ISSN 2706-655X Vol. 7, Issue 4: 77-82, DOI: 10.25236/IJFET.2025.070411

Online Update and Adaptive Optimization Algorithm of Machine Learning Decision Model in Dynamic Environment

Hongyan Qu

King Graduate School, Monroe University, New Rochelle, NY 10801, USA

Abstract: *Dynamic environments bring many problems for machine learning models such as data distribution shift, change in task objectives, and environmental interference. The traditional static models without adaptive mechanism tend to have its performance decline. To solve this problem, this paper gives an online updating and adaptive optimization method for dynamic environment. We create an online update architecture which uses sliding windows along with incremental learning so as to perform real time modification of the model according to changes in the data stream. Adaptive learning rate optimization algorithms for different time scales are designed to balance between fast response and long term stable performance. We add meta-learning strategies so we can adjust model parameters dynamically and also do cross-task transfer. We form a self-evolutionary learning framework and mix it up with a reinforced feedback system so our model could self-improve when interacting with its environment. From experimental results using both synthetic and real world data, we show that this approach has better prediction accuracy and stability than online methods as well as lower resource usage. This kind of dynamic learning framework effectively addresses the problem of model degradation in nonstationary environments and provides a scalable theoretical and engineering foundation for the longterm adaptation and continuous evolution of intelligent systems.*

Keywords: *Adopting adaptive optimization techniques; enhancing feedback*

1. Research Background and Significance of Machine Learning Models for Dynamic Environments

Facing a dynamic environment, the requirement for adaptability of intelligent systems has become one important research direction in machine learning. As applications get more complicated, the environment states for smart making, finance danger regulation, and self-driving cars tend to show non-stationary and dynamic features, mainly due to alterations in data distributions, outside disruptions, and goal shifts [1]. The traditional static model under the i.i.d assumption has fixed parameters after off-line training and no self-adjustment mechanism. It lacks responsiveness to environmental changes, leading to a decline in prediction accuracy and erroneous decision-making. Manual retraining model is expensive, slow to respond, and fragile in high dimensional and heterogeneous data scenarios [2]. Building models which can be updated in real time and optimized adaptively is the key for the stability and efficiency of intelligent systems. From international studies we know that incremental learning on stream data is possible through online gradient descent and perceptron algorithm and distribution change can be handled using drift detection method such as ADWIN and DDM. Meta-learning, reinforcement learning and Bayesian online learning are introduced into the adaptive optimization framework to learn update strategies, adjust learning rates and improve model robustness. In the industries and transportation sectors, it has been proven that adaptive optimization mechanisms are superior in non-stationary environments. Domestic research also focuses on streaming learning, transfer learning and cloud-edge collaboration, and constructs structural evolution and parameter self-tuning models through genetic algorithms and reinforcement learning [3]. The overall trend reveals that research is moving from static parameter updates to system-level adaptive optimization, relying on environmental perception and feedback mechanisms to drive the continuous evolution of models, providing theoretical and technical support for the autonomous decision-making of intelligent systems in complex dynamic environments.

ISSN 2706-655X Vol. 7, Issue 4: 77-82, DOI: 10.25236/IJFET.2025.070411

2. Proposing the problem of machine learning model degradation in dynamic environments

The data distribution in dynamic environments has significant non-stationary and time-varying characteristics, and its changes come from external disturbances, system state transitions, or the evolution of data generation mechanisms [4-5]. Distribution changes are usually manifested as dynamic adjustments of marginal distributions $P(X)$, conditional distributions $P(Y \mid X)$, or joint distributions $P(X,Y)$, causing the relationship between inputs and outputs to evolve over time. For example, in financial and manufacturing systems, environmental factors or equipment aging can cause the sample distribution to drift continuously. To cope with such changes, it is necessary to dynamically characterize data distribution changes through methods such as time series modeling, sliding window sampling, and distribution reweighting , thereby providing a basis for model updates.

Concept drift and task goal drift are key factors that lead to model failure in dynamic environments. Concept drift refers to $P(Y \mid X)$changes in conditional distributions, that is, dynamic shifts in the relationship between inputs and outputs; task goal drift refers to $P(Y)$the evolution of labels or decision goals. Both can be categorized as sudden, gradual, and cyclical, each with varying impacts on model stability. A formal definition is: when t_1,t_2, $P_{t_1}(X,Y)\neq P_{t_2}(X,Y)$, the system is in a drift state. Accurately identifying the type and rate of drift is fundamental to adaptive model updates.

Model degradation is a direct consequence of drift, manifested as decreased prediction accuracy and unstable decision boundaries. As the data distribution shifts, the model's expected loss $\mathrm{E}_{(X,Y)\sim P_t}[L(f_\theta(X),Y)]$increases and generalization performance gradually decreases. The degradation process can be divided into latent, acceleration, and collapse stages. Influenced by factors such as feature drift, target shift, and structural rigidity, long-term accumulation will lead to systematic misjudgments, weakening the model's robustness and credibility [6-8] .

To characterize the above problem, a mathematical model of dynamic learning can be established. Assume that the data distribution is $P_t(X,Y)$, the parameters are θ_t, and the goal is to minimize the time-varying loss function:

$$\min_{\theta_t} \mathrm{E}_{(X,Y)\sim P_t}[L(f_{\theta_t}(X),Y)]+\Omega(\theta_t,\theta_{t-1}),$$

Control parameter updates are made smooth. Data change exhibits Markov property, then we can use hidden state to do advance adjustment and evolve prediction distribution. In this framework, it's assumed that distributions vary continuously and steadily, models' updating speed is slower than environment's changing rate, and there exists an observable feedback signal which can correct deviation. It gives theory basis and modeling basis for online update and adaptive optimization.

3. Online Update & Adaptive Optimization Problem Theoretical Analysis

Data distribution drift detection is important to keep model effectiveness in a dynamic environment. Identify significant changes in input feature or label distribution over time. According to different detection mechanisms, they can be divided into three categories: statistical tests, sliding windows, and model uncertainties [9-10]. Statistical tests such as KS test and MMD depend on distribution difference measurement to perform drift identification, which is appropriate for large-scale data with slow distribution change; Sliding window methods such as ADWIN and DDM adopt dynamic modification of time windows to suit data flow variations and possess strong real-time capabilities; Uncertainty-based detection depends on changes in the model's prediction confidence or entropy, and it can spot drift even without labels or late feedback. During the online model updating phase, it is critical to strike a balance between timeliness and stability[11-13]. If we update too often, it will cause the interaction between feedback and the environment in the adaptive mechanism further enhances the evolutionary ability of the model. Feedback information can be derived from prediction errors, environmental rewards or reinforcement signals to build a closed-loop system of decision-making-response-correction. Reinforcement learning feedback models optimize strategy direction through reward functions, and Bayesian feedback achieves uncertainty correction through posterior updates. This feedback loop ensures the model's ability to self-adjust in the face of mutations and noise, thereby maintaining robustness and optimal long-term performance in dynamic environments. This provides systematic theoretical support for adaptive learning and continuous optimization. Table 1 is used to illustrate the common types of data distribution drift in dynamic environments (mutation, gradual, and periodic) and the applicable conditions and characteristics of their corresponding detection algorithms. By comparing the detection mechanisms, response times,

ISSN 2706-655X Vol. 7, Issue 4: 77-82, DOI: 10.25236/IJFET.2025.070411

computational complexity, and applicable scenarios of different algorithms, a basis is provided for the selection and design of subsequent online update mechanisms[14-15].

