VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**NGUYEN HOANG PHUC – 521H0510**

**FINAL REPORT**

**INTRODUCTION TO**

**MACHINE LEARNING**

**HO CHI MINH CITY, YEAR 2023**

VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**NGUYEN HOANG PHUC – 521H0510**

**FINAL REPORT**

**INTRODUCTION TO**

**MACHINE LEARNING**

Advised by

Assoc. Prof.Le Anh Cuong

**HO CHI MINH CITY, YEAR 2023**

**ACKNOWLEDGMENT**

We would like to express our deepest gratitude to Ths. Mai Van Manh for his invaluable guidance and support throughout the preparation of this report. Your expertise and insights have been instrumental in shaping our understanding and approach to machine learning. Thank you for your time, patience, and dedication.

*Ho Chi Minh City, day 17nd, month 12, year 2023*

*Author*

*(Signature and full name)*

Nguyen Hoang Phuc

**DECLARATION OF AUTHORSHIP**

We hereby declare that this thesis was carried out by ourselves under the guidance and supervision of Assoc. Prof. Le Anh Cuong; and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by my own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

**We will take full responsibility for any fraud detected in our thesis**. Ton Duc Thang University is unrelated to any copyright infringement caused on my work (if any).

*Ho Chi Minh City, day 17nd, month 12, year 2023*

*Author*

*(Signature and full name)*

Nguyen Hoang Phuc

# ABSTRACT

This report examines key elements in machine learning: optimizers and Continual Learning. It begins by exploring different optimizers like Gradient Descent and Adam, explaining how they enhance model learning efficiency and accuracy. The discussion includes a comparative analysis of these optimizers, focusing on their use cases and effectiveness in various scenarios.

The next section addresses Continual Learning, emphasizing its importance in dynamic environments where data continuously evolves. It highlights the challenge of catastrophic forgetting and the necessity of real-world model testing to ensure practical applicability and effectiveness.

Concluding the report, the synergy between advanced optimizers and Continual Learning is discussed, showcasing how their integration leads to more robust, adaptable models. This combination is presented as crucial for the development of sophisticated machine learning models suited for real-world complexities.

# CONTENTS

[ABSTRACT v](#_Toc153207392)

[CONTENTS vi](#_Toc153207393)

[CHAPTER 1. Understanding Optimizers in Machine Learning 1](#_Toc153207394)

[1.1 What Are Optimizers? 1](#_Toc153207395)

[1.2 Types of Optimizers 1](#_Toc153207396)

[1.3 Comparing Optimizers 3](#_Toc153207397)

[CHAPTER 2. Continual Learning and Testing in Production 5](#_Toc153207398)

[2.1 What is Continual Learning? 5](#_Toc153207399)

[2.2 The Challenge of Continual Learning 6](#_Toc153207400)

[2.3 Testing Models in the Real World 6](#_Toc153207401)

[CHAPTER 3. Advanced Integration in Machine Learning 7](#_Toc153207402)

[3.1 Enhancing Model Performance with Optimizers 7](#_Toc153207403)

[3.2 Continual Learning: Evolving Models in Dynamic Environments 8](#_Toc153207404)

[3.3 Synergizing Optimizers with Continual Learning for Robust Models 8](#_Toc153207405)

[PREFERENCES 10](#_Toc153207406)

# Understanding Optimizers in Machine Learning

## What Are Optimizers?

Think of optimizers in machine learning as guides that help a model learn better and faster. Their main job is to adjust the model's internal settings (called weights) to reduce mistakes (or loss). Imagine a model's learning process like walking through a hilly area. The goal is to find the lowest valley, which represents the least amount of error. Optimizers help the model find this valley.

They work by looking at the model's current error and figuring out which way to go to decrease this error. It's like deciding whether to go uphill or downhill to reach lower ground.

The right optimizer can make a big difference. It's like having a good guide who knows the quickest route to the valley. This speeds up the model's learning and helps it perform well.

However, it's a common misunderstanding that optimizers are learning from the data themselves. They are just tools that make the model's learning process more efficient. They don't learn patterns or information from the data; they just help the model to do so in a better way. Understanding this role is key to picking the best optimizer for different learning tasks.

## Types of Optimizers

In the world of machine learning, the choice of an optimizer can be critical. Each optimizer brings its unique flavor to the table, helping models learn from data in different ways:

1. **Gradient Descent:**
   * **The Classic Approach:** A fundamental optimization algorithm that adjusts model parameters based on the entire dataset to minimize loss.
   * **The Mechanism:** It calculates the overall loss function's gradient for the dataset and updates the weights to reduce this loss.
2. **Stochastic Gradient Descent (SGD):**
   * **The Speedster:** Optimizes model parameters using just one data point at a time for faster learning.
   * **The Mechanism:** Selects a random data point at each iteration to update the model, leading to fast but somewhat volatile progress.

* 𝜃: parameters of the model.
* 𝜂: the learning rate.
* : is the gradient of the loss function J according to the parameter 𝜃, calculated at the data point và corresponding label .

