The profit optimization model relies on selecting production output and pricing using deep learning and reinforcement learning techniques

Truong Thi Chi[1], Hoang Do Thanh Tung[2], Ly Van Kien[3] Pham Van Khanh*[2]

[1]Thuong Mai University, chi.tt@tmu.edu.vn

[2] Institute of Information Technology (IoIT), Vietnam Academy of Science and Technology (VAST)

tunghdt@gmail.com, khanhvietdm@gmail.com,

[3] Graduate University of Sciences and Technology (GUST), Vietnam Academy of Science and Technology (VAST)

lyvankien@gmail.com

Abstract

The rapid development of markets and the increasing diversity of consumer preferences have driven businesses to offer products with varying quality levels while optimizing their supply chains to balance production costs with market demand. This study introduces a model that integrates deep learning and reinforcement learning to optimize production and pricing decisions. The model employs deep neural networks to predict product demand with high accuracy, followed by the application of reinforcement learning to make optimal real-time decisions regarding production and pricing. The results demonstrate that this model not only helps businesses optimize profits but also enhances their competitiveness in the highly volatile market environment.

Keyword: *optimization, supply chains, deep learning, reinforcement learning*

1. Introduction

The rapid evolution of markets and the growing diversity of consumer preferences have presented significant challenges for businesses aiming to maintain competitiveness [1]. To meet the changing demands of consumers, companies must optimize their production and pricing strategies, particularly when considering multiple product quality levels. However, optimizing supply chains in such a complex environment requires more than balancing production costs and profits; it also demands accurate market demand forecasting and flexible production strategy adjustments. [3].

In this context, deep learning (DL) and reinforcement learning (RL) emerge as potential technologies that offer comprehensive solutions to supply chain optimization challenges. DL helps businesses accurately predict market demand based on historical data and economic variables, while RL enables optimal production and pricing decisions to maximize profits [2]. The research team presents an integrated model that combines DL and RL to address these challenges. Specifically, the model is designed to accurately predict product demand using a Deep Neural Network (DNN) and simultaneously apply RL to optimize production and pricing strategies in real-time. The ultimate goal is to maximize business profits in a highly competitive and constantly changing market environment. Through this research, we hope to provide a comprehensive analytical and optimization framework for businesses, enabling them to promptly capture market trends and make strategic decisions with the highest efficiency.

Problem: Estimate demand → develop an optimal pricing strategy → achieve maximum profit.

Input:

Historical sales data: Includes product sales volume, pricing, promotions, seasonality, and economic indicators.

Product quality levels: Different quality levels of the product considered for production.

Predicted demand data: Generated by a Deep Neural Network (DNN) based on market factors and historical data.

Current inventory level: Real-time data on the quantity of products in stock.

Cost data: Unit production costs and any other associated costs.

Output:

Product demand forecast: Accurate demand prediction using the DNN model.

Optimal production decision: The quantity of each product quality level to be produced based on the predicted demand.

Optimal pricing strategy: Dynamic pricing decisions that maximize revenue based on current demand, inventory levels, and production costs.

Profit maximization strategy: A reinforcement learning-based approach that adjusts production and pricing over time to maximize cumulative profit.

Policy recommendations: Guidance for businesses on how to flexibly adjust their strategies in response to market changes.

The supply chain optimization model, combining deep learning and reinforcement learning, is illustrated in the diagram below.

2. Deep Learning-Based Demand Prediction Model

With the advancement of machine learning and artificial intelligence, several advanced aggregation algorithms and deep learning-based time series forecasting methods have demonstrated high accuracy and robustness. These algorithms have become increasingly essential in addressing price prediction challenges.

A Deep Neural Network (DNN) model is constructed to predict future product demand based on influencing factors such as product quality, pricing, promotions, seasonality, and economic indicators.

Objective: To predict product demand based on historical data and market factors.

- **Input Layer:** Receives input data (factors influencing demand such as product quality, pricing, promotions, etc.).
- **Hidden Layers:** Consist of multiple consecutive layers where the model learns complex relationships from the data.
- **Output Layer:** Provides the final prediction, which in this case is the product demand forecast.

The output layer has 1 node, representing the predicted product demand value.

Data include:

- Input Data (X):

+ Product Quality Level

+ Price

+ Promotion

+ Seasonality + Economic indicator - Target Data (y):

+ Product Demand

*** Model**: Using Deep Neural Networks to Predict Demand

$\hat{D}_t = f(X_t;\square)$

The function $f(X;\square)$ represents the Deep Neural Network (DNN) model used to predict demand, where f is the neural network function parameterized by $\square$. Specifically:

- **X**: Represents the input factors (such as product quality, pricing, promotions, seasonality, and economic indicators).
- $\square$: Represents the set of parameters (including weights and biases) learned by the model during training. Initially, these parameters θ are randomly initialized. - **Loss Function:** The loss function measures the difference between predicted and actual values. In this case, the loss function used is the **Mean Squared Error** (MSE): $L(\square) = N1 \square_{iN=1} (D_i - f(X_i;\square))^2$

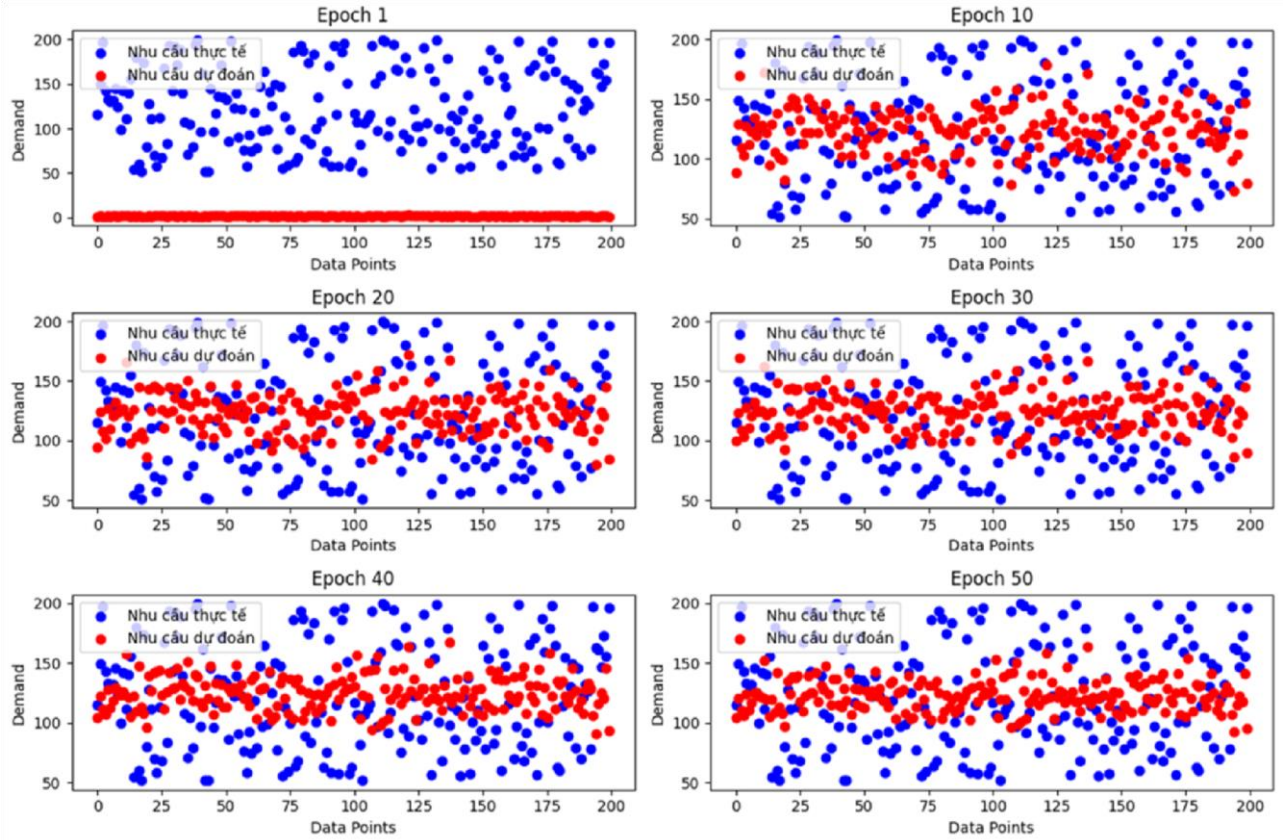
—

Where:

- D: Is the actual demand value that you want to predict..
- $f(X,\square)$ Is the prediction made by the model based on the input factors and parameters θ
- N: Is the number of samples in the dataset..

The model is trained with three hidden layers containing 64, 32, and 16 nodes respectively. These layers use the ReLU activation function, with 50 epochs and a batch size of 32. The results are as follows:

Comparison of actual and predicted demand across different training epochs.



- **Epoch 1:** The predicted demand is far from the actual demand. The red points are primarily clustered around lower values (close to zero), indicating that the model has not been properly trained and is significantly underestimating demand.
- **Epoch 10:** The predicted values begin to spread out and get closer to the actual demand values. However, there are still significant discrepancies, with some predictions being either overestimated or underestimated.
- **Epoch 20:** The predictions improve as the red points become more aligned with the blue points. The model is starting to capture the actual demand pattern, although some gaps still exist.
- **Epoch 30:** The model's predictions continue to improve. The red points are more evenly distributed around the blue points, indicating better prediction accuracy. The difference between predicted and actual demand is decreasing.
- **Epoch 40:** The predictions now align even more closely with actual demand. The red points follow the distribution of the blue points more closely, reflecting that the model is learning and capturing the underlying trends in the data more effectively.
- **Epoch 50:** The predicted demand almost matches the actual demand. Although there are still some minor deviations, the overall performance of the model has become much more consistent.

- **Summary:** The charts clearly illustrate how the model improves with more training epochs. Initially, the model struggles to predict demand accurately, but as it progresses through more epochs, it learns to approximate actual demand more closely. The model effectively reduces prediction errors as training progresses, proving that it is learning and adapting as expected..

3. Reinforcement Learning-Based Decision-Making Model

- **Objective:** To maximize cumulative profit by making production and pricing decisions based on predicted demand.

- Definitions:

+ State (s_t)*:* Includes factors such as current inventory levels, predicted demand, selling price, production costs, and other market factors. The state reflects the business situation at each point in time and is continuously updated.

$$s_t = [I_t, \hat{D}_t, P_t, ...]$$

where:

I: Represents the inventory level of the product at a specific time.

D: Represents the predicted demand for the quantity of products that customers will purchase within a specific cycle.

P: Represents the selling price of the product at a specific time.

Meaning: The state s_t is a vector that includes critical information:

I_t: Current inventory level.

$\hat{D}_t$: Predicted demand at time t.

P_t : Current selling price.

+ Action (a_t)*:* Refers to production quantity and pricing strategy. The action includes deciding the production quantity and pricing strategy for each product level. The model will select actions that optimize profits based on predicted demand and other market factors. a^t is a vector containing two parameters:q^t (production quantity) and p_t (selling Action price) at time t $a_t = [q_t, p_t]$

Where:

q: Production quantity, which determines the output needed to meet demand while optimizing costs.

+ Reward (r_t)*:* The profit at time *t*. It is the profit at each time point, calculated based on revenue (selling price multiplied by the quantity sold) minus production costs. The reinforcement learning model learns from this reward to adjust actions in the future, aiming to maximize cumulative profit.

$$r_t = (p_t.\min(\hat{D}_t, q_t)) - (C.q_t)$$

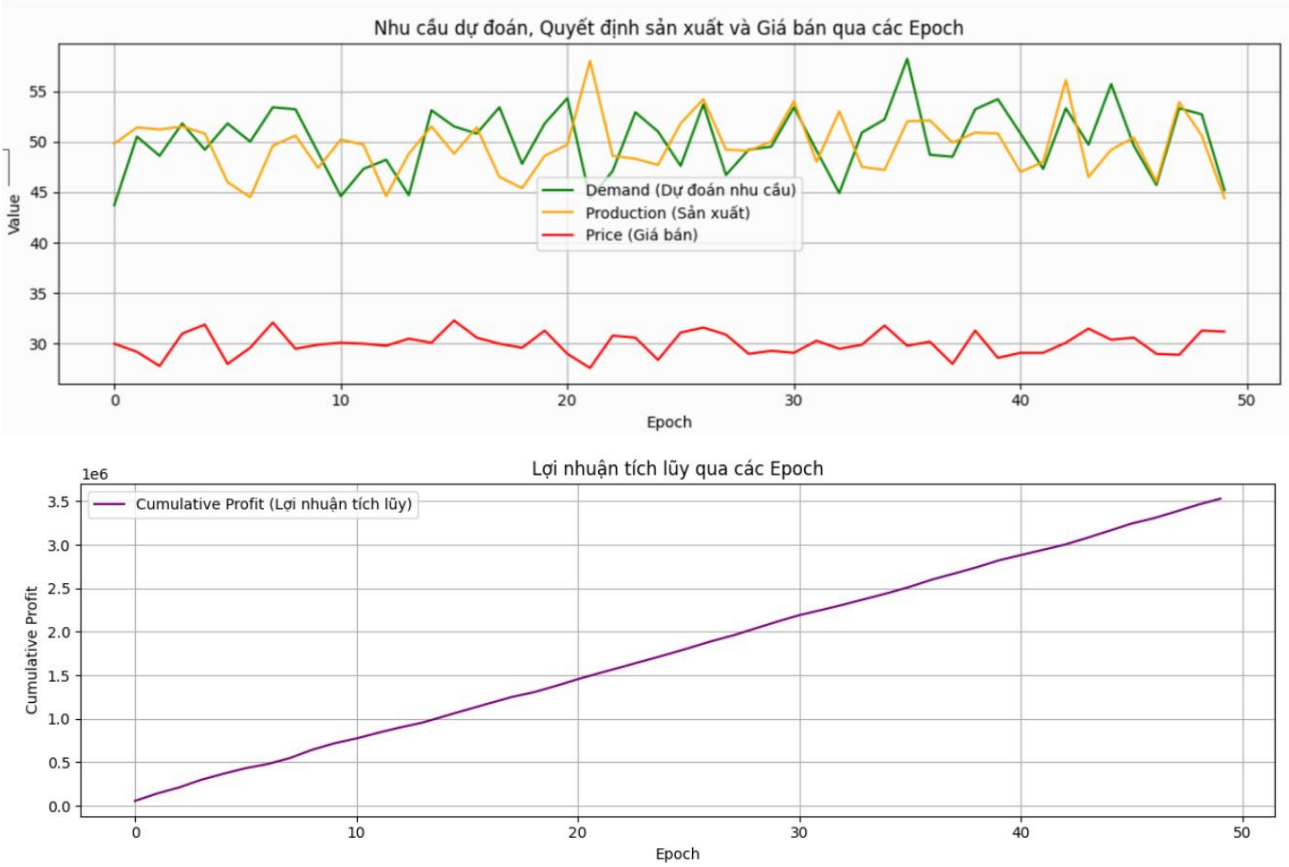
Revenue: $p_t.\min\left(\hat{D}_t, q_t\right)$, Where:

$\hat{D}_t$: Predicted demand.

q_t : Production quantity.

The function $\min(\hat{D}_t, q_t)$ ensures that revenue is only calculated based on the number of products sold, which does not exceed the predicted demand or production quantity. Production Cost: Cq_t, where C is the unit production cost.

Model Training: The model is trained with a unit production cost (C) = 5; 50 epochs, and 100 steps per epoch.



The model demonstrates effective learning over time, with production decisions closely aligning with predicted demand and a stable, optimized pricing strategy. The stability in pricing, combined with adaptive production, indicates that the reinforcement learning aspect of the model is functioning well, leading to consistent decision-making aligned with business objectives.

4. Policy Optimization

- Policy (π) : The policy function is parameterized by θ

$\pi(a_t|s_t;\theta)$ a^{t} based on the current state s_t, with the

objective of

The model learns how to select actions

maximizing cumulative profit. The parameters θ will be optimized during the learning process to find the best policy.

- π**:** This notation represents the policy function. The function determines the probability or decision of selecting action a_t based on the current state s_t.
- a_t**:** This represents the action taken at time t. In the context of supply chain management, this action may involve determining the production quantity and selling price.
- s_t**:** This represents the system's state at time t, including factors such as inventory levels, predicted demand, and selling price. This state provides information about the current situation of the system to support decision-making..
- θ**:** These are the parameters of the policy function. In deep learning, these parameters are typically the weights of the neural network, which are updated during the training process to optimize the policy.

- **Value Function** (V): Represents the expected value of cumulative rewards from state s_t

$$V(s_t) = \mathbb{E}\left[\sum_{k=0} \gamma^k r_{t+k} \middle| s_t\right]$$

Where:

$V(s_t)$**:** This is the value of state s_t at time t. This value function represents the expected s_t and following the current policy. total rewards in the future when starting from state

E**:** Represents the expectation. The expected value is calculated based on all possible future states and actions.

γ

$\sum_{k=0} \gamma^k r_{t+k}$**:** The cumulative rewards over future time periods, where k is the number of steps from the current time:: r_{t+k} **:** The reward received at time t+k.

γ **(discount factor):** A factor between 0 and 1 that indicates the importance of future rewards relative to current rewards. When γ is close to 1, the model will prioritize longterm rewards. Conversely, if γ is close to 0, the model will prioritize short-term rewards.. s_t**:** The current state. The value function $V(s_t)$ indicates the value of state s_t when considering all possible future rewards starting from this state.

- **Objective:** To learn the optimal policy π*\pi^*π* that maximizes the expected cumulative rewards:

$$\theta^* = \arg\max_\theta \mathbb{E}\left[\sum_{t=0}^{T} \gamma_t r_t\right]$$

□□: This is the optimal policy we aim to find. The optimal policy will determine how to choose actions a_t at each state s_t to maximize long-term cumulative rewards. In the context of the supply chain model, this involves determining the optimal strategy for production quantity and pricing based on the current state (inventory, predicted demand, pricing).

*agr*max□: This notation means finding the policy □ that maximizes the expected value of the total discounted rewards. The optimal policy □□ is the one that maximizes this value. *E*□□□: Represents the expectation, calculated based on the different possible outcomes of future rewards. This expectation depends on the current policy π .

T

□□$^t r_t$: This is the total discounted cumulative reward from time t=0 to T=t

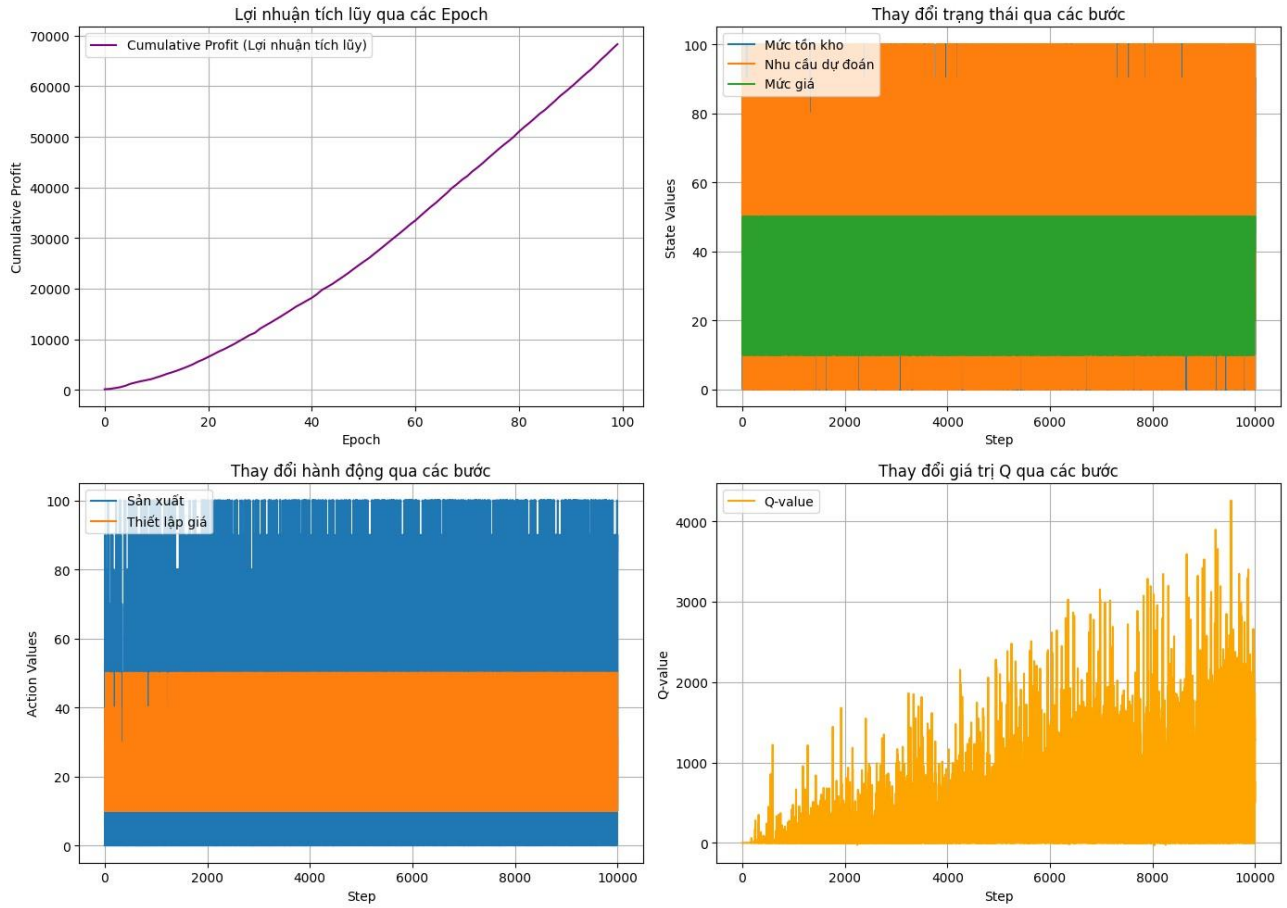
$t=0$

r_t : This represents the profit obtained at time t when the company makes decisions about production and pricing.

□ ^{t}: discount factor, Helps the company weigh the importance of current profits relative to future profits 0 □□□1. . When γ is close to 1, the model will prioritize long-term rewards.

Model Training: The model is trained with the following parameters:

□= 0.95; Learning rate = 0.1; Number of iterations = 100; Steps per iteration = 100; Unit production cost = 5; epsilon=0.1; Initial inventory level 10, Predicted demand 50, Selling price 30



The charts collectively demonstrate that the model has effectively learned an optimal strategy. The consistent growth in cumulative profit, stability in state parameters and actions, and

increasing Q-values all indicate a well-trained reinforcement learning model. This model successfully balances demand forecasting, production, and pricing to achieve maximum profit.

5. Integration of Deep Learning and Reinforcement Learning

Demand Forecasting: This step is crucial because it provides foundational information about the quantity of products that need to be produced. If demand forecasting is inaccurate, subsequent decisions regarding production and pricing will not be optimal, leading to wasted resources or missed market opportunities.

$$\hat{D}_t = f(X_t; \theta^*)$$

Reinforcement Learning: This is the step where the model makes production and pricing decisions based on predicted demand. The reinforcement learning model learns from past experiences, using rewards to adjust actions to maximize profits. This means the model will continuously improve its decisions through trials and adjustments, helping to optimize business efficiency.

State, Action, and Reward: These are the fundamental components of reinforcement learning. The state provides current information, actions are the decisions that the model can make, and rewards are the outcomes of those actions. Policies and value functions help the model determine the best course of action based on the current state.

+ State: $s_t = [I_t, \hat{D}_t, P_t, \ldots]$

+ Action: $a_t = [q_t, p_t]$

+ Reward: $r_t = (p_t . \min(\hat{D}_t, q_t)) - (C . q_t)$

+ Policy: $\pi(a_t | s_t; \theta)$

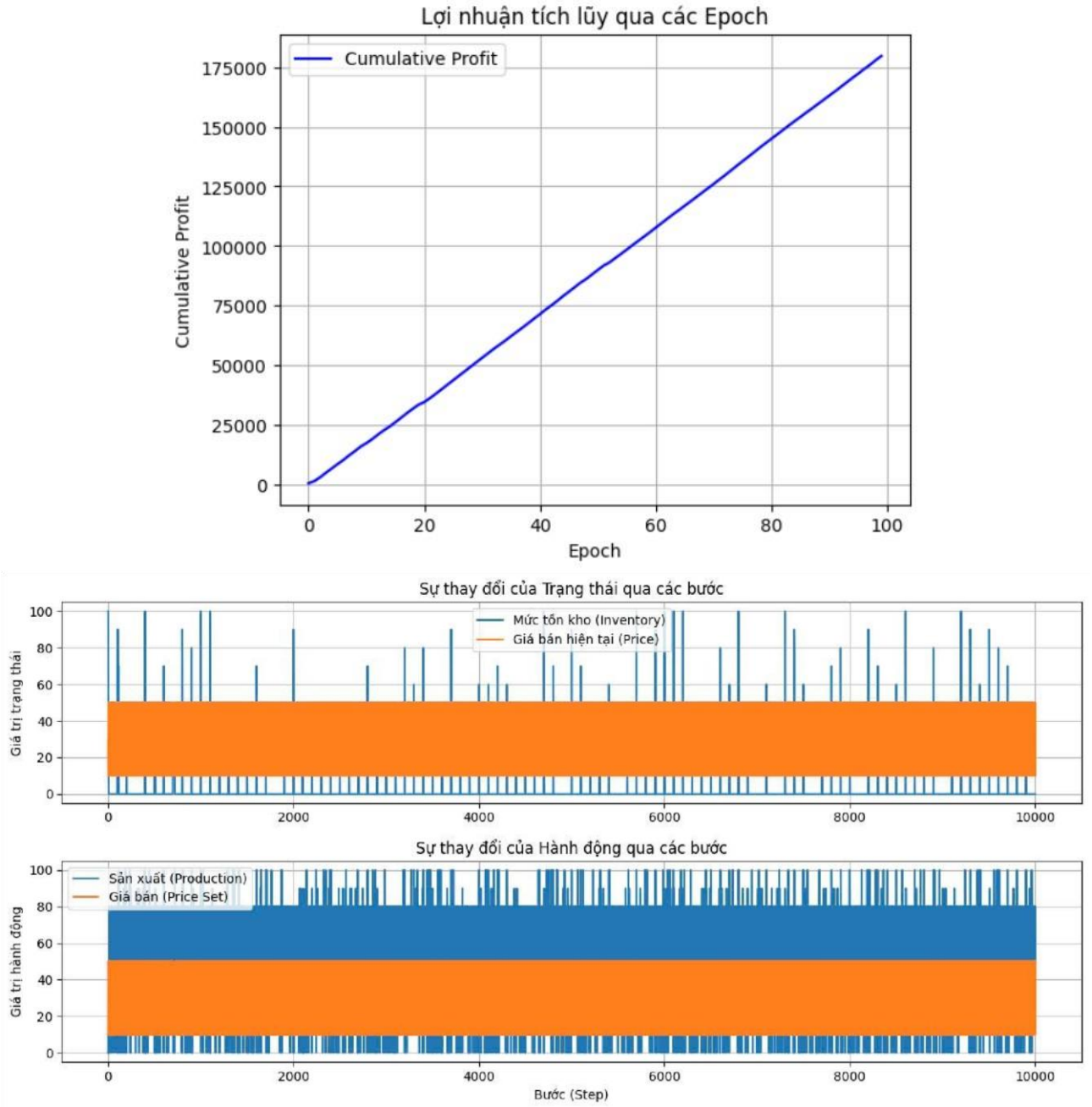
+ Value function: $V(s_t) = \mathbb{E}\left[\sum_{k=0}^{T} \gamma^k r_{t+k} | s_t\right]$

+ Objective: $\theta^* = \arg\max_\theta \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$

Objective: The primary goal of the model is to optimize cumulative profit. This ensures that the model not only focuses on short-term gains but also considers long-term benefits, helping businesses achieve sustainable growth in a volatile market environment.

$$\theta^* = \arg\max_\theta \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Model Training: Train the model with the following parameters: Gamma = 0.95; learning rate = 0.1; epochs = 100; iterations = 100; epsilon = 0.1.



The model's performance in these charts demonstrates a successful learning process. The stable pricing (in both the state and action charts) indicates that the model quickly achieved an optimal pricing strategy. The fluctuations in inventory and production values suggest that the model focuses on dynamically managing production levels to match demand while maintaining profitability. The stability in pricing combined with adaptive production adjustments is a strong indicator of an optimized decision-making process, balancing demand satisfaction and cost efficiency.

6. Conclusion and Future Research Directions

This study introduced an integrated model combining deep learning and reinforcement learning to optimize supply chains in an increasingly competitive market environment with diverse consumer preferences. This is a complex issue that requires businesses not only to accurately predict demand

but also to make flexible and efficient decisions regarding production and pricing. The results indicate that:

Effectiveness of the Integrated Model:

Deep Learning: This was applied to accurately forecast demand based on factors such as sales history, pricing, promotions, and economic indicators. This helps businesses better understand consumption trends and make strategic decisions based on accurate forecasts.

Reinforcement Learning: This was used to determine optimal production and pricing decisions. The model learns from actual outcomes, continuously adjusting its strategy to maximize long-term profits.

Practical Applicability:

The model is designed to fit complex and rapidly changing business environments, particularly when consumer demand is highly volatile.

Businesses can use this model to determine optimal production quantities, pricing levels, and to optimize overall supply chain strategies, thereby enhancing market competitiveness.

Limitations:

Although the model has achieved positive results, there are still some limitations such as parameter selection, the complexity of the model, and scalability issues as data volume increases.
External factors such as policy changes and market volatility may not be fully accounted for in the current model.

Future Research Directions:

Improving Deep Learning Accuracy: Utilizing more advanced techniques such as Transformer models or LSTM variants to enhance predictive capabilities in highly volatile scenarios.

Optimizing Reinforcement Learning: Applying advanced reinforcement learning algorithms like Proximal Policy Optimization (PPO) or Deep Q-Network (DQN) to improve the ability to find optimal policies in a shorter timeframe.

Incorporating External Factors: Integrating external variables such as policy changes, economic events, or unexpected incidents to increase the comprehensiveness of decision-making processes.

Expanding Application Scope: Applying the model to various other industries such as agriculture, food, and healthcare, where supply chains are complex and require high-accuracy forecasting capabilities.

References

[1] Cui, Y., Yao, F. Integrating Deep Learning and Reinforcement Learning for Enhanced Financial Risk Forecasting in Supply Chain Management. J Knowl Econ (2024).

[2] Xiaoya Han, Xin Liu, Equilibrium decisions for multi-firms considering consumer quality preference, International Journal of Production Economics, Volume 227 (2020).

[3] G. Avinash, V. Ramasubramanian, Mrinmoy Ray, Ranjit Kumar Paul, Samarth Godara, G.H. Harish Nayak, Rajeev Ranjan Kumar, B. Manjunatha, Shashi Dahiya, Mir Asif Iquebal, Hidden

Markov guided Deep Learning models for forecasting highly volatile agricultural commodity prices, Applied Soft Computing, Volume 158, 2024, 111557, ISSN 1568-4946.

[4] Fei, Jiang. (2023). How Deep Learning Affect Price Forecasting of Agricultural Supply Chain?. Journal of Information Science and Engineering. 10.6688/JISE,202307_39(4).0007.