Table 1: Comparison of common data distribution drift types and detection algorithms in dynamic environments

Drift Type	Typical characteristics	Representative Detection Algorithm	Response time	Computational complexity	Applicable Scenarios
Mutational drift	The distribution changes dramatically in a short period of time	DDM, Page-Hinkley	Quick response	medium	Real-time monitoring system, financial transaction forecasting
Gradual drift	Data distribution changes slowly	ADWIN, EDDM	Medium-speed response	Lower	Environmental monitoring, energy consumption forecasting
Periodic drift	The distribution fluctuates periodically	HDDM, CUSUM	Predictable response	medium	Traffic flow analysis, meteorological data modeling
Compound drift	There are multiple change modes at the same time	Ensemble-DD, Adaptive Boost	Dynamic response	Higher	High-dimensional complex environments and heterogeneous data scenarios

4. Issues related to online updating of machine learning models and design of adaptive optimization mechanisms in dynamic environments

The main way to deal with changes in data flow in dynamic situations is an online update structure based on sliding windows and incremental learning. Its main idea is to use time windows to control the data sample, so that the model can continuously update even if there is a limit to the resources available [16]. Sliding Window Method: Retains the most recent samples and discards old data by using an adaptive window size. Incremental Learning Mechanism: Uses forgetting factors and weight redistribution to balance between historical knowledge and new knowledge, allowing the model to consider both stability and adaptability when iterating parameters. Multi-time-scale adaptive learning rate optimization method further improves the model's ability to adapt to the rate of change of the environment. A two-layer learning rate scheduling mechanism with fast and slow changes has been introduced, so that the model can quickly catch short-term drifts and still converge in the long run. The learning rate is expressed as a layered function, where the short term represents local disturbance and the long term controls the global trend, this avoids overfitting and slow convergence. Meta-learning based dynamic parameter adjustment strategy for models includes "learning how to learn" concept, it depends on two level optimization structure to realize parameter self-regulation. The inner optimization layer does instant updates; the outer meta-optimizer learns update rules from past gradients and performance info so the model can switch tasks fast. Algorithms such as MAML and Reptile have shown that they can converge quickly and are robust under dynamic distribution. Model self-evolution framework establishes a closed-loop optimization system of decision making and feedback through adding a kind of reinforcement feedback. It's about reinforcement learning, turning model performance indicators into reward signals to do self-correction and restructuring. This mechanism pushes the model to reach policy self-evolution when interacting with the environment, so as to constantly optimize and reach a steady state. Final System Architecture: The final system architecture comprises four layers – Data Stream Access Layer, Drift Detection Layer, Online Update Layer, and Feedback Optimization Layer – creating an adaptive closed loop starting from data acquisition up until model reconstruction. The data stream access layer handles features extraction, the drift detection layer monitors changes in distribution, the online update layer proactively carries out parameter incremental learning, and the feedback layer employs meta-learning and reinforcement learning for strategy modification. The whole system realizes dynamic learning, real-time response and long-term optimization by the cooperation of many parts, giving an engineering solution and theoretical basis for constructing an adaptive and self-evolving intelligent learning system [17-19].

ISSN 2706-655X Vol. 7, Issue 4: 77-82, DOI: 10.25236/IJFET.2025.070411

5. Experimental verification and performance evaluation of adaptive models in dynamic environments

Experimental dataset and scenario creation are the foundations for testing whether machine learning models have the ability to adapt to dynamic environments. And their designs need to contain some important features: changes of data distribution, environmental disturbance and task target drift. To make it general and practical, we could use the experiment method that combines synthesized data and real scene data. Synthetic data depends on creating a time-related distribution shift function (P_t(X)). Sudden, gradual, and cyclical concept drift situations are simulated so as to check if the model adapts well to different types of dynamics. Real world data such as financial market data can be chosen as representative applications. For these tasks, the data has obvious non-stationary and multi-source heterogeneous characteristics, which fully show the model’s strong robustness and generalization performance in complex environments. When building scenarios, pay attention to the influence of external noise and outliers. Keep experimental consisteny via data cleaning and feature normalization. The experimental system should support streaming input and real-time update methods to simulate the dynamism of online decision-making systems. Model response time, drift detection accuracy, and performance improvements following updates across different scenarios will offer vital support for later performance analyses. Table 2 lists out the performances and resources consumption of different models under dynamic environment experiments including prediction accuracy, detection latency, stability index, memory usage and energy consumption ratio. This comparison shows the benefits of the suggested model when considering overall performance, offering proof for later system improvements and engineering implementation.

Table 2: Comparison of experimental model performance and resource efficiency

Model Type	Test scenario	Prediction accuracy (Accuracy/%)	Detection delay (s)	Stability Index (SI)	Memory usage (MB)	Energy consumption ratio (relative value)
Static Random Forest	Baseline scenario	85.4	7.2	0.68	540	1
Online Gradient Descent	Streaming data scenario	88.6	4.1	0.74	460	0.92
Incremental learning model based on sliding window	Dynamic drift scene	91.2	2.9	0.81	420	0.87
Multi-timescale adaptive optimization model	Multitasking scenarios	93.7	2.3	0.85	410	0.84
Meta-learning and reinforcement feedback fusion model	Comprehensive dynamic environment	95.1	1.8	0.89	395	0.79

In order to fully prove the validity and benefits of the proposed model, comparison experiments have been carried out and a set of evaluation metrics has been established. Experimental groups comprised online updating models using sliding windows and incremental learning, multi-timescale learning rate optimization models, and self-evolutionary models that combined meta-learning and reinforcement feedback mechanisms. Control group chose representative static learning model which was traditional random forest, SVM and also popular online learning algorithm such as Hoeffding tree and online gradient descent. Two types of experiment were used, one was distributed drift control experiment, another was multi-task switching experiment. The first kind was to see how the model reacts dynamically when there are different levels of drift, the second kind was to look at how fast it can adapt between tasks. Prediction accuracy is part of the evaluation metric system. To see if it could work in actual systems, they added engineering measures about memory use, time for calculations, and energy use. How to analyze the experimental results? It’s all about looking at how each model does with different kinds of drifts and tasks. Adaptive optimization based models do better than static models and conventional online algorithms when it comes to detecting changes quickly, making accurate predictions, and being sturdy in the face of sudden and slow environmental shifts. Especially in the case of multi-task scenarios, the meta-learning process greatly accelerates the speed at which the model can be retrained, and the improved feedback loop helps to slow down the rate at which performance degrades. Analyzing the model's resource consumption and updating efficiency also shows that it has engineering application value. The adaptive update model cuts down on computing power by about 30% on average compared to the same hardware setup. Windowing and Incremental Mechanisms cut back on memory use quite a bit, and adjusting the learning rate on different time scales helps solve problems related to using the model too many times. In general, from the whole analysis we know that under dynamic and complex environment this kind of model can balance learning accuracy, response speed and resource consumption, which provides theoretical basis and practical reference for the efficient deployment of intelligent systems.

ISSN 2706-655X Vol. 7, Issue 4: 77-82, DOI: 10.25236/IJFET.2025.070411

6. Conclusion

This paper does a thorough study on the online update and adaptive optimization process of machine learning models in changing surroundings. The paper starts with theoretical modeling, followed by algorithm design and ends up with experimental validation, it analyzes the model's capacity to keep learning and evolving over non stationary data streams. From what I see from the research result, it seems traditional static model faces considerable timeliness and structure problems in dynamic environment. Adaptive learning framework put forward in this paper accomplishes dynamic equilibrium and constant convergence of model parameters through sliding window update, incremental learning and multiple timescale optimization techniques. Meta-learning embedding gives the model cross-task migration and quick re-adaptation skills, so it cuts down the reaction time after changes in the environment. Enhanced feedback forms a closed-loop self-evolving system, which allows the model to self-correct and rebuild strategies when interacting with the environment, thus maintaining good performance even when there is concept drift or target drift. Experiment results prove that the proposed method is much better than others in terms of predicting accuracy, drift detection speed, and using resources, especially when the environment is complicated and different kinds of things live together, it shows it works well no matter how difficult or changeable it gets. This study both adds to the theoretical framework within the realm of dynamic machine learning and adaptive optimization, and it gives a practical way for making smart decision-making systems continue to improve. More research could be done on promoting and applying this idea to harder systems with many agents working together, groups learning online at the same time, and changing between different areas, trying to get smarter and learn all by itself in even better ways.

References

[1] Li, Y., Chen, Z., & Wang, H. (2025). Online adaptive learning for dynamic data environments: A meta-reinforcement approach. IEEE Transactions on Neural Networks and Learning Systems, 36(2), 1450–1463.

[2] Zhang, Q., Liu, J., & Dong, F. (2024). Incremental learning with adaptive drift detection in non-stationary environments. Information Sciences, 662, 119908.

[3] Qi, Y. (2025). Data Consistency and Performance Scalability Design in High-Concurrency Payment Systems. European Journal of AI, Computing & Informatics, 1(3), 39-46.

[4] Jiang, Y. (2025). Application and Practice of Machine Learning Infrastructure Optimization in Advertising Systems. Journal of Computer, Signal, and System Research, 2(6), 74-81.

[5] Zou, Y. (2025). Automated Reasoning and Technological Innovation in Cloud Computing Security. Economics and Management Innovation, 2(6), 25-32.

[6] An, C. (2025). Study on Efficiency Improvement of Data Analysis in Customer Asset Allocatior. Journal of Computer, Signal, and System Research, 2(6), 57-65.

[7] Huang, J. (2025). Optimization and Innovation of AI-Based E-Commerce Platform Recommendation System. Journal of Computer, Signal, and System Research, 2(6), 66-73.

[8] Su H, Luo W, Mehdad Y, et al(2025). Llm-friendly knowledge representation for customer support[C]//Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025: 496-504.

[9] Z Zhong. AI-Assisted Workflow Optimization and Automation in the Compliance Technology Field [J]. International Journal of Advanced Computer Science and Applications (IJACSA), 2025, 16(10): 1-5.

[10] Zhang, X. (2025). Optimization of Financial Fraud Risk Identification System Based on Machine Learning. Journal of Computer, Signal, and System Research, 2(6), 82-89.

[11] Wang, Y. (2025). Exploration and Clinical Practice of the Optimization Path of Sports Rehabilitation Technology. Journal of Medicine and Life Sciences, 1(3), 88-94.

[12] Li W.(2025). The Influence of Financial Due Diligence in M&A on Investment Decision Based on Financial Data Analysis[J]. European Journal of AI, Computing & Informatics, 2025, 1(3), 32-38.

[13] Sheng, C. (2025). Innovative Application and Effect Evaluation of Big Data in Cross-Border Tax Compliance Management. Journal of Computer, Signal, and System Research, 2(6), 40-48.

[14] Tu, X. (2025). Optimization Strategy for Personalized Recommendation System Based on Data Analysis. Journal of Computer, Signal, and System Research, 2(6), 32-39.

[15] Zhu, P. (2025). The Role and Mechanism of Deep Statistical Machine Learning In Biological Target Screening and Immune Microenvironment Regulation of Asthma. arXiv preprint arXiv:2511.05904.

[16] Liu, B. (2025). Design and Implementation of Data Acquisition and Analysis System for

ISSN 2706-655X Vol. 7, Issue 4: 77-82, DOI: 10.25236/IJFET.2025.070411

Programming Debugging Process Based On VS Code Plug-In. arXiv preprint arXiv:2511.05825.
[17] Zou, Y. (2025). Design and Implementation of a Cloud Computing Security Assessment Model Based on Hierarchical Analysis and Fuzzy Comprehensive Evaluation. arXiv preprint arXiv:2511.05049.
[18] Chang, C. (2025). Compiling Declarative Privacy Policies into Runtime Enforcement for Cloud and Web Infrastructure. Machine Learning Theory and Practice, 5(1), 76-86.
[19] Sun, J. (2025). Research on Sentiment Analysis Based on Multi-source Data Fusion and Pre-trained Model Optimization in Quantitative Finance. Socio-Economic Statistics Research (2025), 6,(2), 89-98.