1. **Mini Batch Stochastic Gradient Descent (MB-SGD):**
   * **Balancing Act:** Strikes a balance between the classic Gradient Descent and SGD by using small data batches for updates.
   * **The Mechanism:** Updates weights based on small subsets of data, combining the efficiency of SGD with the stability of full-batch learning.
2. **SGD with Momentum:**
   * **The Momentum Factor:** Adds consistency and direction to the standard SGD updates.
   * **The Mechanism:** Incorporates a portion of the previous update in the current one, reducing oscillations and accelerating convergence.
3. **Nesterov Accelerated Gradient (NAG):**
   * **One Step Ahead:** An advanced variant of momentum-based SGD, offering more responsive updates.
   * **The Mechanism:** Updates parameters by preliminarily moving in the accumulated gradient direction, then adjusts after calculating the updated gradient.
4. **Adaptive Gradient (AdaGrad):**
   * **The Customizer:** Tailors learning rates to each parameter, especially beneficial for sparse datasets.
   * **The Mechanism:** Adjusts each parameter's learning rate based on the sum of its past gradients, allowing larger updates for infrequent features.
5. **AdaDelta:**
   * **Overcoming AdaGrad's Limits:** An evolution of AdaGrad, mitigating its rapidly decreasing learning rates.
   * **The Mechanism:** Utilizes a window of recent gradients for updates, ensuring more consistent learning rates over time.
6. **RMSprop:**
   * **The Stabilizer:** Modifies learning rates based on recent trends in gradient magnitudes.
   * **The Mechanism:** Adjusts learning rates for each weight according to the recent magnitude of its gradients, promoting smoother training progress.

* : The model parameter at the time step t.
* 𝜂: Learing rate.
* : mean of the square of the gradient.
* : the gradient of the loss function at the time step t.
* : a small constant to avoid division by zero.

1. **Adam:**
   * **The Best of Both Worlds:** Merges the adaptive learning rate feature of RMSprop with the momentum aspect.
   * **The Mechanism:** Maintains an average of past gradients and adapts learning rates accordingly, providing a versatile approach for various learning challenges.

* : The model parameter at the time step t.
* 𝜂: Learning rate.
* : bias-corrected estimates of the first moment (mean of the gradient) and second moment (mean of the square of the gradient) of the gradient.
* : a constant to avoid division by zero (usually very small, e.g ).

Each optimizer offers a unique way to direct models in learning from data. The selection often depends on the specific requirements and characteristics of the machine learning task at hand.

## Comparing Optimizers

Let's explore a range of optimizers in machine learning, focusing on when they are most effective, what benefits they bring to the table, and the potential challenges they might pose:

1. **Gradient Descent:**
   * **Use Cases:** Best for small to medium-sized datasets where computational resources are not a major constraint.
   * **Advantages:** Simple and straightforward; offers stable convergence.
   * **Disadvantages:** Computationally inefficient for large datasets; converges slowly.
2. **Stochastic Gradient Descent (SGD):**
   * **Use Cases:** Effective for large datasets and online learning scenarios.
   * **Advantages:** Faster convergence than Gradient Descent; capable of escaping local minima.
   * **Disadvantages:** High variance in updates can lead to instability in the convergence path.
3. **Momentum:**
   * **Use Cases:** Useful for deep networks or scenarios with surfaces having ravines or plateaus.
   * **Advantages:** Accelerates SGD; helps navigate through ravines and avoid local minima.
   * **Disadvantages:** Momentum might lead to overshooting the minimum due to accumulated velocity.
4. **Nesterov Accelerated Gradient (NAG):**
   * **Use Cases:** Effective in similar scenarios as Momentum but requires smoother landscapes.
   * **Advantages:** More refined than Momentum with a look-ahead feature, leading to faster convergence.
   * **Disadvantages:** Requires careful tuning of hyperparameters; may overshoot in noisy landscapes.
5. **Adaptive Gradient (AdaGrad):**
   * **Use Cases:** Ideal for sparse data and dealing with feature importance unevenly distributed across data.
   * **Advantages:** Adapts learning rates to each parameter; good for sparse datasets.
   * **Disadvantages:** Learning rate can diminish too quickly, potentially stopping learning early.
6. **AdaDelta:**
   * **Use Cases:** Suitable for problems similar to AdaGrad but where a more robust approach is needed.
   * **Advantages:** Addresses AdaGrad's rapidly diminishing learning rates; more stable.
   * **Disadvantages:** It may require more tuning compared to simpler methods like SGD.
7. **RMSprop:**
   * **Use Cases:** Particularly effective in online and non-stationary problems.
   * **Advantages:** Adjusts learning rate based on recent gradient magnitudes, leading to more stable updates.
   * **Disadvantages:** Hyperparameter tuning is crucial for optimal performance.
8. **Adam:**
   * **Use Cases:** Versatile for a wide range of machine learning problems, especially where fast and efficient convergence is needed.
   * **Advantages:** Combines benefits of adaptive learning rate and momentum; generally efficient in practice.
   * **Disadvantages:** Can be sensitive to hyperparameter settings; sometimes unstable in very noisy problems.

