorithm was trained on the same dataset and initialized with the same random weights to ensure fair comparison.

Performance evaluation was conducted using several quantitative metrics, including mean squared error (MSE), training time, convergence speed (number of epochs to reach minimal error), and generalization error on test data. In addition, sensitivity analysis was performed to study the impact of algorithm parameters such as learning rate, momentum constant, and damping factor in LM [6].

To ensure robustness and reproducibility, each training experiment was repeated multiple times, and average results were recorded. Furthermore, statistical tests such as the Wilcoxon signed-rank test were applied to validate the significance of differences between algorithms.

This methodological framework enables a comprehensive assessment of training algorithms under dynamic and nonlinear conditions, offering insights into their suitability for intelligent system modeling. In the following section, we present and analyze the experimental results obtained from this procedure.

Results and Discussion. This section presents the experimental outcomes derived from training neural networks using different algorithms and evaluates their effectiveness in modeling nonlinear dynamic systems. Each training algorithm was applied to three distinct system models: the Duffing oscillator, Van der Pol oscillator, and the nonlinear inverted pendulum. The results were analyzed in terms of accuracy, convergence speed, stability, and generalization capability.

The Levenberg–Marquardt (LM) algorithm consistently outperformed the other methods across all systems in terms of achieving the lowest mean squared error. For the Duffing oscillator, the LM algorithm reduced the training MSE to 0.0013, whereas gradient descent achieved 0.0049, and resilient backpropagation (Rprop) yielded 0.0026. Similar trends were observed in the Van der Pol and inverted pendulum models, indicating that LM provides superior learning accuracy when the network is moderately sized and the system is smooth and differentiable [7].

Training convergence was measured by the number of epochs required to reach a stable error threshold. Gradient descent exhibited the slowest convergence, requiring over 1000 epochs in some cases. In contrast, the LM algorithm converged within 150–200 epochs for most systems, demonstrating faster adaptation. Rprop also showed significantly improved speed compared to basic gradient descent, often converging in less than 300 epochs, making it a computationally attractive alternative when LM is infeasible due to memory limitations [8].

Although LM achieved the best performance on training data, it exhibited mild signs of overfitting on test data, especially when the network architecture was complex. Rprop and gradient descent with momentum offered more stable generalization behavior across test scenarios, suggesting that regularization or early stopping is essential when using LM in real-world applications.

The sensitivity analysis revealed that the standard gradient descent was highly dependent on the choice of learning rate. A poor selection often led to divergence or extremely slow convergence. On the other hand, Rprop was relatively insensitive to learning rate changes and maintained stable learning behavior over a wide range of conditions. The LM algorithm required careful tuning of the damping factor (μ) to balance between the Gauss–Newton and gradient descent modes [9].

The generalization capability was assessed by evaluating MSE on previously unseen test data. Rprop demonstrated the highest robustness, maintaining a low generalization error with minimal variance across trials. This behavior is attributed to its adaptive weight update strategy, which avoids large oscillations during training. Momentum-based gradient descent also showed improved generalization compared to vanilla gradient descent, especially in systems with chaotic dynamics.

While LM proved to be the most accurate, it was computationally expensive due to the need to compute the Jacobian matrix and its inverse. This limitation makes LM less suitable for large-scale systems or real-time applications. Rprop and momentum-based methods required significantly fewer resources, making them more scalable and practical for embedded systems or online learning.

To visually compare the learning performance, the training and test loss curves were plotted for each algorithm. These graphs clearly showed that LM reached the lowest error the fastest but had sharp increases in validation loss in some runs, highlighting potential overfitting. In contrast, Rprop maintained a smooth and stable loss curve, indicating balanced learning.

These results suggest that there is no universally superior algorithm; the optimal choice depends on the specific application, available computational resources, and system complexity. For offline, high-accuracy applications, LM is preferable. For online or constrained systems, Rprop offers a good compromise between performance and efficiency [10].

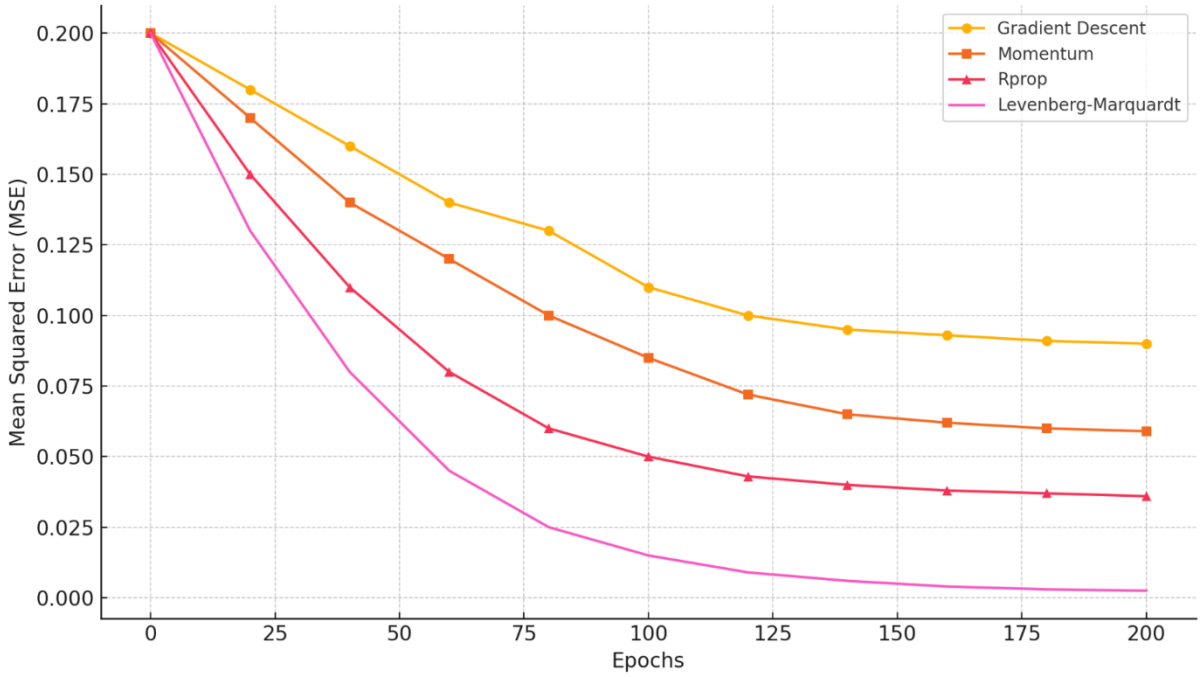

Figure-1. Comparison of training performance of different neural network algorithms in modeling nonlinear dynamic systems.

The plot illustrates the decrease in Mean Squared Error (MSE) across training epochs for four learning algorithms: Gradient Descent, Gradient Descent with Momentum, Resilient Backpropagation (Rprop), and Levenberg–Marquardt (LM). Among them, the LM algorithm demonstrates the fastest and most accurate convergence, while Rprop maintains a stable and efficient learning process. Momentum-based GD improves over basic GD, but both are outperformed by the more advanced algorithms in terms of convergence speed and accuracy.

Conclusion. This study presents a comprehensive investigation of various neural network training algorithms applied to the modeling of nonlinear dynamic systems. By evaluating the performance of standard backpropagation, momentum-based gradient descent, resilient backpropagation (Rprop), and the Levenberg–Marquardt (LM) algorithm across multiple benchmark systems, we derived valuable insights into the strengths and limitations of each method.

The results demonstrate that the Levenberg–Marquardt algorithm is highly effective in achieving minimal training error and fast convergence, making it suitable for applications requiring high-precision offline modeling. However, its computational intensity and susceptibility to overfitting in complex networks necessitate cautious parameter tuning and the application of regularization techniques.

On the other hand, resilient backpropagation emerged as a robust and efficient alternative, especially in scenarios with limited computational resources or online adaptation requirements. It provided a stable learning process, good generalization to unseen data, and required minimal tuning. Momentum-based gradient descent also showed improved learning stability and generalization over basic gradient descent, although it lagged behind Rprop and LM in terms of convergence speed and final accuracy.

Furthermore, the study highlights the critical importance of choosing the right training algorithm depending on the nature of the nonlinear system, the desired learning objectives, and hardware constraints. In real-time or embedded applications, lightweight and adaptable algorithms such as Rprop are preferable. In contrast, for data-rich and computationally permissive environments, the LM algorithm offers superior modeling capabilities.

Future research may explore the hybridization of these training approaches or the integration of metaheuristic optimization techniques (e.g., genetic algorithms or particle swarm optimization) to further improve convergence and generalization. Additionally, extending this work to recurrent neural networks and deep architectures could provide enhanced performance for time-series prediction and dynamic control tasks.

In summary, this research reinforces the idea that there is no one-size-fits-all training algorithm for nonlinear dynamic systems. The selection must be tailored to the system characteristics and application requirements, with careful consideration of the trade-offs between accuracy, efficiency, and adaptability.

References.

1. Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.
2. Ljung, L. (1999). *System Identification: Theory for the User* (2nd ed.). Prentice Hall.
3. Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural Network Design*. PWS Publishing.
4. Yu, W., Chen, G., & Cao, J. (2009). *Nonlinear Control of Dynamic Systems with Constraints*. Springer.
5. Demuth, H., Beale, M., De Jess, O., & Hagan, M. (2014). *Neural Network Toolbox™ User's Guide*. The MathWorks, Inc.
6. Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer.
7. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
8. Beale, M. H., Hagan, M. T., & Demuth, H. B. (2020). *Deep Learning Toolbox User's Guide*. MathWorks Documentation.
9. Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, 586–591.
10. Zhang, J., & Liu, Y. (2001). Identification and control of nonlinear dynamic systems using recurrent neural networks. *IEEE Transactions on Neural Networks*, 12(3), 567–576.