# Continual Learning and Testing in Production

## What is Continual Learning?

Continual Learning in machine learning is a dynamic approach where models constantly adapt and learn from new data while retaining previously acquired knowledge. This concept is essential in a rapidly evolving data environment, as it ensures that models remain current and effective over time.

* **Core Concept:** Unlike traditional models trained on a static dataset, Continual Learning models are designed to integrate new information continually, enhancing their knowledge base without forgetting previous learning.
* **Key Challenge:** The primary focus is to prevent 'catastrophic forgetting,' where new learning can overshadow or erase prior knowledge.
* **Practical Relevance:** It's especially useful in scenarios where data patterns frequently change, such as evolving user preferences in recommendation systems.
* **Advantages:** This approach keeps models up-to-date, reduces the need for retraining, and increases robustness to data variations.

## The Challenge of Continual Learning

Continual Learning in machine learning brings its set of challenges, with the main one being "catastrophic forgetting." This issue arises when a model learns new information but forgets what it previously learned.

* **Catastrophic Forgetting Explained:** This occurs when a model, upon learning new data, overwrites the old information that it was trained on earlier. It's akin to a student forgetting previously studied subjects while preparing for new ones.
* **Balancing Act:** The key difficulty is making sure the model can learn new things without losing valuable knowledge it already has. It's like keeping a balance between remembering the past and adapting to the present.
* **Dealing with Complex Data:** Real-world data is diverse and constantly changing. A continual learning model needs to handle this complexity without getting overwhelmed.
* **Resource Management:** These models must also learn continuously without needing too much computer memory or processing power.

## Testing Models in the Real World

Testing machine learning models in real-world conditions, known as "production testing," is crucial for ensuring they work effectively outside of controlled environments.

* **Real-Life Scenarios:** This testing exposes models to unpredictable and varied real-world data, helping to assess their true performance and adaptability.
* **Continuous Monitoring:** It’s important to keep an eye on the model's performance over time, watching for any changes or issues. This ongoing monitoring helps in understanding how the model responds to new data.
* **Improvement:** Real-world testing provides valuable feedback, which is used to refine and improve the model. Errors or inaccuracies identified during this phase can lead to further adjustments or additional training.
* **Challenges:** Production testing must navigate challenges like maintaining data privacy, ensuring the model's decisions are understandable, and efficiently using computational resources.

Overall, production testing is about verifying that machine learning models are not just theoretically sound but also practically reliable and capable of handling the complexities of real-world data.

# Advanced Integration in Machine Learning

## Enhancing Model Performance with Optimizers

In this section, we delve into how optimizers can significantly enhance the performance of machine learning models. By intelligently adjusting model parameters, optimizers can drastically improve both the speed and accuracy of learning.

* **Optimizers as Performance Boosters:** Discuss how different types of optimizers, from Gradient Descent to Adam, can impact model performance. This includes faster convergence, escaping local minima, and efficient handling of large datasets.
* **Tailoring Optimizers to Model Needs:** Explore strategies for choosing the right optimizer based on specific model requirements, such as dataset size, complexity, and learning speed.
* **Real-World Case Studies:** Provide examples where the choice of optimizer led to substantial improvements in model performance, showcasing their impact in practical scenarios.

## Continual Learning: Evolving Models in Dynamic Environments

Continual Learning represents a paradigm shift in machine learning, allowing models to adapt and evolve in response to new data over time. This section discusses how Continual Learning enables models to stay relevant in dynamic environments.

* **Adaptability of Models:** Explain how Continual Learning equips models to handle changing data patterns, preventing model obsolescence and maintaining accuracy over time.
* **Overcoming Catastrophic Forgetting:** Discuss techniques and strategies used in Continual Learning to balance new learning with the retention of previously acquired knowledge.
* **Application in Fast-Changing Domains:** Illustrate the application of Continual Learning in fields like social media trends, stock market analysis, and customer behavior prediction, where data patterns frequently change.

## Synergizing Optimizers with Continual Learning for Robust Models

The final section focuses on the synergy between optimizers and Continual Learning, emphasizing how their integration can lead to the development of more robust and adaptable models.

* **Combining Strategies for Enhanced Learning:** Discuss how the right optimizer can complement a Continual Learning approach, leading to more effective and efficient adaptation to new data.
* **Challenges and Solutions:** Address the challenges that arise when combining these two techniques, such as managing computational complexity and ensuring stability in learning.
* **Innovative Use Cases:** Present innovative use cases where the combination of advanced optimizers with Continual Learning has led to breakthroughs in machine learning applications, demonstrating the power of this integrated approach.

**PREFERENCES**

[KDnuggets](https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html)

[Fritz](https://fritz.ai/optimizers-in-machine-learning/)

[Krish Naik](https://krishnaik.in/2022/03/28/understanding-all-optimizers-in-deep-learning/)

[The Full Stack](https://fullstackdeeplearning.com/course/2022/lecture-6-continual-learning/)

[Wiki](https://wiki.continualai.org/the-continualai-wiki/introduction-to-continual-learning)

[Adaptive Subgradient Methods for Online Learning and Stochastic Optimization](https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf)